PERFORMANCE TRADEOFF STUDY OF A GPS-AIDED INS FOR A ROCKET TRAJECTORY

THESIS

Muhittin Istanbulluoglu, 1st Lt, TuAF

AFIT/GE/ENG/02M-11

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/GE/ENG/02M-11

PERFORMANCE TRADEOFF STUDY OF A GPS-AIDED INS FOR A ROCKET TRAJECTORY

THESIS

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Engineering and Environmental Management

Muhittin Istanbulluoglu, BS

1st Lt, TuAF

March 2002

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GCE/ENP/02-2

PERFORMANCE TRADEOFF STUDY OF A GPS-AIDED INS FOR A ROCKET TRAJECTORY

Muhittin Istanbulluoglu, BS

1st Lt, TuAF

Approved:

Peter S. Maybeck — 14 Mar 02

Peter S. Maybeck (Chairman) — date

M M Miller — 14 MAR 02

Mikel M. Miller (Member) — date

John F. Raquet — 14 MAR 02

John F. Raquet (Member) — date

Acknowledgments

I would like to thank my thesis advisor, Dr. Peter Maybeck, for his support, patience and guidance through this research. I feel myself lucky for being his thesis student. I would also like to thank my thesis committee members Lt. Col. Mikel M. Miller and Maj. John F. Raquet, for their help and support. Special thanks to Dr. Stanton Musick for his help and support in my studies.

I also would like to thank the great Turkish nation and Turkish Air Force for providing me such a quality Master's Program opportunity. I believe, I have done my best to take full advantage of the experience that I have gotten in this program to serve my country better.

I will not forget to thank my wife for her extreme patience and support, and my parents and all my family members for the values they thought me, which helped me a lot to complete this program.

Table of Contents

Page

Acknowledgments iv

Table of Contents v

List of Figures x

List of Tables xiv

AFIT/GE/ENG/02M-11

Abstract xvi

I Introduction I-1

1.1 Background I-2

1.2 Key Terms I-7

1.3 Literature Review I-8

1.3.1 Benefits of Integrated Systems I-8

1.3.2 INS/GPS Integration Techniques I-10

1.3.3 Integration Kalman Filter Error and Measurement Models I-11

1.3.4 The Chosen Launch Vehicle (Atlas IIAS) for Simulations I-12

1.4 Problem Statement I-14

1.5 Assumptions I-17

1.6 Scope I-22

1.7 Summary I-24

II Theory II-1

2.1 Introduction II-1

2.2 Ring Laser Gyro (RLG) Strapdown INS II-1

2.3 Barometric Altimeter II-3

2.4 Global Positioning System (GPS) II-4

2.4.1 GPS Space Segment II-5

2.4.2 GPS Control Segment II-6

2.4.3 GPS User Segment II-7

2.5 Differential GPS (DGPS) II-8

2.6 Reference Frames II-9

2.6.1 Inertial Frame (x^i , y^i , z^i) II-11

2.6.2 Earth Frame (x^e , y^e , z^e) II-12

2.6.3 Geographic Frame $(x^g , y^g , z^g) = (E, N, U)$ II-13

2.6.4 Navigation Frame (x^n , y^n , z^n) II-14

2.6.5 Body Frame (x^b , y^b , z^b) II-15

2.7 Reference Frames Transformations II-16

2.7.1 Inertial Frame to Earth Frame, C_i^e II-17

2.7.2 Earth Frame to Geographic Frame, C_e^g II-17

2.7.3 Earth Frame to Navigation Frame, C_e^n II-17

2.7.4 Geographic Frame to Navigation Frame, C_g^n II-18

2.7.5 Geographic Frame to Body Frame, C_g^b II-18

2.7.6 Navigation Frame to Body Frame, C_n^b II-19

2.8 Kalman Filter Theory II-19

2.8.1 What is a Kalman Filter? II-19

2.8.2 Kalman Filter Example II-20

2.8.3 Linear Kalman Filter II-26

2.8.4 Linearized and Extended Kalman Filter II-30

2.9 Summary II-34

III Design Methodology and Error Models III-1

3.1 Overview III-1

3.2 Introduction to MatSOFE III-1

3.3 Introduction to PROFGEN III-6

3.4 Satellite Vehicle Data Using WSEM 3.6 .... III-11

3.5 The SNSM Computer Model III-13

3.6 Space Navigation System Model (SNSM) Description III-16

3.6.1. The Inertial Navigation System (INS) Model III-18

3.6.1.1 The 93-State LN-93 Error Model III-19

3.6.1.2 The 39-State INS Truth Model III-20

3.6.1.3 The 11-State INS Filter Model III-21

3.6.2. The Global Positioning System (GPS) Model III-22

3.6.2.1 The 30-State GPS Truth Model III-23

3.6.2.2 The 22-State DGPS Truth Model III-26

3.6.2.3 The 2-State GPS/DGPS Filter Model III-28

3.6.2.4 GPS Measurement Model III-29

3.6.2.5 DGPS Measurement Model III-32

3.7 Chapter Summary....................III-34

IV Results and Analysis....................IV-1

4.1 Validation of MATSOFE....................IV-2

4.1.1. Comparison to Gray's Results [10]....................IV-4

4.1.2. Comparison to Britton's Results [6]....................IV-6

4.2 The Launch Vehicle (Atlas IIAS) Flight Profile....................IV-7

4.3 Development of the Three Types of INS's....................IV-9

4.4 Filter Tuning Process....................IV-10

4.5 Performance Analysis....................IV-12

4.5.1. Baro-Altimeter and Standard GPS Aiding Cases....................IV-12

4.5.1.1 Case I, 0.4 nm/hr CEP INS/Baro-Alt./ Std.GPS....................IV-12

4.5.1.2 Case III, 2.0 nm/hr CEP INS/ Baro-Alt/ Std.GPS....................IV-14

4.5.1.3 Case V, 4.0 nm/hr CEP INS/ Baro-Alt/ Std.GPS....................IV-15

4.5.2. Baro-Altimeter and P-Code DGPS Aiding Cases....................IV-15

4.5.2.1 Case II, 0.4 nm/hr CEP INS/Baro-Alt./P-Code DGPS....................IV-16

4.5.2.2 Case IV, 2.0 nm/hr CEP INS/Baro-Alt./P-Code DGPS....................IV-17

4.5.2.3 Case VI, 4.0 nm/hr CEP INS/Baro-Alt./P-Code DGPS....................IV-18

4.6 Chapter Summary....................IV-19

V. Conclusions and Recommendations....................V-1

5.1 Conclusion....................V-1

5.2 Recommendations....................V-3

Appendix A Error State Definitions for the SNSM Truth and Filter Models....................A-1

Appendix B Dynamics Matrices and Noise Values....................B-1

Appendix C PROF_IN Input File C-1

Appendix D Atlas IIAS Rocket Profile Plots D-1

Appendix E True Ephemeris Generation Process E-1

Appendix F Plots Obtained from WSEM 3.6 .... F-1

Appendix G MatSOFE Validation Plots G-1

Appendix H Tuning Examples H-1

Appendix I Plots of Case I I-1

Appendix J Plots of Case II J-1

Appendix K Plots of Case III K-1

Appendix L Plots of Case IV L-1

Appendix M. Plots of Case V M-1

Appendix N Plots of Case VI N-1

List of Figures

Page

Figure I-1. Loosely coupled INS/GPS integration I-4

Figure I-2. Tightly coupled INS/GPS integration I-4

Figure I-3. The Principle of Satellite Navigation [25] I-5

Figure I-4. Earth's Orbits [26] I-13

Figure I-5. Atlas IIAS I-13

Figure I-6. Space Navigation System Model (SNSM) Simulation I-21

Figure II-1. GPS Major Segments II-4

Figure II-2. GPS Orbital Planes II-6

Figure II-3. GPS master control & monitor stations II-7

Figure II-4. Circle Diagram Relating PROFGEN Frames [29] II-10

Figure II-5. Inertial Frame II-11

Figure II-6. Earth Frame II-12

Figure II-7. Geographic Frame II-13

Figure II-8. Navigation Frame II-14

Figure II-9. Body Frame II-15

Figure II-10. Conditional Density of Position Based on Measured Value z1 .... II-21

Figure II-11. Conditional Density of Position Based on Measurement z2 Alone II-22

Figure II-12. Conditional density of position based on data z1 and z2 .... II-23

Figure III-1. Block Diagram of MatSOFE Hierarchical Structure III-5

Figure III-2. Atlas 2-Dimensional Flight Profile III-11

Figure III-3. Truth and Filter Model Block Diagram..III-15

Figure IV-1. Plot Legend IV-10

Figure C-1. PROF_IN Input File for PROFGEN... C-1

Figure D-1. 2-D Rocket Flight Profile (Alt vs. Time) D-2

Figure D-2. 2-D Rocket Flight Profile (Lat vs. Lon) D-2

Figure D-3. Latitude, Longitude, and Altitude of the Rocket Flight Profile..................D-3

Figure D-4. 2-D Position and Velocity of Rocket Flight Profile D-4

Figure D-5. X,Y, and Z Velocity of the Rocket Flight Profile D-5

Figure D-6. Roll, Pitch, and Heading Change of the Rocket Flight Profile..................D-6

Figure D-7. 2-D Position and Velocity of Tanker Profile..D-7

Figure D-8. Latitude, Longitude, and Altitude of the Tanker Profile D-7

Figure D-9. X,Y, and Z Velocity of the Tanker Profile D-8

Figure D-10. Roll, Pitch, and Heading Change of the Tanker Profile D-8

Figure E-1. Matlab m-file “ADD_SV.M” To Merge INS and GPS data....................... E-2

Figure E-2. Matlab m-file “LOAD_SP3.M” E-4

Figure F-1. PDOP (Start Time and Final Time)..F-2

Figure F- 2. Satellite Elevation/Time Graph F-3

Figure F-3. Satellite Rise/Set Graph F-3

Figure F-4. Number of Visible Satellites F-4

Figure F-5. Bearing/Elevation Graph...F-4

Figure G-1. Latitude, Longitude, and Altitude Errors..G-2

Figure G-2. Latitude, Longitude, and Altitude Errors..G-2

Figure G-3. Latitude, Longitude, and Altitude Errors..G-3

Figure G-4. Latitude, Longitude, and Altitude Errors..G-3

Figure G-5. Latitude, Longitude, and Altitude Errors..G-4

Figure G-6. Latitude, Longitude, and Altitude Errors..G-4

Figure H-1. Latitude and Longitude Errors (Conservative Q-Tuning) H-2

Figure H-2. Latitude and Longitude Errors (Non-Conservative Q-Tuning).................H-3

Figure H-3. East and North Velocity Errors (Conservative Q-Tuning).........................H-4

Figure H-4. East and North Velocity Errors (Non-Conservative Q-Tuning)................H-5

Figure H-5. East, North, and Up Tilt Errors (Conservative Q-Tuning).........................H-6

Figure H-6. East, North, and Up Tilt Errors (Non-Conservative Q-Tuning)................H-7

Figure I-1. Rocket Latitude and Longitude Errors... I-2

Figure I-2. Rocket Altitude and Baro-Altimeter Errors I-3

Figure I-3. Rocket Latitude, Longitude, and Altitude Errors... I-4

Figure I-4. East, North, and Vertical Tilt Errors I-5

Figure I-5. East, North, and Vertical Velocity Errors I-6

Figure I- 6. GPS User Clock Bias and Clock Drift Errors I-7

Figure J-1. Rocket Latitude and Longitude Errors.. J-2

Figure J-2. Rocket Altitude and Baro-Altimeter Errors.. J-3

Figure J-3. Rocket Latitude, Longitude, and Altitude Errors... J-4

Figure J-4. East, North, and Vertical Tilt Errors.. J-5

Figure J-5. East, North, and Vertical Velocity Errors.. J-6

Figure J-6. DGPS User Clock Bias and Clock Drift Errors... J-7

Figure K-1. Rocket Latitude and Longitude Errors K-2

Figure K-2. Rocket Altitude and Baro-Altimeter Errors...K-3

Figure K-3. Rocket Latitude, Longitude, and Altitude Errors K-4

Figure K-4. East, North, and Vertical Tilt Errors..K-5

Figure K-5. East, North, and Vertical Velocity Errors..K-6

Figure K-6. GPS User Clock Bias and Clock Drift Errors...K-7

Figure L-1. Rocket Latitude and Longitude Errors L-2

Figure L-2. Rocket Altitude and Baro-Altimeter Errors L-3

Figure L-3. Rocket Latitude, Longitude, and Altitude Errors....................................... L-4

Figure L-4. East, North, and Vertical Tilt Errors L-5

Figure L-5. East, North, and Vertical Velocity Errors L-6

Figure L-6. DGPS User Clock Bias and Clock Drift Errors L-7

Figure M-1. Rocket Latitude and Longitude Errors.. M-2

Figure M-2. Rocket Altitude and Baro-Altimeter Errors M-3

Figure M-3.   Rocket Latitude, Longitude, and Altitude Errors.................................. M-4

Figure M-4. East, North, and Vertical Tilt Errors M-5

Figure M-5. East, North, and Vertical Velocity Errors M-6

Figure M-6. GPS User Clock Bias and Clock Drift Errors M-7

Figure N-1. Rocket Latitude and Longitude Errors N-2

Figure N-2. Rocket Altitude and Baro-Altimeter Errors..N-3

Figure N-3.   Rocket Latitude, Longitude, and Altitude Errors N-4

Figure N-4. East, North, and Vertical Tilt Errors...N-5

Figure N-5. East, North, and Vertical Velocity Errors...N-6

Figure N-6. DGPS User Clock Bias and Clock Drift Errors...N-7

List of Tables

Page

Table I-1. Case I-VI integration comparison I-15

Table III-1A. PROFGEN Flight Profile Outputs III-8

Table III-1B. PROFGEN Flight Profile Outputs III-9

Table III-2. PROFGEN Segments for "Atlas IIAS" Flight Profile III-10

Table III-3. References for the Sub-Matrices of the SNSM Truth and Filter III-17

Table IV-1. Case I-VI Integration Comparison IV-1

Table IV-2. Atlas IIAS Launch Operation Sequence [15] IV-8

Table IV-3. Q Values for Tilt Errors (before and after fine tuning) IV-11

Table IV-4. Q Values for Velocity Errors (before and after fine tuning) IV-11

Table IV-5. Q Values for Position Errors (before and after fine tuning) IV-12

Table IV-6. Case I 1 Latitude, Longitude and Altitude Errors IV-13

Table IV-7. Case III 1 Latitude, Longitude and Altitude Errors IV-14

Table IV-8. Case V 1 Latitude, Longitude and Altitude Errors IV-15

Table IV-9. Case II 1 Latitude, Longitude and Altitude Errors IV-16

Table IV-10. Britton's MSOFE-generated 1 results for Case II IV-16

Table IV-11. Case IV 1 Latitude, Longitude and Altitude Errors IV-17

Table IV-12. Britton's MSOFE-generated 1 results for Case IV IV-18

Table IV-13. Case VI 1 Latitude, Longitude and Altitude Errors IV-18

Table IV-14. Britton's MSOFE-generated 1 results for Case VI IV-19

Table IV-15. GPS-aided INS Cases Comparison Table IV-20

Table IV-16. DGPS-aided INS Cases Comparison Table...IV-20

Table V-1. Results of All Studied Cases V-2

Table A-1. 39-State INS System Model: First 20 States...A-2

Table A-2. 39-state INS System Model: Second 19 States..A-3

Table A-3. 30-State GPS System Model...A-4

Table A-4. 22-State DGPS System Model..A-5

Table A-5. 13-State Reduced-Order Filter Model A-6

Table B-1. Notation of Variables used in Table B-2 to Table B-4 ....B-2

Table B-2. Elements of the Dynamics Submatrix B-3

Table B-3. Elements of the Dynamics Submatrix $\mathbf{F}_{INS_{t1}}$ B-4

Table B-4. Elements of the Dynamics Submatrix $\mathbf{F}_{INS_{t2}}$ B-5

Table B-5. Elements of the Dynamics Submatrix $\mathbf{F}_{GPS_{t}}$ B-6

Table B-6. Elements of the Dynamics Submatrix $\mathbf{F}_{DGPS_{t}}$ B-6

Table B-7. Elements of Truth Model Process Noise Submatrix for the INS Truth Model B-8

Table B-8. Elements of Truth Model Process Noise for GPS States B-9

Table B-9. Elements of Truth Model Process Noise for DGPS States B-9

Table B-10. Filter Process Noise Q Values for Case Using the 0.4 nm/hr INS B-10

Table B-11. Filter Process Noise Q Values for Cases Using the 2.0 nm/hr INS B-10

Table B-12. Filter Process Noise Q Values for Cases Using the 4.0 nm/hr INS B-10

Table B-13. Truth and Filter Measurement Noise R Values for Cases........................B-11

Table F-1. WSEM Set-Up Parameters F-1

Table F-2. Satellite Bearing/Elevation Table...F-5

AFIT/GE/ENG/02M-11

Abstract

The Turkish Air Force (TuAF) has started a project to launch a satellite using only Turkish resources. Primary motivation behind this research is to assist TuAF's project by keeping up with these innovations in the navigational arena. The basic challenge in navigation system design is to decide which navigation system (or systems) and implementation techniques will be used, depending on accuracy requirements. The two primary navigation systems that will be integrated in this research are the Inertial Navigation System (INS), and the Global Positioning System (GPS). The Kalman filter algorithm is used to integrate INS and GPS. The rocket (Atlas IIAS launch vehicle) flight profile is generated by using PROFGEN and simulated "true" GPS ephemeris data is incorporated into system as GPS measurements. The "modified" alternative system performance analysis tool, MatSOFE, is utilized in this research study. Standard and differential GPS are compared, as are three different grades of INS, in a tradeoff performance analysis.

Performance Tradeoff Study of a GPS-Aided INS for a Rocket Trajectory

I. Introduction

The Turkish Air Force (TuAF) has started a project to launch a satellite using only Turkish resources. In any space program, navigation systems have substantial importance. According to open literature, there are more than 10,000 satellites orbiting around the Earth. Space navigation systems have widely been used in applications for more than thirty years, but the innovations in navigation sensors technology and the implementation techniques of these sensors have been rapidly developing. The primary motivation behind this research is to assist TuAF's project by keeping up with these innovations in the navigational arena.

The basic challenge in navigation system design is to decide which navigation system (or systems) and implementation techniques will be used, depending on accuracy requirements. The two primary navigation systems that will be integrated in this research are the Inertial Navigation System (INS), and the Global Positioning System (GPS). An integrated GPS-aided INS system will be investigated in this study because the strengths and shortcomings of these individual navigation systems are complementary, and therefore a considerable performance gain will be achieved by integrating them together. The Kalman filter algorithm is used to integrate INS and GPS. The integration technique used in an INS/GPS system is named depending on the type of measurements used in the integration Kalman filter: for instance, loosely coupled, tightly coupled or ultra-tightly coupled. The detailed insights related to these systems will be given in the next section, but these terms will be briefly described here. Loosely coupled systems are integrated at the navigation solution level. Both INS position data and GPS position data are taken into the Navigation Kalman filter. Tightly coupled systems are integrated at the pseudorange level, and ultra-tightly coupled systems at the in-phase and quadrature (I and Q) signal level. In this study the tightly coupled integration technique will be used to integrate INS and GPS. The justifications for this choice will be presented in the background section.

It is believed that an integrated INS/GPS system can meet and exceed both the navigation accuracy and integrity requirements for a satellite to launch. In this study, the author will investigate the performance of different types and navigation accuracies of GPS-aided INS systems, and make a tradeoff analysis between their cost, applicability, accuracy, implementation ease, etc. To achieve this goal, 6 different cases, each case involving a separate GPS-aided INS system with different navigation accuracies, were constructed.

This research investigates only the navigation aspect of space navigation systems. The study of designing a control algorithm for a space vehicle could also be studied, but it is considered as the follow-on step, and will not be discussed in this thesis.

1.1 Background

The inertial sensors, like accelerometers and gyros, used in an INS have some errors with stochastic properties. These errors tend to increase over time (long-term instability) with unbounded position error growth and require occasional or continual compensation [17: 526]. Its autonomous navigation capability makes INS the primary navigation system in most applications. It does not need any external aid to navigate, but its accuracy can be improved by using navigational aids (GPS, TACAN, RADAR, etc.). On the contrary, GPS alone does not provide an adequate solution either, “due to low output rate of GPS receivers and the need for redundant information during GPS system failures or simple loss of GPS measurements [44: 2932] ”. GPS has long-term accuracy, but is sensitive under highly dynamic conditions, such that the number of tracked satellites may fall below four, which means the GPS receiver can no longer generate a navigation solution on its own. The integrated INS/GPS system not only provides a very efficient means of navigation due to the short-term accuracy of INS combined with the long-term accuracy of GPS, but also produces a system with performance that exceeds that of the individual sensors.

There are several options to integrate the GPS and INS sensors. Among these options, two traditional integration techniques are widely used in applications: loosely coupled (Figure I-1) and tightly coupled (Figure I-2). In the open literature, there are numerous navigation system designs using these techniques. Both have advantages and disadvantages.

In a loosely coupled system, the INS integration Kalman filter uses GPS-derived position and velocity (the output of a GPS Kalman filter) as a measurement. Loosely coupled integration treats GPS and INS as individual navigation systems, combining the two at the navigation solution level. Raw GPS data is processed first by a Kalman filter, and the integration Kalman filter is then applied to combine INS and GPS navigation solutions. Often only the INS error states are modeled explicitly in the integration Kalman filter. That does not mean that GPS is assumed error-free, only that the dynamic characteristics of the GPS errors are not compensated by the integration filter. The GPS

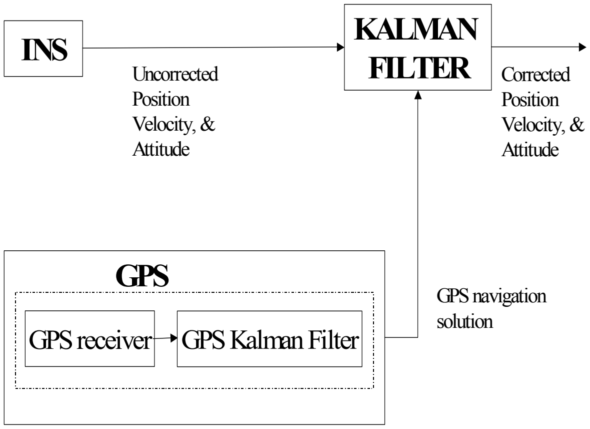

INS
Uncorrected Position Velocity, & Attitude
KALMAN FILTER
Corrected Position Velocity, & Attitude
GPS
GPS receiver
GPS Kalman Filter
GPS navigation solution

Figure I-1. Loosely coupled INS/GPS integration

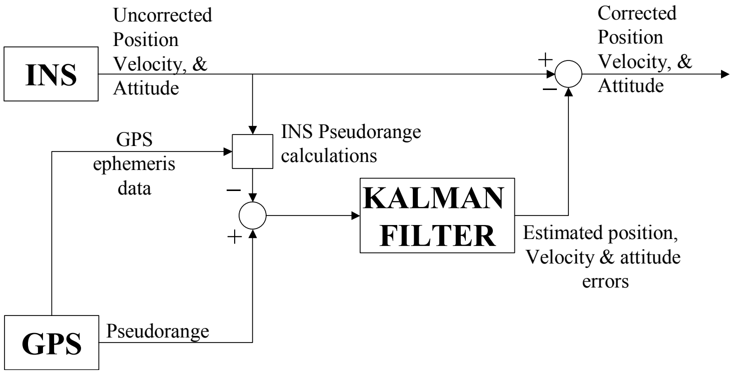
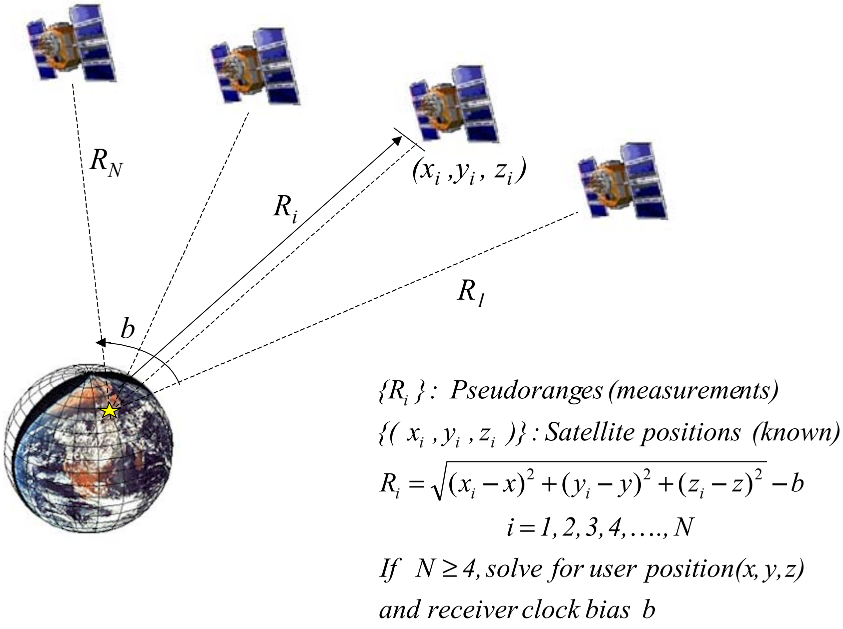

Uncorrected Position Velocity, & Attitude
INS
Corrected Position Velocity, & Attitude
+
−
INS Pseudorange calculations
GPS ephemeris data
−
+
KALMAN FILTER
Estimated position, Velocity & attitude errors
GPS
Pseudorange

Figure I-2. Tightly coupled INS/GPS integration

has its own Kalman filter inside the receiver, which at least models four outputs of the GPS-alone solution, namely three positions and a clock time offset. It is a well-known fact that you need at least four equations to solve four unknowns. Similarly, the GPS receiver provides a three-dimensional position (*x, y, z*) and time estimate provided that a minimum of four satellites is being tracked. See Figure I-3.

Tightly coupled INS/GPS integration combines both INS and GPS measurements in an integrated Kalman filter. The INS/GPS Kalman filter estimates the errors in vehicle



$\{R_i\}$: Pseudoranges (measurements)

$\{(x_i, y_i, z_i)\}$: Satellite positions (known)

$$R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - b$$

$$i = 1, 2, 3, 4, \ldots, N$$

If $N \geq 4$, solve for user position(x, y, z) and receiver clock bias b

Figure I-3. The Principle of Satellite Navigation [25]

position, velocity and attitude along with Inertial Measurement Unit (IMU) errors, such as gyro drift, and accelerometer bias, and GPS errors, such as clock bias and drift rate, by using both INS position solution and GPS pseudorange measurements [41: 773]. Thus, the integration is performed at the pseudorange measurement level rather than at the GPS navigation solution level. Only one Kalman filter is applied in this integration technique. Positioning is performed on the basis of both INS and GPS measurements, even if the number of tracked satellites falls below four. Since they are integrated at the raw in-phase and quadrature phase (I & Q) signal level and provide a position estimate with an accuracy of cm level, and are still under development, ultra-tightly coupled systems are not considered as a concern of this study. Additionally, since the desired position accuracy of a launch vehicle is about 50-80 meters, the author decided not to consider ultra-tightly coupled systems in this study.

There are three main GPS types, depending on GPS receiver range solution technique: (1) Standard GPS, (2) Code Range Differential (DGPS), and (3) Carrier Phase DGPS. For the military applications their *Two-Dimensional Distance Root Mean Square* (2 DRMS) horizontal position accuracies are approximately 6 m., 1-3 m., and 1-2 cm., respectively [34]. 2 DRMS is the root mean square (RMS) of the two-dimensional distance error $\left(DRMS = \sqrt{\left(\sum_{i=1}^{n} \left(x_i^2 + y_i^2 \right) \right) \div n} \right)$. The probability of being inside 2xDRMS circle varies between 95% and 98% (depending upon ratio of σ_x and σ_y). The basic difference between them is the type of information which the GPS receiver uses to generate a navigation solution. The incoming GPS signal consists of three components: carrier (RF sinusoidal signal, which is usually called *phase, or carrier phase*), ranging code (basically zeros and ones, and briefly called *code or PRN code*), and navigation data (a binary coded message) [25]. The differential GPS technique takes advantage of the error correlations between multiple receivers. Its accuracy is anywhere between 5 m. and 1 cm., depending upon the method used. Obviously, carrier phase DGPS provides the highest precision when compared to the others [34]. Taking the accuracy requirements of a launch vehicle into account, the research cases related to carrier-phase DGPS-aided INS systems are removed from the research scope. It was obvious that carrier-phase DGPS would provide much more accuracy than what is needed. Detailed information related to GPS, DGPS, and Carrier-Phase GPS can be found in Key Terms (next page) and Chapter 2.4 of this study.

1.2 Key Terms

Inertial Navigation System (INS): A self-contained, dead reckoning precision navigation system, which uses inertial sensors, such as gyroscopes and accelerometers, to provide navigation information. Typical accuracy: 0.4-4.0 nautical miles/hour circular error probable (CEP). CEP defined as radius of a circle inside which half of the points (errors) fall $\left(CEP = 0.59(\sigma_x + \sigma_y) \pm 3\% \right)$.

Global Positioning System (GPS): A passive, space-based, universal and accurate source of navigation information (three-dimensional position and velocity) and time. The system has been designed, developed, and maintained by the U.S. Department of Defense (DoD) [6,9,10,44]. Accuracy: Stand-Alone GPS is typically 6 meters (military, dual frequency) and 10 meters (civilian) approximate horizontal accuracy (2DRMS) [34].

Differential GPS (DGPS): A ground-based GPS receiver (accurately surveyed) uplinks error corrections to nearby vehicles to be positioned. DGPS takes advantage of the correlation of errors between receivers. The errors associated with the worst error sources are similar for users located “not far” from each other, and they change slowly in time. In other words, errors are correlated both spatially and temporally. Clearly, an error in a measurement can be estimated and removed up to a certain level by using proper DGPS techniques, if the location is known [25]. Accuracy: 1-5 meters, depending upon method used [34].

Carrier-Phase DGPS: A very popular and accurate receiver technique, which is able to measure the incoming satellite-transmitted GPS signal to a fraction of a wavelength. Accuracy: 1-2 centimeter [10, 34].

Kalman Filter: A recursive computer algorithm that uses sampled-data measurements to produce optimal state estimates of a dynamical system, under the assumptions of linear system models and white Gaussian noise models. Developed by R.E. Kalman in the early 1960s, *A New Approach to Linear Filtering and Prediction Problems* [16].

Barometric Altimeter: An altimeter designed to output altitude relative to the pressure difference. They are cheap and widely used as vertical channel aid to INS because of the inherent instability of the vertical channel in an unaided INS [10].

1.3 Literature Review

The following is a brief discussion of the literature reviewed for this research, process, which covers NASA technical reports, previous Air Force Institute of Technology (AFIT) theses, text books, course handouts, and IEEE transactions and conferences related to the INS/GPS integrated navigation system and its space applications. The reviewed sources will be presented in following categories

1.3.1 Benefits of Integrated Systems

The inertial sensors used in an INS have some errors and these errors tend to increase by time. They have to be compensated occasionally or frequently. Although an INS provides good high-frequency navigation information, it has considerable long-term, low frequency errors due to the physical gyro drift rate problem. The use of navigational aids, like GPS and a barometric altimeter, can significantly enhance the navigation accuracy of an INS.

All of the GPS range solution techniques (either stand-alone or differential) suffer from some inadequacy. The most significant are:

- The data rate of GPS receiver is too low for many applications;
- High DGPS accuracy is limited by the distance between the reference station and the user because of the errors associated with the ionosphere, which make it difficult to determine the integer ambiguity resolution on-the-fly [13];
- Influences of high acceleration on the receiver clock, code tracking loop (delay lock loop) and carrier phase loop may become considerable; these tracking loops are used to provide estimates of tracking errors for signal processing purposes in a GPS receiver;
- Signal loss-of-lock and *cycle slips* may occur very frequently due to aircraft high dynamics and other causes [35: 18-12].

INS/GPS integration has proven to be a very efficient means of navigation due to the short term accuracy of INS combined with the long term accuracy of GPS fixes [4,7,8,17,35,41]. INS/GPS integration is the optimal solution to the navigation problem rather than using either system alone, since the Kalman filter includes effects of both GPS and INS errors. “Many researchers have studied the methodology of combining these two navigation sensors. Navigation employing GPS and INS is a synergistic relationship. The integration of these two types of sensors not only overcomes performance issues found in each individual sensor, but also produces a system whose performance exceeds that of the individual sensors.” [17: 526].

INS and GPS are complementary technologies in the sense that the weakness of one is the strength of another. Integration of INS and GPS leads to a particularly attractive and robust system that can produce better navigation information compared to either one.

1.3.2 INS/GPS Integration Techniques

"There are many options to integrate the INS and GPS onboard the spacecraft. The first integration option, also the simplest implementation technique, is resetting the position and velocity output of INS with that of GPS. This option was selected as the baseline INS/GPS integration approach on the NASA/TRW orbit maneuvering vehicle (OMV) [41]."

The second option, called loosely coupled or cascaded integration, basically takes the output of a GPS Kalman filter (position and velocity solution) as measurements into integration Kalman filter. This integration behaves like an INS alone system in case of GPS outages.

The third integration option, called tightly coupled INS/GPS integration, combines the GPS pseudorange and/or range rate measurement and INS raw measurements in an integration Kalman filter. Having only one Kalman filter is not the only difference between loosely and tightly coupled systems. The fundamental difference is using raw measurements instead of the navigation solution for GPS. The INS/GPS integration Kalman filter estimates the errors in spacecraft position, velocity and attitude along with the inertial measurement unit (IMU) instrument errors, such as gyro drift and accelerometer bias and scale factor. This tightly coupled integration option gives the best navigation accuracy of the three techniques just discussed, and it does not suffer from the filter-driving-filter (the GPS receiver Kalman filter-driving-integration Kalman filter) stability concern of the loosely coupled approach [41, 23].

An additional literature review was also conducted to find out whether anything new, especially in the GPS portion of the navigation system and its application in space navigation area, had been accomplished that would provide different, or additional, models to be incorporated into this research. In NASA's technical reports and contractor reports, it is seen that the accuracy expected from a space navigation system (GPS-aided INS and some navigational aids, such as barometric altimeter, star tracker, etc.) is in the level of 50 meters per channel CEP [26,40].

1.3.3 Integration Kalman Filter Error and Measurement Models

Once the error models for the navigational systems are determined, the rest of the Kalman filter design is relatively straightforward. Navigational systems (GPS, INS, barometric altimeter, etc.) error models are named depending on their state number, for instance, an 11-state INS error model or a 30-state GPS error model. Each state within that specified number of states represents a specific sensor error, which we want the Kalman filter to estimate and compensate. The previous AFIT theses accomplished by Gray [10], Britton [6], and White [44], guided this part of literature review. Actually, Gray's thesis was the basis for the other two. The error models and measurement models they used have been analyzed and validated for almost ten years in AFIT theses. These models have proven their accuracy in many applications. However, to be 100% confident about the truthfulness of these models, they are checked once again and matched with the information available in Performance Accuracy (Truth Model/Error Budget) Analysis for the LN-93 Inertial Navigation Unit documentation [18]. The detailed INS, GPS, barometric altimeter error and measurement models will be delineated in Chapter 3 of this study.

1.3.4 The Chosen Launch Vehicle (Atlas IIAS) for Simulations

To accomplish this study, both the GPS and the INS data have to be generated and inserted into the simulation environment. A Fortran-based Flight Profile Generator (PROFGEN) generates the INS data for the rocket flight trajectory [29]. PROFGEN is capable of providing 53 different output variables to the user, and the user can choose the preferred number of output variables. To run this Profile Generator, one has to have the vehicle flight characteristics (take-off coordinates, vehicle initial heading, checkpoints, flight lengths, etc.) in hand and has to put those data into profile input (PROF_IN) file. Detailed information concerning PROFGEN and rocket flight profile generation procedures will be presented in Section 3.3 and 4.2 of this study. As a result of personal interview with Dr. Tregesser [39] from the aerospace department and the reviewed literature [15,19], the Atlas IIAS launch vehicle (Figure I-5) was chosen as the "optimum" in terms of applicability, cost, orbit specifications, flight statistics and thrust [15]. The operational Atlas II family is one of the largest commercial launch vehicles in the United States and is currently operating with a 100% mission success rate. As of June 19, 2001, Atlas IIAS has achieved 22 for 22 successful missions, for a total of 55 consecutive successful Atlas flights [15]. "Based on open-source information, the Federal Aviation Administration (FAA) estimates $ 90-105 million price range for commercial

Atlas IIAS [15]". Atlas IIAS is not only the optimum choice for simulation, but also it is the most realistic Low Earth Orbits (LEO) launch vehicle (see Figure I-4 for Earth orbits diagram). Furthermore, GPS data is extensively being used for space vehicles in LEO's (below 3000 km) [26]. Detailed information concerning Atlas launch vehicles, as depicted in Figure I-5, can be found on the Lockheed Martin Corporation's web site [19] and *International Reference to Space Launch Systems* (third edition), AIAA technical publications, 1999 [15].

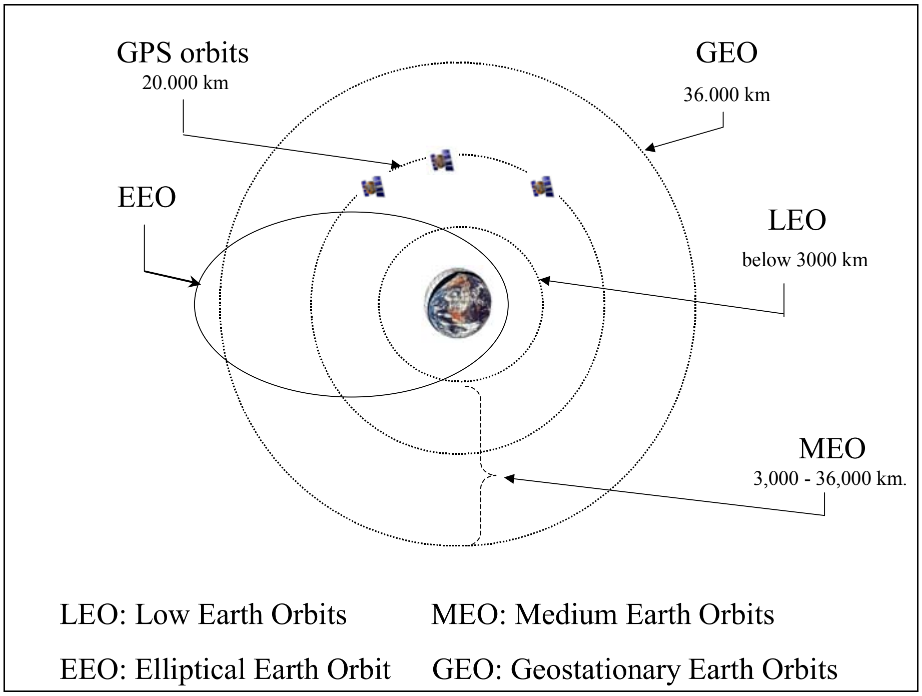


Figure I-4. Earth's Orbits [26]

Figure I-5. Atlas IIAS

1.4 Problem Statement

As mentioned previously, this study has started with examining the theses accomplished by former AFIT students Robert A. Gray [10], Capt. Ryan Britton [6], and 2nd Lt Nathan White [44]. These theses are concerned with accurate GPS/INS integrations for a precision landing or airborne navigation application, rather than for a satellite launch. It has seen that the truth models and filter design error models they used for INS and GPS have been utilized in AFIT for more than 8 years as validated models, and consequently the author decided to use these models in this research.

This research will concentrate on performances of different types and accuracies of GPS-aided INS systems and will make a tradeoff analysis between their cost, applicability, accuracy, implementation ease, etc. to assist TuAF's space project. The tradeoff study will be accomplished based on the performance analysis between 6 differently constructed cases. Each case represents a tightly coupled integration of these two types of GPS (standard single-receiver GPS, dual frequency P-code DGPS) with three grades of INS (0.4, 2.0, and 4.0 nm/hr CEP INS). These three grades of INS's represent military grade, commercial grade, and less expensive (possible MEMS in a few years) Inertial Navigation Systems. The constructed cases are shown in Table I-1.

In Cases I and II, the 0.4 nm/hr CEP INS will be integrated with two different accuracies of GPS data, namely standard single-frequency stand-alone GPS, and P-code DGPS. Cases III through IV and Cases V through VI achieve the same integrations for 2.0nm/hr CEP, and 4.0 nm/hr CEP INS's, respectively.

Table I-1. Case I-VI integration comparison

Case I	Case II	Case III	Case IV	Case V	Case VI
0.4 nm/hr CEP INS	0.4 nm/hr CEP INS	2.0 nm/hr CEP INS	2.0 nm/hr CEP INS	4.0 nm/hr CEP INS	4.0 nm/hr CEP INS
Single-Freq Standard GPS	Dual-Freq P-code DGPS	Single-Freq Standard GPS	Dual-Freq P-code DGPS	Single-Freq Standard GPS	Dual-Freq P-code DGPS
Barometric Altimeter	Barometric Altimeter	Barometric Altimeter	Barometric Altimeter	Barometric Altimeter	Barometric Altimeter

Note: Baro-Altimeter is used up to 80.000 feet. The Atlas launch vehicle reaches that altitude at 53 seconds into the simulation. When the total flight time of 1450 seconds is compared to that, it can be said that the baro-altimeter has no appreciable contribution to the overall performance.

The biggest challenge for this thesis was to establish a “tool”, a simulation program, to generate each of these 6 different cases in a computer environment. The Matlab-based [37] “MatSOFE” (quoted because the author refers to the version first created by former AFIT student William B. Mosle [27] in1993 and modified by Pamela L. Harms [12] in 1995 for Sverdrup Technology, Inc., TEAS Group) Monte Carlo performance analysis tool provided by AFIT/ENG was created for a special case. In this form, it was not comprehensive, designed for a simple specific problem with short flight hours, and especially was not convenient for the cases mentioned above in terms of truth and filter model state dimensions, “Pinson error model” definition, and the GPS data used. The author also received the original version MatSOFE in March 2001. Although it has some written codes for the purpose of GPS-aided INS applications, it was not taking any GPS measurement to update the integrated system. Because of that, most of the effort was spent to modifying the simulation code and writing new code, depending on needs for each integration case. By January 2002, the MatSOFE has reached its present, revised form. The current version of MatSOFE is capable of taking real GPS satellite vehicle ephemeris data by using the System Effectiveness Model for Windows Version 3.6.4. (WSEM 3.6) software [2] for measurement updates. Also, the UD covariance factorization filter principles, developed by Bierman and Thorton [20: 392], were incorporated for propagation and measurement update cycles. U-D factorization algorithm increases the numerical stability in the Kalman filter. The 39-state reduced truth model of Litton 93-state (LN-93) INS [18], which is the most common INS in the market and being used by the F-16 Fighting Falcon aircraft, is also integrated into current version. Detailed information about MatSOFE, error states, Pinson error model, real GPS satellite vehicle ephemeris data, WSEM software program and LN-93 INS definitions will be presented in next chapter.

Even though most of the reviewed subjects were familiar and studied during my AFIT academic education, many additional insights related to INS, GPS, and their integration techniques were gained in the literature review phase. Those insights will be used during both the simulation and tradeoff analysis phase.

1.5 Assumptions

All theses are limited by the assumptions made, and no research can be adequately evaluated unless these assumptions are clearly defined [27]. This section outlines the assumptions that have been made in this thesis [10].

i. All work is to be conducted through computer simulation. "Real world" data is neither collected nor used in the simulations. Instead, all "real world" measurements are accurately simulated. There is true satellite vehicle ephemeris data within the WSEM 3.6 software, but the actual GPS measurements themselves are simulated in the computer environment. The emphasis here is on the model representing the real world conditions in the filter design. The real world data used in the simulation is modeled with a full-order truth error state model. MatSOFE provides a Monte Carlo analysis of a GPS-aided INS Kalman filter design as seen in a realistic "truth model" environment. The full-order "truth models" and reduced-order "filter design models" are presented clearly in Chapter 3 of this thesis study.

ii. The INS will not have the usual 8-minute ground alignment. That feature could be implemented into MatSOFE, but because of the time constraints it is left as a future addition to MatSOFE.

iii. The LN-93 INS system has the natural ability of updating the vertical channel with barometric altimeter output. MatSOFE reflects that characteristic as well. The use of barometric altimeter is included in the modeling of the system, so that the INS platform is assumed to be stabilized with a barometric altimeter. Without a vertical channel aiding device, the INS vertical channel is unstable. In fact, using a barometric altimeter as a vertical channel aid is not the only option, but it is the cheap and the most commonly used method. Unlike GPS aiding, it is not vulnerable to jamming or spoofing.

iv. Also, the barometric altimeter is assumed to be good up to 80,000 feet altitude. Rapid altitude change during the launch is expected, and at 80,000 feet the baro-altimeter will be turned off (in simulations, that happens almost at 53 seconds).

v. The computer-simulated Atlas IIAS launch vehicle profile is generated by using the software "PROFGEN" [29]. PROFGEN has been developed to work with MatSOFE [12] to provide the necessary data files to simulate rocket dynamics. The actual total flight hour for the Atlas is just about 4000 seconds up to perigee entrance. For simplicity and saving computer memory, the flight lasts until second main engine cut off (MECO2). In fact, at about 500 seconds, the rocket is on the orbit and after that point it makes the perigee entrance maneuver. The MatSOFE and PROFGEN are presented in Chapter 3 of this study in details.

vi. In the constructed cases, it is assumed that there will be no GPS measurement outage or failure, because in a space vehicle INS and GPS systems will always be redundant. In reality, from the GPS satellite vehicles' point of view (i.e., with respect to the geometry of the satellites and receiver onboard the rocket), GPS data will almost always be available for LEO space vehicles.

vii. A 4.0 nm/hr CEP INS is assumed as representative of a “micro electromagnetic system (MEMS)” INS. Actually nowadays, their accuracy is almost 6.0 nm/hr CEP, but when this thesis is finished, or shortly thereafter, it is probable to have MEMS INS units with 4.0 nm/hr CEP accuracy. 0.4 nm/hr CEP and 2.0 nm/hr CEP INS’s represent current higher and medium accuracy INS’s respectively.

viii. A sample period of one second has been chosen for the EKF. As a matter of fact, the decision to use a one-second sample period is based primarily on the typical availability of the GPS measurement in the real world. The sample period refers to how often the GPS measurements will be brought into the EKF. The United States Coast Guard Navigation Information Service Bulletin Board distributes post-fit ("precise") GPS orbital ephemeredes ("orbits") computed by National Oceanic and Atmospheric Administration’s (NOAA) National Geodetic Survey (NGS) in two formats. These formats are known as EF18 and SP3. EF18 is a binary file, while SP3 is its ASCII equivalent [32]. Usually, the GPS EF18 and SP3 almanac files used in this study are stored ever 15 minutes or at a higher sample period, and they need to be converted to a different format to alter the sample period. The one-second period of GPS data used in this thesis is obtained by means of the software used to convert EF18 data format into SP3 data format. Though faster GPS outputs are available via “utility software” [32], a one second sample period is chosen as a good, representative design choice.

ix. The computer simulations have been developed using a program called MatSOFE, which has been updated and upgraded by the author. MatSOFE is the Matlab [37] version of the Fortran-based Multi-mode Simulation for Optimal Filter Evaluation (MSOFE) [30]. MSOFE is well-established Air Force software to develop and test linear and extended Kalman filter algorithms.

x. The sv ephemeris data using WSEM 3.6 software program was downloaded from the ARINC, Inc. web page[2]. The ephemeris data is then post-processed by the utility software programs obtained from NOAA’s National Geodetic Survey web page [32] .

xi. Only four GPS satellites were modeled in MatSOFE. In real-world applications, depending on the receiver position on Earth, most likely 6-8 satellites would be visible at a certain time epoch. Four SV are always available. The SEM 3.6 software selects the four best satellites available at a given time based on an average Position Dilution of Precision (PDOP) less than 1.3 and these satellites are used without interruptions.

xii. The MatSOFE runs are conducted using 15-run Monte Carlo analyses. While a larger batch size for the Monte Carlo analysis would be preferable, this value has been chosen to keep the computational burden of the thesis within reasonable bounds, while maintaining adequate confidence that the resulting sample statistics properly reflect the true statistics [10].

xiii. Flight segment durations and accelerations of the real launch vehicle were accurately incorporated in order to generate the Atlas profile realistically. It is also assumed to be launched from Cape Canaveral AS, SLC-41($28.5^0 N, 81.0^0 W$). Available orbit inclinations for that location are $28.5^0 - 55.0^0$ and polar 90.0^0. The author picked 28.5^0 inclination orbit as a fair choice.

xiv. Taylor series approximations truncated at first order are used for linearizing nonlinear equations in the Space Navigation System Model (SNSM) filter. See Figure I-6 for SNSM computer model. The SNSM is presented in detail Section 3.5 of this study.

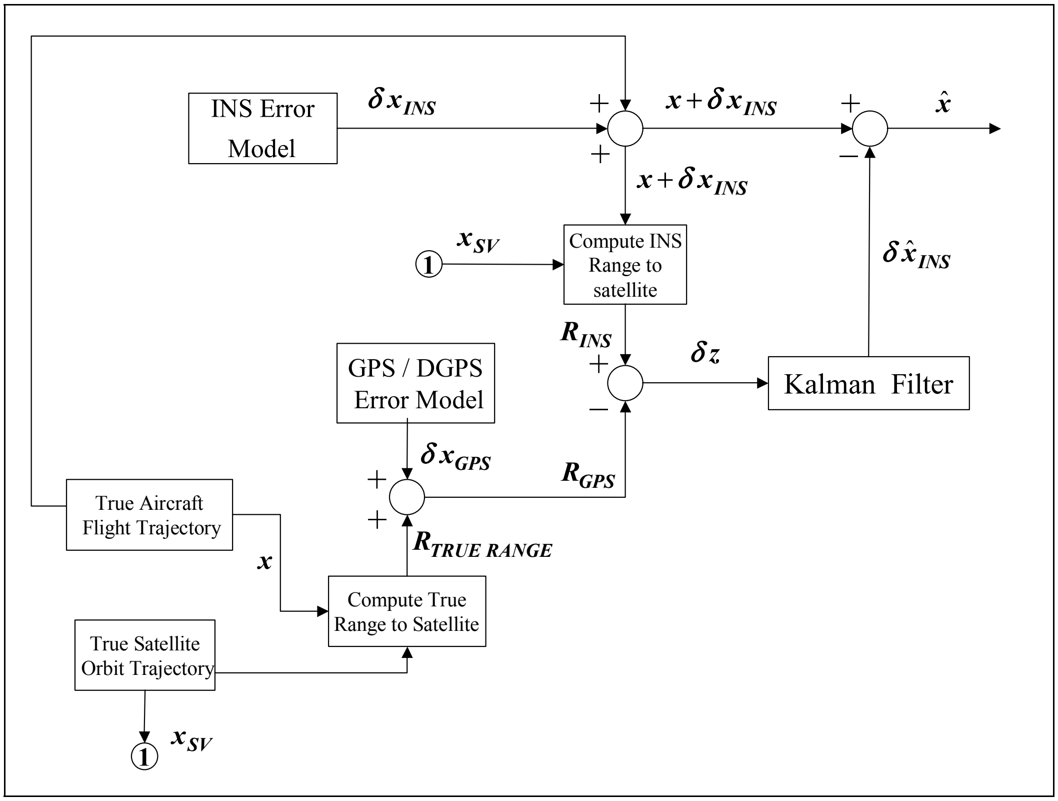


Figure I-6. Space Navigation System Model (SNSM) Simulation

1.6 Scope

This research will focus on a simulation-based tradeoff analysis study of GPS/DGPS integrated with a baro-inertial navigation system for the Atlas IIAS launch vehicle. The study of designing a control algorithm for a launch vehicle could also be studied, but it is considered as the follow-on step.

To accomplish the tradeoff performance analysis of a GPS-aided INS system, one has to make Monte Carlo simulations and/or covariance analyses. MSOFE (Multimode Simulation for Optimal Filter Evaluation (MSOFE) software [30]) is restricted for use by U.S. government agencies and their U.S. contractors. In order to accomplish the analyses, the Matlab-based [37] “MatSOFE” (first created by former AFIT student William B. Mosle [27] in 1993 and modified by Pamela L. Harms [12] in 1995 for Sverdrup Technology, Inc., TEAS Group and finally upgraded and modified by the author) is used as an evaluation software. Not being a compiled language, like Fortran or C, MATLAB causes MATSOFE to be almost eight times slower than MSOFE. This means long run times. (In the simulation phase, it is seen that a 15-Monte Carlo-run simulation takes 8 to 20 hours, depending on the processor speed of the computer being used.) When this constraint is taken into account, accomplishing only the navigational aspect of a space guidance, navigation, and control system becomes a more reasonable scope for this research.

The tasks involved in this research are as follows:

1. Review prior AFIT theses of Negast [33], Mosle [27], Gray [10], Britton [6], and White [44]. Investigate the INS/GPS integrations used in their research.

2. Conduct a literature review concerning the latest innovations in GPS range resolution and recent INS/GPS integration techniques.
3. Study, and if necessary restructure, the current “truth model”: a complete, complex mathematical model that portrays true system behavior very accurately. Justify its sustained use and, update the GPS/DGPS information.
4. Make any adjustments that are required to yield an accurate, validated model.
5. Further validate the truthfulness of the present “truth model” by comparing it to Litton-93 documentation [18].
6. Start simple. First understand the basic principles of the provided MatSOFE software.
    a. Fill the gaps of the provided MatSOFE. Add/rewrite the necessary m-files (matlab script files).
    b. First generate The KC-135 Tanker aircraft flight profile by using the provided PROF_IN file (the input file of PROFGEN describing the desired maneuvers, flight segments, accelerations, etc of the vehicle that was used in previous research) in Gray’s thesis [10].
    c. Make necessary changes in the simulation code in order to take “true” GPS measurement data evaluated by WSEM 3.6 software into the integrated system. Put the GPS satellite vehicle ephemeredes data into a format that Matlab can load it into simulation.
    d. Incorporate the UD covariance factorization filter principles to propagation and measurement update cycles in order to increase the numerical precision and stability in the Kalman filter.

e. Merge FLY_OUT (PFOFGEN output in *formatted* form) and SV_data (final form of processed GPS satellite vehicle (SV) measurements in Litton ECEF coordinates) into a single INPUT file for MatSOFE.

7. Compare MatSOFE results to Mosle's, Gray's and Britton's MSOFE results [6,10,27] to demonstrate that the upgraded MatSOFE is a reliable and easy-to-learn navigation systems performance evaluation tool.
8. Carry out a literature review to decide upon an optimum performance launch vehicle to be used in the simulation. Clearly identify the characteristics of launch vehicle to generate a realistic flight simulation.
9. Perform a Monte Carlo analysis for each suggested case.
10. "Tune" the filter to provide the best possible performance.
11. Analyze each case and compare the results of one to another.

1.7 Summary

This chapter has given a brief overview of the thesis plan to develop an integrated GPS/DGPS, INS, and Barometric Altimeter integrated system for navigation of a rocket launching a satellite. The background for the necessity of such a system, the various system integrations, past research, the project scope, and all assumptions have been presented. The reference frames used in this research, as well as the INS, GPS, DGPS, and barometric altimeter subsystems are presented in Chapter 2, along with a discussion of Kalman filter algorithms. Chapter 3 introduces MatSOFE and PROFGEN, and presents the SNSM system and filter models. Chapter 4 presents the results and analysis of the SNSM. Chapter 5 presents conclusions and recommendations.

II. Theory

2.1 Introduction

The background presented in this section includes the basic theory on a ring laser gyro (RLG), an Inertial Navigation System (INS), a Global Positioning System, a Differential Global Positioning System (DGPS), a barometric altimeter and a radar altimeter. Fundamental Kalman filter and extended Kalman filter (EKF) theory will also be discussed. More information on Kalman filter development and uses may be found in [20,21,22]. Deterministic and stochastic processes used in this section will be presented in roman typeface. Vectors will be displayed in bold-faced type, **x**, and scalars will be shown in normal type, x. Matrices will be displayed in bold-faced upper case, **X.** A particular realization of a variable will be displayed in italics, *x*. The credit for the development of large portions of this chapter belongs to Gray [10] and Britton [6].

2.2 Ring Laser Gyro (RLG) Strapdown INS

A ring laser gyro (RLG) strapdown INS is a precision navigation system, which provides navigation information (position, velocity, attitude) of a vehicle using inertial sensors, such as gyroscopes and accelerometers [17: 526]. " *Inertial navigator* is a self-contained, dead-reckoning, navigation aid using inertial sensors, a reference direction, an initial and/or subsequent fixes to determine direction, distance, and speed; single integration of acceleration provides speed information and a double integration provides distance information [14]."

A strapdown system is mechanized by mounting three gyros and three accelerometers directly to the vehicle for which the navigation function is to be provided. More than three of each can be used to provide enhanced reliability through redundancy (especially in a space navigation system). A digital computer is used to keep track of the vehicle's attitude with respect to a reference frame, based on the information from the gyros. This enables the computer to provide the coordinate transformation necessary to coordinatize the accelerometer outputs in a desired computational reference frame, such as East-North-Up (ENU) or wander azimuth (WA) [6].

The strapdown system is a specific type of inertial navigation system characterized by lack of a gimbal support structure [5]. An advantage of strapdown systems over the gimbaled is that a strapdown system has no moving platform keeping a "stable element" level. Without the moving parts, the system is less prone to failures and cheaper to build. Also, when gyro failures occur in a strapdown system, the gyros may be replaced; the entire inertial measurement unit (IMU) would have to be replaced in a gimbaled system. The disadvantage with a strapdown system is that the platform is physically strapped to the aircraft body. This forces the gyroscopes, accelerometers, and strapdown computer algorithms to be rugged enough to maintain integrity in whatever harsh dynamic environment the aircraft may encounter. The sensors must also provide *precise* measurements over a substantially larger range of values than would a similar sensor on a gimbaled platform. [6,10]

RLG construction basically consists of an optical cavity, a laser device, three or four mirrors, a prism, and a pair of photo detectors. To provide navigation information, the RLG detects and measures angular rates by measuring the frequency difference between two counter-rotating (laser) beams. The two laser beams circulate in a ring-shaped optical cavity at the same time. The beams are reflected around the optical cavity using mirrors. The resonant frequency of a contained laser beam is a function of its optical path length. Since the path traveled by each of the beams is identical when the gyro is at rest, the two laser beams have the same frequencies under these conditions. If the cavity is rotating in an inertial sense, the propagation times of the two light beams are different. The difference in the propagation time reveals itself in the form of a phase shift between the two beams, and a pair of photo detectors detects the phase shift. The magnitude of the phase shift provides a direct indication of the angular rate of rotation of the instrument with respect to inertial space [6,10]

2.3 Barometric Altimeter

The inertial sensors, like accelerometers and gyros, used in INS have some errors with stochastic properties. These errors tend to increase in time (long-term instability) with unbounded position error growth, and require occasional compensation [17: 526]. That long-term instability results in unbounded error growth in vertical position and velocity. By means of aiding the vertical channel with a barometric (or other type of) altimeter, the instability may be controlled [6,3].

The barometric altimeter is probably the simplest way to measure the altitude of an aircraft. The pressure of the Earth's atmosphere decreases as height above the earth increases. Barometric altimeters provide altitude information based on the pressure differences. Barometric altimeters are most inaccurate when ascending or descending at rapid rates but are relatively low in cost [6].

As mentioned earlier in the *Assumptions* section of Chapter I, the barometric altimeter is assumed to be fully functional up to 80,000 feet altitude. After that altitude, the integrated system is GPS-aided INS alone. So, a rapid altitude change is expected during the launch, and after the vehicle passes through 80, 000 ft, some degradation in vertical channel precision is anticipated.

2.4 Global Positioning System (GPS)

GPS is a passive, space-based, universal and accurate source of navigation information (three-dimensional position and velocity) and time system that has three major segments as seen in Figure II-1: Space segment, Control segment, and User segment [6,10,25,44].

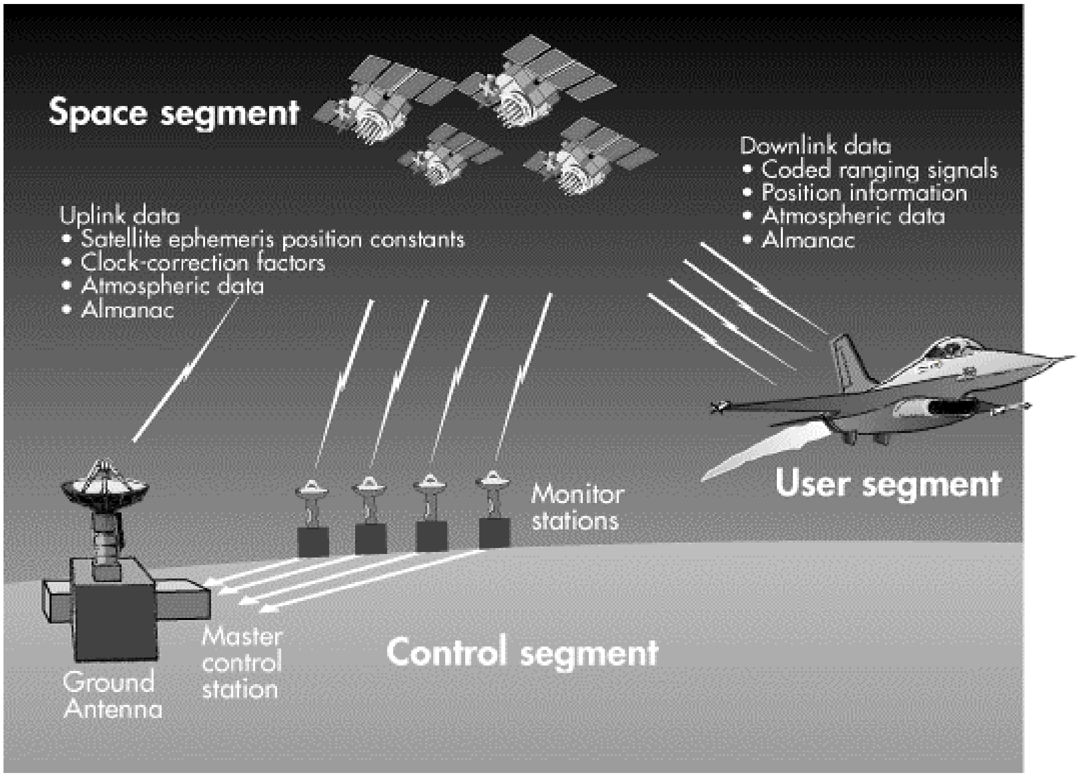


Figure II-1. GPS Major Segments

2.4.1 GPS Space Segment

The GPS *Space* Segment is composed of 24 or more active satellites in six orbital planes. See Figure II-2. The satellites operate in nearly 20,200 km (10,900 NM) orbits at an inclination angle of 55 degrees and with ~ 12-hour period. The spacing of satellites in orbit is arranged so that a minimum of five satellites (more likely 6-8 satellites) will be in view to users worldwide, with a position dilution of precision (PDOP) of six or less. One of the main properties of GPS is the achievable precision, which depends not only on the accuracy of the pseudorange measurements but also on the geometry of the transmitter and the receiver. The accuracy of measurements can be transformed by the geometry from a pseudorange accuracy into a positioning accuracy. This influence is described by the Dilution of Precision (DOP) factors. Appropriate significance is given to the field of view of the GPS satellites [1, 42]. PDOP is a measure of the error contributed by the geometric relationships of the GPS satellites as seen by the GPS receiver [9]. PDOP is mathematically defined as:

$$PDOP = \sqrt{(\sigma_x^2 + \sigma_y^2 + \sigma_z^2)} \tag{2.1}$$

where σ_x^2, σ_y^2 and σ_z^2 are the variances of the *x, y,* and *z* pseudorange measurement-based position error, respectively. A definition for pseudorange measurement is given in Section 2.4.3. Each satellite transmits on two L band frequencies, L1 (1575.42 MHz) and L2 (1227.6 MHz). L1 carries a precise (P) code and a coarse/acquisition (C/A) code. L2 carries only the P code. A navigation data message containing the important information about each satellite is overlaid on these codes. The same navigation data message is carried on both frequencies [10,34].

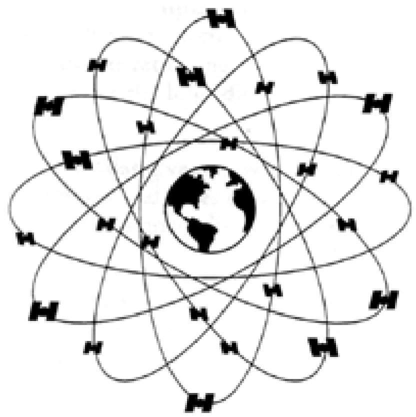


Figure II-2. GPS Orbital Planes

2.4.2 GPS Control Segment

The *Control* Segment has six USAF monitor stations, three of which have uplink capabilities. See Figure II-3. The monitor stations use a GPS receiver to track all satellites in view passively and thus accumulate ranging data from the satellite signals. The information from the monitor stations is processed at the Master Control Station (MCS) to determine satellite orbits and to update the navigation message of each satellite. This updated information is transmitted to the satellites via the ground antennas, which are also used for transmitting and receiving satellite control information. The specific functions of the Control Segment are:

- monitor satellite orbits and health
- maintain GPS time
- predict satellite ephemeredes and clock parameters
- update satellite navigation messages, and
- command small maneuvers of satellites to maintain orbit.

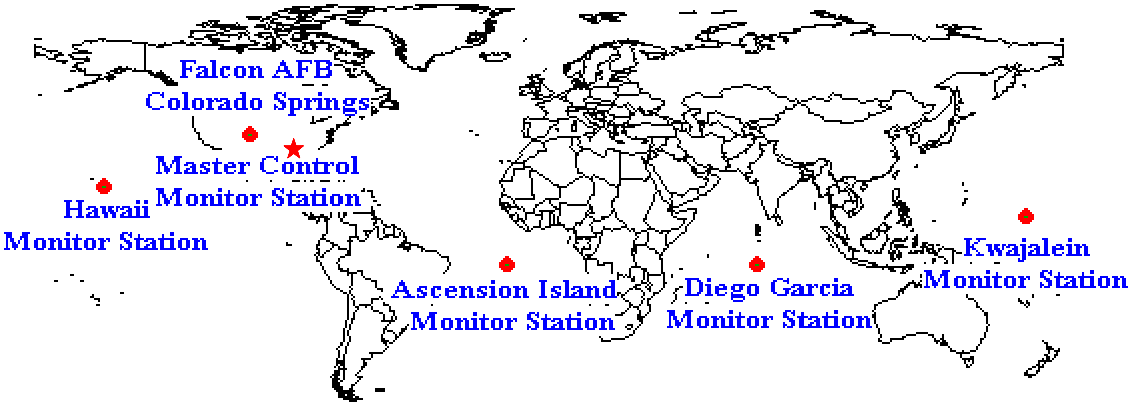


Figure II-3. GPS master control & monitor stations

2.4.3 GPS User Segment

The *User* segment consists of an antenna and receiver processors that provide positions, velocity and precise timing to the respective user. Computing the user's positional information typically requires simultaneous solution of the following four nonlinear position equations [6]:

$$\begin{aligned}(x_1-u_x)^2+(y_1-u_y)^2+(z_1-u_z)^2&=(R_1-C_B)^2\\(x_2-u_x)^2+(y_2-u_y)^2+(z_2-u_z)^2&=(R_2-C_B)^2\\(x_3-u_x)^2+(y_3-u_y)^2+(z_3-u_z)^2&=(R_3-C_B)^2\\(x_4-u_x)^2+(y_4-u_y)^2+(z_4-u_z)^2&=(R_4-C_B)^2\end{aligned}\tag{2.2}$$

where the pseudorange, $R_{i=1,2,3,4}$ to each satellite is defined as

$$\begin{aligned}R_1&=c\Delta t_1\\R_2&=c\Delta t_2\\R_3&=c\Delta t_3\\R_4&=c\Delta t_4\end{aligned}\tag{2.3}$$

and *c* is speed of light

$\Delta t_{i=1,2,3,4}$ is signal transmit time as measured by the receiver

$x_{i=1,2,3,4}$, $y_{i=1,2,3,4}$, $z_{i=1,2,3,4}$ are respective i-th satellite positions

u_x, u_y, u_z is the user position the GPS user equipment is computing numerically and recursively

C_B is the user clock bias (user equipment solves), as expressed in equivalent range offset in Equation (2.2).

Normally the user equipment needs to acquire and maintain lock on at least four satellites in order to compute a 3-D position fix u_x, u_y and u_z [24] and the clock bias C_B. The GPS pseudorange between the user and each satellite is computed based on knowledge of time (the master GPS clock) and the unique signal format, which is broadcast by each satellite. Part of the problem is that the user clock is not identical to the master clock. Once the four pseudoranges are known, a recursive algorithm is solved to compute the user's position [24].

2.5 Differential GPS (DGPS)

Differential GPS is a method of achieving higher GPS positioning accuracies in a local area. The basic principle behind DGPS is based on the fact that pseudorange measurement errors are correlated between two nearby receivers tracking a given satellite. A single DGPS reference station at a known location can compute a range error correction for each GPS satellite in view. The error corrections may then be broadcast to users in the vicinity of the reference receiver. A user can typically improve 3-D positioning accuracy from 10 meters (horizontal DRMS) for a standard GPS down to the

0.1-1 meter level by applying the corrections to the signals received [34]. The accuracies do decrease as the distance between the user and the reference station increases.

Work accomplished by William Negast [33] at AFIT has demonstrated the effectiveness of using differential corrections to increase GPS precision. He was able to eliminate the satellite vehicle (SV) clock error from each pseudorange measurement, and the SV position errors were nearly eliminated. The atmospheric propagation errors can be almost totally eliminated when the two GPS receivers are within 200 miles of each other [33]. The work by Negast [33] provided the new DGPS error model used in this effort. Detailed equations for DGPS and explanations for the errors removed from standard GPS after the application of DGPS corrections are presented in Section 3.6.2.2

2.6 Reference Frames

A navigation "solution" has significance only if the corresponding frame in which the solution is expressed is clearly understood [36]. While the preceding statement may seem obvious, it cannot be overemphasized. Consider that a typical INS "owner's manual" defines *earth* frame, *true* frame, *computer* frame, *platform* frame, *sensor* frame, *accelerometer* frame and the *body* frame [36,5]. From a student's perspective this may at first be overwhelming, but to make matters worse, another INS vendor may well define every frame mentioned above, such as "*earth* frame" in an entirely *different* manner! Therefore, the frames used in this project and the coordinate transformations will briefly be discussed.

Two variants of a five-frame set (inertial, earth, geodetic, wander-level, and body) are implemented in PROFGEN (See Figure II-4). One variant is called Standard

Navigation Units *(SNU)* and is the frame set used in the USAF. The other variant, which is called World Geodetic System (*WGS),* was added to PROFGEN in 1999. In this research the WGS reference coordinate system is preferred in PROFGEN, but the Litton reference coordinate system is used in MatSOFE, because the LN-93 INS mechanization has been defined in this reference system [18]. For instance, the true GPS ephemeredes data are in the WGS 84 "earth-centered-earth-fixed" (ECEF) reference frame but the LN-93 INS uses the "Litton ECEF" reference system. As a matter of fact, the transformation from WGS to Litton reference system is accomplished by only changing the axis orientation. The transformation from WGS ECEF to Litton ECEF is as follows:

$$C_{Litton\_WGS} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.4}$$

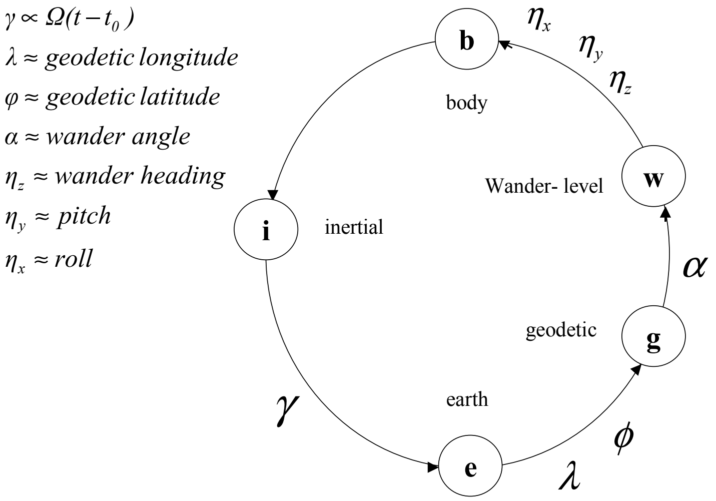

$\gamma \propto \Omega(t-t_0)$
$\lambda \approx$ geodetic longitude
$\varphi \approx$ geodetic latitude
$\alpha \approx$ wander angle
$\eta_z \approx$ wander heading
$\eta_y \approx$ pitch
$\eta_x \approx$ roll
b
body
η_x
η_y
η_z
w
Wander- level
i
inertial
α
g
geodetic
earth
γ
e
ϕ
λ

Figure II-4. Circle Diagram Relating PROFGEN Frames [29]

The GPS data and INS data are merged into a single input m-file named *FLIGHT* and then they are transferred into the Litton ECEF reference frame set in this research.

2.6.1 *Inertial Frame* (x^i , y^i , z^i)

The inertial frame for this research is an orthogonal, right-handed coordinate system; its origin is coincident with the earth's center-of-mass and the frame is oriented as follows. The (x^i, z^i) plane lies in the earth's equatorial plane and does not rotate with respect to the fixed stars. The y^i axis projects from the earth's center-of-mass directly through the North pole.The inertial frame is depicted by the [x^i, y^i, z^i] frame shown in Figure II-5 [10].

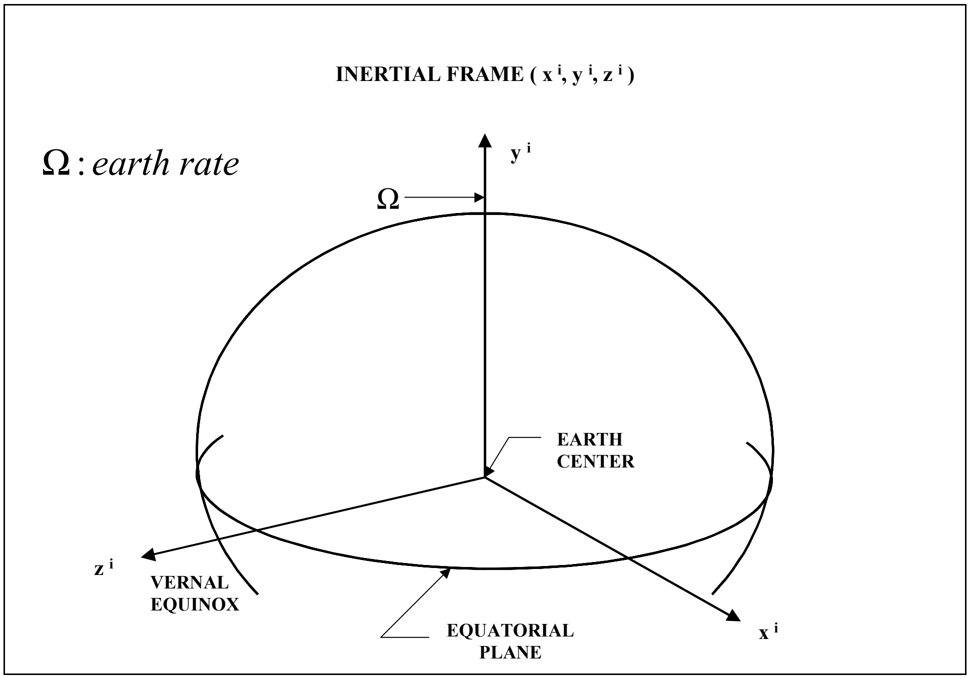


Figure II-5. Inertial Frame

2.6.2 *Earth Frame* (x^e , y^e , z^e)

The earth frame or "earth-centered-earth-fixed" (ECEF) frame is an orthogonal, right-hand coordinate system; its origin is coincident with the earth's center-of-mass, with the (x^e, z^e)plane located in the earth's equatorial plane. The z^e axis is aligned with the Greenwich meridian and rotates at exactly the earth rate, Ω, about the y^e axis, which projects from the earth's center-of-mass directly through the North pole. The Earth frame is depicted as [x^e, y^e, z^e] in Figure II-6 [10].

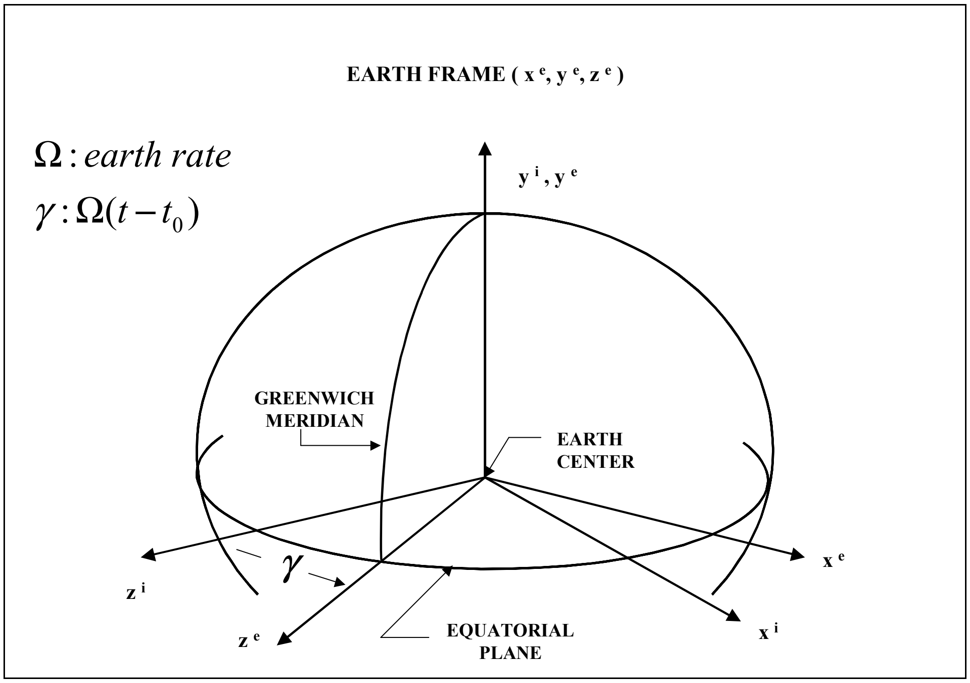


Figure II-6. Earth Frame

2.6.3 *Geographic Frame* (x^g , y^g , z^g) = (E, N, U)

The geographic frame or “local-level” frame is an orthogonal, right-handed coordinate system; its origin is at the location of the INS (or the user), and its axes are aligned with the East, North and Up directions [E, N, U]. The geographic frame remains perpendicular to the earth’s surface with respect to the earth’s gravity field as the user moves over the Earth. The geographic frame is depicted as either [x^g, y^g, z^g] or [E, N, U] in Figure II-7.

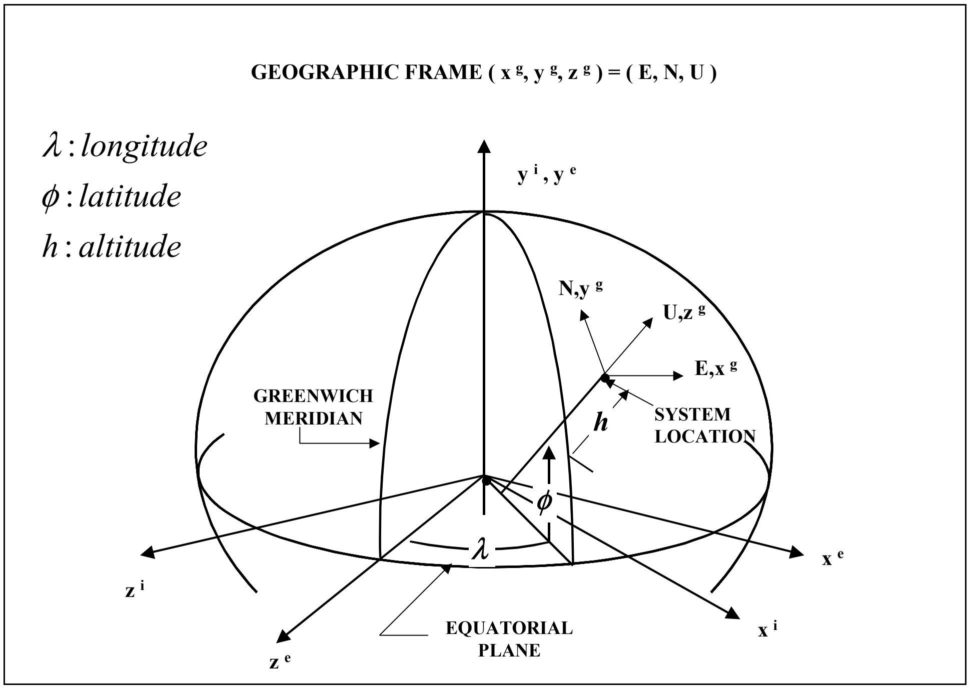


Figure II-7. Geographic Frame

2.6.4 *Navigation Frame* (x^n , y^n , z^n)

The navigation frame or “local-level-wander-azimuth” frame is an orthogonal, right-hand coordinate system; its origin is at the location of the INS (or the user). It is the intended coordinate frame for used by the navigation system internal computations. This frame coincides with the geographic frame when the wander angle, α, equals 0°. The wander angle is a computed angle between a “scribe mark” on a wander azimuth angle platform and North. For gimbaled systems, the platform is purposely not commanded to seek North, due to the high platform angular rates that this would require in polar regions, with resulting performance degradation [6,3,36]. The navigation frame is denoted as [x^n, y^n, z^n] shown in Figure II-8.

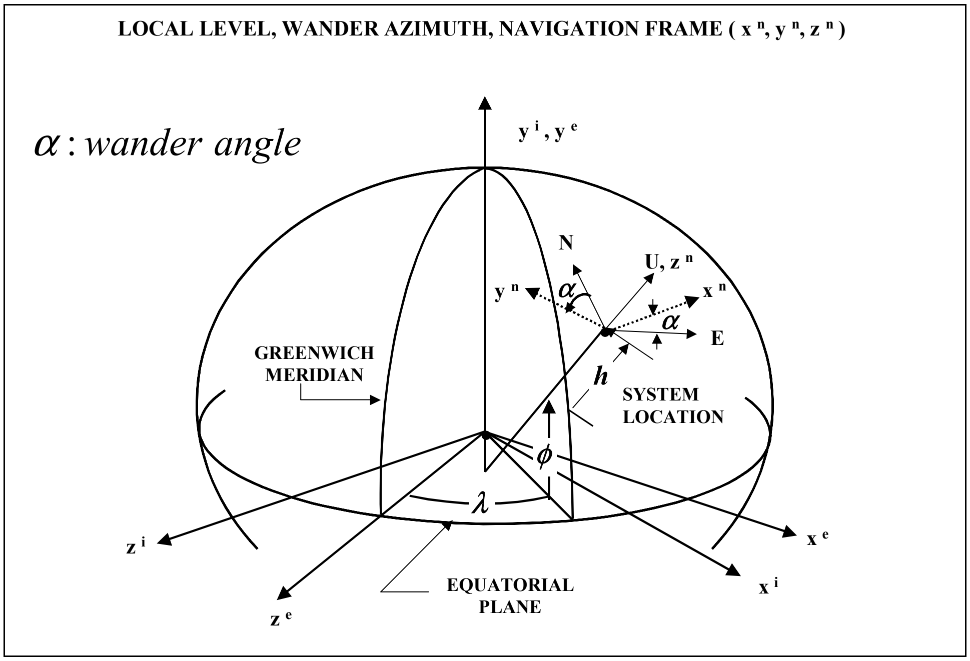


Figure II-8. Navigation Frame

2.6.5 *Body Frame* (x^b , y^b , z^b)

The body frame is an orthogonal, right-hand frame; its origin is at the vehicle (i.e., aircraft) center-of-mass. Its axes are the vehicle's roll, pitch, and yaw axes. The x^b axis points in the forward direction, along the roll axis; the y^b axis points to the right (starboard side) of the vehicle, perpendicular to the roll axis, but along the pitch axis; and the z^b axis is positive out the underside of the vehicle. The body frame is denoted as [x^b, y^b, z^b] and is shown in Figure II-9.

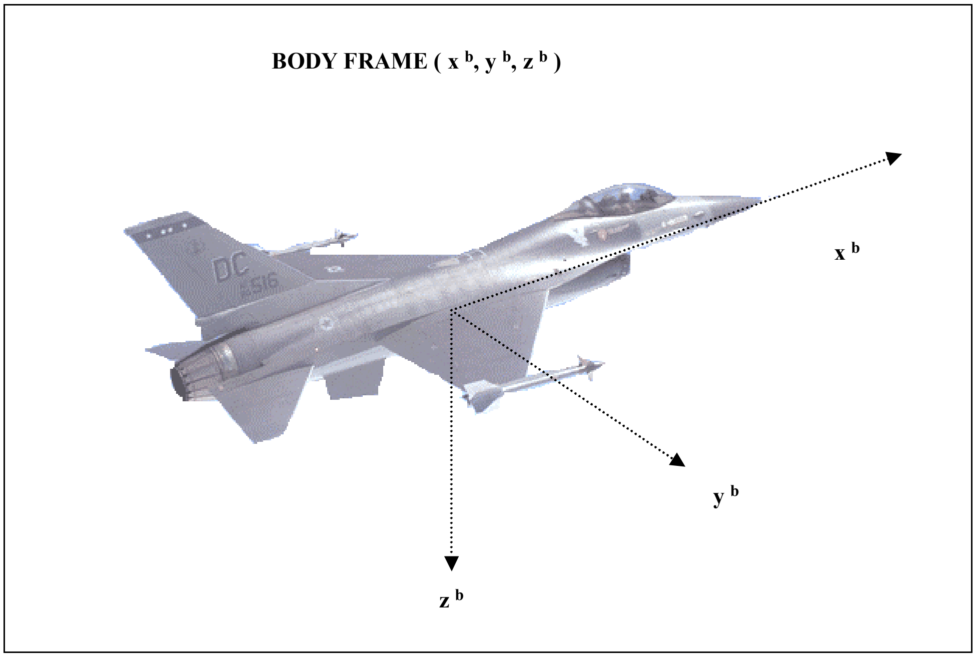


Figure II-9. Body Frame

2.7 Reference Frames Transformations

The RLG INS modeled in this thesis uses the navigation frame or "local-level-wander-azimuth" frame. It is often necessary to express vectors such as position, attitude, velocity or acceleration in terms of several different reference frames. As an example, the INS modeled in this thesis also outputs position error in terms of an *error-angle* vector, $[\delta\theta_x, \delta\theta_y, \delta\theta_z, \delta h]^T$, where $\delta\theta_x$ is the error angle about the local level x^g (or E) axis, $\delta\theta_y$ is the error angle about the local level y^g (or N) axis, $\delta\theta_z$ is the error about the local level z^g (or U) axis, and δh is the altitude error [3,18]. Even though this definition is clear, if the error-angle vector is to have *physical* meaning, it must be transformed into a vector in *navigation* error terminology, $[\delta\phi, \delta\lambda, \delta\alpha, \delta h]^T$, where $\delta\phi$ is the error in latitude, $\delta\lambda$ is longitude error, $\delta\alpha$ is alpha (wander) angle error and δh is again the altitude error.

A *transformation matrix*, $C_{Error\,Angle}^{Navigation\ Error}$, permits compact transformation of the error-angle vector into an equivalent expression in navigation error space. The transformation matrix, $C_{Error\,Angle}^{Navigation\ Error}$ is shown below in Equation (2.5) [3,36]:

$$C_{Error\,Angle}^{Navigation\,Error} = \begin{bmatrix} -\cos\alpha & \sin\alpha & 0 & 0 \\ \sin\alpha\sec\phi & \cos\alpha\sec\phi & 0 & 0 \\ -\sin\alpha\tan\phi & -\cos\alpha\tan\phi & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

ϕ = latitude
α = wander angle

Other transformations are as follows (from [6]):

2.7.1 Inertial Frame to Earth Frame, C_i^e

$$C_i^e = \begin{bmatrix} \cos\Omega t & 0 & -\sin\Omega t \\ 0 & 1 & 0 \\ \sin\Omega t & 0 & \cos\Omega t \end{bmatrix} \tag{2.6}$$

Note that, when $t = 0$, x-axis of ECEF frame equals x-axis of inertial frame

2.7.2 Earth Frame to Geographic Frame, C_e^g

$$\begin{aligned} C_e^g &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\lambda & 0 & -\sin\lambda \\ 0 & 1 & 0 \\ \sin\lambda & 0 & \cos\lambda \end{bmatrix} \\ &= \begin{bmatrix} \cos\lambda & 0 & -\sin\lambda \\ -\sin\phi\sin\lambda & \cos\phi & -\sin\phi\cos\lambda \\ \cos\phi\sin\lambda & \sin\phi & \cos\phi\cos\lambda \end{bmatrix} \end{aligned} \tag{2.7}$$

where

λ = longitude
ϕ = latitude
α = wander angle

2.7.3 Earth Frame to Navigation Frame, C_e^n

$$\begin{aligned} C_e^n &= \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\lambda & 0 & -\sin\lambda \\ 0 & 1 & 0 \\ \sin\lambda & 0 & \cos\lambda \end{bmatrix} \\ &= \begin{bmatrix} \cos\alpha\cos\lambda - \sin\alpha\sin\phi\sin\lambda & \sin\alpha\cos\phi & -\cos\alpha\sin\lambda - \sin\alpha\sin\phi\cos\lambda \\ -\sin\alpha\cos\lambda - \cos\alpha\sin\phi\sin\lambda & \cos\alpha\cos\phi & \sin\alpha\sin\lambda - \cos\alpha\sin\phi\cos\lambda \\ \cos\phi\sin\lambda & \sin\phi & \cos\phi\cos\lambda \end{bmatrix} \end{aligned} \tag{2.8}$$

where

λ = longitude
ϕ = latitude
α = wander angle

2.7.4 Geographic Frame to Navigation Frame, C_g^n

$$C_g^n = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

where

α = wander angle

2.7.5 Geographic Frame to Body Frame, C_g^b

$$C_g^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\rho & \sin\rho \\ 0 & -\sin\rho & \cos\rho \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.10}$$

$$= \begin{bmatrix} \cos\theta\sin\psi & \cos\theta\cos\psi & \sin\theta \\ \sin\rho\sin\theta\sin\psi + \cos\rho\cos\psi & \sin\rho\sin\theta\cos\psi - \cos\rho\sin\psi & -\sin\rho\cos\theta \\ \cos\rho\sin\theta\sin\psi - \sin\rho\cos\psi & \cos\rho\sin\theta\cos\psi + \sin\rho\sin\psi & -\cos\rho\cos\theta \end{bmatrix}$$

where

ρ = roll
θ = pitch
ψ = geographic heading

Actually, the rightmost 3x3 matrix of the C_g^b transformation first transforms from ENU orientation to NED, then the *Navigation-to-Body* transformation is accomplished.

2.7.6 Navigation Frame to Body Frame, C_n^b

$$C_n^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\rho & \sin\rho \\ 0 & -\sin\rho & \cos\rho \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi_P & \sin\psi_P & 0 \\ -\sin\psi_P & \cos\psi_P & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.11}$$

$$= \begin{bmatrix} \cos\theta\sin\psi_P & \cos\theta\cos\psi_P & \sin\theta \\ \sin\rho\sin\theta\sin\psi_P + \cos\rho\cos\psi_P & \sin\rho\sin\theta\cos\psi_P - \cos\rho\sin\psi_P & -\sin\rho\cos\theta \\ \cos\rho\sin\theta\sin\psi_P - \sin\rho\cos\psi_P & \cos\rho\sin\theta\cos\psi_P + \sin\rho\sin\psi_P & -\cos\rho\cos\theta \end{bmatrix}$$

where

ρ = roll
θ = pitch
ψ_P = platform heading ($\psi_P = \psi + \alpha$)

2.8 Kalman Filter Theory

2.8.1 What is a Kalman Filter?

A Kalman filter is simply an optimal recursive data processing algorithm [20] that can be shown to be optimal by essentially any standard, given the appropriateness of several underlying assumptions. These assumptions are that the system in question can be adequately modeled as linear with white, Gaussian system and measurement noises.

One aspect of the word "optimal" is that the Kalman filter can incorporate all information (measurements) provided to it [20]. It processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest with use of (from [20]):

- Knowledge of the system dynamics and measurement device characteristics.

- The statistical description of the system noises, measurement errors and uncertainty in the dynamics models.
- Any available information about initial conditions of the variables of interest.

For example, to determine the velocity of an aircraft, one could use a Doppler radar, or the velocity indications from an inertial navigation system, or the pitot and static pressure and relative wind information in the air data system. Rather than ignore any of these outputs, a Kalman filter could be built to combine all this data and knowledge of the various systems' dynamics and sensor attributes to generate an overall best estimate of velocity. Another way a Kalman filter is optimal is that it obtains the best estimate of desired quantities from data provided by a noisy environment. Here the word "optimal" means that the Kalman filter minimizes mean squared errors and is optimal with respect to essentially all other criteria, and it does so recursively. The word *recursive* means that, unlike certain data processing concepts, the Kalman filter doesn't require all previous data to be kept in storage and reprocessed every time new measurements are taken.

To see how a Kalman filter works, a simple example taken directly from [20] will be presented. It is included here because it helped the author understand the concept of a Kalman Filter in his AFIT studies. Also, the credit for the development of large portions of this section belongs to Gray [10] and Britton [6].

2.8.2 Kalman Filter Example

Suppose that you are lost at sea during the night and have no idea at all of your location. So you take a star sighting to establish your position (for the sake of simplicity, consider a one-dimensional location). At some time t_1 you determine your location to be z_1. However, because of inherent measuring device inaccuracies, human error, and the like, the result of your measurement is somewhat uncertain. Say you decide that the precision is such that the standard deviation (one-sigma value) involved is σ_{z_1} (or equivalently, the variance, or second order central statistic, is $\sigma_{z_1}^2$). Thus, you can establish the conditional probability of $x(t_1)$, your position at time t_1, conditioned on the observed value of the measurement being z_1, as depicted in Figure II-10. This is a plot of the conditional density $f_{\mathrm{x}(t_1)|z(t_1)}(x|z_1)$as a function of the location x: it tells you the probability of being in any one location, based upon the measurement you took. Note that σ_{z_1} is a direct measure of the uncertainty: the larger σ_{z_1} is, the broader the probability peak is, spreading the probability "weight" over a larger range of x values. For a Gaussian density, 68.3% of the probability "weight" is contained within the band σ units to each side of the mean, the shaded portion in Figure II-10.

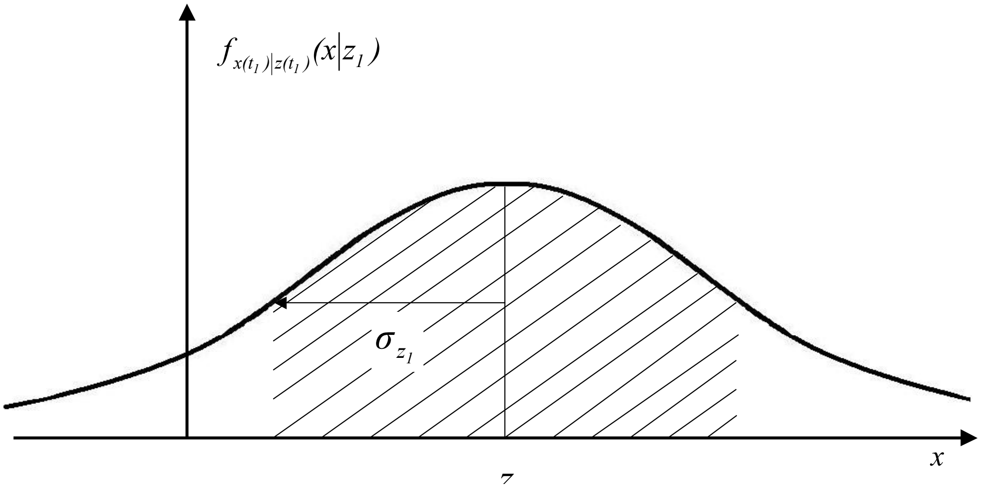


Figure II-10. Conditional Density of Position Based on Measured Value z1

Based on this conditional probability density, the best estimate of your position is

$$\hat{x}(t_1) = z_1 \tag{2.12}$$

and the variance of the error in the estimate is

$$\sigma_x^2(t_1) = \sigma_{z_1}^2 \tag{2.13}$$

Note that $\hat{x}$ is both the mode (value that locates the peak) and the median (value with 1/2 of the probability weight to each side), as well as the mean (center-of-mass).

Now say a trained navigator friend takes an independent fix right after you do, at time $t_2 \cong t_1$ (so that the true position has not changed at all), and obtains a measurement z_2 with a variance $\sigma_{z_2}^2$. Because he has a higher skill, assume the variance in his measurement to be somewhat smaller than in yours. Figure II-11 presents the conditional density of your position at time t_2, based <u>only</u> on the measured value z_2. Note the narrower peak due to smaller variance, indicating that you are rather certain of your position based on his measurement.

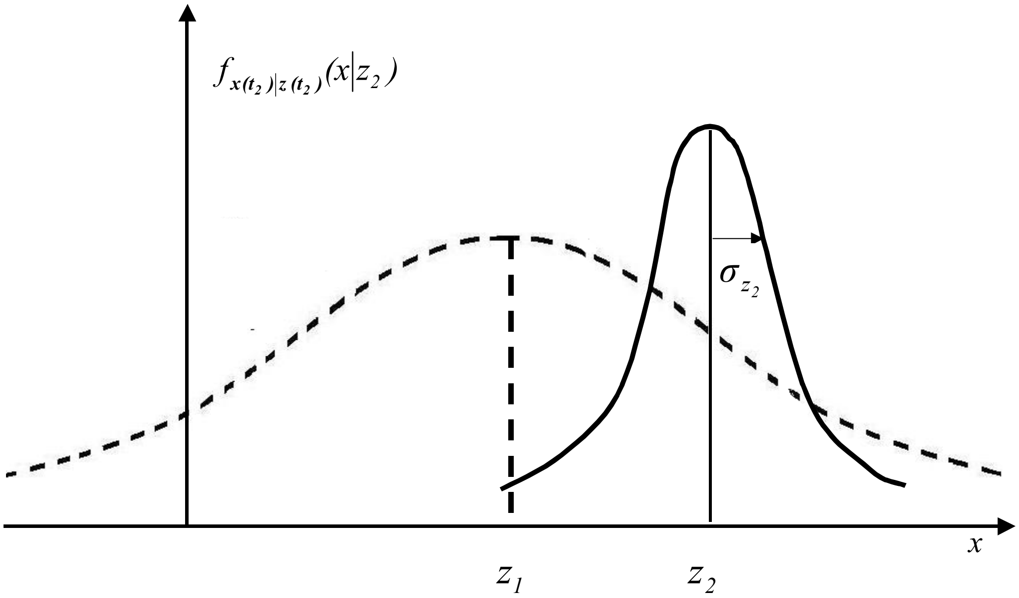


Figure II-11. Conditional Density of Position Based on Measurement z_2 Alone

At this point, you have two measurements available for estimating your position. The question is, how do you combine these data? It can be shown that, based on the assumptions made, the conditional density of your position at $t_2 \cong t_1$, $x(t_2)$, given *both* z_1 *and* z_2, is a Gaussian density with mean μ and variance σ^2 as indicated in Figure II-12 with

$$\mu = \left[\frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right] z_1 + \left[\frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right] z_2 \tag{2.14}$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2} \tag{2.15}$$

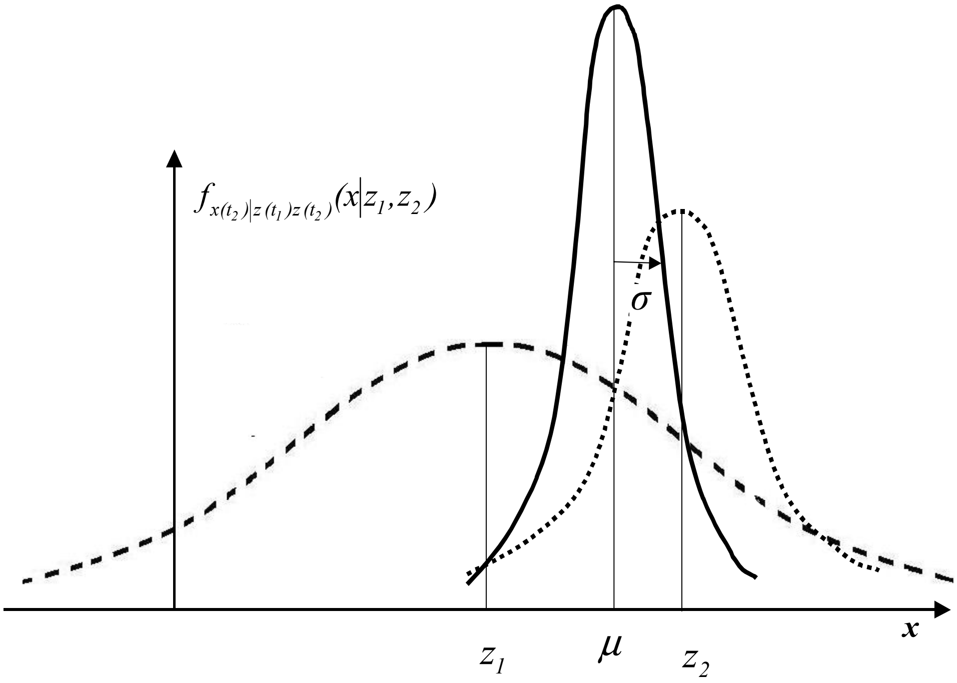


Figure II-12. Conditional density of position based on data z_1 and z_2

Note that, from (2.15), σ is less than either σ_{z_1} or σ_{z_2}, which is to say that the uncertainty in your estimate of position has been decreased by combining the two pieces of information.

Given this density, the best estimate is

$$\hat{x}(t_2) = \mu \tag{2.16}$$

with an associated error variance σ^2. It is the mode, the median and the mean (or, since it is the mean of a conditional density, it is also termed the conditional mean). Furthermore, it is also the maximum likelihood estimate, the weighted least squares estimate, and the linear unbiased estimate with variance that is less than that of any other linear unbiased estimate. In other words, it is the "best" you can do according to just about any reasonable criterion.

After some study, the form of μ given in (2.14) makes good sense. If σ_{z_1} were equal to σ_{z_2}, which is to say you think the measurements are of equal precision, the equation says the optimal estimate of position is simply the average of the two measurements, as would be expected. On the other hand, if σ_{z_1} were larger than σ_{z_2}, which is to say that the uncertainty involved in the measurement z_1 is greater than that of z_2, then the equation dictates "weighting" z_2 more heavily than z_1. Finally, the variance of the estimate is less than σ_{z_1} even if σ_{z_2} is very large: even poor quality data provides some information, and should thus increase the precision of the filter output.

The equation for $\hat{x}(t_2)$can be written as

$$\hat{x}(t_2) = \left[\frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right] z_1 + \left[\frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right] z_2 \tag{2.17}$$

or equivalently as:

$$\hat{x}(t_2) = z_1 + \left[\frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}\right][z_2 - z_1] \tag{2.18}$$

or, in final form that is actually used in Kalman filter implementations (noting that $\hat{x}(t_1) = z_1$),

$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)[z_2 \text{ - } \hat{x}(t_1)] \tag{2.19}$$

where

$$K(t_2) = \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} \tag{2.20}$$

These equations say the optimal estimate at t_2, $\hat{x}(t_2)$, is equal to the best prediction of its value before z_2 is taken, $\hat{x}(t_1)$, plus a correction term of an optimal weighting value times the residual difference between z_2 and the best prediction of its value before it is actually taken, $\hat{x}(t_1)$. It is worthwhile to understand this "predictor-corrector" structure of the filter. Based on all previous information, a prediction of the value that the desired variables and measurement will have at the next measurement time is made. Then, when the next measurement is taken, the difference between it and its predicted value is used to "correct" the prediction of the desired variables.

Using the $K(t_2)$ in Equation (2.20), the variance equation given by (2.15) can be rewritten as

$$\sigma_x^2(t_2) = \sigma_x^2(t_1) - K(t_2)\sigma_x^2(t_1) \tag{2.21}$$

Note that the values of $\hat{x}(t_2)$ and $\sigma_{x_2}^2(t_2)$ embody all the information in the Gaussian density $f_{x(t_2|z(t_1),z(t_2)}(x|z_1, z_2)$. Stated differently, by propagating these two variables, the conditional density of your position at time t_2, given z_1 and z_2, is completely specified.

Thus we have solved the (static) estimation problem example from [20]. This will be of vital importance to the practicality of filter implementation. The filter is plain and simple, “just a computer program in a central processor [20].” If the reader needs a further example detailing dynamics and propagations, see [20].

2.8.3 Linear Kalman Filter

Whenever possible, a system will be modeled as a set of linear differential equations of the form [20]:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.22}$$

where:

x	=	"state" vector (n-dimensional)
F	=	homogenous state dynamics matrix (n x n)
B	=	control input matrix (n x r)
u	=	deterministic control input vector (r-dimensional)
G	=	driving noise input matrix (n x s)
w	=	white Gaussian driving noise vector (s-dimensional)

Because the deterministic control term $\mathbf{B}(t)\mathbf{u}(t)$ is zero in this research, it will be ignored hereafter. The expected value (i.e. *mean*), of the white Gaussian driving noise vector, **w**(t) is:

$$E\{\mathbf{w}(t)\} = \mathbf{0} \tag{2.23}$$

and the noise strength is $\mathbf{Q}(t)$:

$$E\left\{\mathbf{w}(t)\mathbf{w}^T(t+\tau)\right\} = \mathbf{Q}(t)\delta(\tau) \tag{2.24}$$

where $\delta(\cdot)$ is the Dirac delta function.

While Equation (2.22) is written in terms of "whole" value state variables, the models used in the thesis are those of *error* states. This choice of state variable results in simpler dynamics equations [5], and (2.22) may be rewritten as [20]:

$$\boldsymbol{\delta}\dot{\mathbf{x}}(t) = \mathbf{F}(t)\boldsymbol{\delta}\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.25}$$

where $\mathbf{x}(t)$ has been replaced by the error state vector $\delta\mathbf{x}(t)$, and all other quantities retain their previous definitions. The topic of error states is explored more fully in the section on extended Kalman filters.

As previously stated, the Kalman filter incorporates sampled-data measurement information from external measuring devices. Irrespective of the type of measuring device, the equation which is used to describe linear measurements at sample time t_i is of the form [3]:

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.26}$$

or, in the case of error-state models:

$$\boldsymbol{\delta}\mathbf{z}(t_i) = \mathbf{H}(t_i)\boldsymbol{\delta}\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{2.27}$$

where, in both cases above, $\mathbf{H}$ is the *observation* matrix, and $\mathbf{v}$ is a discrete-time zero-mean white Gaussian measurement noise vector with covariance given by [20]:

$$E\left\{\mathbf{v}(t_i)\mathbf{v}^T(t_j)\right\} = \begin{cases} \mathbf{R}(t_i)\ for\ \ t_i = t_j \\ \mathbf{0} \quad\ \ for\ \ t_i \neq t_j \end{cases} \tag{2.28}$$

The Kalman filter "propagates" the error state and its covariance from the instant in time immediately following the most recent measurement update, t_i^+, to the instant in time immediately preceding the next measurement update, t_{i+1}^-, by numerical integration of the following equations [20]:

$$\dot{\hat{\mathbf{x}}}(t/t_i) = \mathbf{F}(t) \quad \hat{\mathbf{x}}(t/t_i) \tag{2.29}$$

$$\dot{\mathbf{P}}(t\ /\ t_i) = \mathbf{F}(t)\mathbf{P}(t\ /\ t_i) + \mathbf{P}(t\ /\ t_i)\mathbf{F}^T(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \tag{2.30}$$

The notation for $\hat{\mathbf{x}}(t\ /\ t_i)$ and associated error covariance $\mathbf{P}(t\ /\ t_i)$ indicate the best estimate of **x** and **P** at time t, based on measurements through time t_i, and t is in the interval from t_i^+ to ${t_{i+1}}^-$. Initial conditions are given as

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i/t_i) \tag{2.31}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i/t_i) \tag{2.32}$$

as provided by the measurement update cycle at time t_i. The variables t_i and t_{i+1} indicate the initial and final times for each integration period, respectively.

After propagation, $\hat{\mathbf{x}}(t_{i+1}^-) = \hat{\mathbf{x}}(t_{i+1}/t_i)$ and $\mathbf{P}(t_{i+1}^-) = \mathbf{P}(t_{i+1}/t_i)$ are "updated" (meaning that state estimates are revised, based on new measurement information). The pivotal element in the update equations for sample time t_i (rather than time t_{i+1}) shown below is the time-varying Kalman filter gain $\mathbf{K}(t_i)$. The $\mathbf{K}(t_i)$ matrix assigns "weights" to the "measurement residual" (the residual consists of the *difference* between the actual measurement and the filter's prediction of the measurement) based on known measurement noise statistics and filter-computed state error covariance from the previous time step. This process is designed to improve the estimate of each element of the state vector. The update equations are [20]:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T(t_i)\left\{\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^T(t_i) + \mathbf{R}(t_i)\right\}^{-1} \tag{2.33}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\{\mathbf{z}_i - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i)]\} \tag{2.34}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{2.35}$$

Although the algorithm shown above is generally applicable to any problem which lends itself to a Kalman filtering solution, it is not necessarily the algorithm which is used in practice. It is often advantageous to use a form of the algorithm known as the *U-D* factorization form [20]. In the *U-D* algorithm, the filter covariance matrix is not propagated and updated as a single square array. The **U** and **D** matrices below representing the pre- and post-measurement filter covariances, respectively, are explicitly computed instead [20]:

$$\mathbf{P}(t_i^-) = \mathbf{U}(t_i^-)\mathbf{D}(t_i^-)\mathbf{U}(t_i^-) \tag{2.36}$$

$$\mathbf{P}(t_i^+) = \mathbf{U}(t_i^+)\mathbf{D}(t_i^+)\mathbf{U}(t_i^+) \tag{2.37}$$

where the **U** matrices are upper triangular and unitary (and thus contain ones along the main diagonal), and the **D** matrices are simply diagonal [20]. This form offers several advantages, including numerical stability, improved precision, and guaranteed non-negativity of the computed covariance's eigenvalues [20]. It is the *U-D* form of the

Kalman filter algorithm which is implemented in the MatSOFE software [12] that is used in this research.

2.8.4 Linearized and Extended Kalman Filter

Not all problems are adequately described with linear systems driven by white Gaussian noise. In many cases, the most appropriate model is nonlinear. The navigation problem at hand falls squarely into the nonlinear category (at least for the measurement model). Fortunately, a relinearized method exists whereby a nonlinear system may be treated in much the same manner as a linear one for a particular class of problems. Suppose that the nonlinear system may be described by [21]:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.38}$$

In this case, the state dynamics vector, $\mathbf{f}[\cdot, \cdot, \cdot]$, is a nonlinear function of the state vector $\mathbf{x}(\cdot)$, the control input (assumed to be zero in this research), and time *t*. The white Gaussian noise is defined exactly as in (2.23) and (2.24), and it still enters the dynamics model linearly. In addition, the measurement equation may also be a nonlinear function of the state vector and time [21]:

$$\mathbf{z}(t_i) = \mathbf{h}[\mathbf{x}(t_i), t_i] + \mathbf{v}(t_i) \tag{2.39}$$

The noise vector **v** is again *zero-mean*, white and Gaussian, entering the measurement equation linearly, and its covariance is described by (2.28).

Recalling that a system must be linear in order to satisfy the assumptions upon which a conventional Kalman filter is based, the nonlinear equations (2.38) and (2.39) must be linearized. The following approach is summarized from [21]:

1. Assume that a nominal state trajectory, $\mathbf{x}_n(t)$, may be generated which satisfies

$$\mathbf{x}_n(t_0) = \mathbf{x}_{n_0} \tag{2.40}$$

and

$$\dot{\mathbf{x}}_n(t) = \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}(t), t] \tag{2.41}$$

where **f** [·, ·, ·] is specified in (2.38), and **u**(t)=**0**.

2. The "nominal" measurements, which accompany the nominal trajectory, are:

$$\mathbf{z}_n(t_i) = \mathbf{h}[\mathbf{x}_n(t_i), t_i] \tag{2.42}$$

3. The "perturbation" of the state derivative is obtained by subtracting the nominal trajectory from the original nonlinear equation:

$$[\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_n(t)] = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), t] - \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}(t), t] + \mathbf{G}(t)\mathbf{w}(t) \tag{2.43}$$

4. The equation above may be approximated to first order by a Taylor series expansion:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}[t; \mathbf{x}_n(t)]\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{2.44}$$

where $\delta\mathbf{x}(\cdot)$ represents a first-order approximation of the process $[\mathbf{x}(\cdot) - \mathbf{x}_n(\cdot)]$, and $\mathbf{F}[t; \mathbf{x}_n(t)]$ is a matrix of partial derivatives of **f** with respect to its first argument, evaluated along the nominal trajectory [21]:

$$\mathbf{F}[t; \mathbf{x}_n(t)] = \left.\frac{\partial\mathbf{f}[\mathbf{x}, t]}{\partial\mathbf{x}}\right|_{\mathbf{x} = \mathbf{x}_n(t)} \tag{2.45}$$

5. The perturbation measurement equation is derived in like fashion and is expressed as [21]:

$$\delta\mathbf{z}(t_i) = \mathbf{H}[t_i; \mathbf{x}_n(t_i)]\delta\mathbf{x}(t) + \mathbf{v}(t_i) \tag{2.46}$$

where

$$\mathbf{H}[t_i ; \mathbf{x}_n (t_i)] = \left. \frac{\partial \mathbf{h}[\mathbf{x}, t_i]}{\partial \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}_n (t_i)} \tag{2.47}$$

With the "error-state" model (2.44)-(2.47) in hand, it is possible to return to the linear filtering theory. An estimate of the whole-valued quantities of interest is obtained from [21]:

$$\hat{\mathbf{x}} = \mathbf{x}_n(t) + \delta\hat{\mathbf{x}}(t) \tag{2.48}$$

The expression above for the linearized Kalman filter is useful, provided that the linearization assumption is not violated. However, if the nominal and "true" trajectories differ by too large an amount, unacceptable errors may result [21]. It is for this reason that extended Kalman filtering is useful in many cases for which perturbation techniques alone do not suffice. Extended Kalman filtering allows for *relinearizing* about newly declared nominals at each sample time, to enhance the adequacy of the linearization process, and thus of the resulting filter performance as well [21].

The extended Kalman filter equations are summarized below. The reader is referred to [21] for details regarding their derivation. The assumed measurement model equation for an extended Kalman filter development is given by Equation (2.39), where $\mathbf{v}(\cdot)$ is once again *zero-mean*, white and Gaussian, with covariance given by (2.28). Measurements are incorporated into the extended Kalman filter via the following set of equations [21]:

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^T[t_i ; \hat{\mathbf{x}}(t_i^-)]\left\{\mathbf{H}[t_i ; \hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-)\mathbf{H}^T[t_i ; \hat{\mathbf{x}}(t_i^-)] + \mathbf{R}(t_i)\right\}^{-1} \tag{2.49}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\left\{\mathbf{z}_i - \mathbf{h}[\hat{\mathbf{x}}(t_i^-); t_i]\right\} \tag{2.50}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}[t_i;\hat{\mathbf{x}}(t_i^-)]\mathbf{P}(t_i^-) \tag{2.51}$$

where

$$\mathbf{H}[t_i;\hat{\mathbf{x}}(t_i^-)] = \left.\frac{\partial \mathbf{h}[\mathbf{x},t_i]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(t_i^-)} \tag{2.52}$$

The state estimate and covariance are propagated from t_i to t_{i+1} by integrating the following equations [21]:

$$\dot{\hat{\mathbf{x}}}(t/t_i) = \mathbf{f}[\hat{\mathbf{x}}(t/t_i),\mathbf{u}(t),t] \tag{2.53}$$

$$\dot{\mathbf{P}}(t/t_i) = \mathbf{F}[t;\hat{\mathbf{x}}(t/t_i)]\mathbf{P}(t/t_i) + \mathbf{P}(t/t_i)\mathbf{F}^T[t;\hat{\mathbf{x}}(t/t_i)] + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^T(t) \tag{2.54}$$

where

$$\mathbf{F}[t;\hat{\mathbf{x}}(t/t_i)] = \left.\frac{\partial \mathbf{f}[\mathbf{x}(t),t_i]}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(t/t_i)} \tag{2.55}$$

and the initial conditions are:

$$\hat{\mathbf{x}}(t_i/t_i) = \hat{\mathbf{x}}(t_i^+) \tag{2.56}$$

$$\mathbf{P}(t_i/t_i) = \mathbf{P}(t_i^+) \tag{2.57}$$

The equations shown above for the extended Kalman filter are programmed into the MatSOFE shell [12] for the problem defined by this thesis. It is the fact that the extended Kalman filter is relinearized about each successive estimate of the state $\hat{\mathbf{x}}(t)$which "enhances the validity of the assumption that deviations from the reference (nominal) trajectory are small enough to allow linear perturbation techniques to be employed" [21].

2.9 Summary

This chapter has presented the basic theory of an RLG-based strapdown INS, GPS, DGPS, and barometric altimeter. Reference frames and coordinate transformations used in this thesis have also been defined. A Kalman filter example from [20] was given and linear, linearized and extended Kalman filter fundamentals were discussed from [21]. The purpose of this chapter was to provide a flavor of Kalman filtering. Chapter 3 will describe the design methodology and error models of the RLG-based strapdown INS, GPS, DGPS, and barometric altimeter used in this thesis for MatSOFE simulations. More information on Kalman filter development and uses may be found in [20,21,22].

III. Design Methodology and Error Models

3.1 Overview

This chapter describes the set-up of the MatSOFE computer simulation for the Space Navigation System Model (SNSM) error model. This chapter also describes the technique used to determine which “real-world” (*true*) satellite vehicle (SV) ephemeris data was used during MatSOFE simulation. A brief description on using PROFGEN [52] to generate a transport aircraft profile and a launch vehicle flight profile will also be discussed. The Tanker profile is used to validate the simulation tool by comparing results to previous research results; the next chapter describes the use of PROFGEN to generate a satellite launch rocket trajectory for this effort. As with Chapter 2, the background works done by Gray [10] and Britton [6] laid the foundations for large portions of this chapter. The credit for the development of large portions of this chapter should be given to them.

3.2 Introduction to MatSOFE

The name "MSOFE" is an acronym meaning "Multimode Simulation for Optimal Filter Evaluation." MSOFE is a general-purpose, multimode simulation program for designing integrated systems that employ optimal Kalman filtering techniques and for evaluating their performance [29]. Its general-purpose construction allows specific user problems to be simulated more quickly and at less cost than without its use. MSOFE has been designed to support a wide variety of system simulation and filter evaluation efforts. It provides two major operating modes:

1) Monte Carlo simulation: to generate multiple sample time histories of system truth states, filter states, and filter estimation errors, including nonlinear effects; usable for linear and extended Kalman filters;

2) Covariance simulation: to generate time histories of the second-order statistics (covariances) of system truth states, filter states, and filter estimation errors, under the assumption of linear (or linearized) models.

Monte Carlo and covariance simulation modes of MSOFE are complementary to one another. The covariance mode can generate filter performance statistics via a single run, whereas the Monte Carlo mode requires several sample runs (say, 15 or more) to generate meaningful statistics for a given scenario. However, the covariance mode is limited to linear (or linearized) systems, whereas the Monte Carlo mode can represent nonlinear as well as linear dynamic and measurement processes. In addition, the Monte Carlo mode provides better visibility into the detailed workings of the filter models and computation processes, and can easily be reduced to a deterministic mode (by suppressing noise sampling) when required [10,6].

MSOFE has been the primary tool used in the design and evaluation of the Kalman filters by AFIT students at Wright-Patterson AFB, Ohio. As stated earlier, MSOFE is restricted to use by U.S. government agencies and their U.S. contractors. As an international officer, the author is not eligible to use MSOFE in the role of a tool for Kalman filter evaluation. For that reason the Matlab based [37] "MatSOFE" is used in this research to conduct navigation performance analysis of a launch vehicle. MatSOFE was first created by former AFIT student William B. Mosle [27], GE/93D, in 1993 and modified by Pamela L. Harms [12] in 1995 for Sverdrup Technology, Inc., TEAS Group most recently before the author received it. Before the author received it, MatSOFE was the analysis tool that had been used on the Exploitation of Differential Global Positioning System (GPS) for Guidance Enhancement (EDGE) Program to conduct a navigation performance analysis in TEAS group. That received version will be called the EDGE-version MatSOFE in this thesis.

"MatSOFE is written entirely in Matlab [37] code. Matlab is an interactive system and programming language for general scientific and technical computation. MatSOFE has been written as a set of Matlab scripts and functions using m-files. The m-file is a feature of Matlab used in the MatSOFE tool" [12]. This feature allows the user to execute sequences of commands that are stored in files with separate names and extension of ".m", as in "Matsofe_in.m" which is in fact an m-file in which the filter process strength (Q) and measurement noise covariance (R) are stated along with all the constants used in simulation.

MatSOFE is designed only for Monte Carlo simulations. It does not have the capability of making a covariance analysis. Such a capacity could be added to MatSOFE performance evaluation as well, but as is discussed clearly earlier, the covariance analysis is limited to linear (or linearized) systems, whereas the Monte Carlo simulations can represent nonlinear as well as linear dynamic and measurement processes. Besides that, the "real world" is itself nonlinear for this application. 15 runs of Monte Carlo simulations (unless otherwise noted) are made in this research by using MatSOFE.

The latest version MatSOFE is the outcome of a collective effort. Many people struggled to reflect their knowledge in Matlab and Kalman filtering before it reached today's capabilities. At the time of delivery, the EDGE version MatSOFE had 53 m-files, containing a 48-state strapdown INS/GPS model, which includes 21 states for INS and 27 states for GPS (with 5 modeled GPS SV errors) along with a 17-state filter model. It had a limited capability. The GPS orbit model was a fairly simplistic model and was not very realistic for anything other than extremely short flight trajectories (less than two minutes) During this research, 21 m-files have been deleted, 18 new m-files have been added, and almost all others have been modified to be adequate for a realistic launch vehicle simulation. Currently, MatSOFE has a 69-state (61-state for DGPS) LN-93 INS/GPS integration truth model, which includes a 39-state reduced order LN-93 INS model and 30-state GPS (22-state for DGPS) error model. The filter model contains 13 states (9 states for Pinson error variables' definition, 2 states for barometric stabilization, and 2 states for GPS). Unlike the EDGE version, it uses "simulated true GPS data", and the U-D factorization algorithm in order to increase the numerical stability in the Kalman filter.

Similar to MSOFE, MatSOFE permits the user to design integrated systems through Kalman filtering techniques and provides a reliable evaluation tool for that purpose. The power of Matlab makes MatSOFE an excellent software tool for "realistic" implementation of Kalman filter performance evaluation tool analogous to the Fortran-based MSOFE. One advantage of MatSOFE is that it is capable of making matrix operations versus the Fortran-based program, MSOFE, which performs scalar operations. Additional advantages of MatSOFE include access to data, simplicity of programming, ease to build models, simplicity to obtain outputs, familiarization of AFIT students with Matlab and relative ease in troubleshooting and debugging. On the other hand, using a compiled language MSOFE is a faster and well-established Kalman filter performance evaluation tool.

As a performance evaluation tool, MatSOFE (see flow chart below) first takes all the constants and filter tuning parameters (**Q_s, Q_f , R_s , R_f**) provided by the "Matsofe_in.m" file. In the "Setup2.m" m-file, all the desired simulation scope information (simulation time, number of measurements and their rates, etc) is declared. "Initrun2.m" sets the initial conditions (**x**(t_0), **P**(t_0) and simulation time controls initialization). All computations (U-D factorization, propagation and update cycles, the process of storing and keeping track of the evaluated parameters, etc., takes places in this m.file. Probably 95% of the simulation time is consumed in "Simrun2.m". Endrun2 and Endsim.m are called at t he end of each run and at the end of simulation respectively.

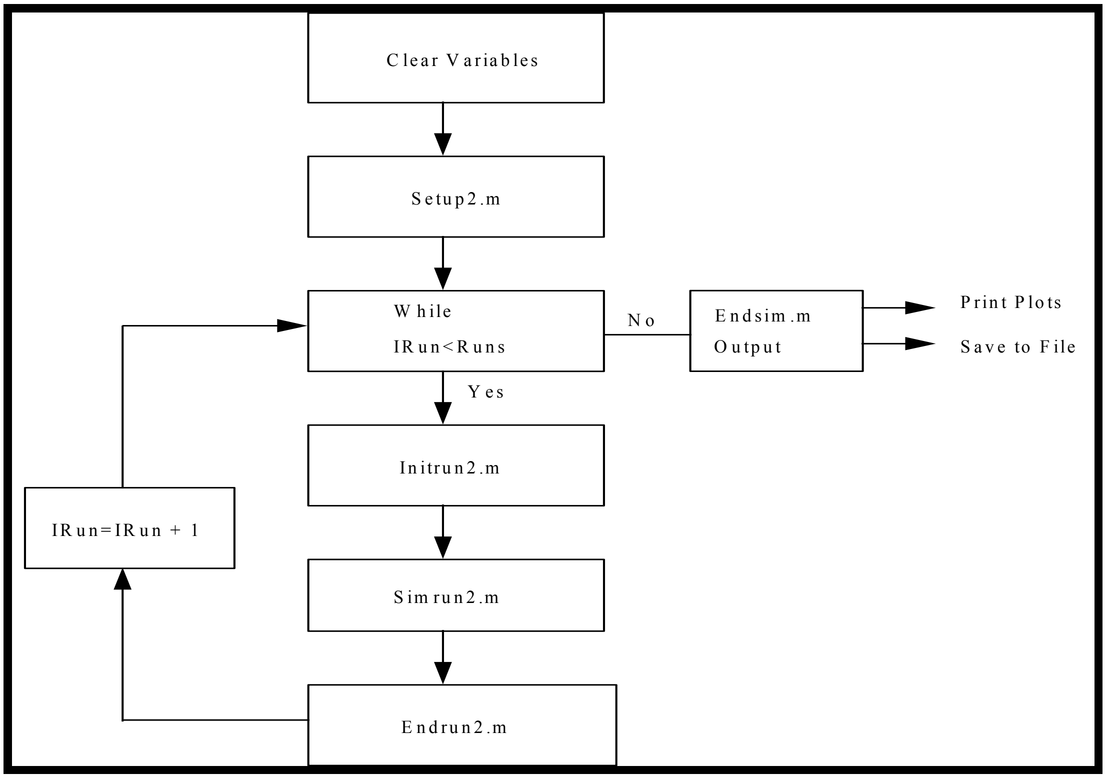


Figure III-1. Block Diagram of MatSOFE Hierarchical Structure

3.3 Introduction to PROFGEN

PROFGEN calculates kinematic data for a simulated air vehicle (aircraft, missile, rocket) moving in free space over the earth. Kinematics is concerned with space and time, and with the time rate of change of quantities that describe the geometry of motion. Kinematic data include translational variables like position and velocity, and rotational variables like angle and angular rate, which portray the aircraft's dynamic state [29].

PROFGEN models the aircraft as a point mass body. There are no actuators, no control surfaces, no aerodynamics, and no propulsion system. Position is given as (geographic) latitude, longitude and altitude. Velocity with respect to earth is coordinatized and presented in a local-vertical frame. Acceleration consists of velocity rates-of-change summed with Coriolis effects and gravity. Attitude consists of roll, pitch and yaw: the Euler angles [6]. PROFGEN tool utilities are stand-alone interactive programs that (mostly) prompt for their inputs. To use PROFGEN, one must supply an input file that specifies the desired trajectory.

PROFGEN builds a complete trajectory from a sequence of flight segments, one segment following another in time. The final values of the variables for segment "k" become the initial values for the segment "k+1", thereby maintaining continuous-time histories for all output quantities. Each flight segment executes one "maneuver" at a time, with six kinds of maneuvers being possible: straight flights, rolls, vertical turns, horizontal turns, jinking and free fall. Since it cannot do several maneuvers simultaneously, PROFGEN cannot be made to perform intricate actions like Immelmanns or barrel rolls. This will be limiting for some applications [29]. Flight segments should not exceed 53.

PROFGEN's kinematic computations track the motion of the aircraft in six degrees of freedom (6-DOF). The vehicle may move anywhere in 3-space and may achieve any attitude except vertical flight. The vector variables can be output in any of four coordinate frames (inertial, earth, wander-level, body). Direction cosine matrices between pairs of frames can also be output. See Table III-1 for PROFGEN flight profile outputs.

PROFGEN has been used to create trajectories for a variety of aircraft including bombers, fighters, transporters, and helicopters. But it is also able to produce trajectories other than aircraft. For example, it has been used to simulate missile flights, bombs in free fall, satellites in Earth orbit, and busses driving highways.

For this study the latest version of PROFGEN (Version 8.1) is used to generate trajectories. In order to validate MatSOFE as a reliable Kalman filtering performance evaluation tool, the work accomplished by Gray [10], Britton [6], and Mosle [27] are regenerated in the MatSOFE environment. To do this job, the KC-135 Tanker aircraft flight profile of these previous research efforts is generated in PROFGEN by using the provided PROF_IN code in Gray's thesis. After MatSOFE is validated, at the second stage, the Atlas IIAS launch profile is generated in PROFGEN. The Atlas flight profile has 8 flight segments. The segments for the "Tanker" flight profile used in this thesis are shown in Table III-2. The parameters shown in Table III-2 are computed by the author taking the actual flight characteristics into account.   Appendix C shows the Fortran-based PROF_IN file for the Atlas launch vehicle and Figure III-2 shows the plot of 2-dimensional PROFGEN-generated rocket profile.

Table III-1A. PROFGEN Flight Profile Outputs

#	Variable Name	Dim	Printed Name	Printed to FLY_OUT	Printed to PROF_OUT
0	time	1	TIME	YES	YES
1	geodetic longitude	1	GLON	YES	YES
2	geodetic latitude	1	GLAT	YES	YES
3	altitude	1	ALT	YES	YES
4	celestial longitude	1	CLON		
5	wander angle	1	ALPHA	YES	YES
6	heading	1	HEAD		
7	roll	1	ROLL	YES	YES
8	pitch	1	PITCH	YES	YES
9	yaw	1	YAW	YES	YES
10	terrestrial longitude rate	1	dGLON		
11	geographic latitude rate	1	dGLAT		
12	altitude rate	1	dALT		
13	celestial longitude rate	1	dCLON		
14	wander angle rate	1	dALPHA	YES	YES
15	heading rate	1	dHDG		
16	roll rate	1	dROLL	YES	YES
17	pitch rate	1	dPITCH	YES	YES
18	yaw rate	1	dYAW	YES	YES
19	signed earth velocity magnitude	1	Vs		
20	signed earth velocity magnitude rate	3	dVs		
21	position in frame i	3	Ri		
22	earth velocity in frame i	3	Vi		
23	inertial velocity in frame i	3	VIi		
24	gravitation in frame i	3	GNi		
25	specific force in frame i	3	FIi		

Table III-1B. PROFGEN Flight Profile Outputs

#	Variable Name	Dim	Printed Name	Printed to FLY_OUT	Printed to PROF_OUT
26	angular rate b/i in frame i	3	Wbi_i		
27	DCM to inertial from body	3x3	Cib		
28	DCM to inertial from earth	3x3	Cie		
29	angular rate e/i in frame e	3	Wei_e		
30	position in frame e	3	Re		
31	earth velocity in frame e	3	Ve		
32	inertial velocity in frame e	3	VIe		
33	gravity in frame e	3	Ge		
34	specific force in frame e	3	FIe		
35	angular rate b/i in frame e	3	Wbi_e		
36	DCM to earth from body	3x3	Ceb		
37	DCM to earth from wander-level	3x3	Cew		
38	angular rate n/e inframe w	3	Wwe_w		
39	position in frame w	3	Rw		
40	earth velocity in frame w	3	Vw	YES	YES
41	inertial velocity in frame w	3	VIw		
42	gravity in frame w	3	Gw		
43	specific force in frame w	3	FIw	YES	YES
44	angular rate b/i in frame w	3	Wbi_w		
45	DCM to wander-level from body	3x3	Cwb		
46	DCM to wander-level from inertial	3x3	Cwi		
47	angular rate b/w in frame b	3	Wbw_b		
48	position in frame b	3	Rb		
49	earth velocity in frame b	3	Vb		
50	inertial velocity in frame b	3	VIb		
51	gravity in frame b	3	Gb		
52	specific force in frame b	3	FIb		
53	angular rate b/i in frame b	3	Wbi_b		

Table III-2. PROFGEN Segments for "Atlas IIAS" Flight Profile

	MANEUVER DEFINITION	MAN.	SEGLNT	PACC	TACC	DELHED	DELPIT	DELROL	PO_DT	FO_DT
1	Ground-lit SRB ignition and liftoff	Strt.	1	1	0	0	0	0	1	1
2	Ground-lit SRB burnout & Air-lit SRB ignition	Vert.	60	1.9	0.5	0	-5	0	5	1
3	Air-Lit SRB burnout and jettison	Vert.	57	0.25	2.22	0	-20	0	5	1
4	Atlas booster engine cutoff and booster package jettison	Vert.	50	0.2	2.22	0	-20	0	5	1
5	Payload fairing jettison	Vert.	48	0.18	1.1	0	-5	0	5	1
6	Sustainer engine cutoff	Vert.	85	0.4	1.3	0	-10	0	10	1
7	Atlas/Centaur seperation	Vert.	192	0.3	1.7	0	-20	0	50	1
8	Centaur main engine cutoff (MECO2)	Vert.	985	0.3	0.8	0	-10	0	50	1

MAN: Maneuver type

SEGLNT: Segment length (sec.)

PACC: Path acceleration (gees)

TACC: Turn acceleration, maximum centrifugal component (gees)

DELHEL: Desired change in heading for horizontal turn (deg.)

DELPIT: Desired change in pitch angle for vertical turn (deg.)

DELROL: Desired change in roll angle for roll maneuver (sec.)

PO_DT: Time interval between synchronous formatted prints to PROF_OUT (sec.)

FO_DT: Time interval between synchronous unformatted writes to FLY_OUT (-)

Note: These definitions are extracted from the PROFGEN user manual [29]

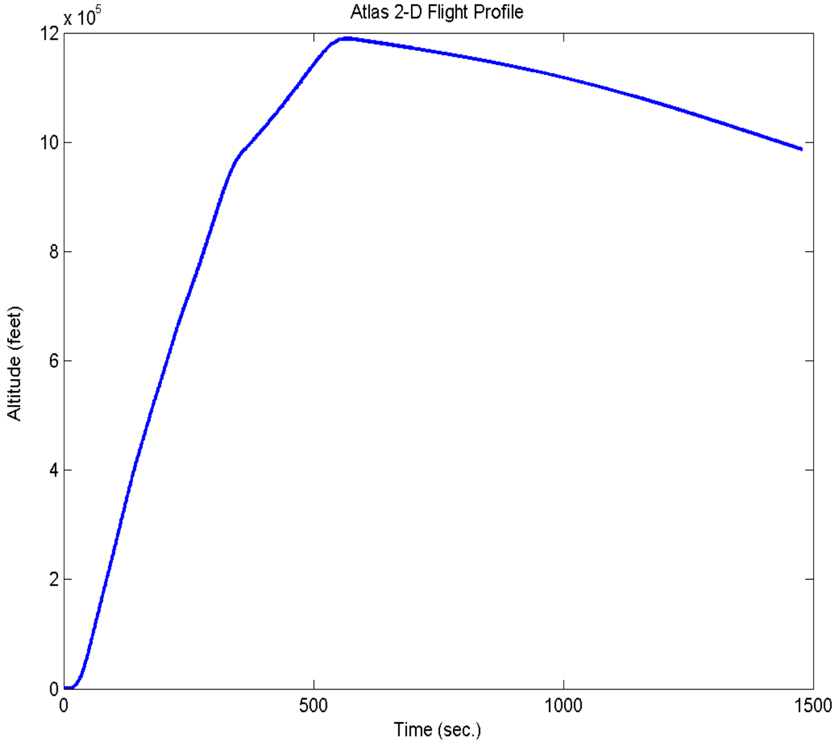


Figure III-2. Atlas 2-Dimensional Flight Profile

3.4 *Satellite Vehicle Data Using WSEM 3.6*

The PROFGEN output file, FLY_OUT.asc, simulates and provides necessary variables which MatSOFE needs for the estimation process. Another file that MatSOFE must read for this study is the simulated true SV ephemeris data (for GPS measurement simulations) provided by the information obtained from [2,32 38].

At AFIT, for the first time, Gray [10] made use of true satellite vehicles (SV) ephemeris data as an input to MSOFE rather than utilizing the MSOFE subroutine to generate generic SV data. The System Effectiveness Model for Windows (WSEM) software program is part of a wide-ranging collection of engineering tools developed to analyze GPS system performance and mission effectiveness.

The WSEM [2] program incorporates a simple computer model of the GPS system. This software system model includes a simulated Earth based on the World Geodetic System 1984 (WGS-84) definitions. It also includes up to 32 simulated Navstar SVs that orbit around the Earth and various types of GPS user equipment (UE) simulations which the WSEM operator may select and define to be located at any point or set of points on or near the simulated Earth surface. As part of its GPS system model, WSEM automatically computes the current Earth-centered, Earth-fixed (ECEF) positions of each Navstar SV in the constellation defined by the selected almanac data file. "Almanac," as used here, refers to a set of orbital parameters that define where a satellite is and where it is headed. The WSEM's SV almanacs and related mathematical algorithms are implemented exactly as in the actual GPS system, except that provisions for geostationary satellite almanacs have also been included. Thus, WSEM is compatible with the information that the real GPS SVs broadcast on-orbit [2].

The "SEM almanac" used in this study is obtained from the National Geodetic Survey(NGS)'s continuously operating reference stations (CORS) web site [32]. The SV ephemeris used in the SNSM simulation was selected based upon the best four SV available for a random day (the current day when SEM almanac was downloaded) by the WSEM 3.6 Software [2]. The four SVs chosen were based on:

- Random day (28 July 2001) selected.

  (GPS week 1124, Day of year 209).
- GPS Almanac data file "072801A.AL3" (See Appendix E) was obtained (downloaded) from United States Coast Guard web sites [32, 38].

- LAT/LONG/ALT along the Tanker flight profile was noted and entered in WSEM V3.6.
    - Best 4 SV based on PDOP algorithm
    - 10° mask angle (SV visibility rejection criteria)
    - Scenario duration: 1 hour [Begin time 04:00 UTC (Coordinated Universal Time or 08:00 Eastern time)]

The best four SVs are then numerically displayed to the user. The mask angle was chosen as 10° based on the author's experience. Thus, all satellites in view above a 10° angle made from the GPS antenna surface (mask angle) on 28 July 2001 between the hours of 04:00 - 06:00 Greenwich mean time are available for use. For further information on SEM 3.6 software and the SEM 3.6 output plot format, see [2].

3.5 The SNSM Computer Model

The Landing System Model (LSM) and Differential Landing System Model (DLSM) model [10,6] were used as a reference to model the Space Navigation System Model (SNSM) computer model. The SNSM is divided into two parts for computer modeling, the truth model and the filter design model. The truth model represents the computer-generated simulation of the real-world error characteristics found in avionics "black boxes" and the environment in which the units operate. The research was accomplished through computer simulation; therefore, the truth model will simulate the errors in true avionics hardware (INS, GPS/DGPS, Baro, Radar Altimeter) black boxes. The truth model generates the measurement updates for the SNSM filter, the true flight profile of the aircraft, and a state variable baseline for evaluating filter performance

[10,6]. Detailed explanations related to truth model, filter model, measurement model, GPS and DGPS error models, etc. will be presented in Sections 3.6.1 and 3.6.2. The truth model consists of 69 error states for GPS-aided INS and 61error states for DGPS-aided INS about their nominal values. The filter model represents the SNSM as it could be hosted on-board an aircraft computer. Briefly, the SNSM filter model is a 13-state extended Kalman filter developed through order reduction of the original 93-state truth model of [27,33] for just INS, plus 30 more states for GPS (22 for DGPS). Section 3.6.1 and 3.6.2 of this section explains INS and GPS error models in detail, respectively. An advantage of using only 13 states is that the current state-of-the-art aircraft host computers can handle the computational requirements.

The block diagram, Figure III-3, explains the interaction of the filter and truth model in MatSOFE. A simulated flight profile is provided by PROFGEN [29], and the U.S. National Geodetic Survey (NGS)'s continuously operating reference stations (CORS) web page [32] provides true SV ephemeris data for any SV. Use of the "real-world" ephemeris replaced almost six of the prior m-files functions used in old versions of MatSOFE. The best four SV, which have the best (lowest) position dilution of precision (PDOP), were chosen by using System Effectiveness Model for Windows (WSEM) software [2]. The WSEM36 program requires GPS almanac files to run. Current Almanac files can be found at the ARINC web site [2] or at the United States Coast Guard web sites [32, 38]. With this information, the truth model is able to simulate a real world INS navigation solution, $\boldsymbol{x} + \boldsymbol{\delta x}_{INS}$, and generate the real world GPS/DGPS measurements, $\mathbf{R}_{\text{GPS}}$ and $\mathbf{R}_{\text{DGPS}}$ respectively. The Kalman filter block in Figure III-3 represents the SNSM filter. Corrections from the SNSM filter are subtracted from the

INS navigation solution to generate the best possible navigation solution available, $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\delta x}_{INS} - \boldsymbol{\delta \hat{x}}_{INS}$ [27,33]. Now that the MatSOFE implementation of the SNSM filter has been explained, the truth and filter models for the GPS, DGPS, and the INS subsystems will be described.

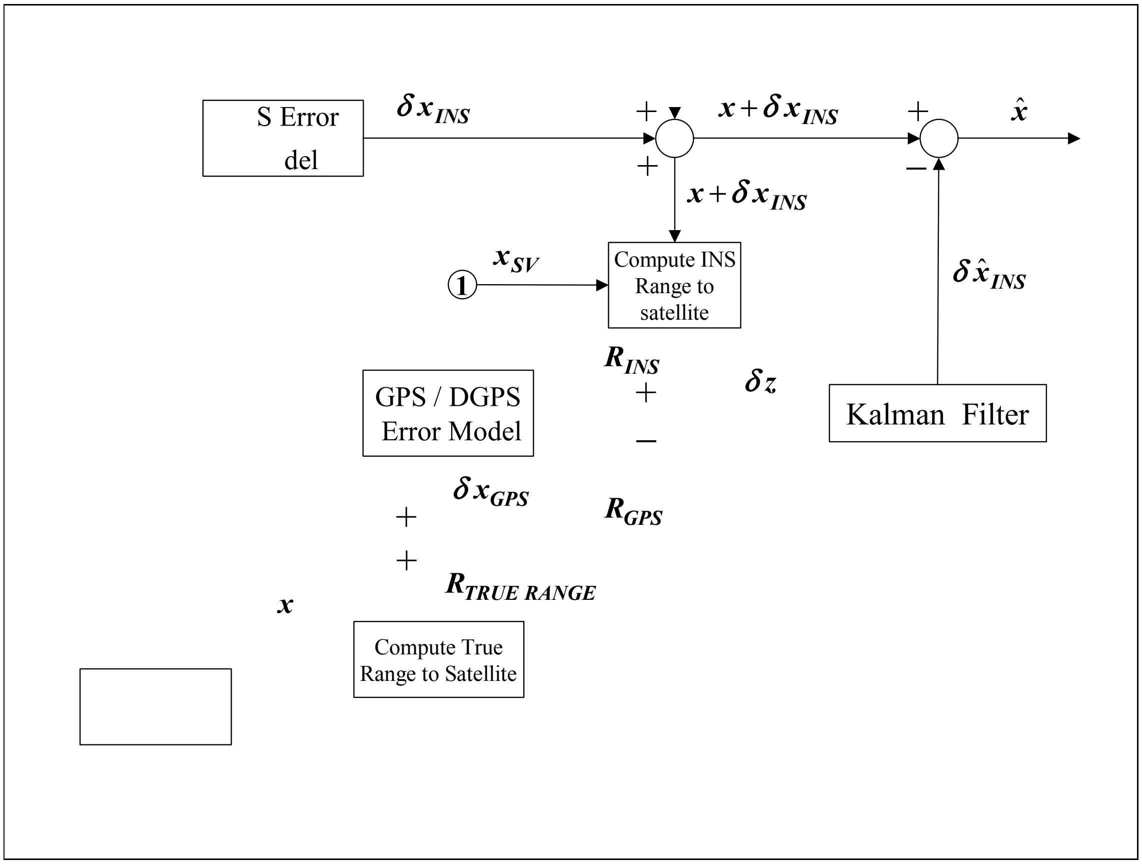


Figure III-3. Truth and Filter Model Block Diagram

3.6 Space Navigation System Model (SNSM) Description

This section presents the truth and filter model propagation and measurement equations, (2.38) and (2.39), respectively. The following presentation will be divided up by navigation subsystems with most parts taken directly from [27]. First the INS portion of the equations will be presented, then the GPS followed by the DGPS. Before the different navigation subsystems are individually described, the high-level state and measurement equations for the SNSM filter are provided, followed by those for the truth model. Equations (3.1) and (3.2) show how the different navigation subsystems models combine to form a single SNSM filter model:

$$\begin{bmatrix} \delta\dot{x}_{INS_f} \\ \delta\dot{x}_{GPS_f} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{INS_f} & 0 \\ 0 & \boldsymbol{F}_{GPS_f} \end{bmatrix} \boldsymbol{\delta x}_f + \begin{bmatrix} \boldsymbol{w}_{INS_f} \\ \boldsymbol{w}_{GPS_f} \end{bmatrix} \tag{3.1}$$

$$\boldsymbol{\delta z}_f = \begin{bmatrix} \boldsymbol{H}_{INS_f} & \boldsymbol{H}_{GPS_f} \end{bmatrix} \boldsymbol{\delta x}_f + \boldsymbol{v}_f \tag{3.2}$$

where the process and measurement noises are described as

$$E\{\boldsymbol{w}_f(t)\boldsymbol{w}_f^T(t+\tau)\} = \boldsymbol{Q}_f \; \delta(\tau) \tag{3.3}$$

$$E\{\boldsymbol{v}_f(t_i)\boldsymbol{v}_f^T(t_j)\} = \boldsymbol{R}_f \; \delta(\tau) \tag{3.4}$$

As stated earlier, the overall filter model consists of 13 states: 11 for the INS and 2 for the GPS/DGPS. A description of the 13-state vector, $\boldsymbol{\delta x}_f$, implemented in the filter model can be found in Table A-5 in Appendix A. References to further descriptions of the sub-matrices in the filter equations can be found in Table III-3. The barometric altimeter aiding measurements are considered to be INS measurements (represented through appropriate elements in the F matrix to denote continuous time updating, rather than treating them as explicit measurements to be handled by the Kalman filter), while the GPS/DGPS measurements are the respective updates for the baro/inertial system from the GPS/DGPS.

Table III-3. References for the Sub-Matrices of the SNSM Truth and Filter

Filter Model	Location of Description	Truth Model	Location of Description
$\boldsymbol{F}_{INS_f}$	Section 3.6.1.3	$\boldsymbol{F}_{INS_{Filter}}$	Section 3.6.1.1, 3.6.1.2
-	-	$\boldsymbol{F}_{INS_{t1}}$	Section 3.6.1.2
-	-	$\boldsymbol{F}_{INS_{t2}}$	Section 3.6.1.2
$\boldsymbol{F}_{GPS_f}$	Section 3.6.2.1	$\boldsymbol{F}_{GPS_t}$	Section 3.6.2.1
$\boldsymbol{F}_{DGPS_f}$	Section 3.6.2.2	$\boldsymbol{F}_{DGPS_t}$	Section 3.6.2.2
$\boldsymbol{w}_{INS_f}$	Section 3.6.1.3	$\boldsymbol{w}_{INS_t}$	Section 3.6.1.1
$\boldsymbol{w}_{GPS_f}$	Section 3.6.2.1	$\boldsymbol{w}_{GPS_t}$	Section 3.6.2.1
$\boldsymbol{w}_{DGPS_f}$	Section 3.6.2.2	$\boldsymbol{w}_{DGPS_t}$	Section 3.6.2.2
$\boldsymbol{H}_{INS_f}$	Section 3.6.1.4	$\boldsymbol{H}_{INS_t}$	Section 3.4.1.4
$\boldsymbol{H}_{GPS_f}$	Section 3.6.2.4	$\boldsymbol{H}_{GPS_t}$	Section 3.6.2.4
$\boldsymbol{H}_{DGPS_f}$	Section 3.6.2.5	$\boldsymbol{H}_{DGPS_t}$	Section 3.4.2.5

The propagation and measurement equations for the SNSM *truth* model is presented in similar fashion below:

$$\boldsymbol{\delta\dot{x}}_t = \begin{bmatrix} \boldsymbol{F}_{Filter} & \boldsymbol{F}_{INS_{t1}} & 0 \\ 0 & \boldsymbol{F}_{INS_{t2}} & 0 \\ 0 & 0 & \boldsymbol{F}_{GPS_{t1}} \end{bmatrix} \boldsymbol{\delta x}_t + \begin{bmatrix} \boldsymbol{w}_{Filter} \\ \boldsymbol{w}_{INS_{t1}} \\ \boldsymbol{w}_{GPS_{t1}} \end{bmatrix} \tag{3.5}$$

$$\boldsymbol{\delta z}_t = \begin{bmatrix} \boldsymbol{H}_{INS_t} & \boldsymbol{H}_{GPS_t} \end{bmatrix} \boldsymbol{\delta x}_t + \boldsymbol{v}_t \tag{3.6}$$

The SNSM truth model consists of the original thirteen states of the filter model (represented by $\mathbf{F}_{Filter}$ and $\mathbf{w}_{Filter}$), augmented by additional INS and GPS/DGPS states. The total number navigation subsystems state is 69: 39 INS states, and 30 GPS states for GPS integrated systems, and 61: 39 INS states, and 22 DGPS states for DGPS integrated systems. Table A-1through Table A-4, in Appendix A, provide a full description of each individual state of the truth model. Table B-1 through Table B-6, and Table B-7 through Table B-9 in Appendix B have a complete listing of the components of the **F** elements and the **Q** noise strengths associated with the **w** vector components in Equation (3.5).

There is one crucial difference between the first thirteen states of the filter model and the first thirteen states of the truth model [27,33]. The filter model dynamics driving noise and measurement noise do not correlate exactly with those of the first thirteen states of the truth model. To achieve good tuning against the truth model, the filter model noise statistics values ($\mathbf{Q_f}$ and $\mathbf{R_f}$) have been altered [27,33]. The following sections will provide a detailed presentation into the exact make-up of the truth and filter model propagation and measurement equations for all navigation subsystems used in this thesis.

3.6.1 The Inertial Navigation System (INS) Model

This section presents the truth and filter models used for the INS. The INS model is a strapped-down wander azimuth system that senses aircraft motion via gyros and accelerometers and is used as the primary source for navigation [27]. The INS model has been derived from a medium accuracy ring laser gyro (RLG) INS 93-state model [18,33]. First, the original 93-state model will be presented, followed by the reduced-ordered

39-state truth and 11-state filter models. After the INS truth and filter state equations have been defined, barometric altimeter measurement equations will be presented.

3.6.1.1 The 93-State LN-93 Error Model

The 93-state Litton INS computer model has been generated by the Wright Laboratories, Avionics Directorate, Avionics System Integration and Research Team (ASIRT). Their development uses both past AFIT research and INS vendor [18] documentation to "fine-tune" past modeling efforts [33,36,43]. The 93-state model generates a high number of documented error sources that are found in the Litton wander-azimuth LN-93 INS [18]. These errors are described using six categories of states [27,33]:

$$\boldsymbol{\delta x} = \left[\boldsymbol{\delta x}_1^{\mathsf{T}}\ \boldsymbol{\delta x}_2^{\mathsf{T}}\ \boldsymbol{\delta x}_3^{\mathsf{T}}\ \boldsymbol{\delta x}_4^{\mathsf{T}}\ \boldsymbol{\delta x}_5^{\mathsf{T}}\ \boldsymbol{\delta x}_6^{\mathsf{T}} \right]^{\mathsf{T}} \tag{3.7}$$

where $\boldsymbol{\delta x}$ is a 93 × 1 column vector and:

$\boldsymbol{\delta x}_1$: represents the "general" error vector containing 13 position, velocity, attitude, and vertical channel errors (representative of a Pinson 9-state model of INS error characteristics augmented with 4 altimeter error states).

$\boldsymbol{\delta x}_2$: consists of 16 gyro, accelerometer, and baro-altimeter exponentially time-correlated errors, and "trend" states. These states are modeled as first order Markov processes in the truth (system) model.

$\boldsymbol{\delta x}_3$: represents gyro bias errors. These 18 states are modeled as random constants in the truth model.

$\boldsymbol{\delta x}_4$: is composed of the accelerometer bias error states. These 22 states are modeled in exactly the same manner as the gyro bias states.

$\boldsymbol{\delta x}_5$: depicts accelerometer and gyro initial thermal transients. The 6 thermal transient states are first order Markov processes in the system model.

$\boldsymbol{\delta x}_6$: models the gyro compliance errors. These 18 error states are modeled as biases in the system model.

The 93-State Litton model state space differential equation is given by:

$$\begin{Bmatrix} \boldsymbol{\delta \dot{x}_1} \\ \boldsymbol{\delta \dot{x}_2} \\ \boldsymbol{\delta \dot{x}_3} \\ \boldsymbol{\delta \dot{x}_4} \\ \boldsymbol{\delta \dot{x}_5} \\ \boldsymbol{\delta \dot{x}_6} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{F}_{11} & \boldsymbol{F}_{12} & \boldsymbol{F}_{13} & \boldsymbol{F}_{14} & \boldsymbol{F}_{15} & \boldsymbol{F}_{16} \\ \mathbf{0} & \boldsymbol{F}_{22} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{F}_{55} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\delta x_1} \\ \boldsymbol{\delta x_2} \\ \boldsymbol{\delta x_3} \\ \boldsymbol{\delta x_4} \\ \boldsymbol{\delta x_5} \\ \boldsymbol{\delta x_6} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{w_1} \\ \boldsymbol{w_2} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.8}$$

A full description of the sub-matrices for this equation is given in the Litton LN-93 manual [18]. This large state model represents the most accurate model available for the LN-93 navigation errors [27,33].

3.6.1.2 The 39-State INS Truth Model

The 93-state model is a very accurate representation of the INS error characteristics, but the high dimensionality of the state equations makes the model very CPU-intensive for “first-look” projects. The intent of this thesis is to evaluate performance characteristics associated with a particular *class* of INS (medium precision or lower precision). Previous AFIT theses have demonstrated that reduced-ordered truth models can be used in place of the 93-state truth model without losing a significant degree of accuracy [27,33]. Therefore the INS truth model has been reduced to a 39-state model. The reduced-ordered model retains only the truly essential states from Equation (3.8). The truth model state space equation is defined in Equation (3.9):

$$\begin{Bmatrix} \delta\dot{x}_1 \\ \delta\dot{x}_2 \\ \delta\dot{x}_3 \\ \delta\dot{x}_4 \end{Bmatrix} = \begin{bmatrix} \boldsymbol{F}_{11} & \boldsymbol{F}_{12} & \boldsymbol{F}_{13} & \boldsymbol{F}_{15} \\ \mathbf{0} & \boldsymbol{F}_{22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{3.9}$$

It should be noted that the INS truth state vector $\boldsymbol{\delta x}$ is a 39-state vector. The four components of $\boldsymbol{\delta x}$ in Eq.(3.9) do not directly correlate to the first four components of the 93-state Litton model [27,33]. For a complete listing of the 39 states and how they relate to those in [18], see Table A-1 and Table A-2 in Appendix A

3.6.1.3 The 11-State INS Filter Model

The INS filter model retains the essential states from the 39-state truth model. Through past AFIT research, the 11-state INS filter has been shown to perform adequately when given frequent GPS/DGPS measurement updates [6,10,27,33]. The 11 states are composed of the standard 9 states for the Pinson INS error model, and two states (clock bias and clock drift) for the GPS or DGPS. Table A-5 in Appendix A shows the 11 states used for the INS filter model. The final INS filter dynamics submatrix, **F**, as well as process noise strength **Q** and measurement noise covariance **R,** can be found in Appendix B

3.6.2 The Global Positioning System (GPS) Model

The GPS navigation system used is based on electromagnetic signals transmitted from orbiting GPS satellites. This model has been developed throughout research at AFIT, and many of its fundamental concepts are addressed in a variety of sources [6,10,27,33,36]. GPS generates navigation information by acquiring the range to multiple satellites of known position, called "*pseudoranges*". Inherent in the pseudorange are errors caused by ionospheric and tropospheric delays, satellite clock biases, receiver noise, and ephemeris errors [25,34]. These errors work together to dilute the accuracy of standard GPS to a level of a couple of meters, depending on the range resolution technique being used. By incorporating the *differential* corrections to the standard GPS pseudoranges, one can achieve much higher navigation precision. Several error sources can be eliminated or significantly reduced because these errors are common to both the reference station receiver and the user's receiver. These errors are composed of the satellite's clock error, errors in the satellites's broadcasted ephemeris data, and signal propagation delays that are not accounted for by the receiver's measurements or modeling. When two receivers in the same vicinity (within about 100 nautical miles) are using the same set of four satellites, the above errors will be common to both and can be removed or essentially eliminated by differential techniques [25,34]. The navigation information passed to the SNSM filter is the respective range and ephemeris data position to each of four satellites, with differential corrections applied to provide more accurate information [33]. The next four sections present all the necessary equations to define the GPS and DGPS truth, and associated filter error models fully.

3.6.2.1 The 30-State GPS Truth Model

There are five types of error sources that are modeled in the GPS truth model state equations. The first two states represent the errors in the user clock and are modeled as follows:

$$\begin{Bmatrix} \delta\dot{R}_{clk_U} \\ \delta\dot{D}_{clk_U} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{clk_U} \\ \delta D_{clk_U} \end{Bmatrix} \tag{3.10}$$

where

δR_{clk_U} = range equivalent of user clock bias

δD_{clk_U} = velocity equivalent of user clock drift

The initial state estimates and covariances for these states were chosen to be consistent with previous AFIT research, [6,10,27,33,36]. and are:

$$\begin{Bmatrix} \delta\hat{R}_{clk_U}(t_0) \\ \delta\hat{D}_{clk_U}(t_0) \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.11}$$

and

$$\boldsymbol{P}_{\delta R_{clk_U},\delta D_{clk_U}}(t_0) = \begin{bmatrix} 9.0\times10^{14}\ ft^2 & 0 \\ 0 & 9.0\times10^{10}\ ft^2/sec^2 \end{bmatrix} \tag{3.12}$$

Because these error sources are a function of the user equipment, they are common to all the satellite vehicles. The remaining five sources of errors are unique to each satellite vehicle (SV), based on their individual equipment and their position with respect to the user. The first SV-specific error source for GPS is the code loop error, δR_{cloop}. Although the code loop is part of the user equipment shared by all the SV's, its error magnitude is relative to each SV. The second and third SV-specific errors are the atmospheric interference with the EM signals, δR_{ion} and δR_{trop}, as related to the ionospheric and tropospheric delay, respectively, in the signal's propagation. The code loop error, the tropospheric delay and ionospheric delay are all modeled as first order Markov processes with time constants shown in Equation (3.13), consistent with previous AFIT research [10,27,33,36]. Both are driven by zero-mean white Gaussian noise with strengths shown in Equation (3.16). The fourth SV-specific error source is due to inaccuracies in the clocks on board the SV's, δR_{Sclk}. By using differential corrections, this error source was also removed from the DGPS model. The final error source was based on line-of-sight errors between the SV's and the receiver, δx_{s_i}, δy_{s_i}, δz_{s_i}, respectively. The model for these states is shown in Equation (3.13):

$$\begin{Bmatrix} \delta\dot{R}_{cloop} \\ \delta\dot{R}_{trop} \\ \delta\dot{R}_{ion} \\ \delta\dot{R}_{Sclk} \\ \delta\dot{x}_{s_i} \\ \delta\dot{y}_{s_i} \\ \delta\dot{z}_{s_i} \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/1500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{cloop} \\ \delta R_{trop} \\ \delta R_{ion} \\ \delta R_{Sclk} \\ \delta x_{s_i} \\ \delta y_{s_i} \\ \delta z_{s_i} \end{Bmatrix} + \begin{Bmatrix} w_{cl} \\ w_{trop} \\ w_{ion} \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \qquad (3.13)$$

where the initial covariance for the states is given by:

$$\mathbf{P}_{GPS}(t_0)=\begin{bmatrix} 0.25ft^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0ft^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0ft^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25ft^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25ft^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 25ft^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 25ft^2 \end{bmatrix} \tag{3.14}$$

and mean values and strengths of the dynamics driving noise are given by:

$$E\{\boldsymbol{w}_{GPS}(t)\}=\boldsymbol{0} \tag{3.15}$$

$$E\left\{\boldsymbol{w}_{GPS}(t)\boldsymbol{w}_{GPS}^{T}(t+\tau)\right\} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.004 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.004 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\delta(\tau)\frac{ft^2}{sec} \tag{3.16}$$

Equations (3.13) - (3.16) are repeated for each of four satellites and then this is added to Equations (3.10) - (3.12) to get 30 states.

A quick reference of the truth model non-zero GPS dynamics matrix components is provided in Table B-5 of Appendix B. This ends the description of the 30-state truth model for standard GPS.

3.6.2.2 The 22-State DGPS Truth Model

Like the GPS truth model, there are five types of error sources that are modeled in the DGPS truth model state equations. The first two states represent the errors in the user clock and are identical to the first two states in the previous section. They are modeled as follows:

$$\begin{Bmatrix} \delta\dot{R}_{clk_U} \\ \delta\dot{D}_{clk_U} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{clk_U} \\ \delta D_{clk_U} \end{Bmatrix} \tag{3.17}$$

where

δR_{clk_U} = range equivalent of user clock bias

δD_{clk_U} = velocity equivalent of user clock drift

The initial state estimates and covariances for these states were chosen to be consistent with previous AFIT researches [6,27,33,36], and are:

$$\begin{Bmatrix} \delta\hat{R}_{clk_U}(t_0) \\ \delta\hat{D}_{clk_U}(t_0) \end{Bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.18}$$

and

$$\mathbf{P}_{\delta R_{clk_U}, \delta D_{clk_U}}(t_0) = \begin{bmatrix} 9.0\times 10^{14}\ ft^2 & 0 \\ 0 & 9.0\times 10^{10}\ ft^2 / sec^2 \end{bmatrix} \tag{3.19}$$

Because these error sources are a function of the user equipment, they are common to all the satellite vehicles. The remaining five sources of errors are unique to each satellite vehicle (SV), based on their individual equipment and their position with respect to the user. The first SV-specific error source for GPS is the code loop error, δR_{cloop}. Although the code loop is part of the user equipment shared by all the SV's, its error magnitude is relative to each SV. The work done by Negast [33] has shown that, with differential corrections applied, this error can be removed from the DGPS model. The second and third SV-specific errors are the atmospheric interference with the EM signals, δR_{ion} and δR_{trop}, as related to the ionospheric and tropospheric delay in the signal's propagation. The tropospheric delay and ionospheric delay are both modeled as first order Markov processes with time constants shown in Equation (3.20), consistent with previous AFIT research [6,27,33,36]. Both are driven by zero-mean white Gaussian noise with strengths shown in Equation (3.23). The fourth SV-specific error source is due to inaccuracies in the clocks on board the SV's, δR_{Sclk}. By using differential corrections, this error source was also removed from the DGPS model. The final error source was based on line-of-sight errors between the SV's and the receiver, δx_{s_i}, δy_{s_i}, δz_{s_i}, respectively. The model for these states is shown in Equation (3.20):

$$\begin{Bmatrix} \delta\dot{R}_{trop} \\ \delta\dot{R}_{ion} \\ \delta\dot{x}_{s_i} \\ \delta\dot{y}_{s_i} \\ \delta\dot{z}_{s_i} \end{Bmatrix} = \begin{bmatrix} -\frac{1}{500} & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{1500} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{trop} \\ \delta R_{ion} \\ \delta x_{s_i} \\ \delta y_{s_i} \\ \delta z_{s_i} \end{Bmatrix} + \begin{Bmatrix} w_{trop} \\ w_{ion} \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{3.20}$$

where the initial covariance for the states is given by:

$$\mathbf{P}_{DGPS}(t_0) = \begin{bmatrix} 1.0\,ft^2 & 0 & 0 & 0 & 0 \\ 0 & 1.0\,ft^2 & 0 & 0 & 0 \\ 0 & 0 & 0.35\,ft^2 & 0 & 0 \\ 0 & 0 & 0 & 0.35\,ft^2 & 0 \\ 0 & 0 & 0 & 0 & 0.35\,ft^2 \end{bmatrix} \tag{3.21}$$

and mean values and strengths of the dynamics driving noise are given by:

$$E\left\{\boldsymbol{w}_{DGPS}(t)\right\} = \boldsymbol{0} \tag{3.22}$$

$$E\left\{\boldsymbol{w}_{DGPS}(t)\boldsymbol{w}_{DGPS}^{T}(t+\tau)\right\} = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0.0004 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \delta(\tau)\frac{ft^2}{\text{sec}} \tag{3.23}$$

The reduced dynamic driving noise strengths from standard GPS are indicative of the error reduction for these remaining states when *differential* corrections are applied. A quick reference of the truth model non-zero DGPS dynamics matrix components is provided in Table B-6 of Appendix B This ends the description of the 22-state truth model for DGPS. Now the filter model will be presented.

3.6.2.3 The 2-State GPS/DGPS Filter Model

Various research efforts have shown that two states provide a sufficient model for GPS and DGPS [6 10,27,33,36,43]. The primary argument is that the errors modeled by the other states in the previous two sections are small when compared to the two states common to all SV's. By adding dynamics driving noise, of strength **Q**, and re-tuning the filter, the overall performance of the DLSM can be maintained with the significantly reduced-order model of Equation (3.24):

$$\begin{Bmatrix} \delta\dot{R}_{clk_U} \\ \delta\dot{D}_{clk_U} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_{clk_U} \\ \delta D_{clk_U} \end{Bmatrix} + \begin{Bmatrix} w_{R_{clk}} \\ w_{D_{clk}} \end{Bmatrix} \tag{3.24}$$

The values implemented for the dynamics driving noise strengths can be found in Tables B-10 through B-12 of Appendix B. It should be noted that, in the tuning process, the measurement noise covariance values **R** (as shown in Appendix B) have also been adjusted to achieve adequate tuning of the filter [20]. This completes the description of the GPS/DGPS filter model. The next section presents the GPS/DGPS measurement equations for both the truth and the filter models.

3.6.2.4 GPS Measurement Model

There are four GPS measurement updates, one for each of the satellite pseudorange signals received by the SNSM filter. These measurement updates are once again *difference* measurements. First the DGPS truth model difference measurement will be fully presented, followed by a brief description of the filter measurement model.

The GPS difference measurement is formed by taking the difference of the INS-calculated pseudorange, R_{INS} and actual pseudorange, R_{GPS}:

$$\delta z_{GPS} = R_{INS} - R_{GPS} \tag{3.25}$$

The real pseudorange, R_{GPS} is the sum of the true range from the user to the satellite plus all the errors in the pseudorange signal propagation. The measurement equation is modeled as:

$$R_{GPS} = R_t + \delta R_{cloop} + \delta R_{trop} + \delta R_{ion} + \delta R_{Sclk} + \delta R_{Uclk} - v \tag{3.26}$$

where

R_{GPS} = GPS pseudorange measurement, from SV to user

R_t = true range, from SV to user

δR_{cloop} = range error due to code loop error

δR_{trop} = range error due to tropospheric delay

δR_{ion} = range error due to ionospheric delay

δR_{Sclk} = range error due to SV clock error

δR_{Uclk} = range error due to user clock error

v = zero-mean white Gaussian measurement noise

Note that, since v is assumed *zero*-mean, white Gaussian noise, statistically speaking, one can choose v with either a plus "+" or minus "-" coefficent in Equation (3.26). The author chooses the coefficient carefully so that the end result shows a plus "+ v" sign in Equation (3.31) and Equation (3.32).

The second source of a range measurement is the INS itself, R_{INS} [27,33]. R_{INS} is the difference between the SNSM-calculated position, $\boldsymbol{X_U}$ and the satellite position from the ephemeris data $\boldsymbol{X_S}$. This difference vector is represented below in the ECEF frame

$$R_{INS} = \left|\boldsymbol{X}_U - \boldsymbol{X}_S\right| = \left| \begin{Bmatrix} x_U \\ y_U \\ z_U \end{Bmatrix}^e - \begin{Bmatrix} x_S \\ y_S \\ z_S \end{Bmatrix}^e \right| \tag{3.27}$$

An equivalent form for Equation (3.27)) is

$$R_{INS} = \sqrt{(x_U - x_S)^2 + (y_U - y_S)^2 + (z_U - z_S)^2} \tag{3.28}$$

Based on Assumption *xiv* from Chapter 1, Equation (3.28) can be approximated and rewritten in terms of the true range and a truncated first-order Taylor series, with perturbations representing the errors in $\boldsymbol{X_U}$ and $\boldsymbol{X_S}$:

$$R_{INS} = R_t + \left.\frac{\partial R_{INS}(\boldsymbol{X}_S, \boldsymbol{X}_U)}{\partial \boldsymbol{X}_U}\right|_{(\boldsymbol{X}_{S,}\boldsymbol{X}_U)_{nom}} \cdot \delta \boldsymbol{X}_U + \left.\frac{\partial R_{INS}(\boldsymbol{X}_S, \boldsymbol{X}_U)}{\partial \boldsymbol{X}_S}\right|_{(\boldsymbol{X}_{S,}\boldsymbol{X}_U)_{nom}} \cdot \delta \boldsymbol{X}_S \quad (3.29)$$

The solution for R_{INS} is found by substituting Equations (3.28) into Equation (3.29) and evaluating the partial derivatives to get [27,33].

$$\begin{aligned} R_{INS} = R_t &- \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_U \\ &+ \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_S + \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_S + \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_S \end{aligned} \quad (3.30)$$

Finally, the GPS pseudorange truth model *difference* measurement is given as:

$$\begin{aligned} \delta z_{GPS_t} &= R_{INS} - R_{GPS} \\ &= -\left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_U \\ &\quad + \left[\frac{x_S - x_U}{|R_{INS}|}\right] \cdot \delta x_S + \left[\frac{y_S - y_U}{|R_{INS}|}\right] \cdot \delta y_S + \left[\frac{z_S - z_U}{|R_{INS}|}\right] \cdot \delta z_S \\ &\quad - [1]\delta R_{trop} - [1]\delta R_{ion} - [1]\delta R_{Uclk} - [1]\delta R_{cloop} - [1]\delta R_{Sclk} + v \end{aligned} \quad (3.31)$$

The user position errors in Equation (3.31) can be derived from the first three states of the filter or truth model using an orthogonal transformation [27,33].

The filter design model for the GPS measurement will now be derived. Since the filter model does not contain the states for the errors in the satellite position, these terms are removed from the equation. The filter model measurement equation can therefore be written as:

$$\begin{aligned}\delta z_{GPS_f} &= R_{INS} - R_{GPS} \\ &= -\left[\frac{x_S - x_U}{|R_{INS}|}\right]\cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right]\cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right]\cdot \delta z_U \\ &\quad -[1]\delta R_{Uclk} + v\end{aligned} \tag{3.32}$$

The filter measurement noise variance, R, will be tuned to attain adequate performance despite the reduction in order from the truth model and despite the approximation of the truncated Taylor series. The measurement noise variances for both the filter and the truth model equations are provided in Table B-13 of Appendix B. This completes the description of the GPS measurement equations and the entire SNSM filter and truth model equations for a standard GPS-aided INS.

3.6.2.5 DGPS Measurement Model

There are four *differentially* corrected GPS measurement updates, one for each of the satellite range signals received by the SNSM filter. These measurement updates are once again *difference* measurements. First the DGPS truth model difference measurement will be fully presented, followed by a brief description of the filter measurement. The DGPS difference measurement is formed by taking the difference of the INS-calculated pseudorange, R_{INS} and actual pseudorange, R_{DGPS}.

$$\delta z_{DGPS} = R_{INS} - R_{DGPS} \tag{3.33}$$

The real pseudorange, R_{DGPS} is the sum of the true range from the user to the satellite plus all the errors in the pseudorange signal propagation. After differential corrections are applied, the measurement equation is modeled as:

$$R_{DGPS} = R_t + \delta R_{trop} + \delta R_{ion} + \delta R_{Uclk} - v \tag{3.34}$$

where

R_{DGPS} = Differentially corrected GPS pseudorange measurement, from SV to user

R_t = true range, from SV to user

δR_{trop} = range error due to tropospheric delay

δR_{ion} = range error due to ionospheric delay

δR_{Uclk} = range error due to user clock error

v = zero-mean white Gaussian measurement noise

A development analogous to equations (3.27) through (3.30) can be accomplished, and finally, the DGPS pseudorange truth model *difference* measurement is given as:

$$\begin{aligned} \delta z_{DGPS_t} &= R_{INS} - R_{DGPS} \\ &= -\left[\frac{x_S - x_U}{|R_{INS}|}\right]\cdot\delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right]\cdot\delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right]\cdot\delta z_U \\ &\quad + \left[\frac{x_S - x_U}{|R_{INS}|}\right]\cdot\delta x_S + \left[\frac{y_S - y_U}{|R_{INS}|}\right]\cdot\delta y_S + \left[\frac{z_S - z_U}{|R_{INS}|}\right]\cdot\delta z_S \\ &\quad -[1]\delta R_{trop} - [1]\delta R_{ion} - [1]\delta R_{Uclk} + v \end{aligned} \tag{3.35}$$

As before, the user position errors in Equation (3.35) can be derived from the first three states of the filter or truth model using an orthogonal transformation [27,33].

The filter design model for the DGPS measurement will now be derived. Since the filter model does not contain the states for the errors in the satellite position, these terms are removed from the equation. The filter model measurement equation can therefore be written as:

$$\delta z_{DGPS_f} = R_{INS} - R_{DGPS}$$

$$= -\left[\frac{x_S - x_U}{|R_{INS}|}\right]\cdot \delta x_U - \left[\frac{y_S - y_U}{|R_{INS}|}\right]\cdot \delta y_U - \left[\frac{z_S - z_U}{|R_{INS}|}\right]\cdot \delta z_U \qquad (3.36)$$
$$-[1]\delta R_{Uclk} + v$$

The filter measurement noise variance, *R*, will be tuned to attain adequate performance despite the reduction in order from the truth model and despite the approximation of the truncated Taylor series. The measurement noise variances for both the filter and the truth model equations are provided in Table B-13 of Appendix B. This completes the description of the DGPS measurement equations and the entire SNSM filter and truth model equations.

3.7 Chapter Summary

This chapter presents the set-up of the SNSM MatSOFE computer simulation. An introduction to MatSOFE, the WSEM software, and PROFGEN is provided. The truth model and filter model propagation and measurement equations are described for the INS/Baro, GPS, and DGPS subsystems. The INS/Baro, GPS, and DGPS truth model is located in tabular form in Appendix A. The dynamic submatrices $\mathbf{F}_{Filter}$, $\mathbf{F}_{INSt1}$, $\mathbf{F}_{INSt2}$, $\mathbf{F}_{GPS}$, $\mathbf{F}_{DGPS}$, and process noise strength and measurement noise covariance matrices for filter and truth models are presented in Appendix B Results and analysis of the SNSM simulation are presented in Chapter 4.

IV. Results and Analysis

This chapter presents results and analysis of the following items:

- Validation of MatSOFE (comparison of MatSOFE results to previous AFIT researchers' [6,10,27,33] results for the same navigation problem).
- "Rocket" trajectory (detailed Atlas IIAS flight profile) created using PROFGEN [29].
- GPS satellite ephemeris data processing procedures (GDOP, PDOP, HDOP, number of satellites, etc.) using WSEM 3.6 software [2].
- Development and simulation of three INS's in MatSOFE.
- Filter tuning process and a tuning example.
- Performance results and analysis of the GPS/DGPS-aided INS system integrations involved in this research. Table IV-1 shows these system integration comparisons.
- Chapter summary

Table IV-1. Case I-VI Integration Comparison

Case I	Case II	Case III	Case IV	Case V	Case VI
0.4 nm/hr CEP INS	0.4 nm/hr CEP INS	2.0 nm/hr CEP INS	2.0 nm/hr CEP INS	4.0 nm/hr CEP INS	4.0 nm/hr CEP INS
Single-Freq Standard GPS	Dual-Freq P-code DGPS	Single-Freq Standard GPS	Dual-Freq P-code DGPS	Single-Freq Standard GPS	Dual-Freq P-code DGPS
Barometric Altimeter	Barometric Altimeter	Barometric Altimeter	Barometric Altimeter	Barometric Altimeter	Barometric Altimeter

4.1 Validation of MATSOFE

This research is a follow-on effort of Gray [10] and Britton [11]. There are some similarities along with variations between their studies and this study. They both studied a GPS/DGPS-aided LN-93 INS [18] integrated systems for precision landing approaches. It has seen that the truth models and filter design error models they used for INS and GPS have been utilized in AFIT for more than 8 years as validated models, and consequently they are used as validated and reliable references in terms of truth and filter model for both INS and GPS/DGPS including truth/filter measurements models in this research. On the contrary, this research dealt with evaluating position accuracy for a launch vehicle rather than accurate GPS/INS integrations for a precision landing or airborne navigation application.

To accomplish the performance analysis of a GPS-aided INS system, one has to make Monte Carlo simulations and/or covariance analyses. As an international officer, the author was not eligible to make use of the MSOFE software [30]. MSOFE is restricted to use by U.S. government agencies and their U.S. contractors. At the beginning it was not a considerable concern, but as time passed, generating and validating a performance evaluation tool became the biggest issue of this study. That is because, up to now, AFIT students have been accustomed to utilize MSOFE [30] as a performance evaluation tool. For that reason, first of all, the modified MatSOFE needs to be validated as a reliable performance evaluation tool. The most effective way of validating MatSOFE was to take previous researchers' data as an input to MatSOFE. If it gives comparable outputs to the same inputs, then it would be validated.

To start, the original MatSOFE was revised to handle more complicated navigation system problems, like INS/GPS/BARO integrated navigation system with higher-state and realistic truth model definitions, rather than an INS only navigation system designed for a simple specific problem with short flight durations. It especially was not convenient for this study in terms of truth and filter model state dimensions, "Pinson error model" definition, and the GPS data used. For more information about MatSOFE and revisions made to MatSOFE, see Chapter 3.2, Introduction to MatSOFE.

The next step was generating necessary simulated INS and GPS ephemeris data (Sections 4.2 and 4.3 discuss the INS and GPS data simulation process in detail). For that reason, the exact tanker aircraft profile, *FLY_OUT* file (FLY_OUT is the output file of the PFOFGEN) is regenerated by using the provided *PROF_IN* file (PROF_IN is the input file for profile generator, PROFGEN [29]) in Gray's thesis [10]. This regenerated tanker aircraft flight profile has been used throughout the MatSOFE validation process. The first MatSOFE simulation run is accomplished by using all Gray's [10] inputs, same FLY_OUT file for tanker aircraft and the same day of simulated "true" GPS ephemeris data. A similar approach is tracked for Britton [6] inputs, concerning the cases with DGPS integration, by using the same simulation model except for implementing DGPS into system rather than standard GPS. Then the validation scope is extended utilizing Mosle's [27] and Negast's [33] inputs as well. The next and succeeding sections will discuss comparison between original outputs they have presented in their theses by utilizing MSOFE as an evaluation tool, and MatSOFE generated outputs for the same set of circumstances. By saying "the same set of circumstances", the author means utilizing the same **F**, **Q**, **H**, and **R** values as used by those previous researchers for both filter and the truth model. As mentioned earlier, the same error models for LN-93 inertial navigation system, PROFGEN output (FLY_OUT file), and simulated GPS data are also generated for comparison cases.

4.1.1 Comparison to Gray's Results [10]

A GPS-integrated navigation system, using a 0.4 nm/hr CEP INS with baro-altimeter incorporated, was a common study case to almost all previous AFIT researchers being tracked [6,10,27,33]. Gray's first research case and this research's first case are exactly the "same" study cases, except for the flight profile (PROFGEN data) used. Actually, creating a simulation case close enough to his case depends on the information he declared in his thesis. During the comparison phase, some additional information was needed to duplicate their work to reflect the same characteristics, like USOFE code (USOFE is the user-controlled portion of MSOFE), filter and truth model initial state and covariance values, etc. It is almost impossible to reproduce the identical or exactly same simulation conditions for validation purposes, but the author has confidence that the reproduced models represent their work effectively.

Figure G-1 and Figure G-2 in Appendix G show the MatSOFE-generated and MSOFE-generated outputs for the same simulation case, respectively. Only position error plots are shown in Appendix G. For more detailed comparison, see Gray's thesis [10]. Under the same simulation conditions, MatSOFE consistently generates positions errors that are almost twice as accurate as the MSOFE results. The encouraging thing about this result is that it is not close enough to say "same" in terms of position accuracy, but at least it is not worse. This phenomenon has also seen in comparison to Mosle's [27] work. The first approach to solve that problem was to regenerate the simulated GPS ephemeris data because this result has not been seen in comparison to Britton's results (Britton did the same study as Gray except for using DGPS instead of GPS). Regenerating GPS data didn't solve the problem. There are two reasonable explanations for what might cause that difference between MatSOFE results, and that of Gray's and Mosle's cases. Mosle implemented the USOFE subroutines " ORBIT" and "CALCDOP" for GPS data simulation rather than simulate true GPS ephemeris data in his research. It is believed that might cause that disagreement in comparison. In contrast, Gray was the first researcher in AFIT to put the simulated true GPS ephemeris data into practice, but there are inconsistencies found between what he stated in his thesis and what was written in his USOFE code (e.g., some of his thesis filter process noise ($\mathbf{Q}_f$) values don't agree either with his USOFE code values or with Britton's values.) That might be the explanation for the disagreement in comparison with Gray's study. Insufficient conversion from MSOFE into MatSOFE could also be accepted as another reasonable explanation to the disagreement experienced, because, as mentioned before, reproducing someone else's study environment is highly dependent on the information gathered, related to this study. The author has never been able to look at the original MSOFE code, with the result that he could not ensure the removal of any discrepancies between it and the MatSOFE implementation. MatSOFE outputs for the same case as Mosle's [27] values are presented later in Figure G-5 of Appendix G.

4.1.2 Comparison to Britton's Results [6]

The inconsistency faced in the GPS-related comparison to Gray has not been experienced with Britton's DGPS-aiding 0.4 nm/hr CEP INS case with internal baro-altimeter aiding. As a matter of fact, Britton had the same study case just like others (Gray, Mosle, and Negast) except for P-code dual frequency DGPS aiding the INS rather standard GPS. The MatSOFE-generated results for the DGPS-aided INS integrated navigation system evaluation are close enough to be declared as accurate as the MSOFE-generated one (see Figure G-3 and Figure G-4). This accuracy agreement with Britton study was convincing enough to believe that MatSOFE is a reliable performance evaluation tool for navigation system problems. For now, pronouncing that MatSOFE is an evaluation tool as powerful as MSOFE would be an extremely optimistic conclusion. But, Matlab-based MatSOFE can be upgraded to the same or even higher level as MSOFE in the future, because Matlab is quite a powerful programming language to handle difficult computations.

One common thing faced in all the comparison cases was that the previous researchers preferred slightly conservative tuning for their performance evaluations. Mosle [27] explains the reason for this choice in his thesis as: " The conservative nature of this tuning proved to be necessary to maintain good tracking of the most important navigation states." This choice of tuning may be necessary for a precise and intensive navigation problem, like the precision landing system, but fine (non-conservative) tuning is preferred for the rocket navigation problem studied in this research.

4.2 The Launch Vehicle (Atlas IIAS) Flight Profile

It was a requirement to regenerate the “Tanker” aircraft profile for validation of MatSOFE to duplicate the previous studies in MatSOFE. Therefore, a *PROF_IN* file (PROF_IN is the input file for profile generator software, PROFGEN [29]) is created first based on the inputs that Gray [10] provided in his thesis (the actual PROF_IN file used to generate Tanker profile can be found in Gray’s thesis in Appendix C of Gray’s thesis.)

Although a lot of insights into PROFGEN were gained in the Tanker aircraft profile generation, a personal interview was conducted with Dr. Tragesser [39] of the AFIT Department of Aeronautics and Astronautics and a sample rocket (Delta II) PROF_IN file was requested from Dr. Stanton Musick [31] of the Air Force Research Laboratory prior to generating the launch vehicle profile. PROFGEN reads in the PROF_IN file, and outputs a binary flight profile output variables file (user-selected variables, see Table III-1) called “FLY_OUT.asc”.

As discussed earlier, the launch vehicle is simulated by assuming it to be launched from Cape Canaveral AS, SLC-41, with the following initial conditions:

- Initial latitude $28.5^0 N$
- Initial longitude $81.0^0 W$
- Initial altitude 0 feet
- Initial ground path heading 90.0^0

All initial conditions, flight segments and durations, and user-selected input and output variables and their units, etc. are all presented in Appendix C. Also, Table IV-2 shows the actual launch operation sequence of the Atlas IIAS. With the assumption of executing launch operations from Cape Canaveral AS, two available orbit inclination options allow the choice of any inclination between $28.5^0 - 55.0^0$ or 90.0^0. The 90.0^0 inclination orbit is usually preferred for the missions which require watching every part of the Earth 39]. The author picked 28.5^0 inclination orbit as a fair choice for a satellite launch into low earth orbit (LEO).

Table IV-2. Atlas IIAS Launch Operation Sequence [15]

EVENT	Time (sec)
Liftoff	0.0
Ground-Lit SRB Burnout	54.7
Air-Lit SRB Ignition	60.0
Ground-Lit SRB Jettison	77.1
Air-Lit SRB Burnout	115.3
Air-Lit SRB Jettison	117.2
Atlas Booster Engine Cutoff	163.3
Booster Package Jettison	166.4
Payload Fairing Jettison	214.5
Atlas Sustainer Engine Cutoff	289.2
Atlas/Centaur Separation	293.3
Centaur Main Engine Start (MES1)	309.8
Centaur Main Engine Cutoff (MECO1)	584.8
Start Turn To Main Engine Start 2 (MES2)	1180.8
Centaur Main Engine Start (MES2)	1475.8
Centaur Main Engine Cutoff (MECO2)	1571.9
Start Alignment To Separation	1573.9
Begin Spinup	1691.9
Separate Spacecraft	1798.9
Start Turn To Collision& Contamination Avoidance Maneuver (CCAM)	1918.9
Centaur End Of Mission	4239.9

Note: SRB stands for Solid Rocket Booster.

4.3 Development of the Three Types of INS's

The three grades of INS (0.4, 2.0, and 4.0 nm/hr CEP INS) are assumed to be good representative of military grade, commercial grade, and less expensive (possible MEMS in a few years) Inertial Navigation Systems. For navigation problems like rocket launch, it is believed and assumed that INS and GPS sensors will always be redundant. From this assumption, and taking the chosen orbit (LEO) into account, there comes another assumption: no GPS outages. Now it can be clearly said that the GPS will be the dominant system, and overall position accuracy will be dictated by GPS accuracy performance. Despite the facts mentioned above, the three grades of INS are included in the research scope. It is believed that the results obtained in each of the cases will be a useful reference for ongoing research and development efforts.

Gray [10] says that he developed good models for the three grades of INS with the modifications he made to the initial system covariance matrix by only altering the random constant shaping filter states (and not changing the 1st order Gauss-Markov drift states, etc.). Unfortunately, inconsistencies are found between what he stated in his thesis and what was found in his USOFE code (user-controlled , problem-specific portion of MSOFE code for simulation). He says that he altered the initial covariance conditions of the system, but they are all zeroed out in his USOFE code. When what he said in his thesis is exactly applied in MatSOFE, the filter estimation goes unstable after about100 sec of flight. Then, after personal discussions with Dr. Maybeck (thesis advisor) and Dr. Musick [31], it was decided that system process noise values ($\mathbf{Q}_s$) should be altered rather than initial covariance to simulate these different grades of INS's. In this thesis, this method is applied. The truth and filter process noise values are shown in Appendix B.

4.4 Filter Tuning Process

The reduced order truth model (69 states for GPS-aided INS and 61 for DGPS-aided INS) requires a large amount of computer processing . One of the objectives in this thesis is to minimize the computer processor time (especially taking into account the fact that Matlab [37] is not a compiled language) and memory cost. For that reason, the 13-state filter model is chosen as the truth model subset (usually called the filter design model, and it should satisfactorily model the truth model). Remembering that the truth model represents the real world, the filter designer's goal is to make sure that the filter design model represents the truth model well (this process is called tuning). One of the tools in filter designer's hand is the dynamics noise strength of the white noises added to differential equations (see Eq. (3.5)). These are usually called “**Q**” values of the filter or briefly “$\mathbf{Q}_f$”. Another tool called in order to tune the filter is called measurement noises parameter “**R**”. It is actually the covariance value of the measurement noise vector “***v***”, in measurement equations (see Eq.(3.6)). In this example only “**Q**” tuning will be discussed because “**R** tuning follows a similar procedure.

Consider the position, velocity and tilt error states of Case II (0.4 nm/hr INS/DGPS/Baro-Altimeter) that were tuned using the legend in Figure IV-1.

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)

Figure IV-1. Plot Legend

In this thesis the tuning strategy followed was: tuning the three tilt error states first, then velocity, and finally the other states. When you look at Figure H-5 in Appendix H, you will see non-conservatively tuned tilt errors compared to Figure H-6 in Appendix H (the true error and the mean error $\pm\sigma_{true}$, are well within the filter computed, 0 $\pm\sigma_{filter}$, error). Figure H-5 has been tuned by decreasing **Q** value as shown in Table IV-3.

Table IV-3. Q Values for Tilt Errors (before and after fine tuning)

SMSM State Number	SNSM State Name	Q-value Before	Q-value After
4	East-Tilt	5.712e-13	1.428e-13
5	North-Tilt	4.760e-13	1.900e-13
6	Vert(Up)-Tilt	8.568e-13	0.952e-13

Figures H-1 and H-2, and Figures H-3 and H-4 show the conservative and non-conservative tuning examples for position and velocity errors, respectively. Also the "before" and "after fine tuning:" **Q** values for position and velocity are shown in Table IV-4 and, Table IV-5, respectively.

Table IV-4. Q Values for Velocity Errors (before and after fine tuning)

SMSM State Number	SNSM State Name	Q-value Before	Q-value After
7	East-velocity	9.262e-8	1.021e-8
8	North- velocity	2.573e-8	0.253e-8
9	Vert(Up)- velocity	3.607e-3	5.152e-5

Table IV-5. Q Values for Position Errors (before and after fine tuning)

SMSM State Number	SNSM State Name	Q-value Before	Q-value After
1	Latitude	7.520e-16	0.750e-16
2	Longitude	7.092e-16	0.704e-16
3	Altitude	0	0

4.5 Performance Analysis

This section discusses the results for Cases I - VI for the satellite launch profile. The plots for these cases are located in Appendix I through Appendix N.

4.5.1 Baro-Altimeter and Standard GPS Aiding Cases

As discussed before, the baro-altimeter aids the INS up to a flight altitude of 80,000 feet, whereas GPS measurements are always available for updating the inertial system. The rocket flight profile was utilized to provide real world information for the simulation cases. As explained before, it was assumed that there were no satellite vehicle (SV) outages for the entire flight of the rocket profile. Several investigations were conducted by using WSEM software determine whether any GPS outage would be experienced in any phase of the flight profile or not. The answer was definitely not.

4.5.1.1 Case I, 0.4 nm/hr CEP INS/Baro-Alt./ Std.GPS

A main concern related to this research was vertical channel instability. While the GPS sensor onboard the rocket is the redundant and the dominant navigation sensor, the obtained positioning results were more accurate than needed. For instance, while the desired vertical position accuracy was at the level of 50-80 meters, the average vertical position error achieved from the GPS-aided INS (mean error $\pm\sigma_{true}$) for Case I was less than10 feet. Table IV-6 shows the 1σ Latitude, Longitude and Altitude Errors for Case I.

Table IV-6. Case I 1σ Latitude, Longitude and Altitude Errors

SMSM Error State Number	Position Error State	Case I Mean True Error (feet)	Case I True σ_{true} Error (feet)	Case I Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	0.48	2.46	2.84
2	δlongitude	0.84	2.45	2.92
3	δaltitude	0.25	4.45	6.86

Upon careful inspection of Figure I-2, which shows baro-altimeter error, it will be noticed that somehow the filter is being told that real world errors would grow (the filter over-estimates the actual error) and it increases the filter-predicted 1σ boundaries up to almost 550 sec of flight. At that point of time filter-predicted 1σ value converges to normal values (begins to match with true 1σ value). As a matter of fact, there is nothing wrong with the actual real-world error at any time in the entire profile. The first diagnostic attempt to solve that problem was to investigate the vehicle dynamics to see whether anything was wrong with the rocket flight profile output (FLY_OUT) file or not. The same phenomenon happened for both tanker and rocket profiles. It even happened in the artificial scenario of a flight profile simulating an aircraft flying wings-level and straight ahead for 1 hour. The second attempt entailed investigating the GPS data. More than 6 different true GPS ephemeris data are simulated for random day selections. That approach was not a solution either. The encouraging aspect is that the real world errors and other states of the filter were not impacted appreciably by this erroneous 1σ value and best of all, they are not being affected by the wrong prediction by the filter of this state. Finally, after it is carefully discussed with Dr. Maybeck (thesis advisor) and Dr. Musick, it was left as is, with little anticipated impact on the performance analysis.

The plots related to each error state of Case I are shown in Appendix I. They amplify the trends already seen in Table IV-6.

4.5.1.2 Case III, 2.0 nm/hr CEP INS/ Baro-Alt/ Std.GPS

Case III simulates the same scenario with Case I except for the 2.0 nm/hr CEP INS replacing the 0.4nm/hr CEP one. The same wrong filter-predicted 1σ problem was experienced in Case III also, and will be experienced in Case V as well. Actually, that has been seen in all the GPS-aided (vs. DGPS-aided) INS simulation cases which have been carried out in this research. Because of the reasons mentioned earlier, GPS is the dominant navigation sensor of the integration, and overall position accuracy is essentially determined by GPS. So, as expected in terms of positioning, Case I and Case III provide fundamentally the same performance. But if you look at the tilt and velocity error states, it is obvious that the 0.4 nm/hr CEP INS has smaller errors when compared to the 2.0 nm/hr CEP INS. Appendix K shows the plots related to each error state of Case III. Also Table IV-7 shows the 1σ Latitude, Longitude and Altitude Errors for Case III.

Table IV-7. Case III 1σ Latitude, Longitude and Altitude Errors

SMSM Error State Number	Position Error State	Case III Mean True Error (feet)	Case III True σ_{true} Error (feet)	Case III Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	0.54	2.37	3.28
2	δlongitude	0.71	2.43	3.56
3	δaltitude	0.18	4.48	6.97

4.5.1.3 Case V, 4.0 nm/hr CEP INS/ Baro-Alt/ Std.GPS

The same consideration discussed for Case III applies for Case V too. The difference between Case V from other two GPS-aiding INS cases is that Case V simulates the 4.0 nm/hr CEP INS in the integrated system.

Appendix M. shows the plots related to each error state of Case V. Also, Table IV-8 shows the 1σ Latitude, Longitude and Altitude Errors for Case V.

Table IV-8. Case V 1σ Latitude, Longitude and Altitude Errors

SMSM Error State Number	Position Error State	Case V Mean True Error (feet)	Case V True σ_{true} Error (feet)	Case V Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	0.25	2.83	3.76
2	δlongitude	0.10	2.94	4.09
3	δaltitude	0.19	4.91	6.99

4.5.2 Baro-Altimeter and P-Code DGPS Aiding Cases

The DGPS-aided INS cases are, as expected, more accurate than GPS-aided INS cases. They are not only more accurate but also more consistent with previous AFIT researcher Britton's [6] research results than the GPS-aided INS results were with Gray's [10] work. The baro-altimeter is assumed to be functioning until 80,000 feet and DGPS measurements are always available for updating the INS. One more nice point about DGPS cases is that we don't see the strange baro-altimeter state filter prediction error that has been seen in previous section.

4.5.2.1 Case II, 0.4 nm/hr CEP INS/Baro-Alt./P-Code DGPS

A Dual-Frequency Military P-Code DGPS is assumed to be onboard for DGPS-aided case simulations. As in the GPS cases, DGPS is the dominant navigation sensor onboard. The position accuracy of DGPS will dictate overall performance of the navigation system. DGPS-aided 0.4nm/hr CEP INS simulation case results are shown in Appendix J. Table IV-9 shows 1σ Latitude, Longitude and Altitude Errors for Case II. Also for comparison purpose Britton's MSOFE generated results for the same simulation condition position errors are also presented in Table IV-10. Remember that, Britton [6], like Gray [10], studied a precision approach problem except for updating the INS with DGPS. Britton utilized the same tanker aircraft profile Gray used in his thesis, rather than a rocket launch profile as in this effort .

Table IV-9. Case II 1σ Latitude, Longitude and Altitude Errors

SMSM Error State Number	Position Error State	Case II Mean True Error (feet)	Case II True σ_{true} Error (feet)	Case II Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	0.13	1.08	1.48
2	δlongitude	0.08	0.80	1.07
3	δaltitude	0.10	1.92	3.75

Table IV-10. Britton's MSOFE-generated 1σ results for Case II

SMSM Error State Number	Position Error State	Case II True σ_{true} Error (feet)	Case II Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	1.04	1.38
2	δlongitude	0.75	1.62
3	δaltitude	4.55	5.96

4.5.2.2 Case IV, 2.0 nm/hr CEP INS/Baro-Alt./P-Code DGPS

Case IV simulates the same scenario as Case II, except for the 2.0 nm/hr CEP INS replacing the 0.4nm/hr CEP one in the system integration. Again, DGPS is the dominant navigation sensor. So, as expected in terms of positioning, Case II and Case IV provide basically the same performance. But if one considers the tilt and velocity error states, it is obvious that the 0.4 nm/hr CEP INS case has smaller errors when compared to the 2.0 nm/hr CEP INS case. Appendix L shows the plots related to each error state of Case IV, and Table IV-11 shows the 1σ Latitude, Longitude and Altitude Errors for Case IV. Also, for comparison purpose, Britton's MSOFE-generated results for the same simulation condition position errors are also presented in Table IV-12. Again, remember that Britton [6], like Gray [10], studied a precision approach problem and that he utilized the same tanker aircraft profile Gray used in his thesis.

Table IV-11. Case IV 1σ Latitude, Longitude and Altitude Errors

SMSM Error State Number	Position Error State	Case IV Mean True Error (feet)	Case IV True σ_{true} Error (feet)	Case IV Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	0.03	1.39	1.79
2	δlongitude	0.13	0.95	1.34
3	δaltitude	0.04	2.27	4.01

Table IV-12. Britton's MSOFE-generated 1σ results for Case IV

SMSM Error State Number	Position Error State	Case IV True σ_{true} Error (feet)	Case IV Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	1.29	1.69
2	δlongitude	0.94	2.09
3	δaltitude	3.69	6.37

4.5.2.3 Case VI, 4.0 nm/hr CEP INS/Baro-Alt./P-Code DGPS

The same consideration discussed for Case II and IV applies for Case VI too. The difference of Case VI from other two DGPS-aiding INS cases is that Case VI simulates the 4.0 nm/hr CEP INS in the integrated system.

Appendix N shows the plots related to each error state of Case VI, and Table IV-13 shows the 1σ Latitude, Longitude and Altitude Errors for Case VI. As before, Britton's MSOFE-generated results for the same simulation condition position errors are also presented in Table IV-14, keeping in mind that Britton [6] used a tanker aircraft profile in his research versus a rocket launch trajectory.

Table IV-13. Case VI 1σ Latitude, Longitude and Altitude Errors

SMSM Error State Number	Position Error State	Case VI Mean True Error (feet)	Case VI True σ_{true} Error (feet)	Case VI Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	0.01	1.49	2.05
2	δlongitude	0.04	1.04	1.50
3	δaltitude	0.02	2.19	3.98

Table IV-14. Britton's MSOFE-generated 1σ results for Case VI

SMSM Error State Number	Position Error State	Case VI True σ_{true} Error (feet)	Case VI Filter $0 \pm 1\sigma_{filter}$ Error (feet)
1	δlatitude	1.43	1.87
2	δlongitude	1.08	2.31
3	δaltitude	3.32	6.26

4.6 Chapter Summary

This chapter presented the results of six different MatSOFE-generated simulation cases for the rocket flight profile. Note that all cases met and exceeded the desired positioning accuracy. The overall comparison of GPS-aided INS cases and DGPS-aided INS cases are presented in Tables IV-15 and IV-16, respectively. Almost 10 months ago, when the first performance analysis cases were planned, 16 different simulation cases were constructed. But studying Carrier-Phase DGPS-aiding INS cases (3 cases, one for each grade of INS) quickly was seen to be a substantial overkill in terms of meeting position error specifications. Four cases related to implementing a star tracker as an additional measurement into the navigation system were postponed to study as the last study cases if time permitted. Three cases constructed to study some GPS outages during flight were canceled due to the gathered data out of WSEM software [2] indicating that such outages were not to be anticipated.

Table IV-15. GPS-aided INS Cases Comparison Table

	SNSM Position Error States	Mean True Error	True Error $1\sigma_{\text{TRUE}}$	Filter $0 \pm 1\sigma_{\text{FILTER}}$
CASE I	δlatitude	0.48	2.46	2.84
	δlongitude	0.84	2.45	2.92
	δaltitude	0.25	4.45	6.86
CASE III	δlatitude	0.54	2.37	3.28
	δlongitude	0.71	2.43	3.56
	δaltitude	0.18	4.48	6.97
CASE V	δlatitude	0.25	2.83	3.76
	δlongitude	0.10	2.94	4.09
	δaltitude	0.19	4.91	6.99

Table IV-16. DGPS-aided INS Cases Comparison Table

	SNSM Position Error States	Mean True Error	True Error $1\sigma_{\text{TRUE}}$	Filter $0 \pm 1\sigma_{\text{FILTER}}$
CASE II	δlatitude	0.13	1.08	1.48
	δlongitude	0.08	0.80	1.07
	δaltitude	0.10	1.92	3.75
CASE IV	δlatitude	0.03	1.39	1.79
	δlongitude	0.13	0.95	1.34
	δaltitude	0.04	2.27	4.01
CASE VI	δlatitude	0.01	1.49	2.05
	δlongitude	0.04	1.04	1.50
	δaltitude	0.02	2.19	3.98

V. Conclusions and Recommendations

5.1 Conclusion

This chapter presents recommendations and conclusions based on the assumptions made, problems faced, and the results obtained, which were presented in Chapter 4. This includes a basic introduction on the focus of the thesis, the conclusions which were drawn from the experience gained by the study cases, and recommendations for future work which may provide a better scope to MatSOFE.

This thesis focused on the investigation of using standard and differentially corrected GPS data to improve the accuracy of the integrated navigation system for a satellite launch vehicle. A major aspect of this thesis is the use of the alternative performance evaluation tool, MatSOFE, as compared to the more typically used MSOFE tool. Students who have utilized MSOFE in their theses or class projects complained that they were uncomfortable with Fortran and that most of the effort of the study was spent trying to understand the working principles of MSOFE [11]. Simulated true GPS and DGPS measurements data implementation, and U-D factorization capability are augmented to MatSOFE in this research. Also, KC-135 tanker aircraft and Atlas IIAS launch profiles are implemented (using PROFGEN [29]), along with LN-93 INS [18] error model definitions. Furthermore, many former AFIT students' theses are reconsidered and their efforts are duplicated in a Matlab environment. This research is the most comprehensive Matlab-based extended Kalman Filter integration analysis tool utilized up to this point at AFIT. All position accuracy stipulations are met in the simulations. It has also seen that DGPS provides noticeable improvement over standard

GPS. Most of the time in this research has been consumed by modifying MatSOFE to its current status, to make sure that it can handle this thesis requirements.

The position errors of all studied cases are presented in Table V-1. When we take a quick look at the results, we can easily say that all cases reached and exceeded desired position accuracy. The GPS-aided INS study cases results, namely Case I, III, and V results, are very close to each other. The reason for that is GPS is the dominant navigation sensor in the system. If we did not assume that there would be no GPS outages during flight, then these position errors would be more separated from each other, with more impact of the different grade INS's becoming evident . The same discussion is true for the DGPS-aiding INS cases.

Table V-1. Results of All Studied Cases

	SNSM Position Error States	Mean True Error	True Error $1\sigma_{\text{TRUE}}$	Filter $0 \pm 1\sigma_{\text{FILTER}}$
CASE I	δlatitude δlongitude δaltitude	0.48 0.84 0.25	2.46 2.45 4.45	2.84 2.92 6.86
CASE II	δlatitude δlongitude δaltitude	0.13 0.08 0.10	1.08 0.80 1.92	1.48 1.07 3.75
CASE III	δlatitude δlongitude δaltitude	0.54 0.71 0.18	2.37 2.43 4.48	3.28 3.56 6.97
CASE IV	δlatitude δlongitude δaltitude	0.03 0.13 0.04	1.39 0.95 2.27	1.79 1.34 4.01
CASE V	δlatitude δlongitude δaltitude	0.25 0.10 0.19	2.83 2.94 4.91	3.76 4.09 6.99
CASE VI	δlatitude δlongitude δaltitude	0.01 0.04 0.02	1.49 1.04 2.19	2.05 1.50 3.98

5.2 Recommendations

This thesis has started with examining the work Gray [10] and Britton [6] accomplished. Their achievements for GPS and DGPS integrated extended Kalman filter integrations are considered as the bases of this research.

- MatSOFE is written without any access to MSOFE. Therefore, it is believed by the author that, as a future effort MatSOFE can be made more reliable and robust Kalman filter evaluation tool, as opposed to current capabilities it has.
- The real emphasis on this research was position accuracy, and it has seen that the problem related to baro-altimeter error state has no appreciable impact on other error states. It is also believed that this phenomenon could be a result of not being able to see the original MSOFE code.
- The INS performance is a little bit disregarded since no GPS outage during flight was one of the assumptions and GPS accuracy is superior over INS.
- It was not necessary for this thesis, but for future usage, carrier-phase GPS signals can be implemented to increase the accuracy of the GPS measurements to the centimeter level for necessary navigation problems. A viable carrier-phase model was not available for the MatSOFE package during this study. If time allowed, developing and utilizing such an accurate measurement system, would, perhaps make MatSOFE one of the best and most versatile performance evaluation tools available.
- According to reviewed literature, the navigation sensor used in Atlas IIAS is only INS [15]. Also, there are some other launch vehicles using GPS-aided INS navigation system. At the very beginning of this study, it was quite clear

that even the worst case of this study (CASE I, GPS-aided 0.4 nm/hr CEP INS) would meet and exceed the position requirements. The only reason for incorporating DGPS measurements into system was to make MatSOFE a more useful and reliable filter performance evaluation tool in the future. In reality incorporating DGPS measurements would not be enough. The reference receiver positioning and means of differential corrections should also be included inside the research scope, but this was not pursued here because of the good non-differential GPS-aided INS performance.

Appendix A

Error State Definitions for the SNSM Truth and Filter Models

Tabular listings of the truth and filter models are presented. Tables A-1 and A-2 show the 39 INS states for the truth model, with the SNU 84-1 [Litton,18] and LSM [10] and DLSM [6] state numbers given for cross-reference. Table A-3 lists the GPS states and Table A-4 lists DGPS states respectively, and finally Table A-5 lists the states in the reduced-ordered SNSM filter model.

Table A-1. 39-State INS System Model: First 20 States

SNSM State	State Symbol	Definition	LITTON State	LSM State	DLSM State
1	$\delta\theta_X$	X-component of vector angle from true to computer frame	1	1	1
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame	2	2	2
3	$\delta\theta_Z$	Z-component of vector angle from true to computer frame	3	3	3
4	ϕ_X	X- component of vector angle from true to platform frame	4	4	4
5	ϕ_y	Y- component of vector angle from true to platform frame	5	5	5
6	ϕ_Z	Z- component of vector angle from true to platform frame	6	6	6
7	δV_X	X-component of error in computed velocity	7	7	7
8	δV_y	Y-component of error in computed velocity	8	8	8
9	δV_Z	Z-component of error in computed velocity	9	9	9
10	δh	Error in vehicle altitude above reference ellipsoid	10	10	10
11	δh_B	Total baro-altimeter correlated error	23	11	11
14	δh_L	Error in lagged inertial altitude	11	14	14
15	δS_3	Error in vertical channel aiding state	12	15	15
16	δS_4	Error in vertical channel aiding state	13	16	16
17	Δx_C	X-component of accelerometer and velocity quantizer correlated noise	17	17	17
18	Δy_C	Y-component of accelerometer and velocity quantizer correlated noise	18	18	18
19	Δz_C	Z-component of accelerometer and velocity quantizer correlated noise	19	19	19
20	δg_X	X-component of gravity vector errors	20	20	20
21	δg_y	Y-component of gravity vector errors	21	21	21
22	δg_Z	Z-component of gravity vector errors	22	22	22

Note: SNSM state 12 and state 13 are located in Table A-3 and Table A-4

Table A-2. 39-state INS System Model: Second 19 States

SNSM State	State Symbol	Definition	LITTON State	LSM State	DLSM State
23	b_x	X-component of gyro drift rate repeatability	30	23	23
24	b_y	Y-component of gyro drift rate repeatability	31	24	24
25	b_z	Z-component of gyro drift rate repeatability	32	25	25
26	S_{gx}	X-component of gyro scale factor error	33	26	26
27	S_{gy}	Y-component of gyro scale factor error	34	27	27
28	S_{gz}	Z-component of gyro scale factor error	35	28	28
29	∇_{bx}	X-component of accelerometer bias repeatability	48	29	29
30	∇_{by}	Y-component of accelerometer bias repeatability	49	30	30
31	∇_{bz}	Z-component of accelerometer bias repeatability	50	31	31
32	S_{ax}	X-component of accelerometer and velocity quantizer scale factor error	51	32	32
33	S_{ay}	Y-component of accelerometer and velocity quantizer scale factor error	52	33	33
34	S_{az}	Z-component of accelerometer and velocity quantizer scale factor error	53	34	34
35	S_{QAx}	X-component of accelerometer and velocity quantizer scale factor asymmetry	54	35	35
36	S_{QAy}	Y-component of accelerometer and velocity quantizer scale factor asymmetry	55	36	36
37	S_{QAz}	Z-component of accelerometer and velocity quantizer scale factor asymmetry	56	37	37
38	μ_1	X accelerometer misalignment about Z-axis	66	38	38
39	μ_2	Y accelerometer misalignment about Z-axis	67	39	39
40	μ_3	Z accelerometer misalignment about X-axis	68	40	40
41	σ_3	Z accelerometer misalignment about Y-axis	69	41	41

Table A-3. 30-State GPS System Model

SNSM State	State Symbol	Definition	LSM State	DLSM State
12	δR_{clku}	User clock bias	12	12
13	δD_{clku}	User clock drift	13	13
42	δR_{cloop1}	SV 1 code loop error	42	-
43	δR_{trop1}	SV 1 tropospheric error	43	43
44	δR_{ion1}	SV 1 ionospheric error	44	44
45	δR_{clksv1}	SV 1 clock error	45	-
46	δx_{sv1}	SV 1 x-component of position error	46	46
47	δy_{sv1}	SV 1 y-component of position error	47	47
48	δz_{sv1}	SV 1 z-component of position error	48	48
49	δR_{cloop2}	SV 2 code loop error	49	-
50	δR_{trop2}	SV 2 tropospheric error	50	50
51	δR_{ion2}	SV 2 ionospheric error	51	51
52	δR_{clksv2}	SV 2 clock error	52	-
53	δx_{sv2}	SV 2 x-component of position error	53	53
54	δy_{sv2}	SV 2 y-component of position error	54	54
55	δz_{sv2}	SV 2 z-component of position error	55	55
56	δR_{cloop3}	SV 3 code loop error	56	-
57	δR_{trop3}	SV 3 tropospheric error	57	57
58	δR_{ion3}	SV 3 ionospheric error	58	58
59	δR_{clksv3}	SV 3 clock error	59	-
60	δx_{sv3}	SV 3 x-component of position error	60	60
61	δy_{sv3}	SV 3 y-component of position error	61	61
62	δz_{sv3}	SV 3 z-component of position error	62	62
63	δR_{cloop1}	SV 4 code loop error	63	-
64	δR_{trop4}	SV 4 tropospheric error	64	64
65	δR_{ion4}	SV 4 ionospheric error	65	65
66	δR_{clksv4}	SV 4 clock error	66	-
67	δx_{sv4}	SV 4 x-component of position error	67	67
68	δy_{sv4}	SV 4 y-component of position error	68	68
69	δz_{sv4}	SV 4 z-component of position error	69	69

Table A-4. 22-State DGPS System Model

SNSM State	State Symbol	Definition	LSM State	DLSM State
12	δR_{clku}	User clock bias	12	12
13	δD_{clku}	User clock drift	13	13
42	δR_{trop1}	SV 1 tropospheric error	43	43
43	δR_{ion1}	SV 1 ionospheric error	44	44
44	δx_{sv1}	SV 1 x-component of position error	46	46
45	δy_{sv1}	SV 1 y-component of position error	47	47
46	δz_{sv1}	SV 1 z-component of position error	48	48
47	δR_{trop2}	SV 2 tropospheric error	50	50
48	δR_{ion2}	SV 2 ionospheric error	51	51
49	δx_{sv2}	SV 2 x-component of position error	53	53
50	δy_{sv2}	SV 2 y-component of position error	54	54
51	δz_{sv2}	SV 2 z-component of position error	55	55
52	δR_{trop3}	SV 3 tropospheric error	57	57
53	δR_{ion3}	SV 3 ionospheric error	58	58
54	δx_{sv3}	SV 3 x-component of position error	60	60
55	δy_{sv3}	SV 3 y-component of position error	61	61
56	δz_{sv3}	SV 3 z-component of position error	62	62
57	δR_{trop4}	SV 4 tropospheric error	64	64
58	δR_{ion4}	SV 4 ionospheric error	65	65
59	δx_{sv4}	SV 4 x-component of position error	67	67
60	δy_{sv4}	SV 4 y-component of position error	68	68
61	δz_{sv4}	SV 4 z-component of position error	69	69

Table A-5. 13-State Reduced-Order Filter Model

SNSM State	State Symbol	Definition	LSM State	DLSM State
1	$\delta\theta_x$	X-component of vector angle from true to computer frame	1	1
2	$\delta\theta_y$	Y-component of vector angle from true to computer frame	2	2
3	$\delta\theta_z$	Z-component of vector angle from true to computer frame	3	3
4	ϕ_x	X- component of vector angle from true to platform frame	4	4
5	ϕ_y	Y- component of vector angle from true to platform frame	5	5
6	ϕ_z	Z- component of vector angle from true to platform frame	6	6
7	δV_x	X-component of error in computed velocity	7	7
8	δV_y	Y-component of error in computed velocity	8	8
9	δV_z	Z-component of error in computed velocity	9	9
10	δh	Error in vehicle altitude above reference ellipsoid	10	10
11	δh_B	Total baro-altimeter correlated error	11	11
12	δclk_b	User clock bias	12	12
13	δclk_{dr}	User clock drift	13	13

Appendix B

Dynamics Matrices and Noise Values

B.1 Definition of Dynamics Matrices

In Chapter 3, the truth and filter model dynamics are defined by the submatrices, $\mathbf{F}_{Filter}$, $\mathbf{F}_{INS_{t1}}$, $\mathbf{F}_{INS_{t2}}$, $\mathbf{F}_{GPS_t}$, and $\mathbf{F}_{DGPS_t}$ of Equation (3.5). The $\mathbf{F}_{Filter}$ represents the filter dynamics matrix, which is also a submatrix of the larger truth model dynamics matrix [27]. The other three matrices represent the additional truth model non-zero portions of the **F** matrix that simulate the real world [27,33]. Tables B-2, B-3, B-4, B-5, and B-6 contain the non-zero elements of the dynamics submatrices $\mathbf{F}_{Filter}$, $\mathbf{F}_{INS_{t1}}$, $\mathbf{F}_{INS_{t2}}$, $\mathbf{F}_{GPS_t}$, and $\mathbf{F}_{DGPS_t}$, respectively. All undeclared variables shown in the following tables are defined in the LN-93 technical report, along with their units [18,27]. The structure of the dynamics matrices in this chapter correspond to the truth model state definitions in Appendix A and to the AFIT thesis (LSM/DLSM model) by [6,10]. The notation used in Table B-2 through Table B-4in this Appendix is defined in Table B-1.

Table B-1. Notation of Variables used in Table B-2 to Table B-4

ρ_x, ρ_y, ρ_z	Components of angular rate of navigation frame with respect to the earth (craft rate), coordinatized in Litton True frame
$\Omega_x, \Omega_y, \Omega_z$	Components of earth sidereal rate vector (earth rate), coordinatized in Litton ECEF, with respect to inertial space (15.041067 deg/hr)
a	Equatorial radius of the earth (6378388 meters)
g_o	Equatorial gravity magnitude (32.08744 ft/sec^2)
$\omega_{itx}, \omega_{ity}, \omega_{itz}$	Components of angular rate of body frame with respect to inertial space (spatial rate), coordinatized in Litton True frame
V_x, V_y, V_z	Components of vehicle velocity vector with respect to earth-fixed coordinates
A_x, A_y, A_z	Components of specific force, coordinatized in Litton True frame
C_{RX}, C_{RY}	Components of earth spheroid inverse radii of curvature
$\omega_{ibx}, \omega_{iby}, \omega_{ibz}$	Components of angular rate of body frame with respect to inertial space (spatial rate), coordinatized in Litton Body frame
$C_{i,j}$	Elements of the transformation matrix C_{body}^{nav}
$\beta\delta_{hc}$	Barometer inverse correlation time (600 seconds)
$\beta\nabla_{xc}, \beta\nabla_{yc}, \beta\nabla_{zc}$	Accelerometer inverse correlation time constants (5 minutes)
$\beta\delta_{gx}, \beta\delta_{gy}, \beta\delta_{gz}$	Gravity vector error inverse correlation time constants (Velocity/correlation distance)
$\sigma_{\delta hc}^2$	Variance of barometer correlated noise
$\sigma_{\nabla xc}^2, \sigma_{\nabla yc}^2, \sigma_{\nabla zc}^2$	Variances of accelerometer correlated noise
$\sigma_{\delta gx}^2, \sigma_{\delta gy}^2, \sigma_{\delta gz}^2$	Variances of gravity vector correlated noise
$\sigma_{\eta bx}^2, \sigma_{\eta by}^2, \sigma_{\eta bz}^2$	Power spectral density value of gyro drift rate white noise
$\sigma_{\eta Ax}^2, \sigma_{\eta Ay}^2, \sigma_{\eta Az}^2$	Power spectral density value of accelerometer white noise
k_1, k_2, k_3, k_4	Vertical channel gains (see figure 2 of [18])

Table B-2. Elements of the Dynamics Submatrix $\mathbf{F}_{Filter}$

Element	Variable		Element	Variable
(1,3)	-ρy		(1,8)	-CRY
(2,3)	ρx		(2,7)	CRX
(3,1)	ρy		(3,2)	-ρx
(4,2)	-Ωz		(4,3)	Ωy
(4,5)	ωitz		(4,6)	-ωity
(4,8)	-CRY		(5,1)	Ωz
(5,3)	-Ωx		(5,4)	-ωitz
(5,6)	ωitx		(5,7)	CRX
(6,1)	-Ωy		(6,2)	Ωx
(6,4)	ωity		(6,5)	-ωitx
(7,1)	-2VyΩy-2VzΩz		(7,2)	2VyΩx
(7,3)	2VzΩx		(7,5)	-Az
(7,6)	Ay		(7,7)	-VzCRX
(7,8)	2Ωz		(7,9)	-ρy-2Ωy
(8,1)	2VxΩy		(8,2)	2VxΩx-2VzΩz
(8,3)	2VzΩy		(8,4)	Az
(8,6)	-Ax		(8,7)	-2Ωz
(8,8)	-VzCRY		(8,9)	ρx+2Ωx
(9,1)	2VxΩz		(9,2)	2VyΩz
(9,3)	-2VyΩy-2VxΩx		(9,4)	-Ay
(9,5)	Ax		(9,7)	ρy2Ωy+VxCRX
(9,8)	-ρx-2Ωx+VyCRY		(9,10)	2go/a
(10,9)	1		(9,11)	k_2
(10,11)	k_1		(11,11)	-βδhc
(12,13)	1		--------	--------

Table B-3. Elements of the Dynamics Submatrix $\mathbf{F}_{INS_{t1}}$

Element	Variable		Element	Variable		Element	Variable
(9,14)	$-k_2$		(9,15)	-1		(9,16)	k_2
--------	--------		(10,14)	$-k_1$		(10,16)	k_1-1
(14,10)	1		(14,14)	-1		(15,14)	k_3
(15,16)	$-k_3$		(16,10)	k_4		(16,14)	$-k_4$
(7,17)	C_{11}		(7,18)	C_{12}		(7,19)	C_{13}
(7,20)	1		(8,17)	C_{21}		(8,18)	C_{22}
(8,19)	C_{23}		(8,21)	1		(9,17)	C_{31}
(9,18)	C_{32}		(9,19)	C_{33}		(9,22)	1
(16,11)	$k_4/600$		(16,16)	k_4-1		(15,11)	$-k_3$
(4,23)	C_{11}		(4,24)	C_{12}		(4,25)	C_{13}
(4,26)	$C_{11}\omega_{ibx}$		(4,27)	$C_{12}\omega_{iby}$		(4,28)	$C_{13}\omega_{ibz}$
(5,23)	C_{21}		(5,24)	C_{22}		(5,25)	C_{23}
(5,26)	$C_{21}\omega_{ibx}$		(5,27)	$C_{22}\omega_{iby}$		(5,28)	$C_{23}\omega_{iby}$
(6,23)	C_{31}		(6,24)	C_{32}		(6,25)	C_{33}
(6,26)	$C_{31}\omega_{ibx}$		(6,27)	$C_{32}\omega_{iby}$		(6,28)	$C_{33}\omega_{iby}$
(7,29)	C_{11}		(7,30)	C_{12}		(7,31)	C_{13}
(7,32)	$C_{11}A_x^B$		(7,33)	$C_{12}A_y^B$		(7,34)	$C_{13}A_z^{B'}$
(7,35)	$C_{11}\left\|A_x^B\right\|$		(7,36)	$C_{12}\left\|A_y^B\right\|$		(7,37)	$C_{13}\left\|A_z^{B'}\right\|$
(7,38)	$C_{12}A_y^B$		(7,39)	$C_{11}A_x^B$		(7,40)	$C_{13}A_y^B$
(7,41)	$C_{13}A_x^B$		(8,29)	C_{21}		(8,30)	C_{22}
(8,31)	C_{23}		(8,32)	$C_{21}A_x^B$		(8,33)	$C_{22}A_y^B$
(8,34)	$C_{23}A_z^{B'}$		(8,35)	$C_{21}\left\|A_x^B\right\|$		(8,36)	$C_{22}\left\|A_y^B\right\|$
(8,37)	$C_{23}\left\|A_z^{B'}\right\|$		(8,38)	$C_{22}A_y^B$		(8,39)	$C_{21}A_x^B$
(8,40)	$C_{23}A_y^B$		(8,41)	$C_{23}A_x^B$		(9,29)	C_{31}
(9,30)	C_{32}		(9,31)	C_{33}		(9,32)	$C_{31}A_x^B$
(9,33)	$C_{32}A_y^B$		(9,34)	$C_{33}\,A_z^{B'}$		(9,35)	$C_{31}\left\|A_x^B\right\|$
(9,36)	$C_{32}\left\|A_y^B\right\|$		(9,37)	$C_{33}\left\|A_z^{B'}\right\|$		(9,38)	$C_{31}A_x^B$
(9,39)	$-C_{32}A_x^B$		(9,40)	$C_{33}A_y^B$		(9,41)	$C_{33}A_x^B$

Table B-4. Elements of the Dynamics Submatrix $\mathbf{F}_{INS_{t2}}$

Element	Variable
(17,17)	-β∇xc = -3.33E-3 $\sec^{-1}$
(18,18)	-β∇yc = -3.33E-3 $\sec^{-1}$
(19,19)	-β∇zc = -3.33E-3 $\sec^{-1}$
(20,20)	-βδgx = -8.22E-6*$\sqrt{V_x^2 + V_y^2 + V_z^2} \quad sec^{-1}$
(21,21)	-βδgy = -8.22E-6*$\sqrt{V_x^2 + V_y^2 + V_z^2} \quad sec^{-1}$
(22,22)	-βδgz = -8.22E-6*$\sqrt{V_x^2 + V_y^2 + V_z^2} \quad sec^{-1}$

Note: For the above element definitions (Tables B-2, 3, 4) $t_0 = 0$

-β∇xc,yc,zc = Accelerometer inverse time constant (5 min)

-βδgx,gy,gz = Gravity vector inverse correlation time constant (V/20 NM)

-βδhc = Barometer inverse correlation time (600 sec)

$A_{x,y,z}^{B}$ = Components of acceleration in the body frame

$A_z^{B'}$ = Specific force component

C = Transformation matrix from body frame to navigation frame, C_b^n

Table B-5. Elements of the Dynamics Submatrix $\mathbf{F}_{GPS_t}$

Element	Variable		Element	Variable		Element	Variable
(42,42)	-1 $ft^2 / \sec$		(43,43)	-1/500 $ft^2 / \sec$		(44,44)	-1/1500 $ft^2 / \sec$
(49,49)	-1 $ft^2 / \sec$		(50,50)	-1/500 $ft^2 / \sec$		(51,51)	-1/1500 $ft^2 / \sec$
(56,56)	-1 $ft^2 / \sec$		(57,57)	-1/500 $ft^2 / \sec$		(58,58)	-1/1500 $ft^2 / \sec$
(63,63)	-1 $ft^2 / \sec$		(64,64)	-1/500 $ft^2 / \sec$		(65,65)	-1/1500 $ft^2 / \sec$

Table B-6. Elements of the Dynamics Submatrix $\mathbf{F}_{DGPS_t}$

Element	Variable		Element	Variable
(43,43)	-1/500 $ft^2 / \sec$		(44,44)	-1/1500 $ft^2 / \sec$
(50,50)	-1/500 $ft^2 / \sec$		(51,51)	-1/1500 $ft^2 / \sec$
(57,57)	-1/500 $ft^2 / \sec$		(58,58)	-1/1500 $ft^2 / \sec$
(64,64)	-1/500 $ft^2 / \sec$		(65,65)	-1/1500 $ft^2 / \sec$

B.2 Elements of the Process Noise and Measurement Noise Matrices

This section defines the dynamic noise strengths and measurement noise variances for the truth and filter models. The truth model non-zero dynamics noise strengths are defined in Tables B-7 through B-12. These noise strengths correspond to the driving noises $\boldsymbol{w}_{INS_t}$, $\boldsymbol{w}_{GPS_t}$, $\boldsymbol{w}_{DGPS_t}$ and $\boldsymbol{w}_{Filter}$ in Equation (3.5). Note that the (4,4) through (9,9) σ^2 terms in Table B-7 are variable names as defined in the Litton technical report [18] and do not represent variance terms typically associated with the notation σ^2 [6,10]. They are actually the heights of Power Spectral Density (PSD) functions associated with each white noise. However, the σ^2 terms in "$2\beta_i\ \sigma_i^2$" in (11,11) through (24,24) are real variances (of the outputs of first order lag shaping filters). The filter dynamics driving noise terms implemented after filter tuning for each respective INS integration (0.4 nm/hr, 2.0 nm/hr and 4.0 nm/hr) is listed in Table B-10, Table B-11, and Table B-12. Finally, the measurement noise variances used in the truth and filter models are presented in Table B-13.

Table B-7. Elements of Truth Model Process Noise Submatrix for the INS Truth Model

Element	Variable		Element	Variable		Element	Variable
(4,4)	$\sigma^2_{\eta_{bx}}$ = 190.4E-15 $[\,ft^2/\sec^5]$		(5,5)	$\sigma^2_{\eta_{by}}$ = 190.4E-15 $[\,ft^2/\sec^5]$		(6,6)	$\sigma^2_{\eta_{bz}}$ = 190.4E-15 $[\,ft^2/\sec^5]$
(7,7)	$\sigma^2_{\eta_{Ax}}$ = 102.9E-9 $[\,ft^2/\sec^5]$		(8,8)	$\sigma^2_{\eta_{Ay}}$ = 102.9E-9 $[\,ft^2/\sec^5]$		(9,9)	$\sigma^2_{\eta_{Az}}$ = 102.9E-9 $[\,ft^2/\sec^5]$
(19,19)	$2\beta_{\nabla xc}\sigma^2_{\nabla xc}$ = 2.75E-11 $[\,ft^2/\sec^5]$		(20,20)	$2\beta_{\nabla yc}\sigma^2_{\nabla yc}$ = 2.75E-11 $[\,ft^2/\sec^5]$		(21,21)	$2\beta_{\nabla zc}\sigma^2_{\nabla zc}$ = 2.75E-11 $[\,ft^2/\sec^5]$
(22,22)	$2\beta_{\delta gx}\sigma^2_{\delta gx}$ = 3.10E-13 $[\,ft^2/\sec^5]$		(23,23)	$2\beta_{\delta gy}\sigma^2_{\delta gy}$ = 3.10E-13 $[\,ft^2/\sec^5]$		(24,24)	$2\beta_{\delta gz}\sigma^2_{\delta gz}$ = 3.10E-13 $[\,ft^2/\sec^5]$
			(11,11)	$2\beta_{\delta hc}\sigma^2_{\delta hc}$ = 33.34 $[\,ft^2/\sec\,]$			

Table B-8. Elements of Truth Model Process Noise for GPS States

Element	Variable		Element	Variable		Element	Variable
(42,42)	0.5 $ft^2 / \sec$		(43,43)	0.004 $ft^2 / \sec$		(44,44)	0.004 $ft^2 / \sec$
(49,49)	0.5 $ft^2 / \sec$		(50,50)	0.004 $ft^2 / \sec$		(51,51)	0.004 $ft^2 / \sec$
(56,56)	0.5 $ft^2 / \sec$		(57,57)	0.004 $ft^2 / \sec$		(58,58)	0.004 $ft^2 / \sec$
(63,63)	0.5 $ft^2 / \sec$		(64,64)	0.004 $ft^2 / \sec$		(65,65)	0.004 $ft^2 / \sec$

Table B-9. Elements of Truth Model Process Noise for DGPS States

Element	Variable		Element	Variable
(43,43)	0.001 $ft^2 / \sec$		(44,44)	0.0004 $ft^2 / \sec$
(50,50)	0.001 $ft^2 / \sec$		(51,51)	0.0004 $ft^2 / \sec$
(57,57)	0.001 $ft^2 / \sec$		(58,58)	0.0004 $ft^2 / \sec$
(64,64)	0.001 $ft^2 / \sec$		(65,65)	0.0004 $ft^2 / \sec$

Table B-10. Filter Process Noise Q Values for Case Using the 0.4 nm/hr INS

Element	Variable		Element	Variable
(1,1)	7.5E-16 rad^2/sec		(2,2)	7.0E-16 rad^2/sec
(3,3)	0.0 rad^2/sec		(4,4)	190.4E-15 rad^2/sec
(5,5)	190.4E-15 rad^2/sec		(6,6)	85.2E-15 rad^2/sec
(7,7)	5.15E-8 ft^2/sec^3		(8,8)	2.06E-8 ft^2/sec^3
(9,9)	25,000.0 ft^2/sec^3		(10,10)	450.0 ft^2/sec
(11,11)	833.5 ft^2/sec		(12,12)	3.5 ft^2/sec
(13,13)	5.0E-15 ft^2/sec^3			

Table B-11. Filter Process Noise Q Values for Cases Using the 2.0 nm/hr INS

Element	Variable		Element	Variable
(1,1)	7.5E-16 rad^2/sec		(2,2)	7.5E-16 rad^2/sec
(3,3)	0.0 rad^2/sec		(4,4)	2.85E-11 rad^2/sec
(5,5)	2.85E-11 rad^2/sec		(6,6)	2.85E-9 rad^2/sec
(7,7)	2.06E-5 ft^2/sec^3		(8,8)	2.06E-5 ft^2/sec^3
(9,9)	5,000.0 ft^2/sec^3		(10,10)	850.0 ft^2/sec
(11,11)	833.5 ft^2/sec		(12,12)	3.5 ft^2/sec
(13,13)	5.0E-15 ft^2/sec^3			

Table B-12. Filter Process Noise Q Values for Cases Using the 4.0 nm/hr INS

Element	Variable		Element	Variable
(1,1)	7.5E-16 rad^2/sec		(2,2)	7.5E-16 rad^2/sec
(3,3)	0.0 rad^2/sec		(4,4)	3.33E-10 rad^2/sec
(5,5)	3.33E-10 rad^2/sec		(6,6)	5.7E-13 rad^2/sec
(7,7)	2.57E-5 ft^2/sec^3		(8,8)	2.57E-5 ft^2/sec^3
(9,9)	5,000.0 ft^2/sec^3		(10,10)	700 ft^2/sec
(11,11)	933.5 ft^2/sec		(12,12)	4.0 ft^2/sec
(13,13)	5.0E-16 ft^2/sec^3			

Table B-13. Truth and Filter Measurement Noise R Values for Cases

Measurement	Truth Noise	Filter Noise
Satellite Vehicles (GPS)	16 ft^2	75 ft^2
Satellite Vehicles (DGPS)	9 ft^2	30 ft^2

Appendix C

PROF_IN Input File

```
"================================   prof_in ==================================
 GROUP 1, General Information
 -----------------------------------------------------------------------------

   * prof_in is the input file for PROFGEN.  This self-documenting file
     contains controls that specify flight conditions and maneuvers.

     prof_in is divided into three GROUPs with contents as follows:
       GROUP 1:  General Information;
       GROUP 2:  Problem Control Parameters, echoed in Table 1 of PROF_OUT;
       GROUP 3:  Segment Control Parameters, echoed in Table 2 of PROF_OUT.
     All data are entered in list directed format.

   * Each GROUP is composed of a REMINDER followed by several parameters.
     The REMINDER describes the GROUP and should not be altered.
     Assign parameter values appropriate to your specific problem.

   * In GROUPs 1 and 2, parameters are presented in a NOTE-value style.
     The NOTE, delimited in single quotes, briefly defines the parameter
     and the value follows the NOTE.  Together they look like this:
          'NAME   TYPE   DESCRIPTION   UNITS   DEFAULT'     value

   * In GROUP 3, parameters are presented in a worksheet style.
 ============================================================================="

'PTITLE  CH*80  PROFGEN run title                           - "no title" '
'Atlas IIAS Two-stage rocket mission, Cape Canaveral to low-Earth orbit, low
fidelity'

"=============================================================================
 GROUP 2, Problem Control Parameters
 -----------------------------------------------------------------------------
   * Logical parameters IN_MET, PO_MET and FO_MET specify which of two units
     systems to use for I/O.  When true, meters, kilograms and seconds (m_kg_s)
     are used, and when false, feet, pounds-mass and seconds (ft_p_s) are used.

   * This table summarizes how the three XX_MET parameters are applied:
     --------------------     --------------------     --------------------
     IN_MET |  prof_in        PO_MET |  PROF_OUT       FO_MET |  FLY_OUT
     -------+------------     -------+------------     -------+------------
     true   |  m_kg_s_deg     true   |  m_s_deg        true   |  m_s_rad
     false  |  ft_p_s_deg     false  |  ft_s_deg       false  |  ft_s_rad
     --------------------     --------------------     --------------------
```

Figure C-1. PROF_IN Input File for PROFGEN

```
     * Parameters PO_VAR and FO_VAR specify which of 53 variables to output
       to PROF_OUT and FLY_OUT.  Indices for all 53 available variables are
       given in the following ordered list:
       ------------------------------------------------------------------------
        1       GLON   GLAT   ALT    CLON   ALPHA  HEAD   ROLL   PITCH  W_HDG
       10       dGLON  dGLAT  dALT   dCLON  dALPHA dHEAD  dROLL  dPITCH dW_HDG
       19       Vs     dVs
       21       Ri     Vi     VIi    GNi    FIi    WBIi   Cib    Cie    WEIe
       30       Re     Ve     VIe    Ge     FIe    WBIe   Ceb    Cen    WNEn
       39       Rn     Vn     VIn    Gn     FIn    WBIn   Cnb    Cni    WBNb
       48       Rb     Vb     VIb    Gb     FIb    WBIb


  -----------------------------------------------------------------------

   NAME    TYPE  DESCRIPTION                       UNITS    DEFAULT       value
   ==============================================================================="

  "Problem specs
   -------------------------------------------------------------------------------"
  'IN_MET  LG    prof_in metric, else British        -       .T.       '  /
  'WAMECH  IN    wander angle mechanization index:
                  1 constant alpha   2 alpha wander
                  3 unipolar         4 free azimuth  -        1        '  2
  'RHUMB   LG    rhumb line path, else great circle  -       .T.       '  .f.
  'RTOL    DP    relative tolerance for integration  -     1.D-12      '  /
  'ATOL    DP    absolute tolerance for integration  -     1.D-12      '  /


  "Earth constants
   -------------------------------------------------------------------------------"
  'ESQ     DP    earth eccentricity squared         -  6.69437999014D-3'  /
  'REQ     DP    earth semimajor axis              m     6.378137D6   '  /
  'WEI     DP    earth sidereal rate             rad/s  7.292115D-5  '  /
  'GM      DP    earth gravitational constant      m3/s2 3.986004418D14'  /
  'GEE     DP    scales PACC&TACC to accel units  m/s2/gee    9.8      '  /
  'RHOSL   DP    air density at sea level         kg/m3     1.225     '  /
  'ZETA    DP    exponent in air density function,
                  rho = RHOSL * exp(-ZETA * alt)    1/m    1.1385D-4   '  /


  "Vehicle specs
   -------------------------------------------------------------------------------"
  'ROLRAT  DP    maximum roll rate                 deg/s     1.D0      '  /
  !   no vehicle parameter is used
  'ROLTC   DP    roll-axis time constant            sec      1.D0      '  /
  'VMASS   DP    vehicle mass                       kg       1.D0      '  /
  'RAREA   DP    reference area for drag            m2       1.D0      '  /
  'CDRAG   DP    coefficient of drag                 -       1.D0      '  /
```

Figure C-1 (cont).   PROF_IN Input File for PROFGEN

```
"State initial conditions
 ----------------------------------------------------------------------------"
'TSTART  DP    initial time of trajectory         sec      0.D0      '  /
'VsO     DP    initial speed                      m/s      0.D0      '  /
'ROLLO   DP    initial roll angle                 deg      0.D0      '  /
'PITCHO  DP    initial pitch angle                deg      0.D0      '  89.
'HEADO   DP    initial ground path heading        deg      0.D0      '  90.
'ALPHAO  DP    initial wander angle               deg      0.D0      '  /
'GLATO   DP    initial geodetic latitude          deg      0.D0      '  28.5
'GLONO   DP    initial geodetic longitude         deg      0.D0      ' -81.0
'CLONO   DP    initial celestial longitude        deg      0.D0      '  /
'ALTO    DP    initial altitude                    m       0.D0      '  /


"Output controls
 ----------------------------------------------------------------------------"
'PO_X    IN    print (PROF_OUT file) control       -        1        '  1
'FO_X    IN    write (FLY_OUT  file) control       -        0        '  1
'PO_MET  LG    PROF_OUT metric, else British       -       .T.       '  .f.
'FO_MET  LG    FLY_OUT metric, else British        -       .T.       '  .f.
'DELR    DP    spread in roll between outputs     deg      0.D0      '  /
'DELP    DP    spread in pitch between outputs    deg      0.D0      '  /
'DELY    DP    spread in heading between outputs  deg      0.D0      '  /
'PO_SEP  DP    min separation in PROF_OUT outputs sec      0.D0      '  /
'FO_SEP  DP    min separation in FLY_OUT outputs  sec      0.D0      '  /

'WGSaxes LG    coordinate frame set                -       .T.
             true:  e= WGS-84, g= E-N-U, ... b= nose-rt_wing-down
             false: e= SNU-84, g= N-W-U, ... b= nose-rt_wing-down '  /

'PO_VAR IN(53) indices of variables to PROF_OUT            53*-1     '
        1,2,3,40,43,5,7,8,9,14,16,17,18/  { lon,lat,alt,Vw,FIw,
alfa,rol,pit,yaw,dalfa,droll,dpitch,dyaw }

'FO_VAR IN(53) indices of variables to FLY_OUT             53*-1     '
        1,2,3,40,43,5,7,8,9,14,16,17,18/  { lon,lat,alt,Vw,FIw,
alfa,rol,pit,yaw,dalfa,droll,dpitch,dyaw }


"==============================================================================
 GROUP 3, Segment Control Parameters
 -----------------------------------------------------------------------------
   * Each pair of lines in GROUP 3 defines a segment in the flight profile.
     The first line of the pair is the segment title, and the second line
     specifies nine parameters that define the maneuver.  Up to MXSEG
     segments are allowed (see file dimens.prm).
```

Figure C-1 (cont).   PROF_IN Input File for PROFGEN

```
   * Parameter Definitions
     ----------------------------------------------------------------------
     MANUVR   CH*4()   maneuver type: strt, roll, vert, horz, jink, fall
     SEGLNT   DP()     segment length
     PACC     DP()     path acceleration
     TACC     DP()     turn acceleration, maximum centrifugal component
     DELHED   DP()     desired change in heading angle for horizontal turn,
                         perturbation in heading angle for jinking maneuver
     DELPIT   DP()     desired change in pitch angle for vertical turn,
                         period of ground wave for jinking maneuver
     DELROL   DP()     desired change in roll angle for roll maneuver
     PO_DT    DP()     time interval for formatted prints to PROF_OUT
     FO_DT    DP()     time interval for unformatted writes to FLY_OUT
     ----------------------------------------------------------------------

   * Parameter Values

 =============================================================================
 MANUVR SEGLNT  PACC    TACC   DELHED DELPIT DELROL   PO_DT    FO_DT
  -      sec    gee     gee    deg    deg-s  deg      sec      sec
 ============================================================================="

'1 Ground-Lit SRB Ignition and Liftoff'
's'   1.     1.0    0.     0.     0.     0.     1.     1.

'2 Ground-Lit SRB Burnout & Air-lit SRB Ignition'
'v'   60     1.9    0.5    0.     -5.    0.     5.     1.

'3 Air-Lit SRB  Burnout & Jettison'
'v'   57.    0.25   2.22   0.     -20.   0.     5.     1.

'4 Atlas Booster Engine Cutoff and Booster Pacakage Jettison'
'v'   50.    0.2    2.22   0.   -20.    0.     5.     1.

=============================================================================
 MANUVR SEGLNT  PACC    TACC   DELHED DELPIT DELROL   PO_DT    FO_DT
  -      sec    gee     gee    deg    deg-s  deg      sec      sec
 ============================================================================="

'5 Payload Fairing Jettison'

'v'   48.    0.18   1.1    0.     -5.    0.     5.     1.

'6 Sustainer Engine Cutoff '
'v'   85.    0.4    1.3    0.     -10.   0.     10.    1.


'7 Atlas/Centaur Seperation'
'v'   192.   0.3    1.7    0.     -20.   0.     50.    1.

'8 Centaur Main Engine Start - Cutoff'
'v'   985.   0.3    0.8    0.     -10.   0.     50.    1.
```

Figure C-1 (cont).   PROF_IN Input File for PROFGEN

Appendix D

Atlas IIAS Rocket Profile Plots

Although this Appendix is named *Atlas IIAS Rocket Profile Plots,* The Tanker Aircraft Flight Profile is also included in this section (Figures D-8 and D-9), because it is used in this research for MatSOFE validation purposes. Notice that in Figure D-1, the rocket is losing altitude after about 600 sec. of flight, which seems surprising; but in reality, it shows *correctly* that a peak altitude is reached and the rocket is making an orbit entry maneuver.

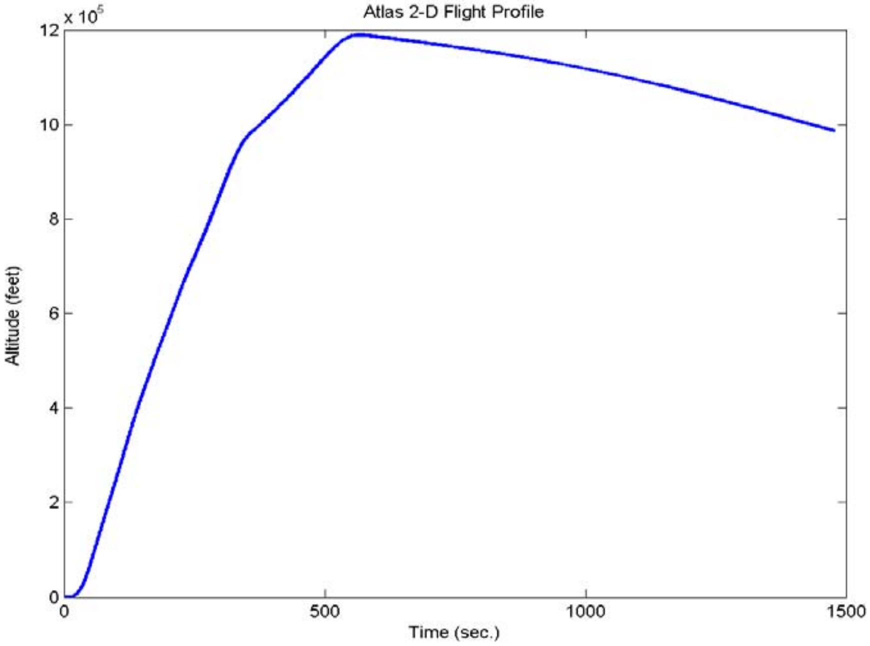


Figure D-1. 2-D Rocket Flight Profile (Alt vs. Time)

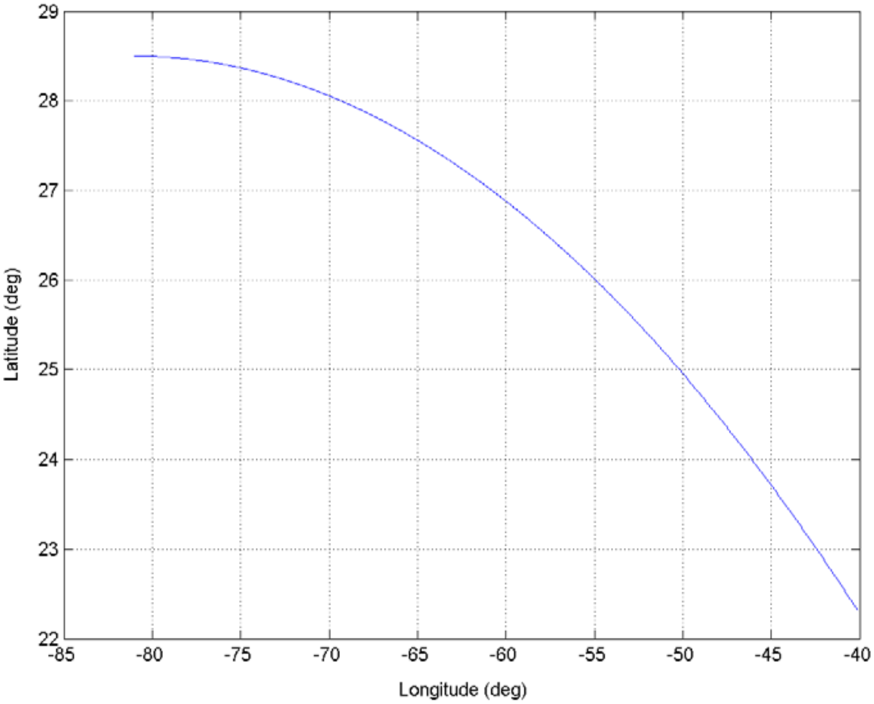


Figure D-2. 2-D Rocket Flight Profile (Lat vs. Lon)

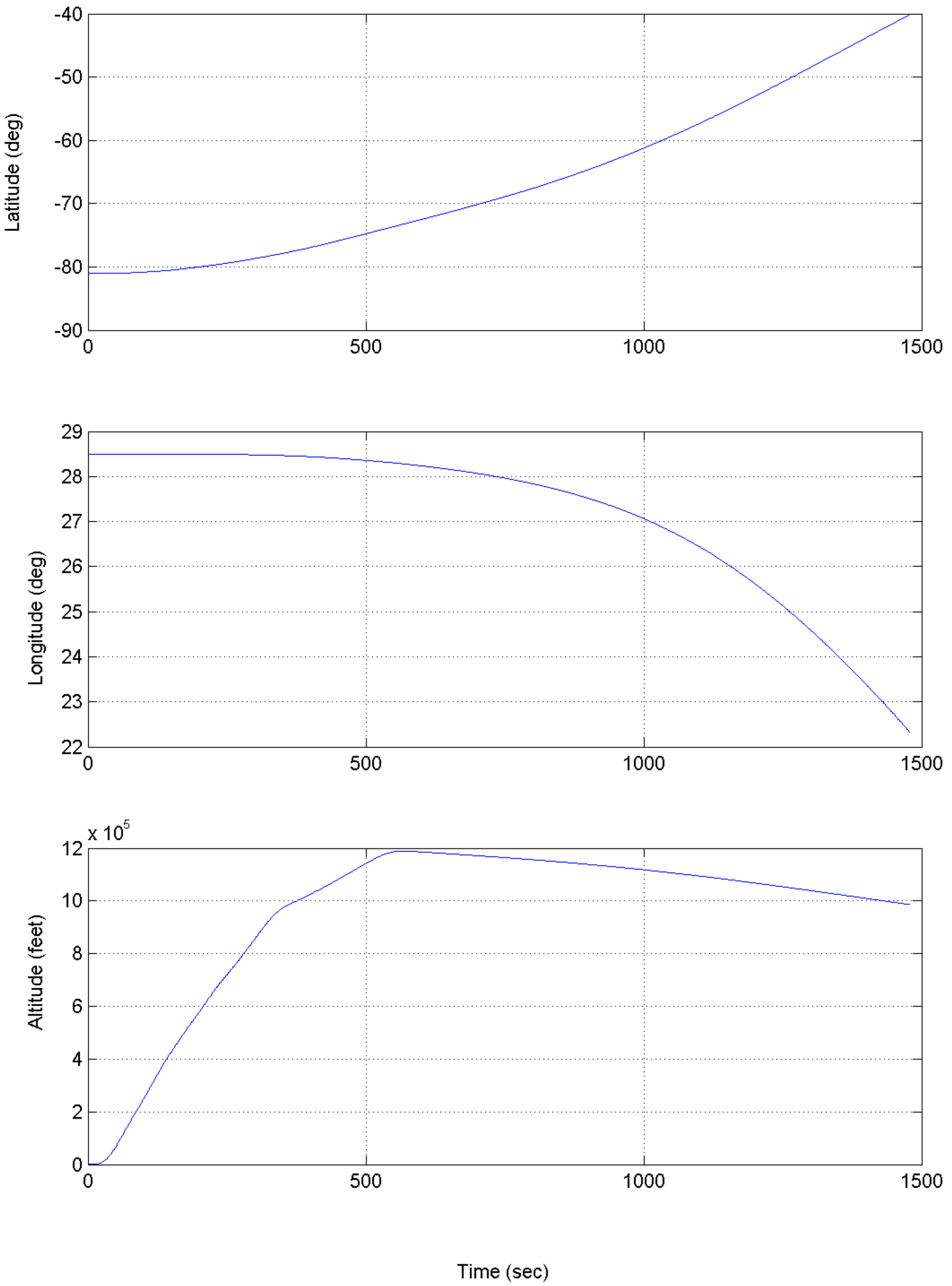


Figure D-3. Latitude, Longitude, and Altitude of the Rocket Flight Profile

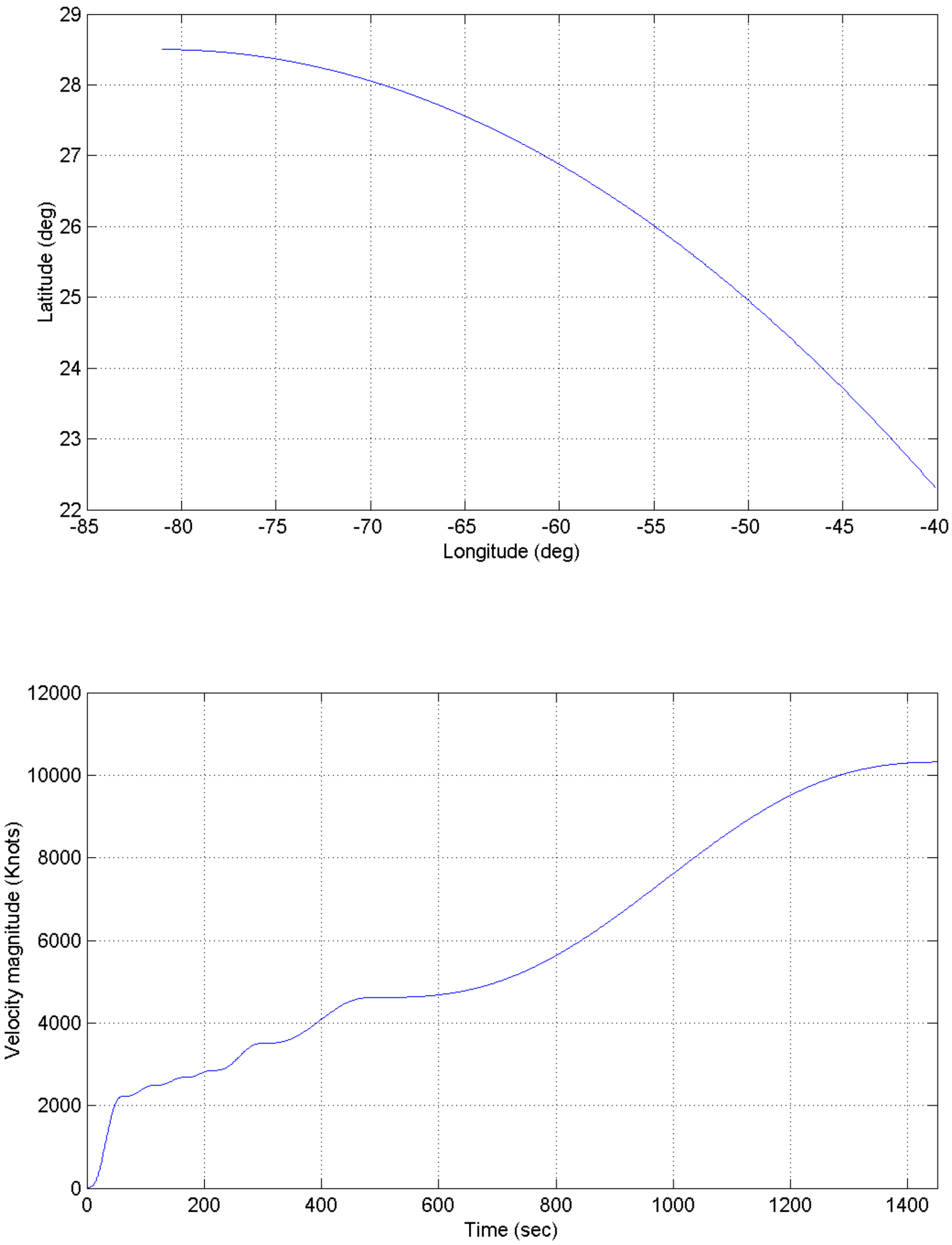


Figure D-4. 2-D Position and Velocity of Rocket Flight Profile

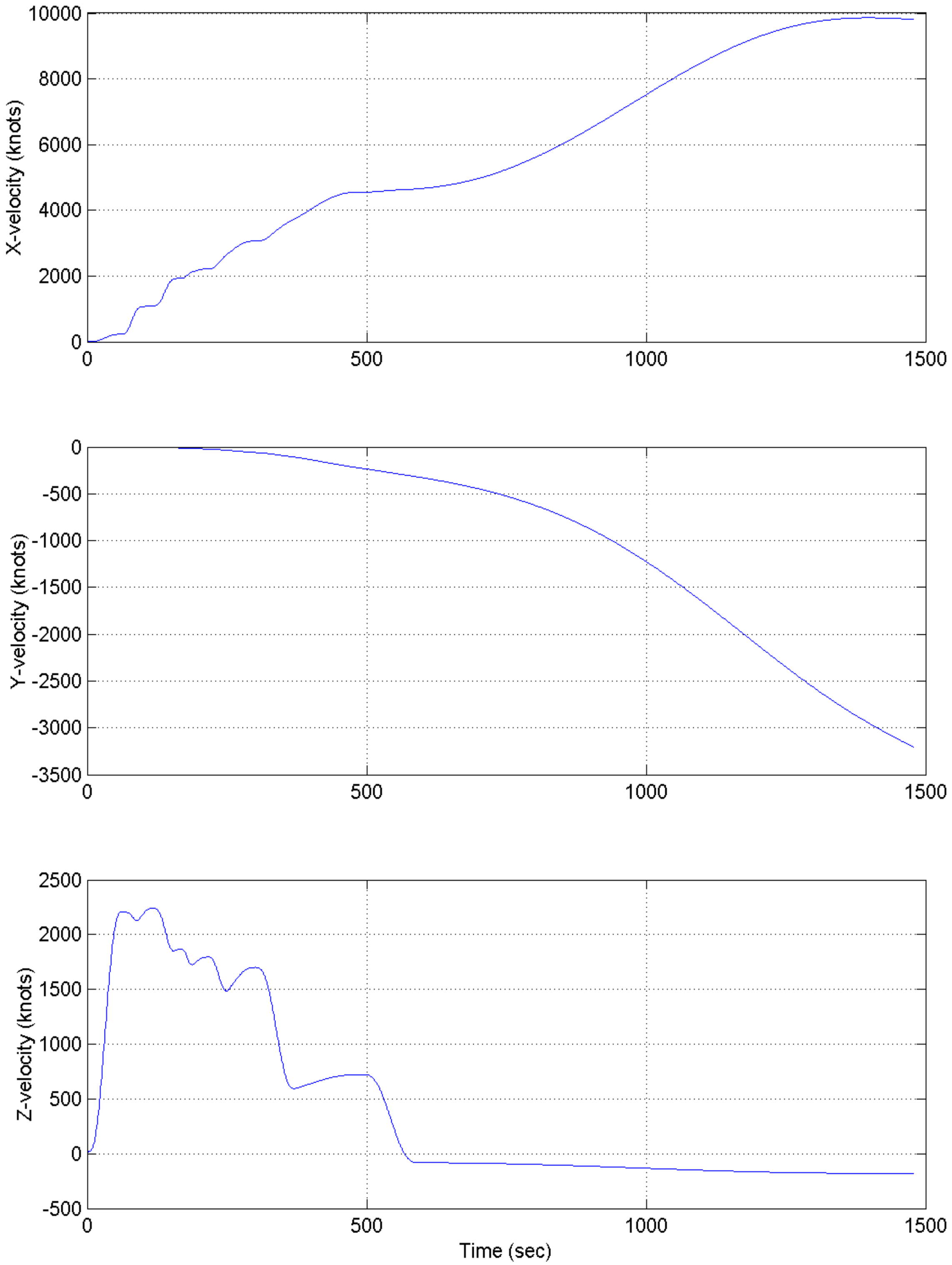


Figure D-5. X,Y, and Z Velocity of the Rocket Flight Profile

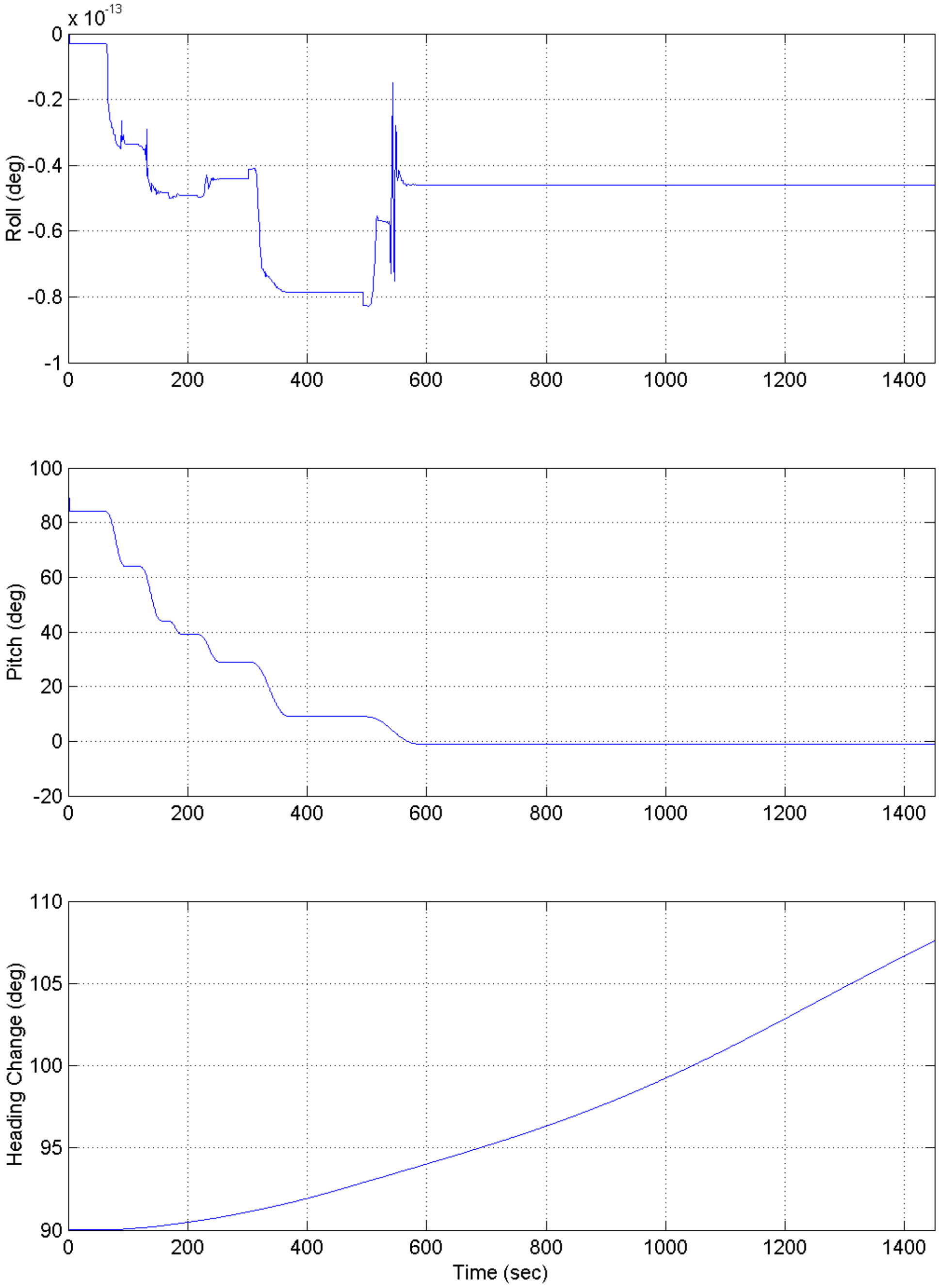


Figure D-6. Roll, Pitch, and Heading Change of the Rocket Flight Profile

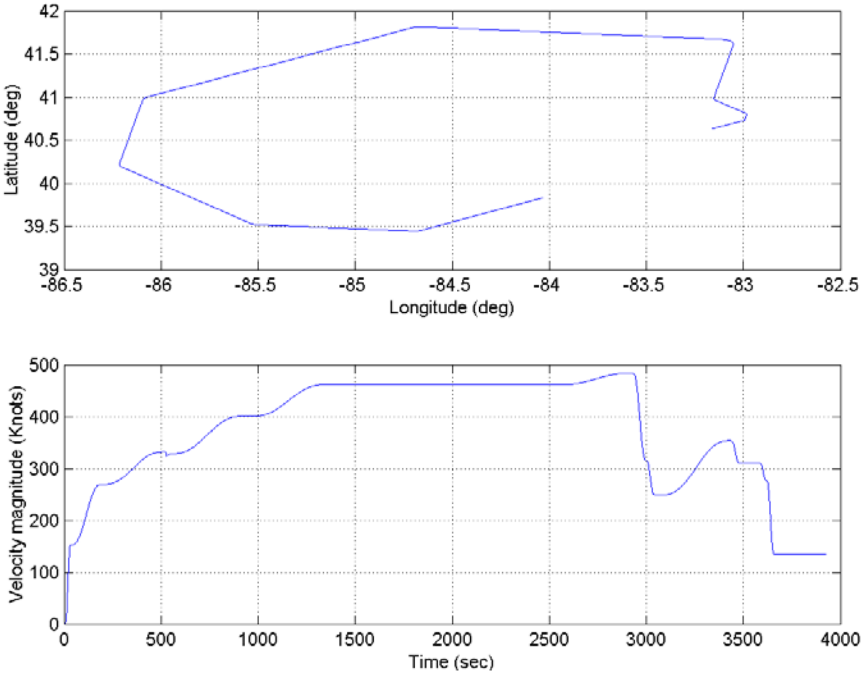


Figure D-7. 2-D Position and Velocity of Tanker Profile

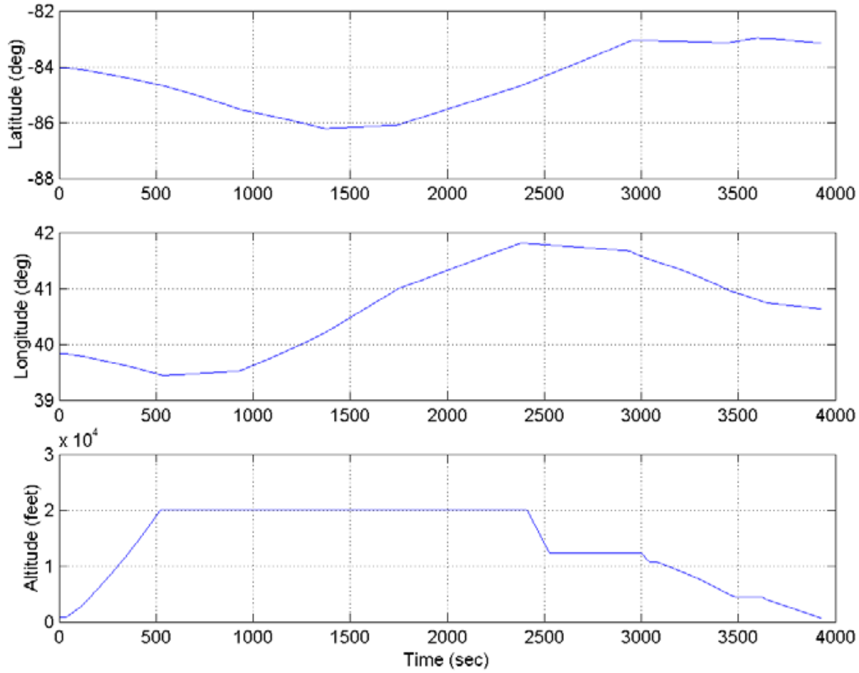


Figure D-8. Latitude, Longitude, and Altitude of the Tanker Profile

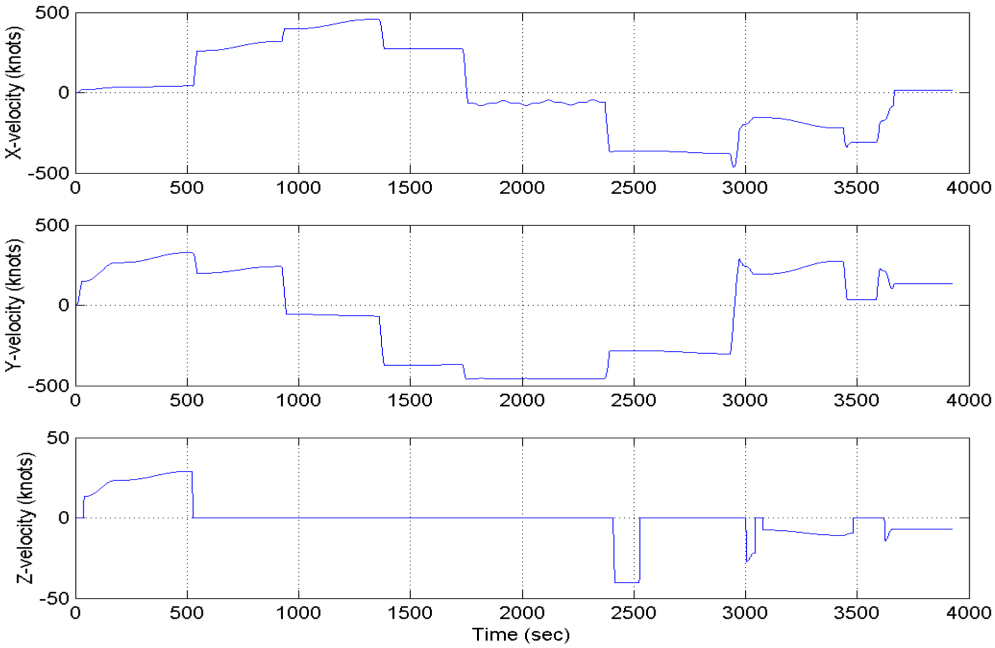


Figure D-9. X,Y, and Z Velocity of the Tanker Profile

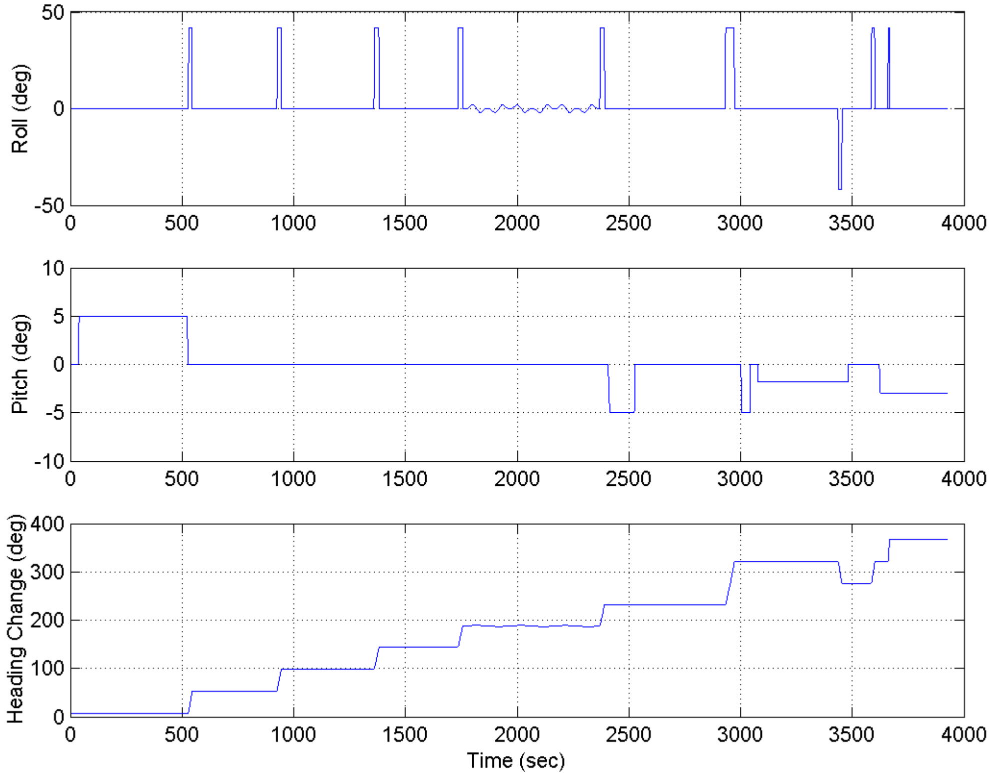
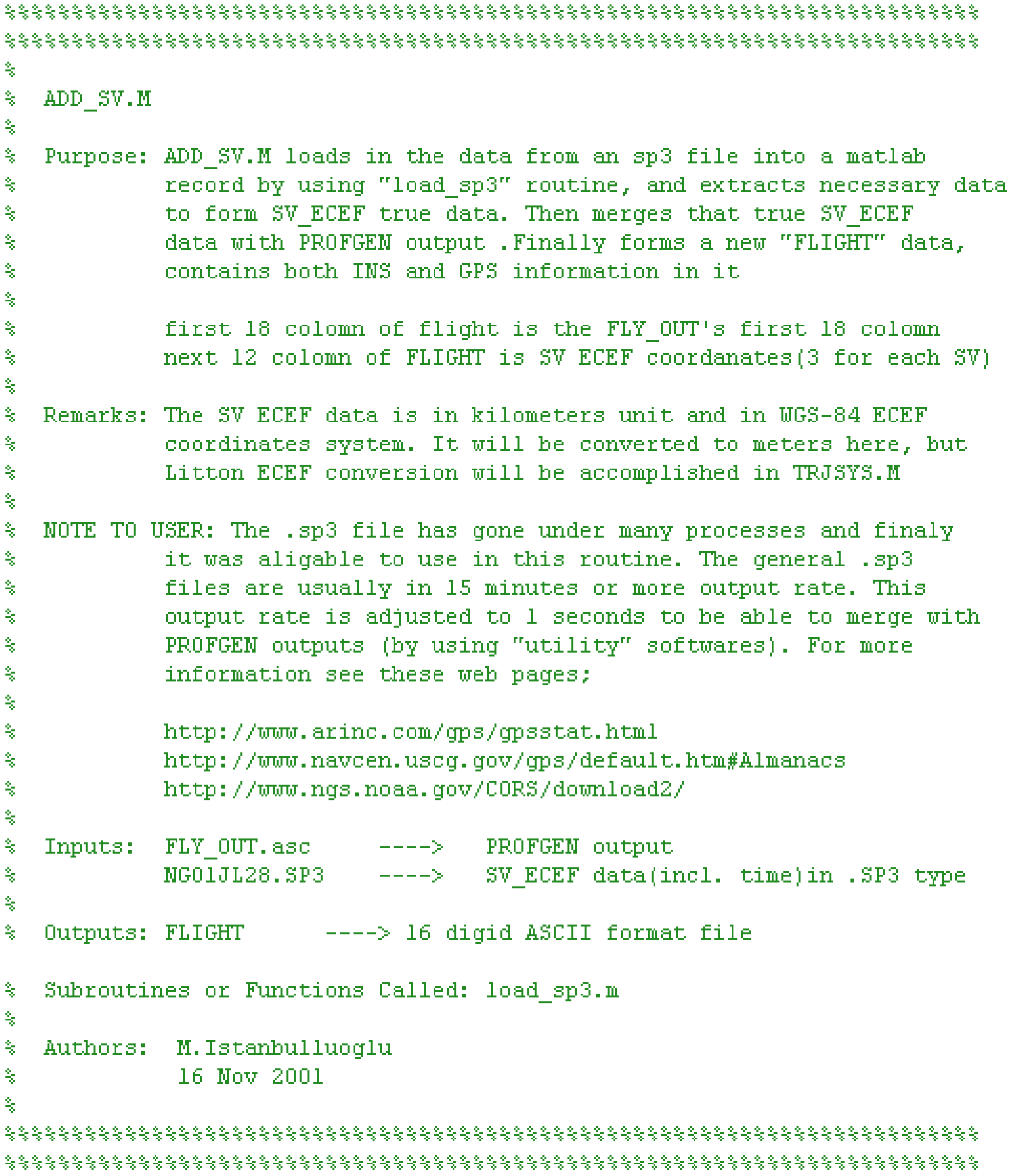


Figure D-10. Roll, Pitch, and Heading Change of the Tanker Profile

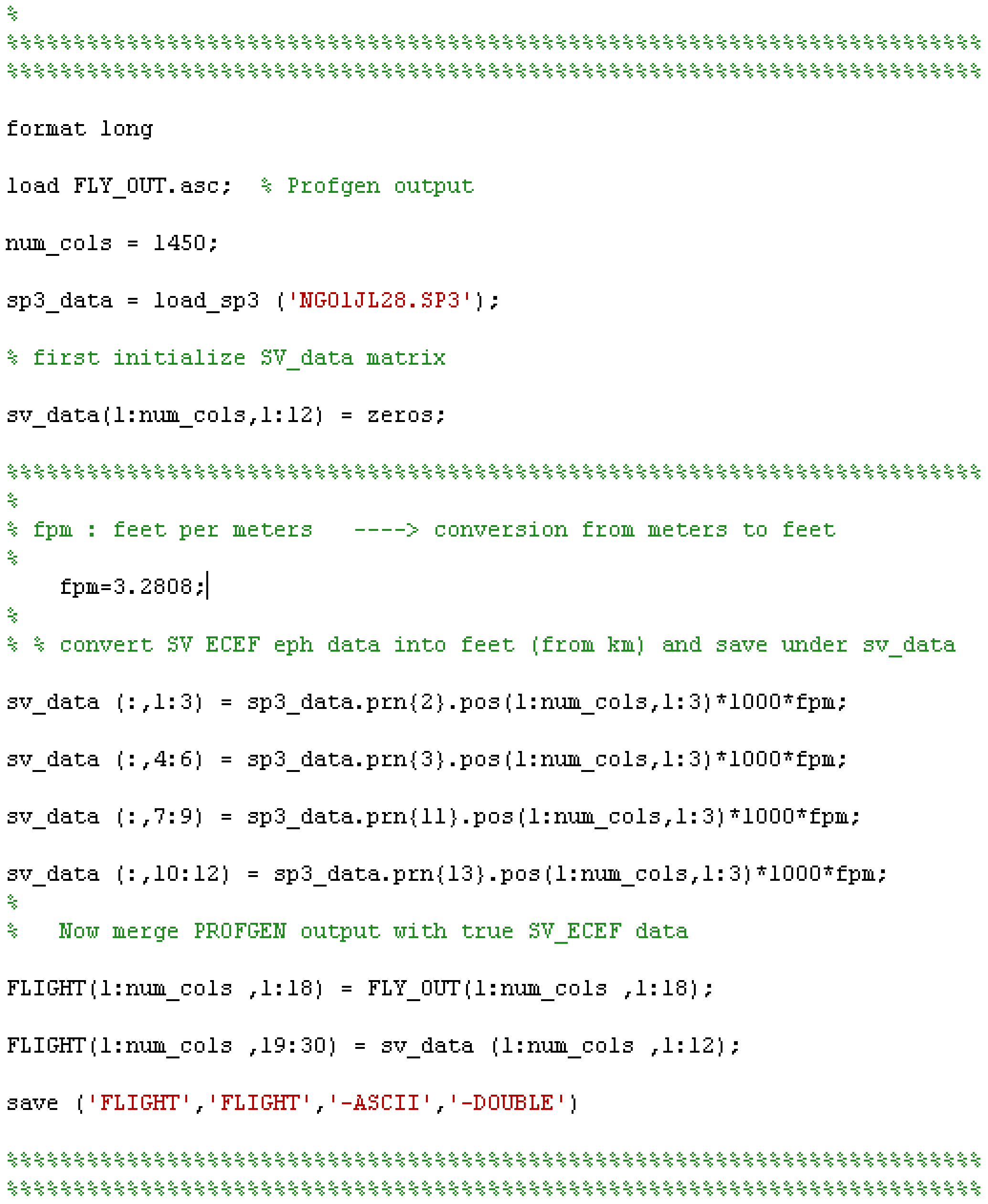
Appendix E

True Ephemeris Generation Process



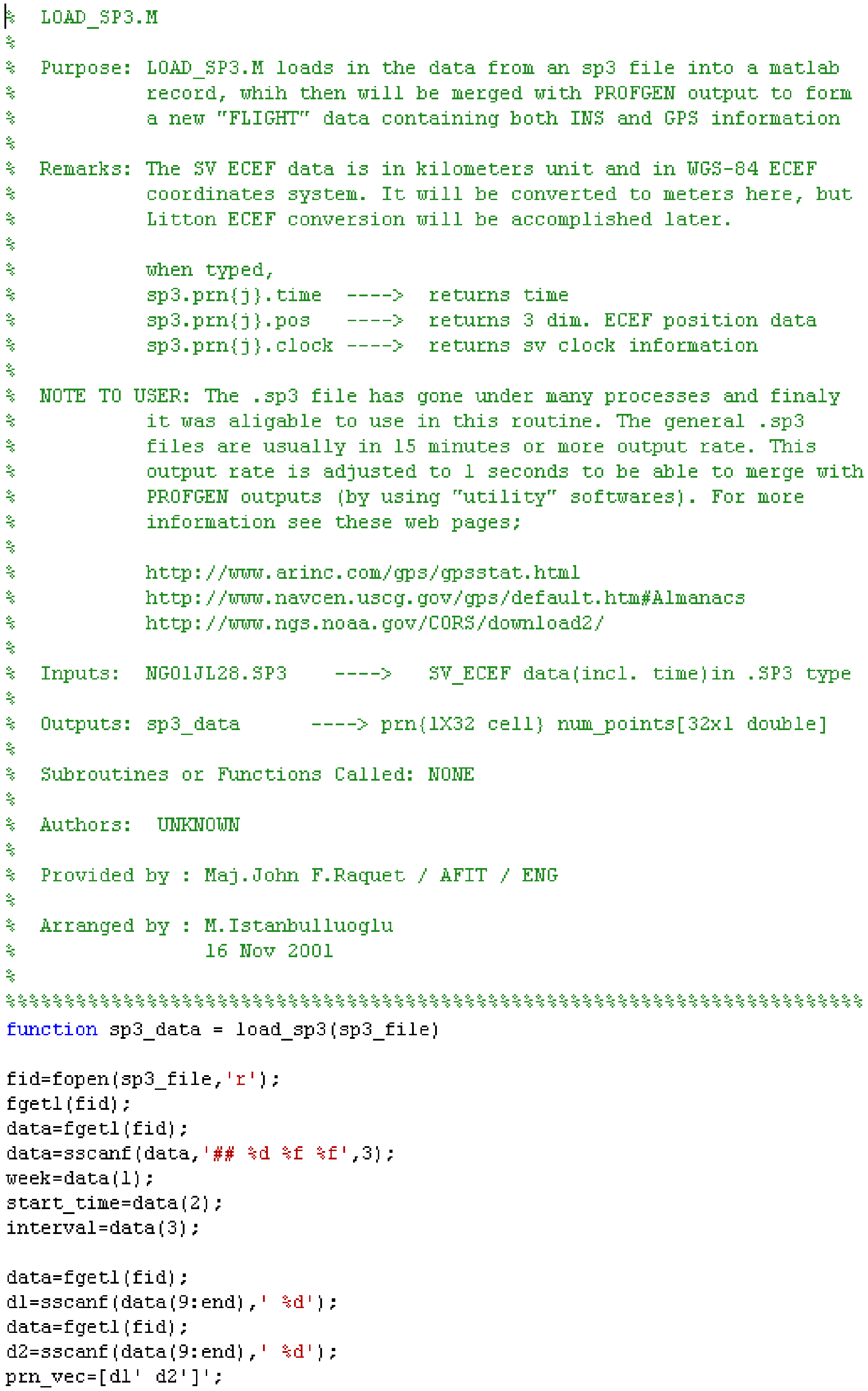
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%  ADD_SV.M
%
%  Purpose: ADD_SV.M loads in the data from an sp3 file into a matlab
%           record by using "load_sp3" routine, and extracts necessary data
%           to form SV_ECEF true data. Then merges that true SV_ECEF
%           data with PROFGEN output .Finally forms a new "FLIGHT" data,
%           contains both INS and GPS information in it
%
%           first 18 colomn of flight is the FLY_OUT's first 18 colomn
%           next 12 colomn of FLIGHT is SV ECEF coordanates(3 for each SV)
%
%  Remarks: The SV ECEF data is in kilometers unit and in WGS-84 ECEF
%           coordinates system. It will be converted to meters here, but
%           Litton ECEF conversion will be accomplished in TRJSYS.M
%
%  NOTE TO USER: The .sp3 file has gone under many processes and finaly
%           it was aligable to use in this routine. The general .sp3
%           files are usually in 15 minutes or more output rate. This
%           output rate is adjusted to 1 seconds to be able to merge with
%           PROFGEN outputs (by using "utility" softwares). For more
%           information see these web pages;
%
%           http://www.arinc.com/gps/gpsstat.html
%           http://www.navcen.uscg.gov/gps/default.htm#Almanacs
%           http://www.ngs.noaa.gov/CORS/download2/
%
%  Inputs:  FLY_OUT.asc     ---->   PROFGEN output
%           NG01JL28.SP3    ---->   SV_ECEF data(incl. time)in .SP3 type
%
%  Outputs: FLIGHT      ----> 16 digid ASCII format file

%  Subroutines or Functions Called: load_sp3.m
%
%  Authors:  M.Istanbulluoglu
%            16 Nov 2001
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure E-1. Matlab m-file “ADD_SV.M” To Merge INS and GPS data



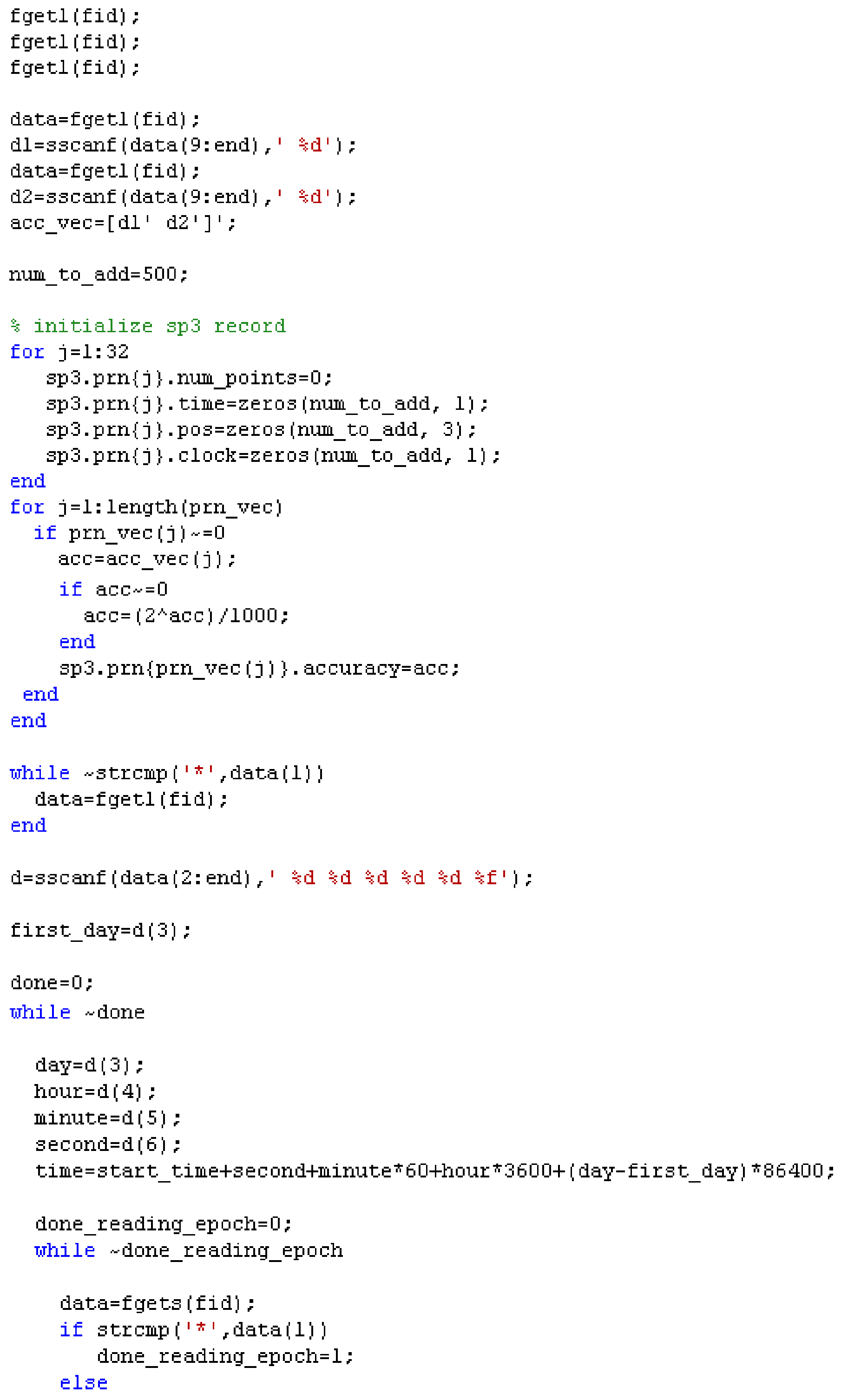
```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

format long

load FLY_OUT.asc;  % Profgen output

num_cols = 1450;

sp3_data = load_sp3 ('NG01JL28.SP3');

% first initialize SV_data matrix

sv_data(1:num_cols,1:12) = zeros;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% fpm : feet per meters   ----> conversion from meters to feet
%
    fpm=3.2808;
%
% % convert SV ECEF eph data into feet (from km) and save under sv_data

sv_data (:,1:3) = sp3_data.prn{2}.pos(1:num_cols,1:3)*1000*fpm;

sv_data (:,4:6) = sp3_data.prn{3}.pos(1:num_cols,1:3)*1000*fpm;

sv_data (:,7:9) = sp3_data.prn{11}.pos(1:num_cols,1:3)*1000*fpm;

sv_data (:,10:12) = sp3_data.prn{13}.pos(1:num_cols,1:3)*1000*fpm;
%
%   Now merge PROFGEN output with true SV_ECEF data

FLIGHT(1:num_cols ,1:18) = FLY_OUT(1:num_cols ,1:18);

FLIGHT(1:num_cols ,19:30) = sv_data (1:num_cols ,1:12);

save ('FLIGHT','FLIGHT','-ASCII','-DOUBLE')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure E-1 (cont.) Matlab m-file “ADD_SV.M” To Merge INS and GPS data



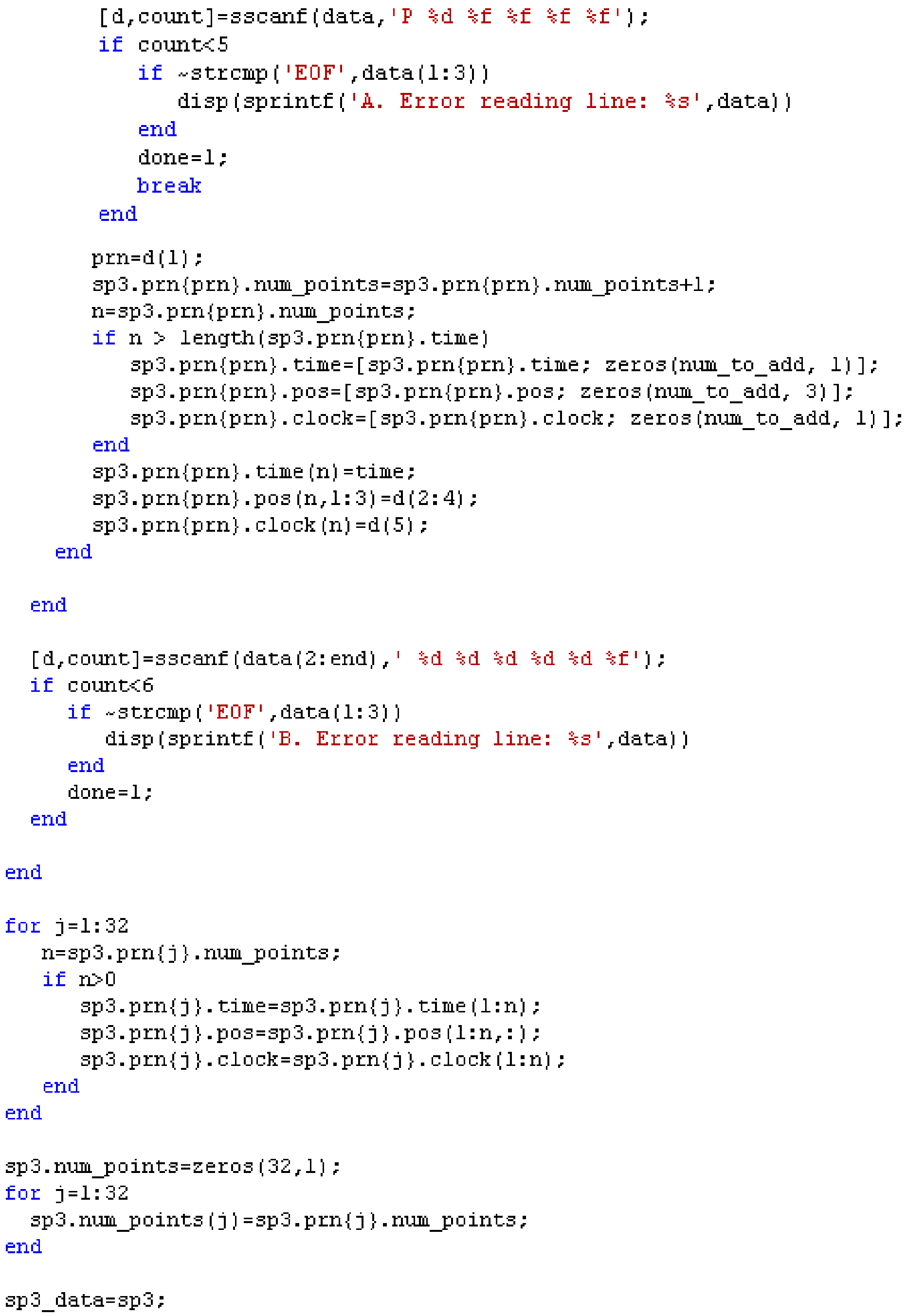
```
%  LOAD_SP3.M
%
%  Purpose: LOAD_SP3.M loads in the data from an sp3 file into a matlab
%           record, whih then will be merged with PROFGEN output to form
%           a new "FLIGHT" data containing both INS and GPS information
%
%  Remarks: The SV ECEF data is in kilometers unit and in WGS-84 ECEF
%           coordinates system. It will be converted to meters here, but
%           Litton ECEF conversion will be accomplished later.
%
%           when typed,
%           sp3.prn{j}.time  ---->  returns time
%           sp3.prn{j}.pos   ---->  returns 3 dim. ECEF position data
%           sp3.prn{j}.clock ---->  returns sv clock information
%
%  NOTE TO USER: The .sp3 file has gone under many processes and finaly
%           it was aligable to use in this routine. The general .sp3
%           files are usually in 15 minutes or more output rate. This
%           output rate is adjusted to 1 seconds to be able to merge with
%           PROFGEN outputs (by using "utility" softwares). For more
%           information see these web pages;
%
%           http://www.arinc.com/gps/gpsstat.html
%           http://www.navcen.uscg.gov/gps/default.htm#Almanacs
%           http://www.ngs.noaa.gov/CORS/download2/
%
%  Inputs:  NG01JL28.SP3    ---->   SV_ECEF data(incl. time)in .SP3 type
%
%  Outputs: sp3_data      ----> prn{1X32 cell} num_points[32x1 double]
%
%  Subroutines or Functions Called: NONE
%
%  Authors:  UNKNOWN
%
%  Provided by : Maj.John F.Raquet / AFIT / ENG
%
%  Arranged by : M.Istanbulluoglu
%                16 Nov 2001
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function sp3_data = load_sp3(sp3_file)

fid=fopen(sp3_file,'r');
fgetl(fid);
data=fgetl(fid);
data=sscanf(data,'## %d %f %f',3);
week=data(1);
start_time=data(2);
interval=data(3);

data=fgetl(fid);
d1=sscanf(data(9:end),' %d');
data=fgetl(fid);
d2=sscanf(data(9:end),' %d');
prn_vec=[d1' d2']';
```

Figure E-2. Matlab m-file “LOAD_SP3.M”



```
fgetl(fid);
fgetl(fid);
fgetl(fid);

data=fgetl(fid);
d1=sscanf(data(9:end),' %d');
data=fgetl(fid);
d2=sscanf(data(9:end),' %d');
acc_vec=[d1' d2']';

num_to_add=500;

% initialize sp3 record
for j=1:32
   sp3.prn{j}.num_points=0;
   sp3.prn{j}.time=zeros(num_to_add, 1);
   sp3.prn{j}.pos=zeros(num_to_add, 3);
   sp3.prn{j}.clock=zeros(num_to_add, 1);
end
for j=1:length(prn_vec)
  if prn_vec(j)~=0
    acc=acc_vec(j);
    if acc~=0
      acc=(2^acc)/1000;
    end
    sp3.prn{prn_vec(j)}.accuracy=acc;
 end
end

while ~strcmp('*',data(1))
  data=fgetl(fid);
end

d=sscanf(data(2:end),' %d %d %d %d %d %f');

first_day=d(3);

done=0;
while ~done

  day=d(3);
  hour=d(4);
  minute=d(5);
  second=d(6);
  time=start_time+second+minute*60+hour*3600+(day-first_day)*86400;

  done_reading_epoch=0;
  while ~done_reading_epoch

    data=fgets(fid);
    if strcmp('*',data(1))
       done_reading_epoch=1;
    else
```

Figure E-2 (cont.) Matlab m-file “LOAD_SP3.M”



```
        [d,count]=sscanf(data,'P %d %f %f %f %f');
        if count<5
           if ~strcmp('EOF',data(1:3))
              disp(sprintf('A. Error reading line: %s',data))
           end
           done=1;
           break
        end

       prn=d(1);
       sp3.prn{prn}.num_points=sp3.prn{prn}.num_points+1;
       n=sp3.prn{prn}.num_points;
       if n > length(sp3.prn{prn}.time)
          sp3.prn{prn}.time=[sp3.prn{prn}.time; zeros(num_to_add, 1)];
          sp3.prn{prn}.pos=[sp3.prn{prn}.pos; zeros(num_to_add, 3)];
          sp3.prn{prn}.clock=[sp3.prn{prn}.clock; zeros(num_to_add, 1)];
       end
       sp3.prn{prn}.time(n)=time;
       sp3.prn{prn}.pos(n,1:3)=d(2:4);
       sp3.prn{prn}.clock(n)=d(5);
    end

  end

  [d,count]=sscanf(data(2:end),' %d %d %d %d %d %f');
  if count<6
     if ~strcmp('EOF',data(1:3))
        disp(sprintf('B. Error reading line: %s',data))
     end
     done=1;
  end

end

for j=1:32
   n=sp3.prn{j}.num_points;
   if n>0
      sp3.prn{j}.time=sp3.prn{j}.time(1:n);
      sp3.prn{j}.pos=sp3.prn{j}.pos(1:n,:);
      sp3.prn{j}.clock=sp3.prn{j}.clock(1:n);
   end
end

sp3.num_points=zeros(32,1);
for j=1:32
  sp3.num_points(j)=sp3.prn{j}.num_points;
end

sp3_data=sp3;
```

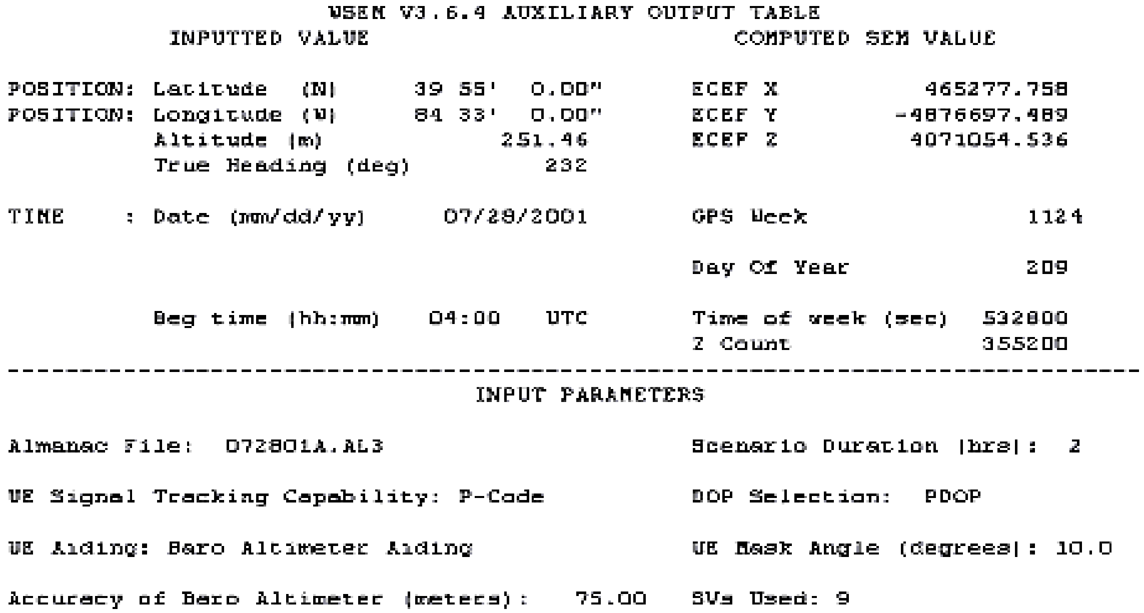
Figure E-2 (cont.) Matlab m-file “LOAD_SP3.M”

Appendix F

Plots Obtained from WSEM 3.6

The following Tables show how WSEM software helps determine the four best visible satellites. The tables presented in this section are the actual WSEM outputs for the satellites chosen and used in this research.

Table F-1. WSEM Set-Up Parameters



```
                    WSEM V3.6.4 AUXILIARY OUTPUT TABLE
           INPUTTED VALUE                         COMPUTED SEM VALUE

POSITION: Latitude  (N)      39 55'  0.00"      ECEF X          465277.758
POSITION: Longitude (W)      84 33'  0.00"      ECEF Y        -4876697.489
          Altitude (m)            251.46        ECEF Z         4071054.536
          True Heading (deg)         232

TIME    : Date (mm/dd/yy)      07/28/2001       GPS Week                1124

                                                Day Of Year              209

          Beg time (hh:mm)    04:00   UTC       Time of week (sec)   532800
                                                Z Count              355200
------------------------------------------------------------------------------
                              INPUT PARAMETERS

Almanac File:  072801A.AL3                      Scenario Duration (hrs):  2

UE Signal Tracking Capability: P-Code           DOP Selection:  PDOP

UE Aiding: Baro Altimeter Aiding                UE Mask Angle (degrees): 10.0

Accuracy of Baro Altimeter (meters):   75.00    SVs Used: 9
```

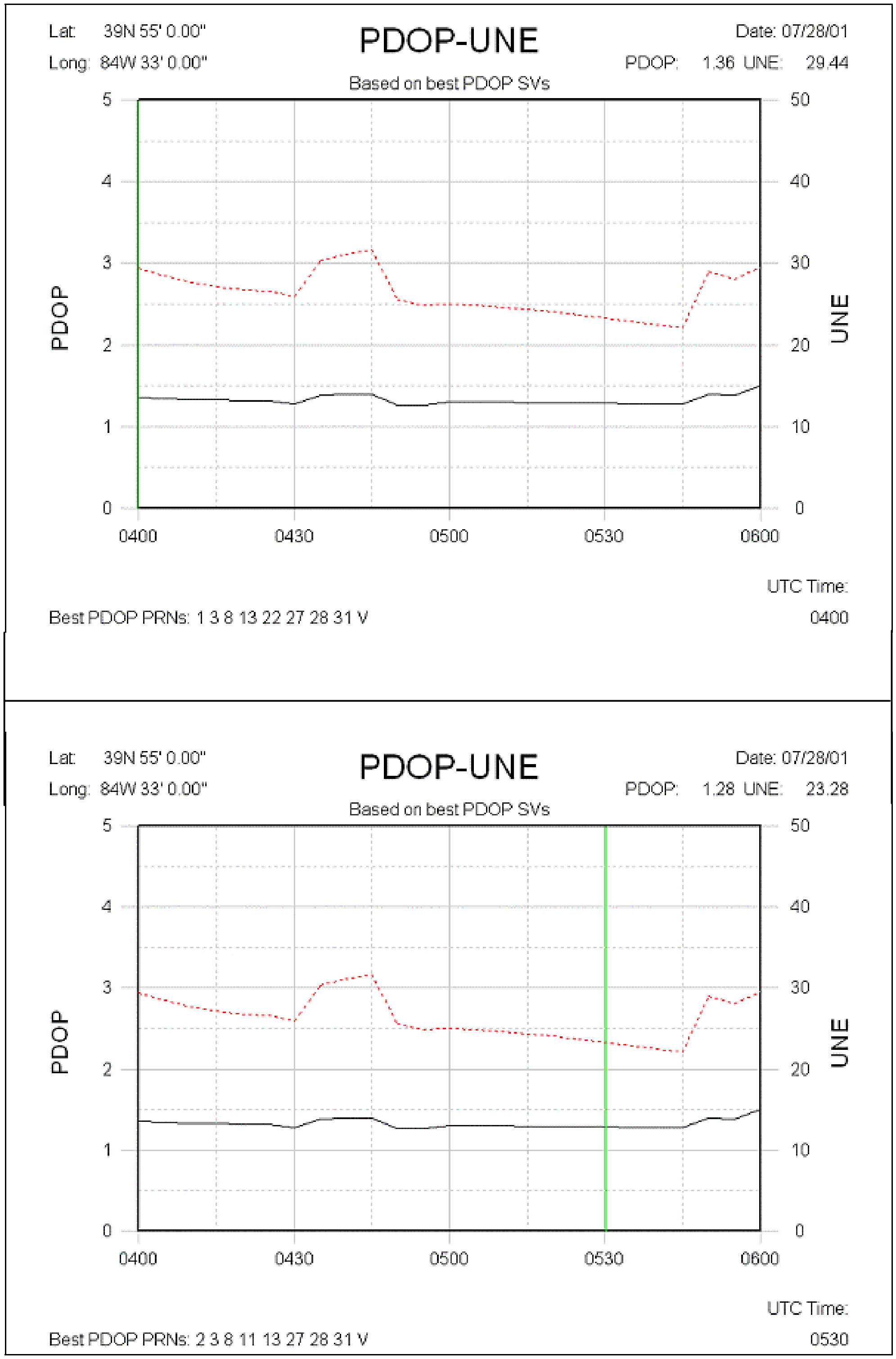


Figure F-1. PDOP (Start Time and Final Time)

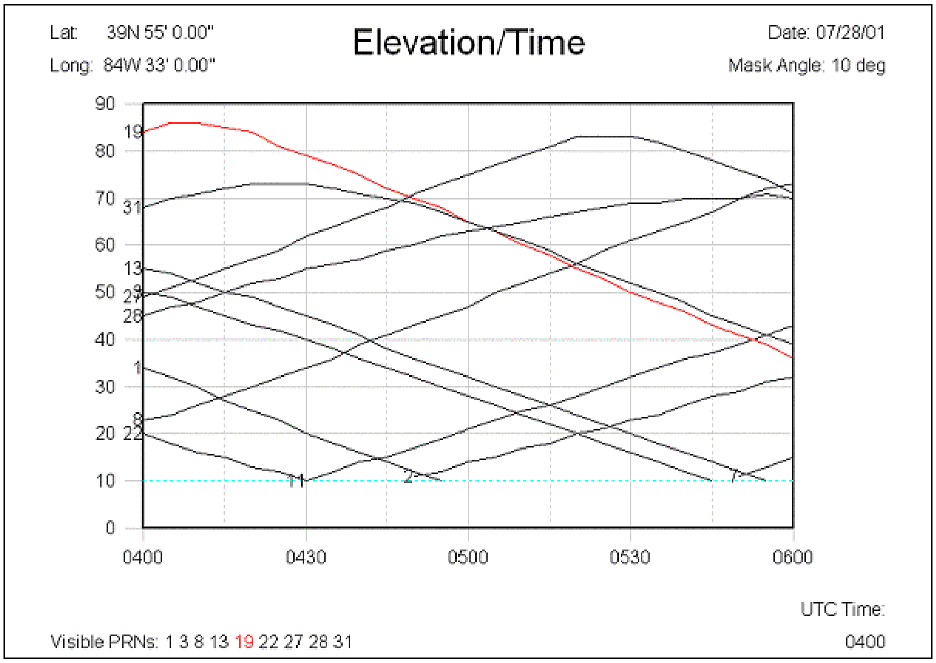


Figure F- 2. Satellite Elevation/Time Graph

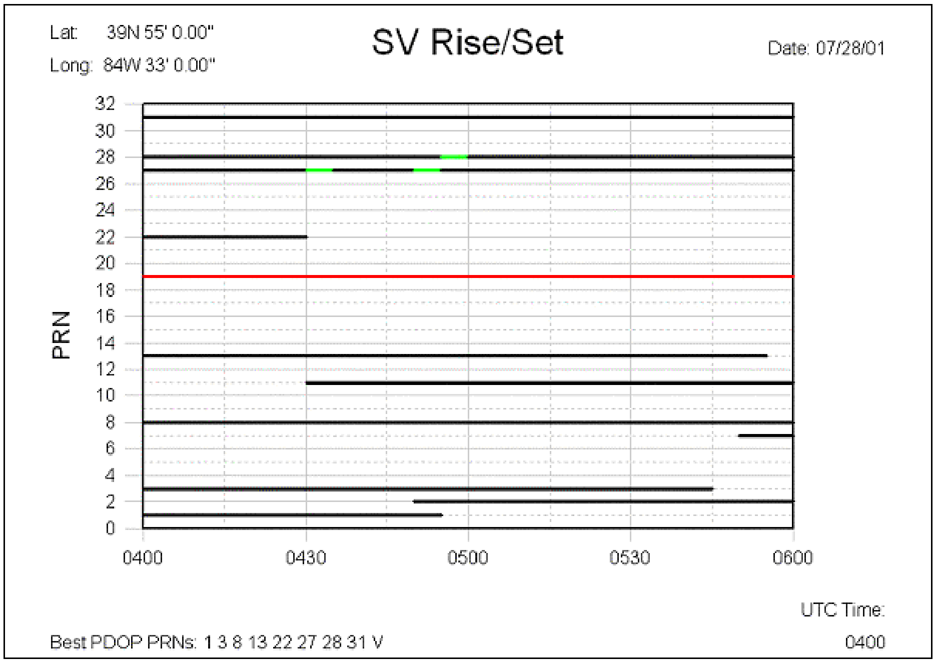


Figure F-3. Satellite Rise/Set Graph

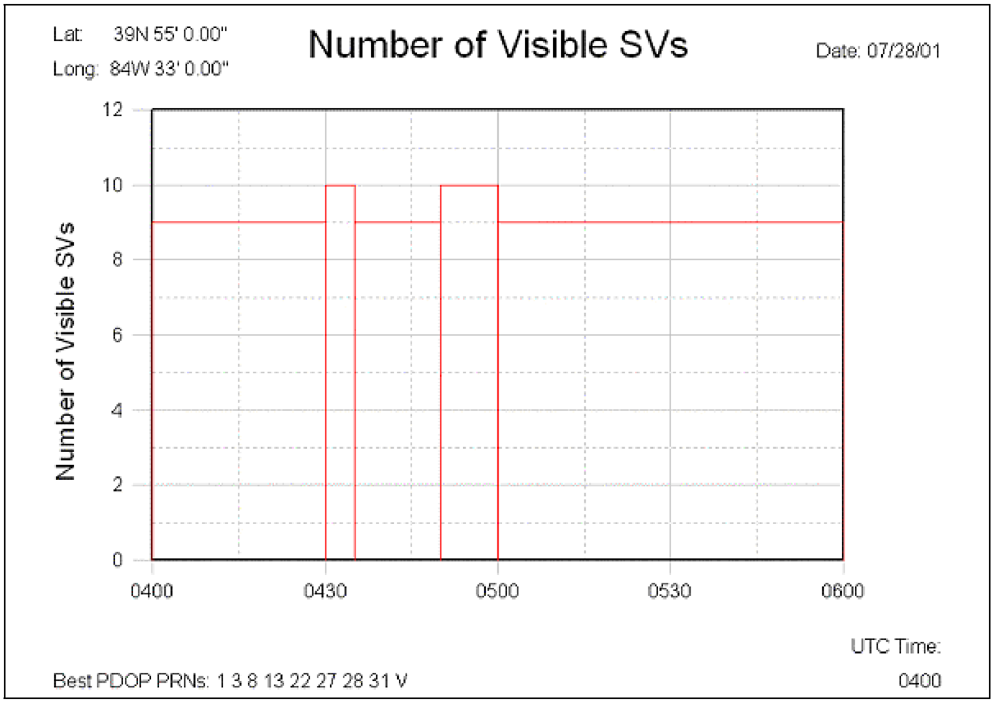


Figure F-4. Number of Visible Satellites

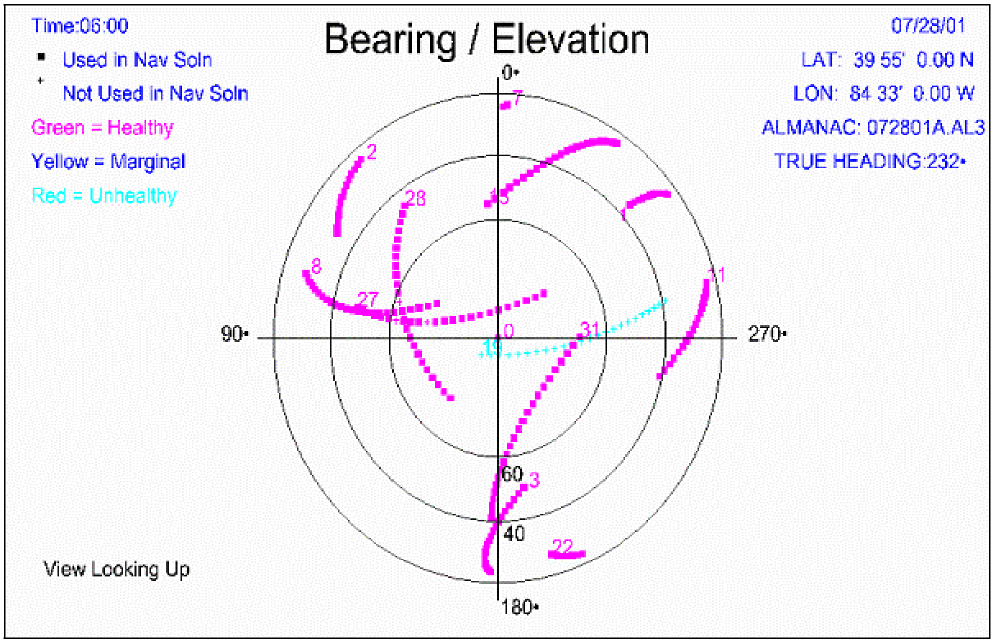
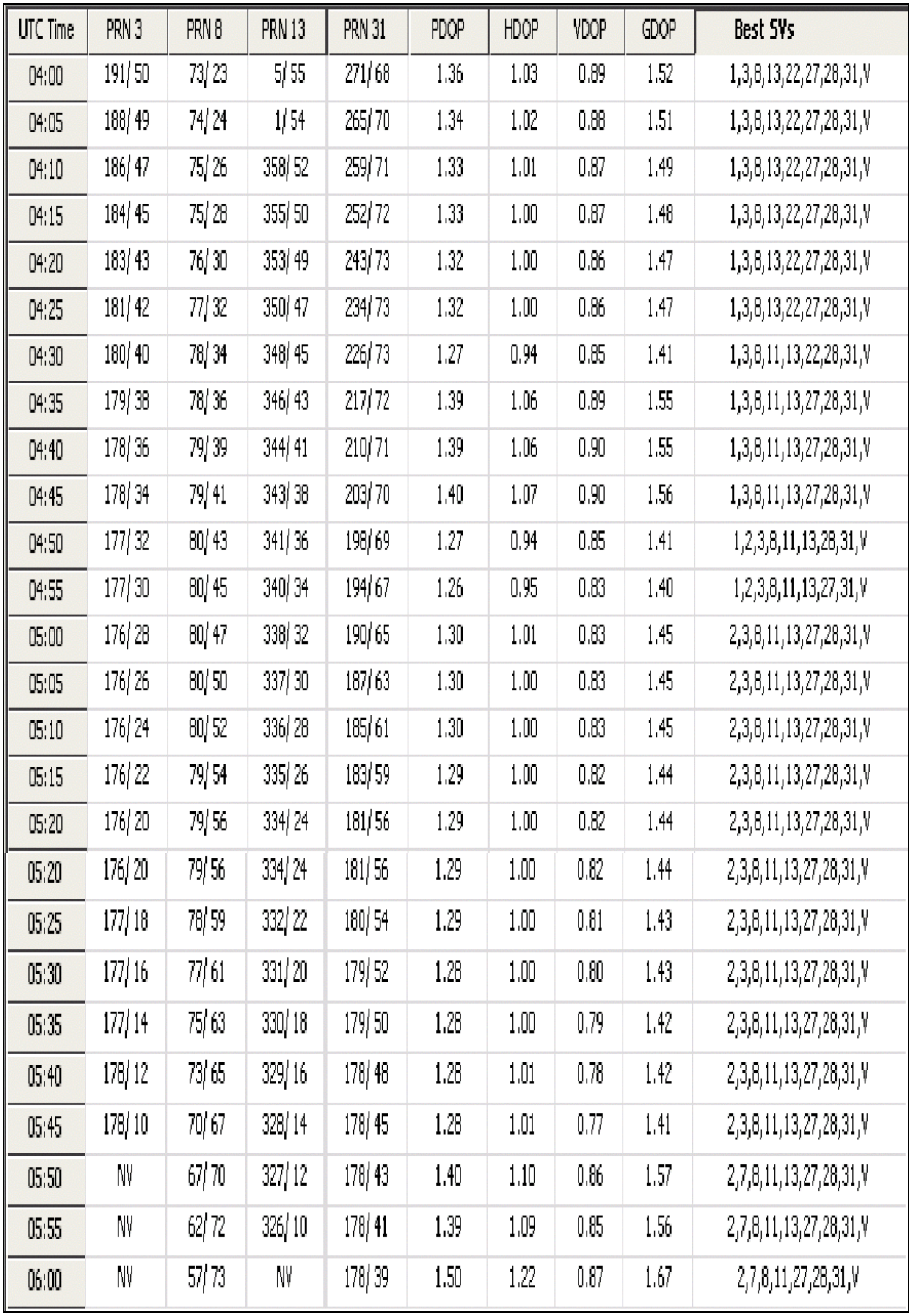


Figure F-5. Bearing/Elevation Graph

Table F-2. Satellite Bearing/Elevation Table



UTC Time	PRN 3	PRN 8	PRN 13	PRN 31	PDOP	HDOP	VDOP	GDOP	Best SVs
04:00	191/ 50	73/ 23	5/ 55	271/ 68	1.36	1.03	0.89	1.52	1,3,8,13,22,27,28,31,V
04:05	188/ 49	74/ 24	1/ 54	265/ 70	1.34	1.02	0.88	1.51	1,3,8,13,22,27,28,31,V
04:10	186/ 47	75/ 26	358/ 52	259/ 71	1.33	1.01	0.87	1.49	1,3,8,13,22,27,28,31,V
04:15	184/ 45	75/ 28	355/ 50	252/ 72	1.33	1.00	0.87	1.48	1,3,8,13,22,27,28,31,V
04:20	183/ 43	76/ 30	353/ 49	243/ 73	1.32	1.00	0.86	1.47	1,3,8,13,22,27,28,31,V
04:25	181/ 42	77/ 32	350/ 47	234/ 73	1.32	1.00	0.86	1.47	1,3,8,13,22,27,28,31,V
04:30	180/ 40	78/ 34	348/ 45	226/ 73	1.27	0.94	0.85	1.41	1,3,8,11,13,22,28,31,V
04:35	179/ 38	78/ 36	346/ 43	217/ 72	1.39	1.06	0.89	1.55	1,3,8,11,13,27,28,31,V
04:40	178/ 36	79/ 39	344/ 41	210/ 71	1.39	1.06	0.90	1.55	1,3,8,11,13,27,28,31,V
04:45	178/ 34	79/ 41	343/ 38	203/ 70	1.40	1.07	0.90	1.56	1,3,8,11,13,27,28,31,V
04:50	177/ 32	80/ 43	341/ 36	198/ 69	1.27	0.94	0.85	1.41	1,2,3,8,11,13,28,31,V
04:55	177/ 30	80/ 45	340/ 34	194/ 67	1.26	0.95	0.83	1.40	1,2,3,8,11,13,27,31,V
05:00	176/ 28	80/ 47	338/ 32	190/ 65	1.30	1.01	0.83	1.45	2,3,8,11,13,27,28,31,V
05:05	176/ 26	80/ 50	337/ 30	187/ 63	1.30	1.00	0.83	1.45	2,3,8,11,13,27,28,31,V
05:10	176/ 24	80/ 52	336/ 28	185/ 61	1.30	1.00	0.83	1.45	2,3,8,11,13,27,28,31,V
05:15	176/ 22	79/ 54	335/ 26	183/ 59	1.29	1.00	0.82	1.44	2,3,8,11,13,27,28,31,V
05:20	176/ 20	79/ 56	334/ 24	181/ 56	1.29	1.00	0.82	1.44	2,3,8,11,13,27,28,31,V
05:20	176/ 20	79/ 56	334/ 24	181/ 56	1.29	1.00	0.82	1.44	2,3,8,11,13,27,28,31,V
05:25	177/ 18	78/ 59	332/ 22	180/ 54	1.29	1.00	0.81	1.43	2,3,8,11,13,27,28,31,V
05:30	177/ 16	77/ 61	331/ 20	179/ 52	1.28	1.00	0.80	1.43	2,3,8,11,13,27,28,31,V
05:35	177/ 14	75/ 63	330/ 18	179/ 50	1.28	1.00	0.79	1.42	2,3,8,11,13,27,28,31,V
05:40	178/ 12	73/ 65	329/ 16	178/ 48	1.28	1.01	0.78	1.42	2,3,8,11,13,27,28,31,V
05:45	178/ 10	70/ 67	328/ 14	178/ 45	1.28	1.01	0.77	1.41	2,3,8,11,13,27,28,31,V
05:50	NV	67/ 70	327/ 12	178/ 43	1.40	1.10	0.86	1.57	2,7,8,11,13,27,28,31,V
05:55	NV	62/ 72	326/ 10	178/ 41	1.39	1.09	0.85	1.56	2,7,8,11,13,27,28,31,V
06:00	NV	57/ 73	NV	178/ 39	1.50	1.22	0.87	1.67	2,7,8,11,27,28,31,V

Appendix G

MatSOFE Validation Plots

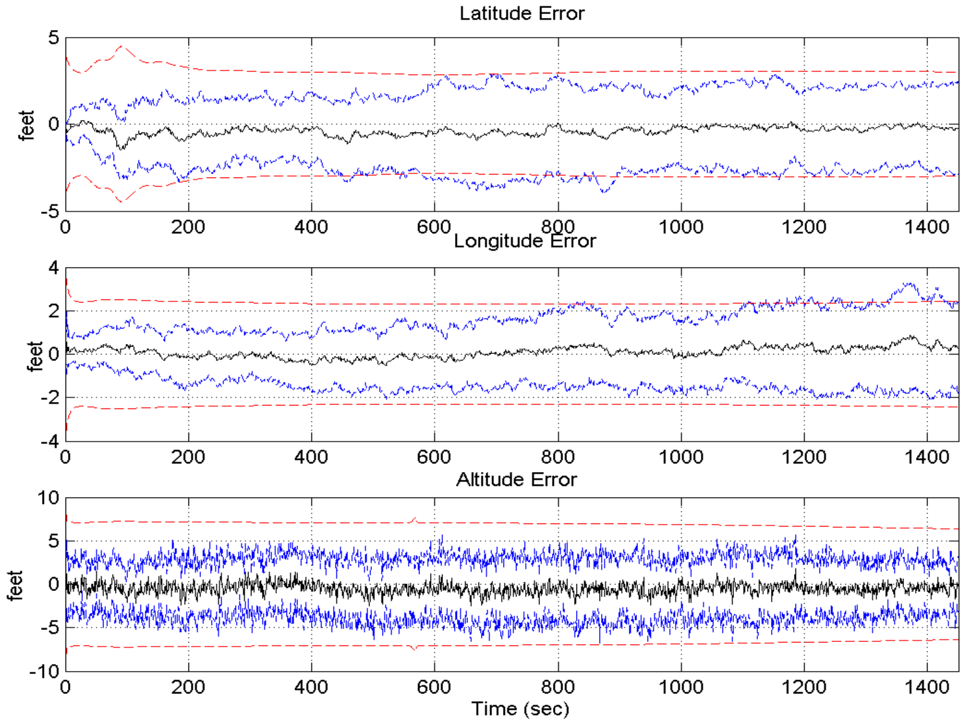


Figure G-1. Latitude, Longitude, and Altitude Errors

(MatSOFE output for Gray's study case)

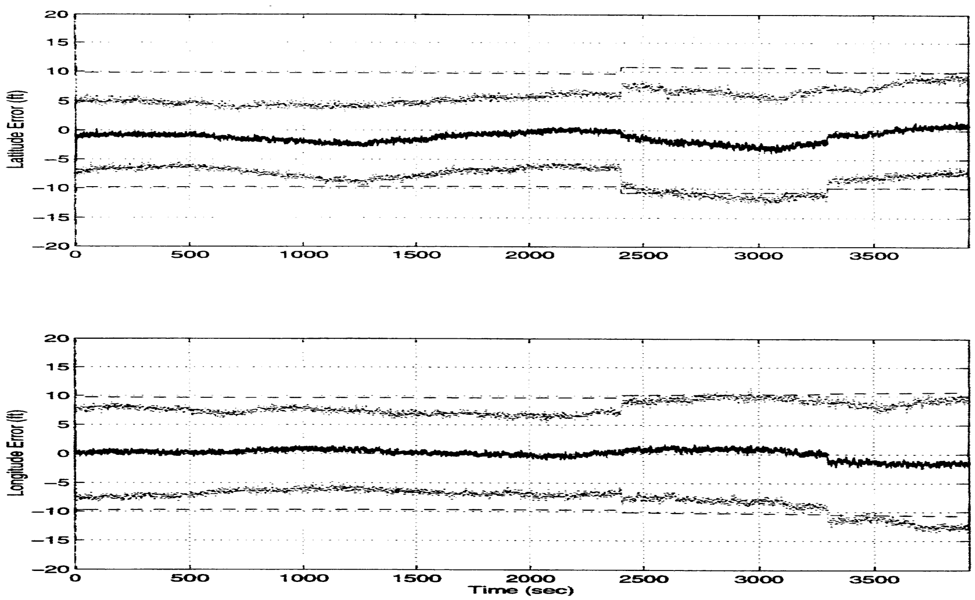


Figure G-2. Latitude, Longitude, and Altitude Errors

(MSOFE output for Gray's study case)

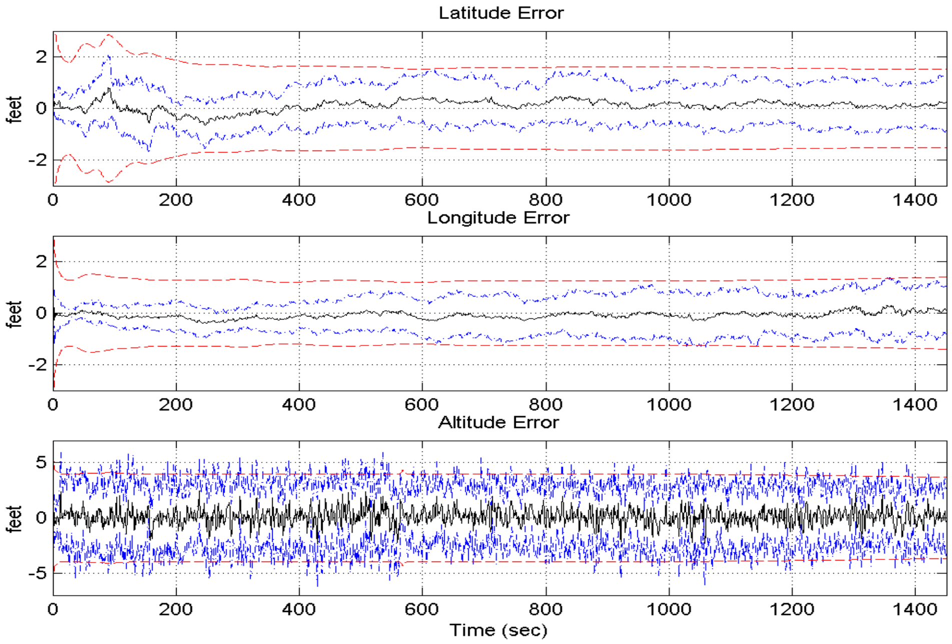


Figure G-3. Latitude, Longitude, and Altitude Errors
(MatSOFE output for Britton's study case)

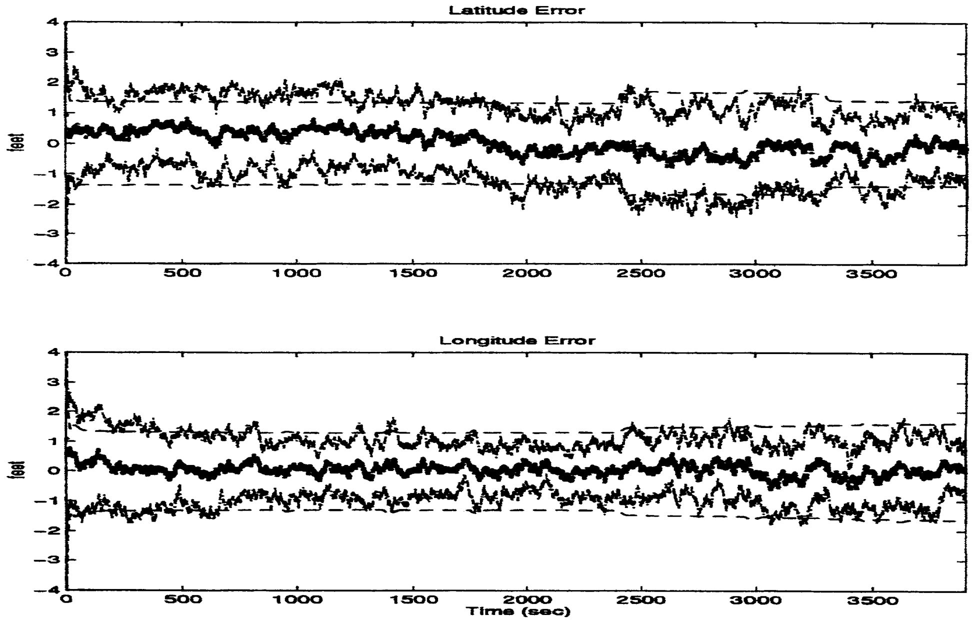


Figure G-4. Latitude, Longitude, and Altitude Errors
(MSOFE output for Britton's study case)

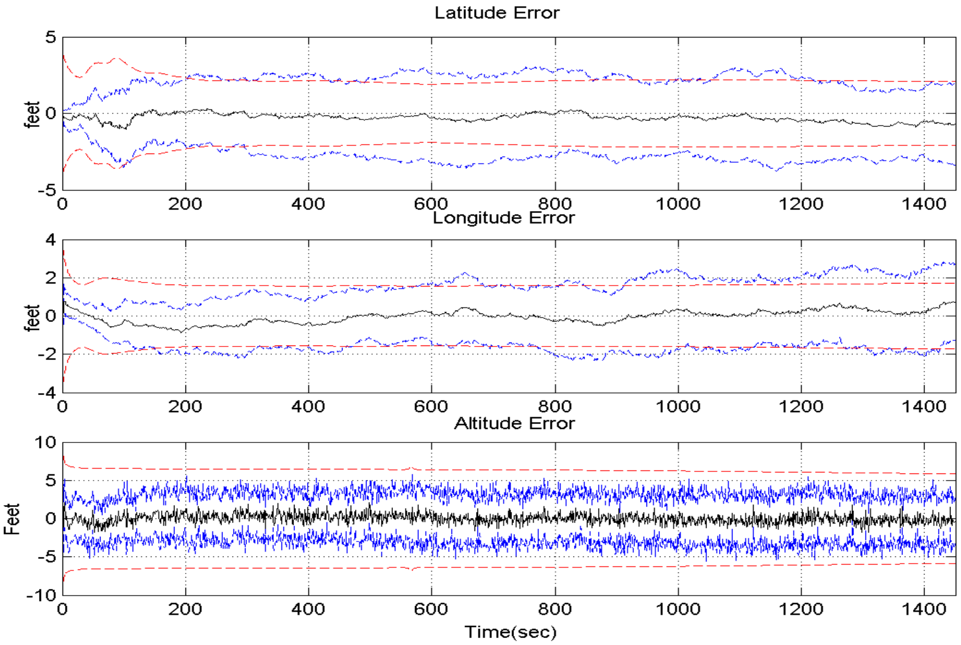
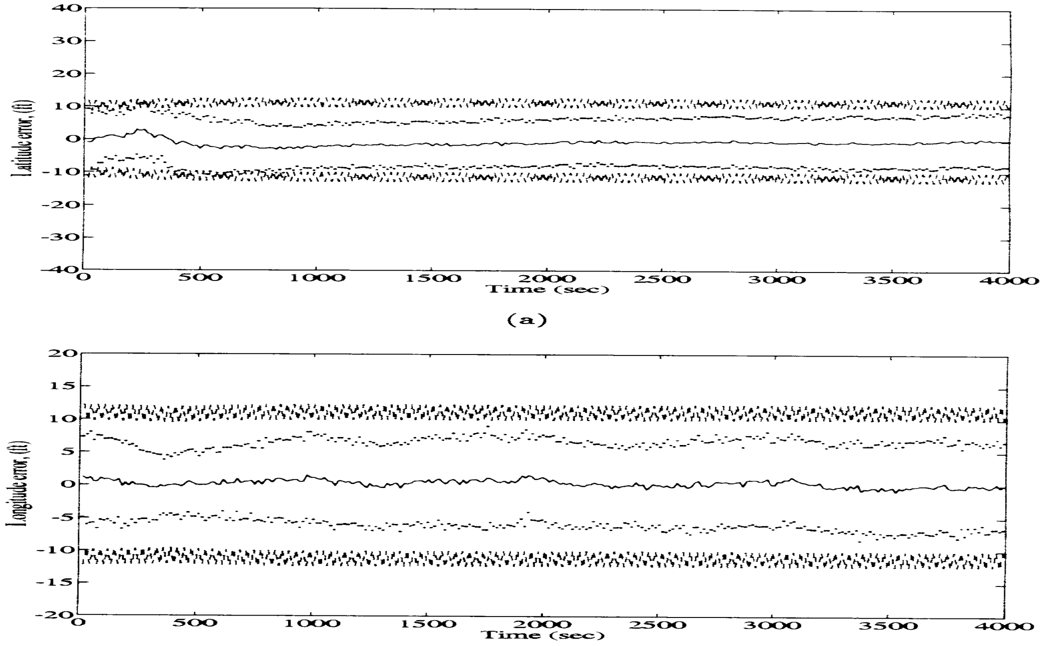


Figure G-5. Latitude, Longitude, and Altitude Errors
(MatSOFE output for Mosle's study case)



(a)

Figure G-6. Latitude, Longitude, and Altitude Errors
(MSOFE output for Mosle's study case)

Appendix H

Tuning Examples

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)

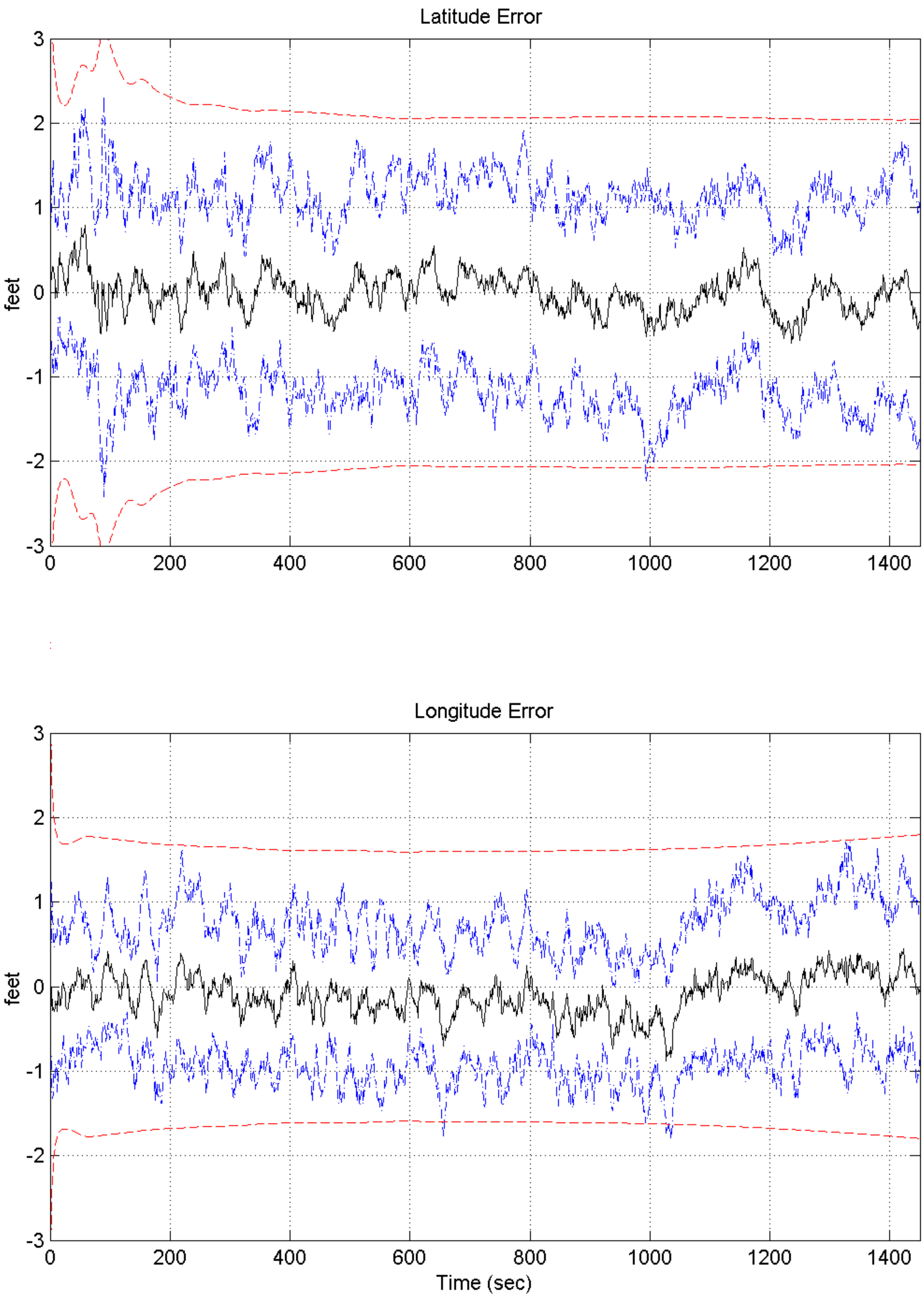


Figure H-1. Latitude and Longitude Errors (Conservative Q-Tuning)

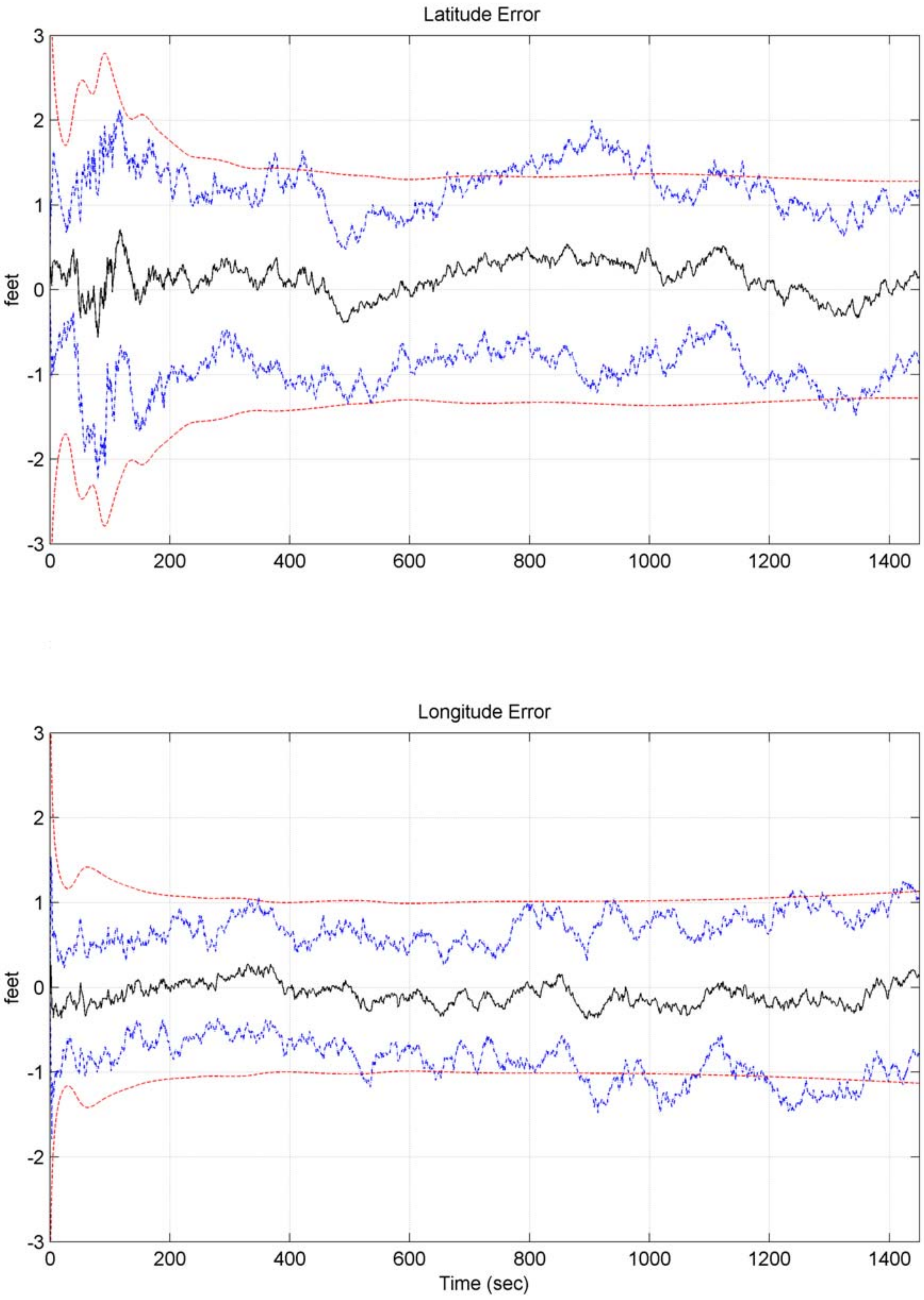
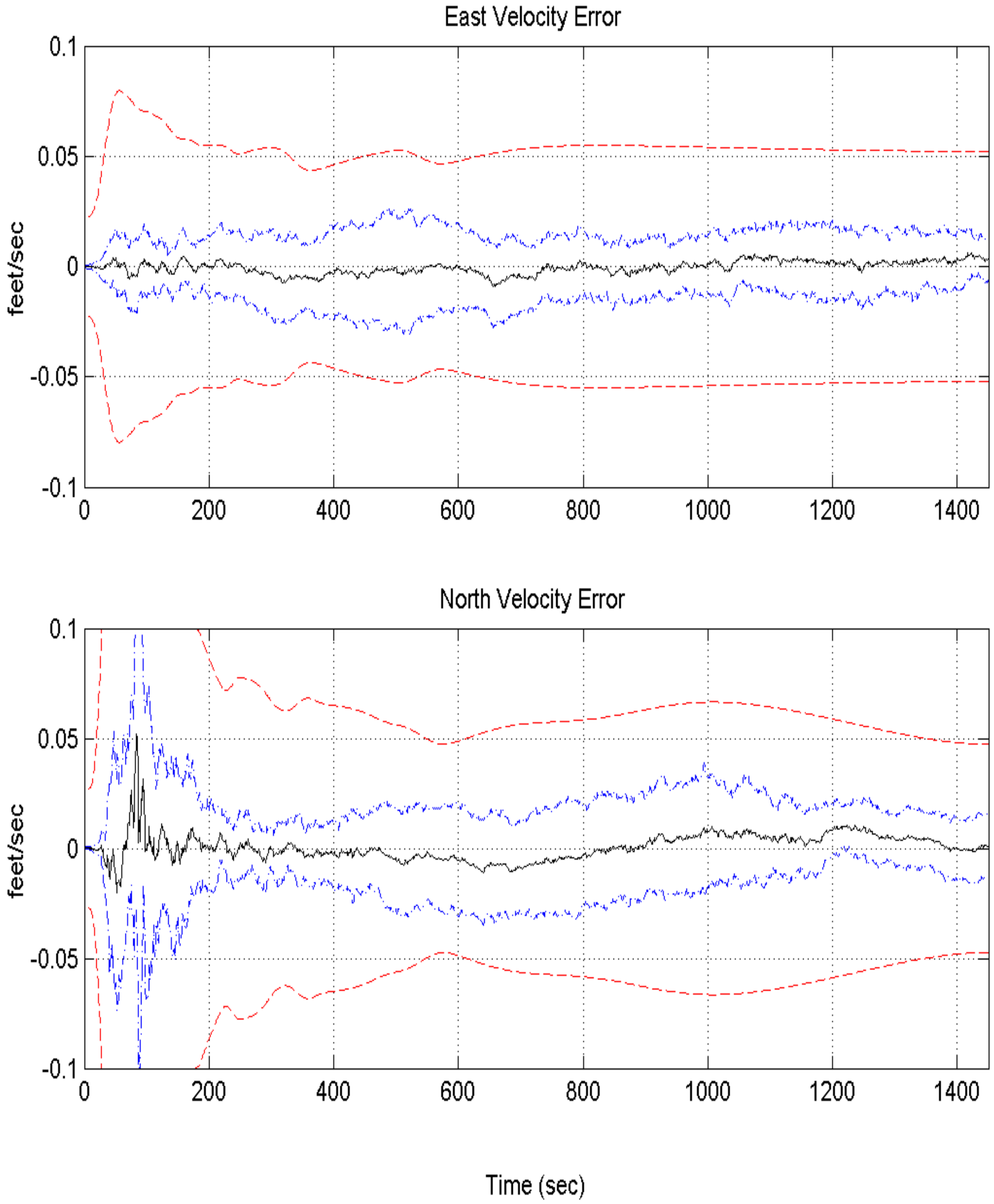


Figure H-2. Latitude and Longitude Errors (Non-Conservative Q-Tuning)



East Velocity Error

North Velocity Error

Figure H-3. East and North Velocity Errors (Conservative Q-Tuning)

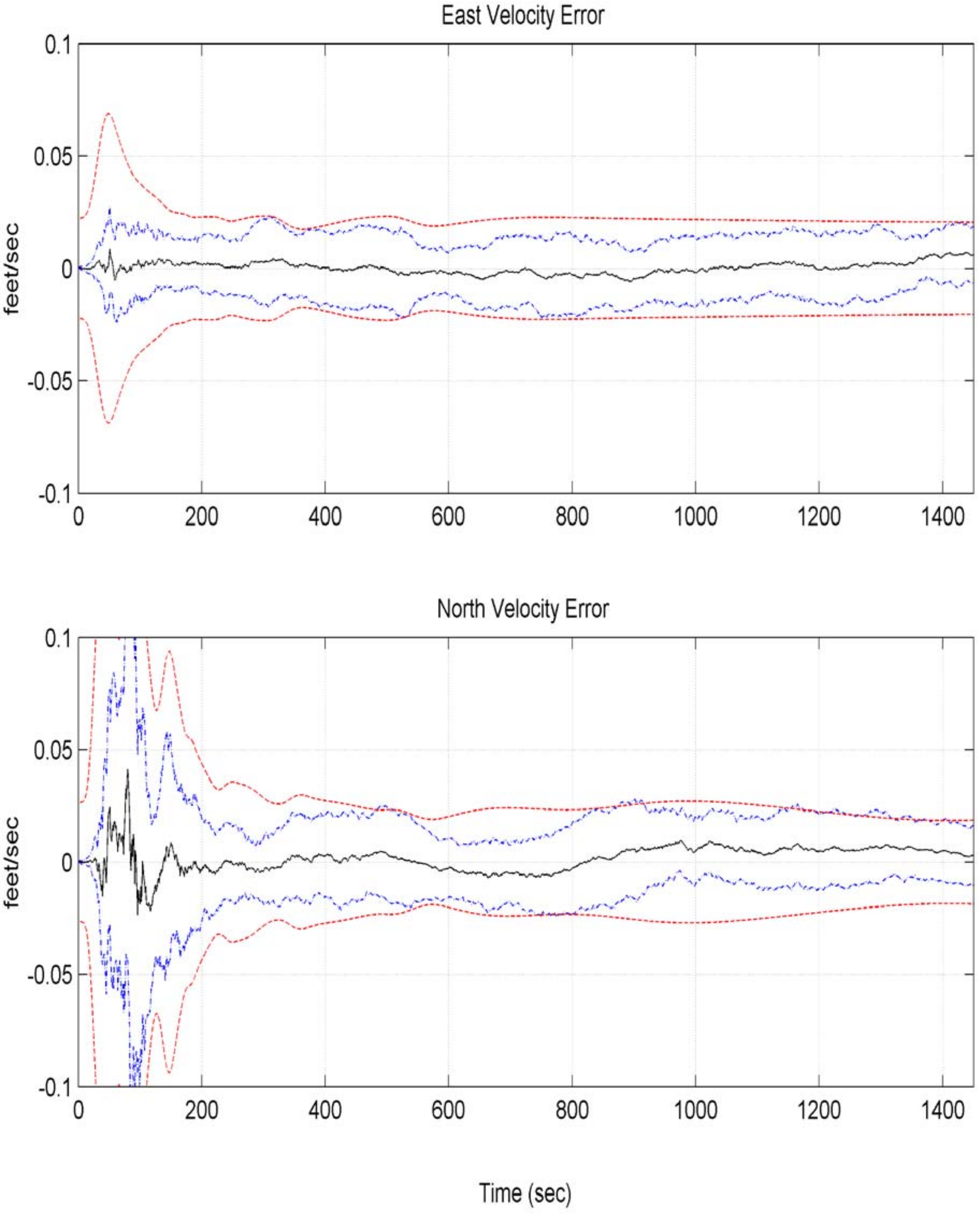


Figure H-4. East and North Velocity Errors (Non-Conservative Q-Tuning)

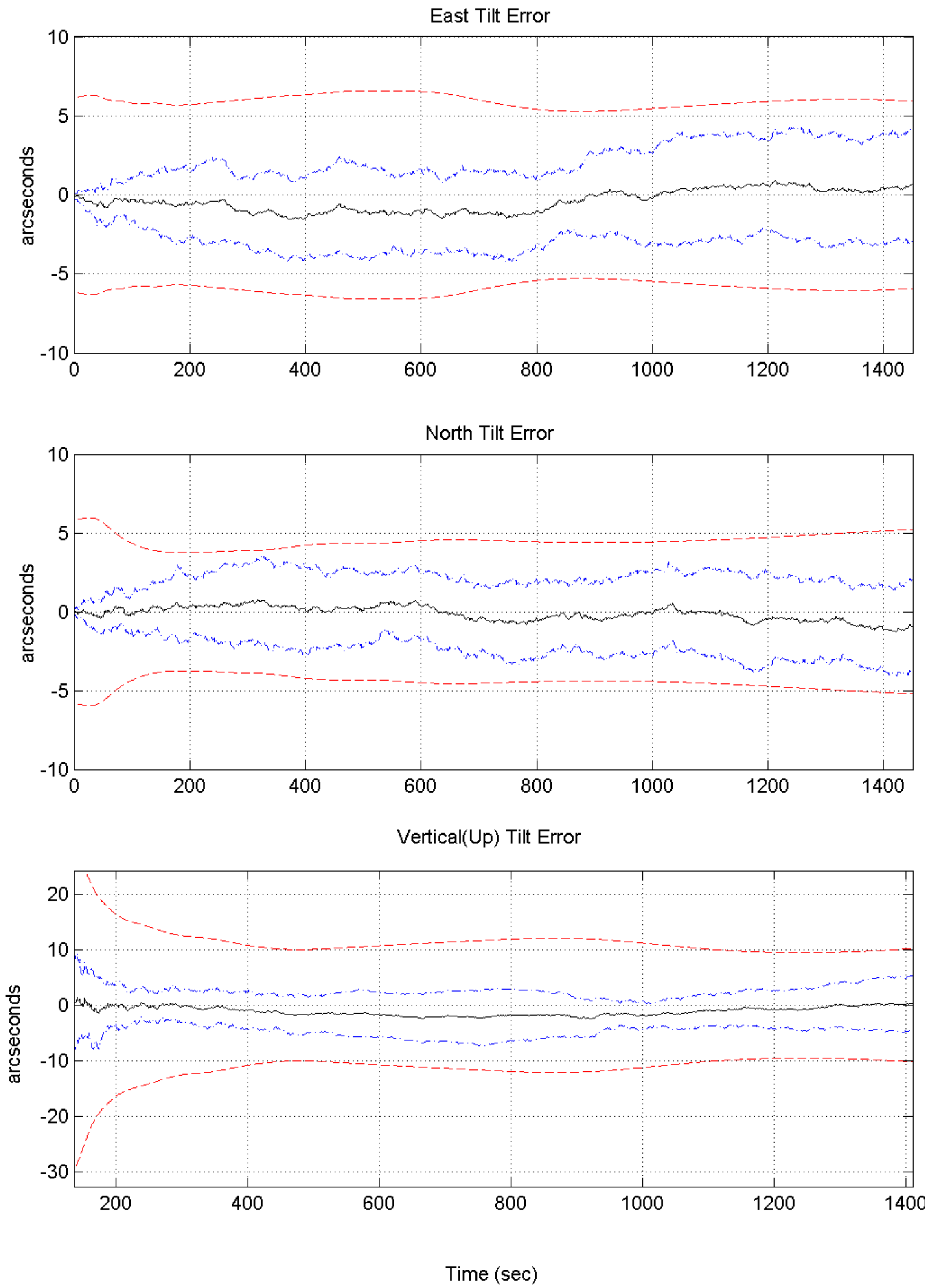


Figure H-5. East, North, and Up Tilt Errors (Conservative Q-Tuning)

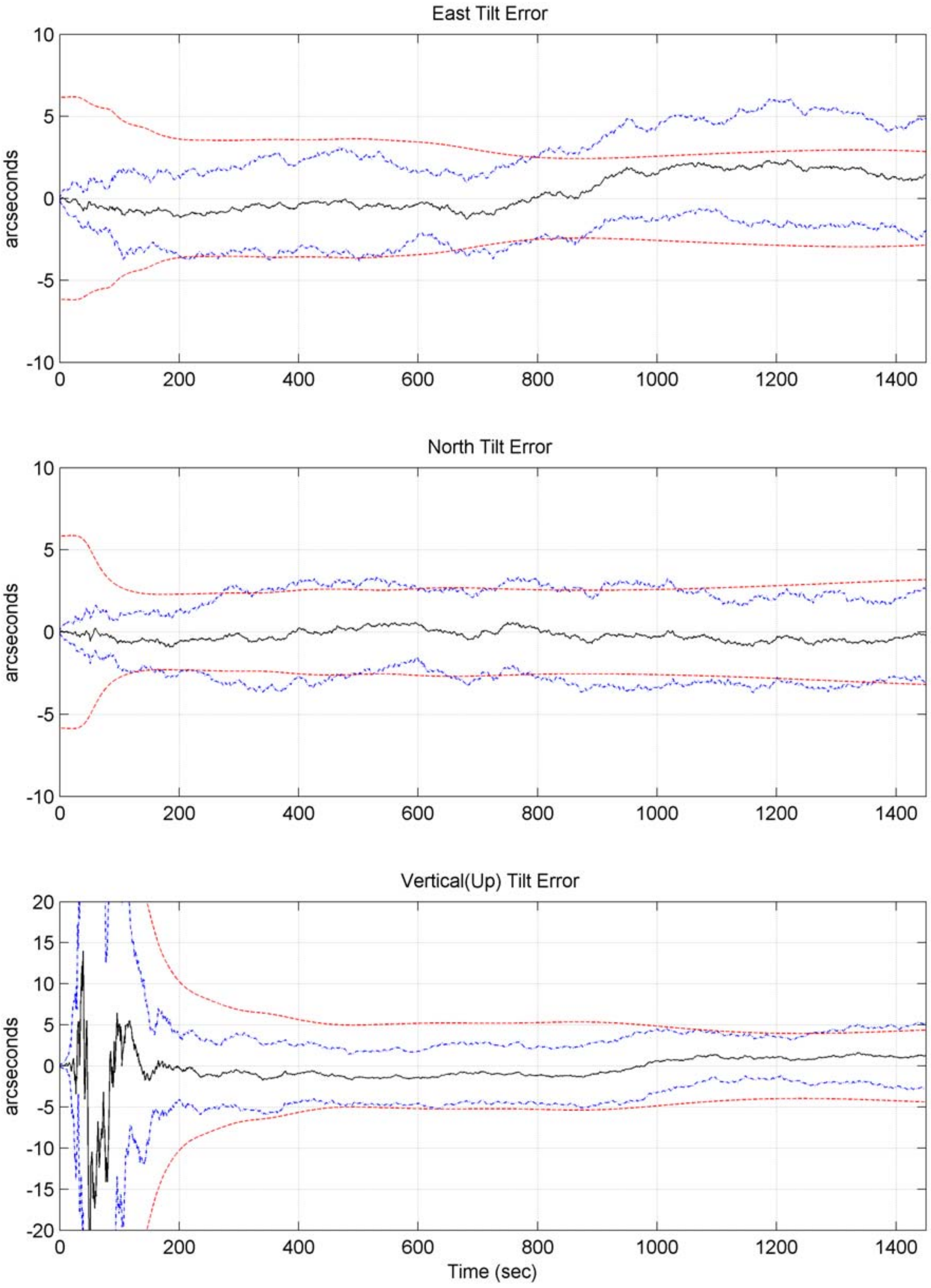


Figure H-6. East, North, and Up Tilt Errors (Non-Conservative Q-Tuning)

Appendix I

Plots of Case I

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)

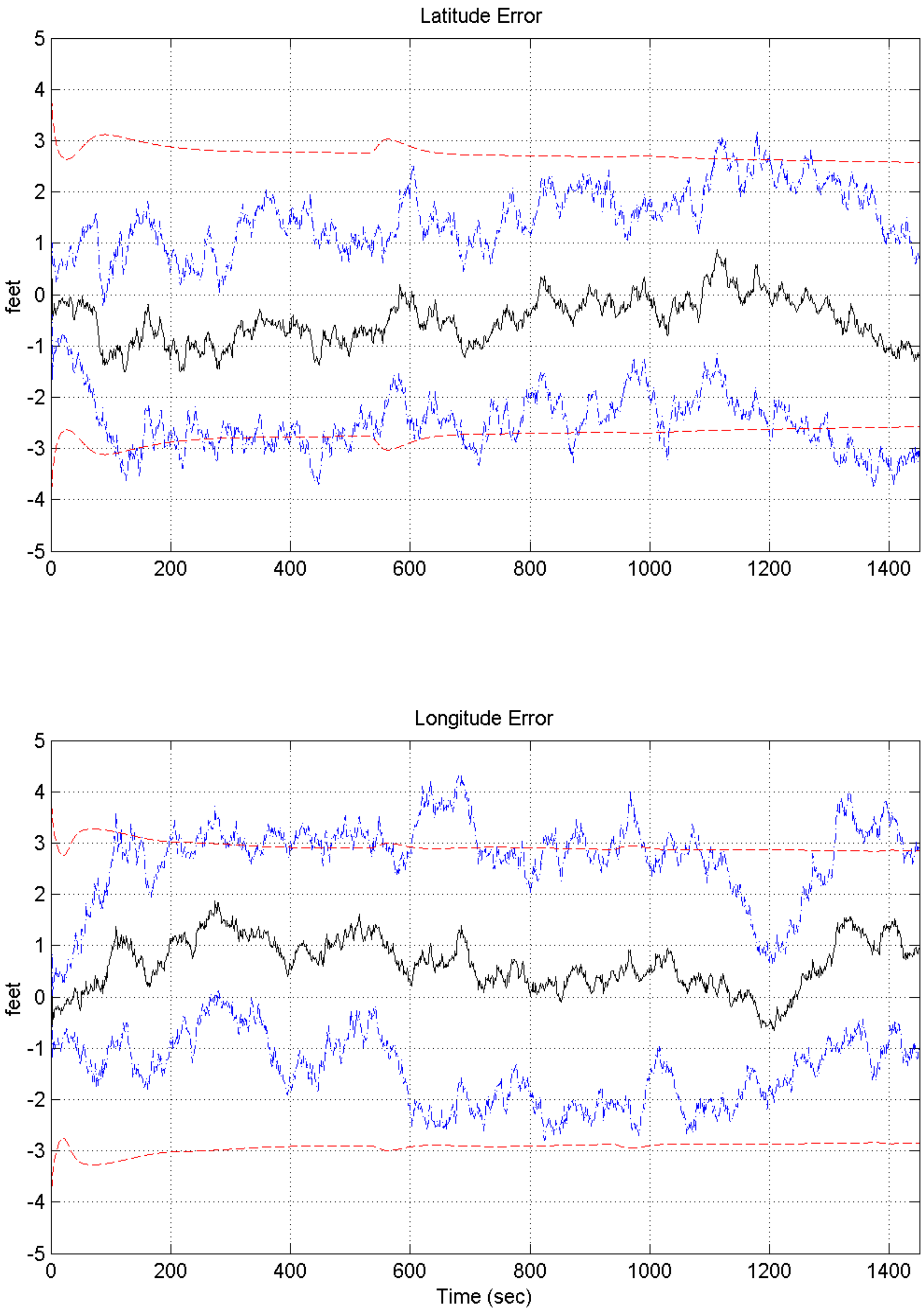


Figure I-1. Rocket Latitude and Longitude Errors

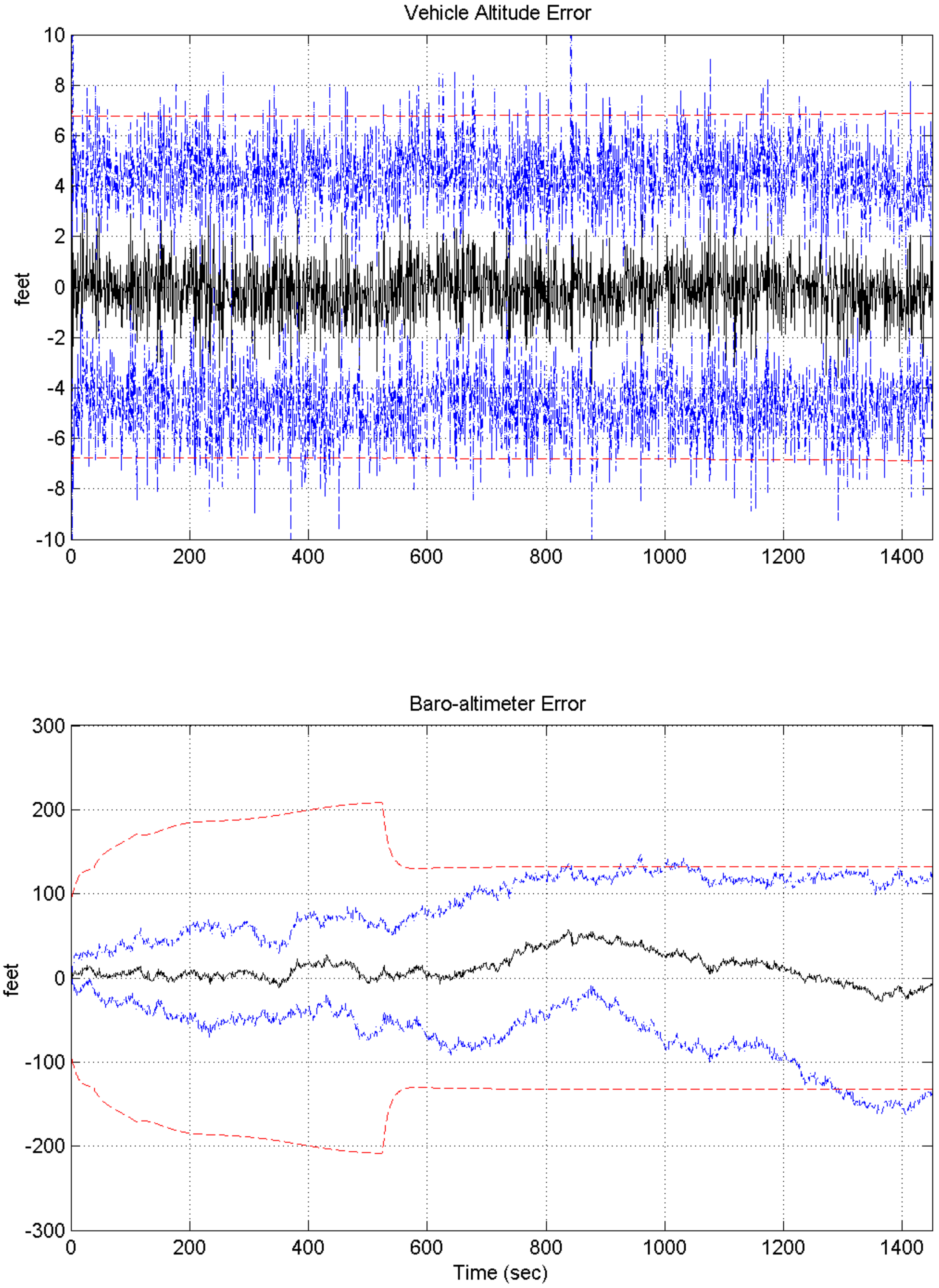


Figure I-2. Rocket Altitude and Baro-Altimeter Errors

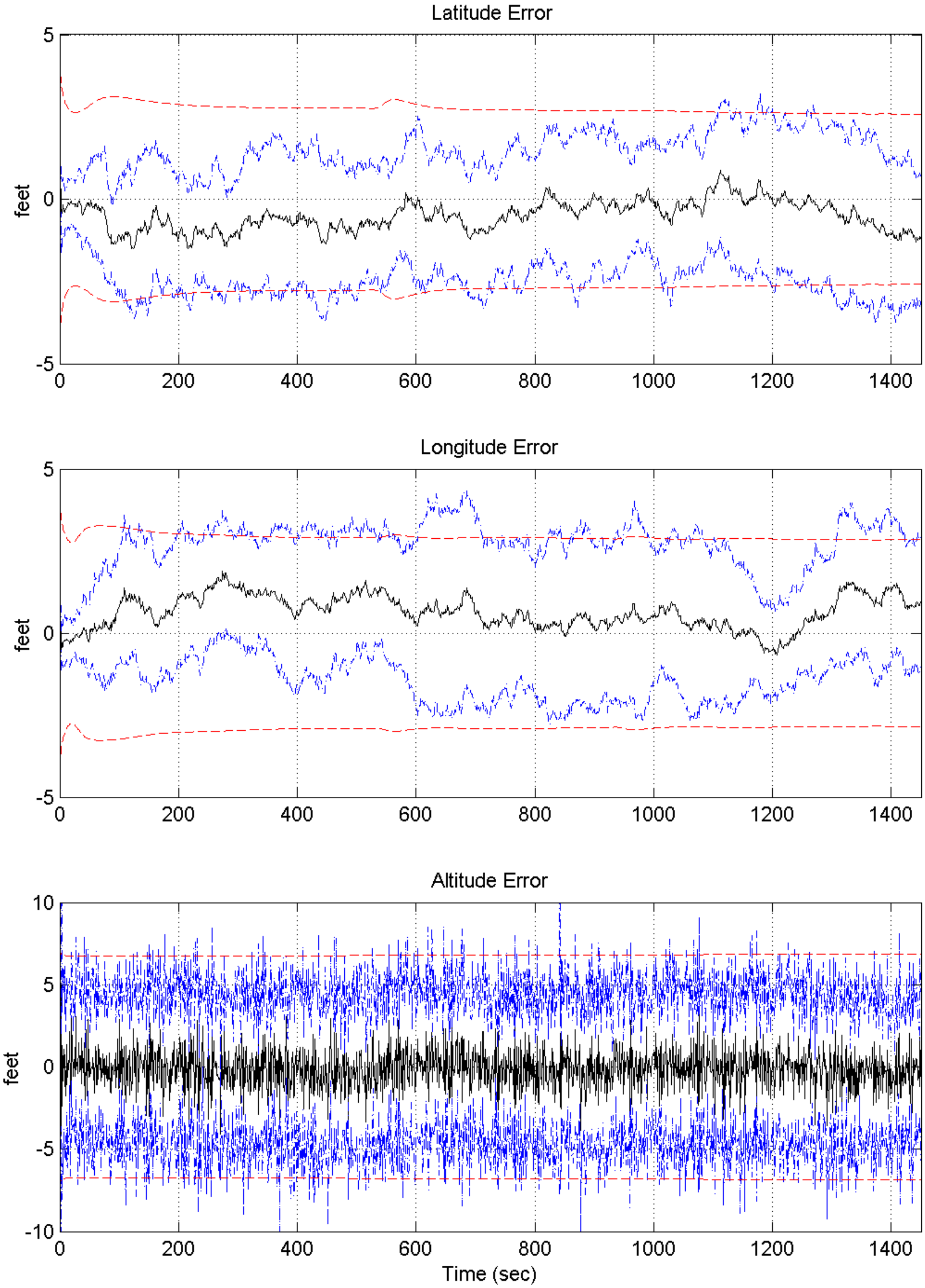


Figure I-3. Rocket Latitude, Longitude, and Altitude Errors

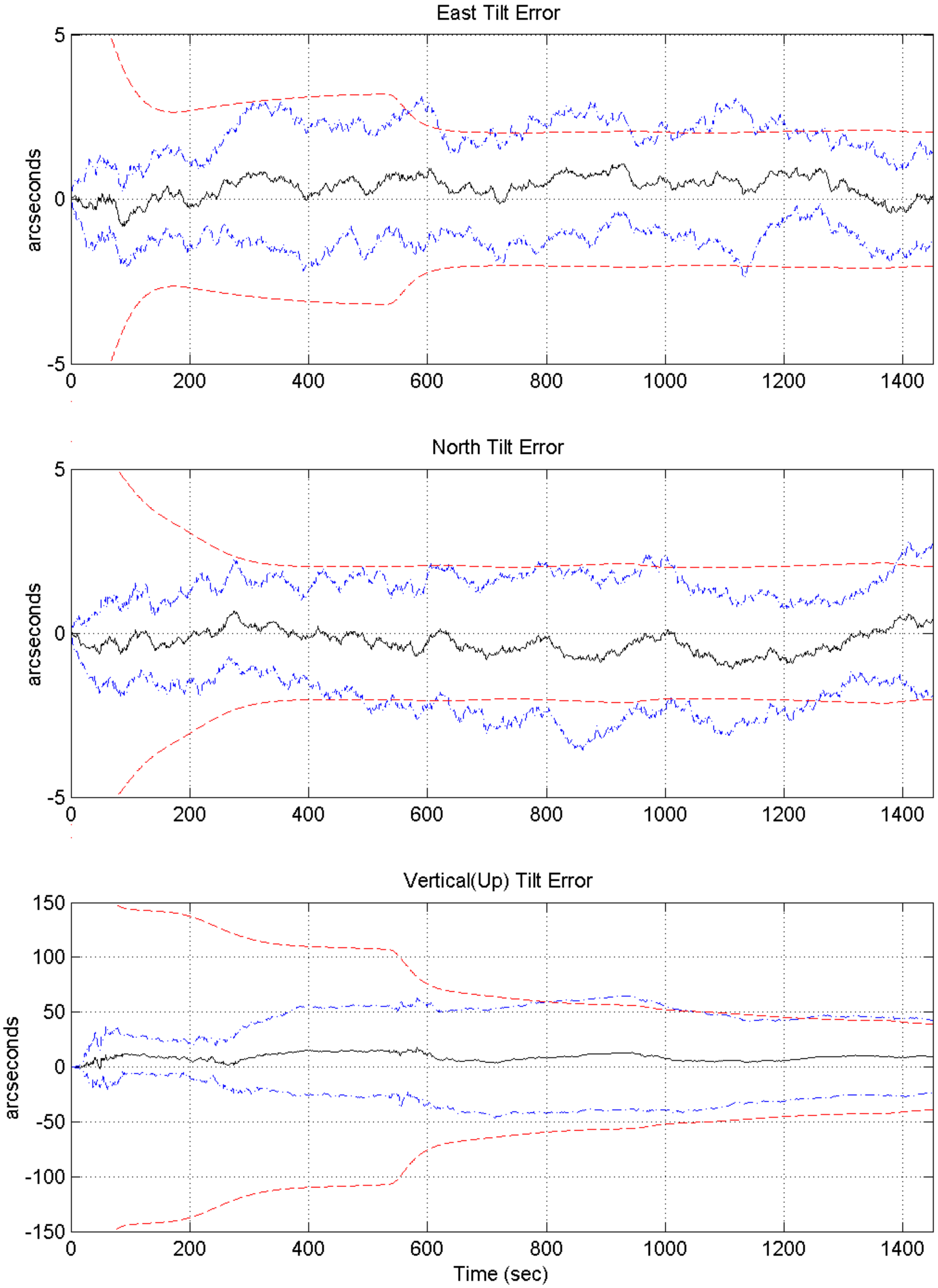


Figure I-4. East, North, and Vertical Tilt Errors

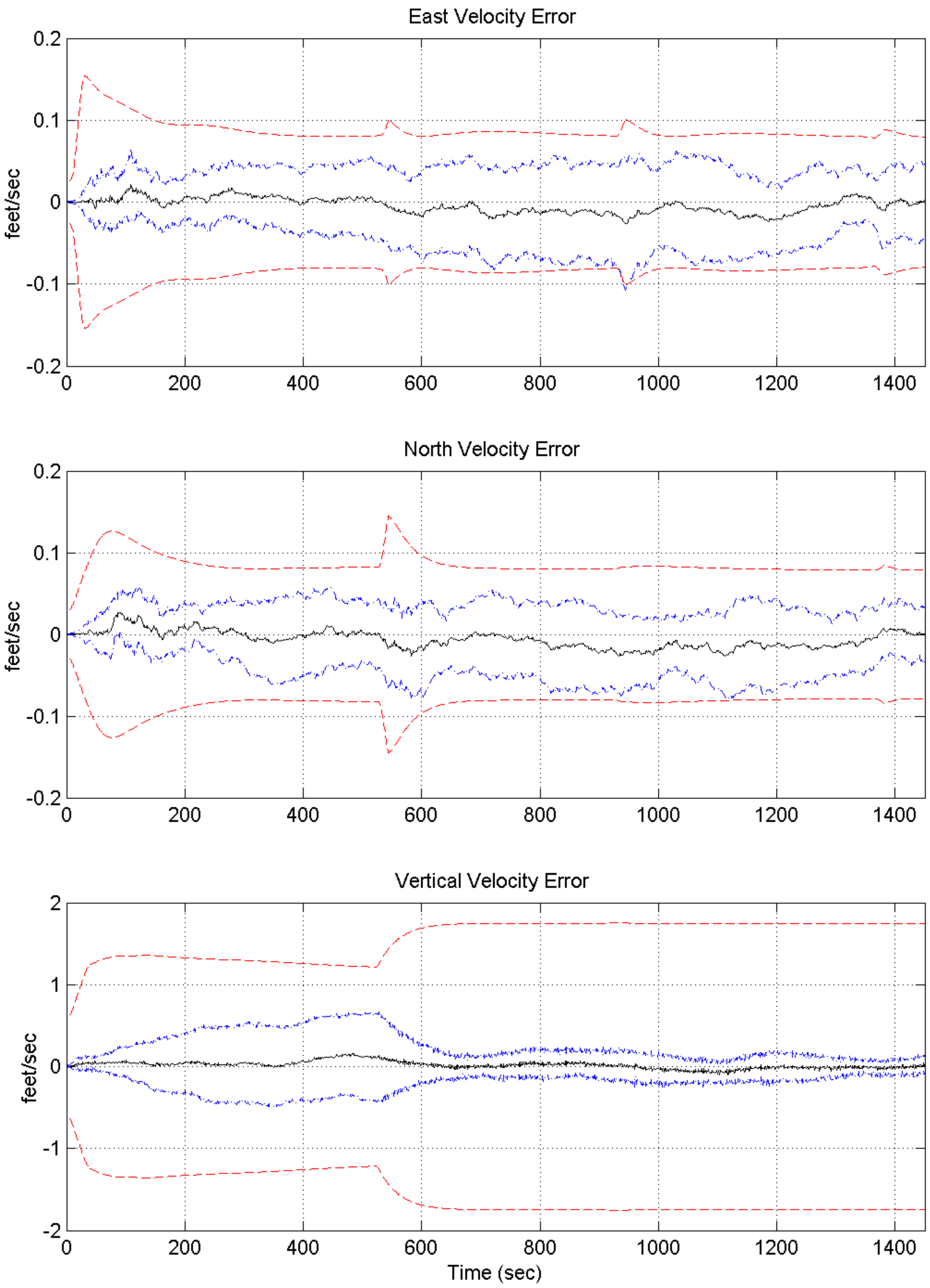


Figure I-5. East, North, and Vertical Velocity Errors

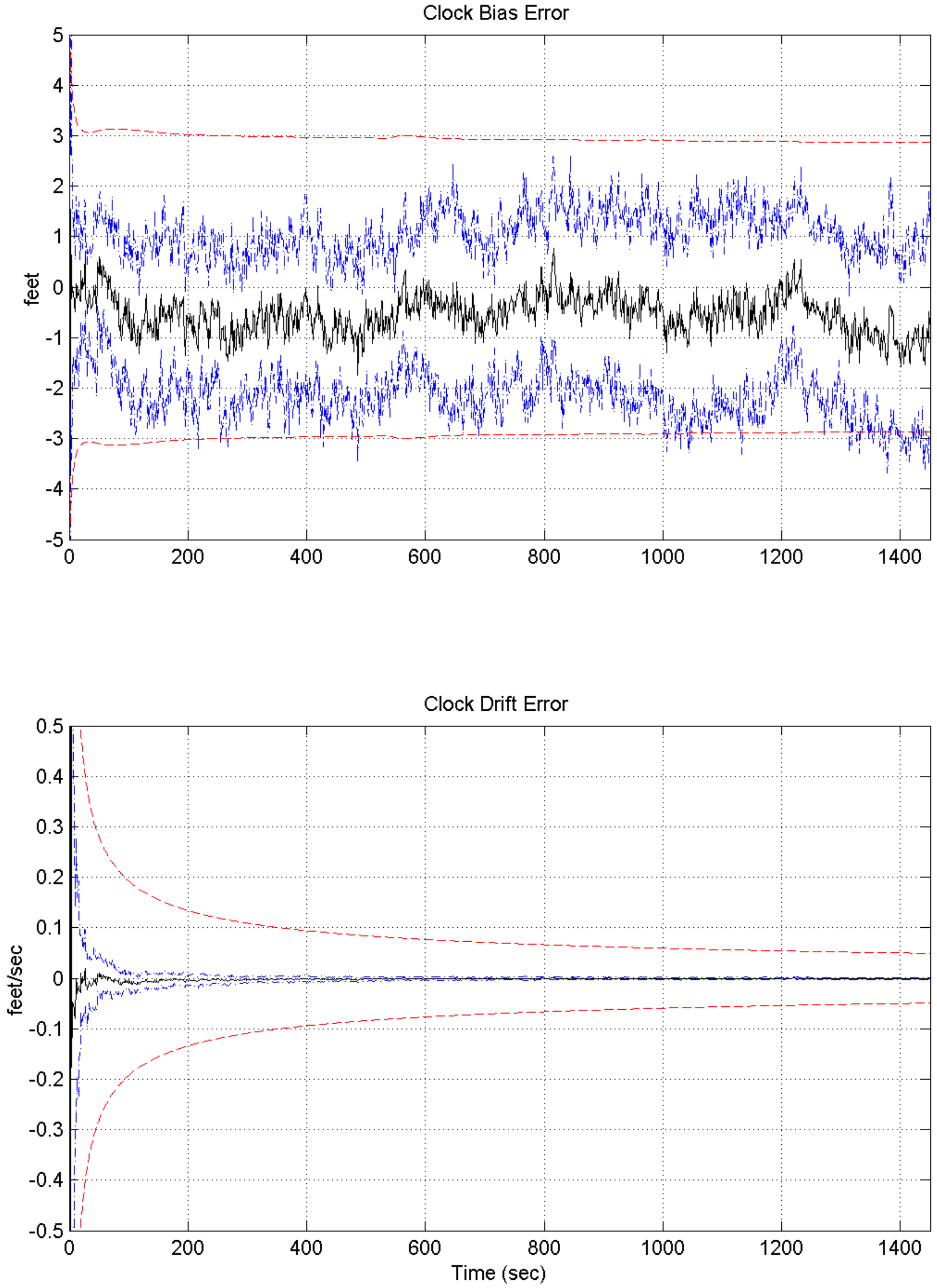


Figure I- 6. GPS User Clock Bias and Clock Drift Errors

Appendix J

Plots of Case II

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)

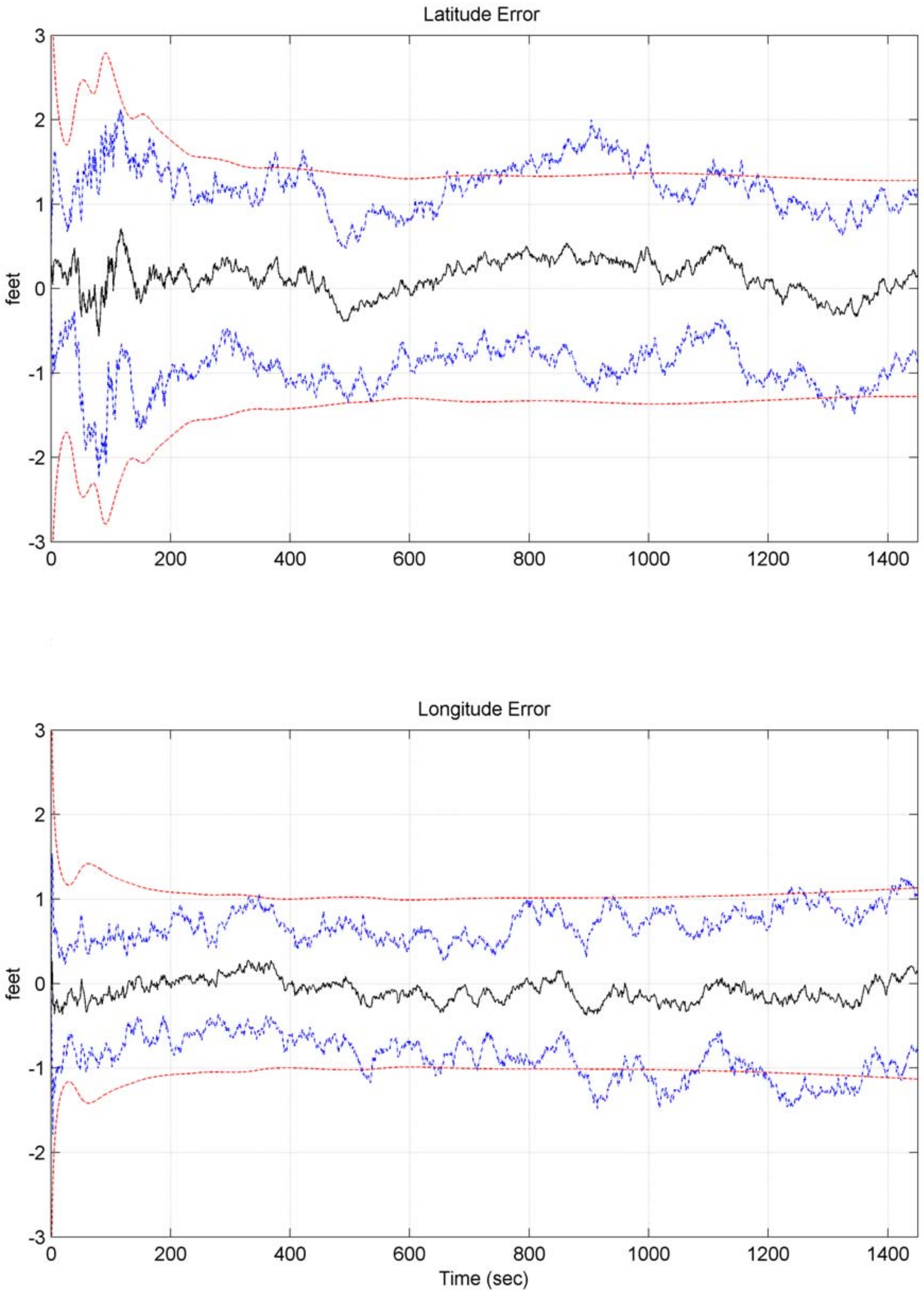


Figure J-1. Rocket Latitude and Longitude Errors

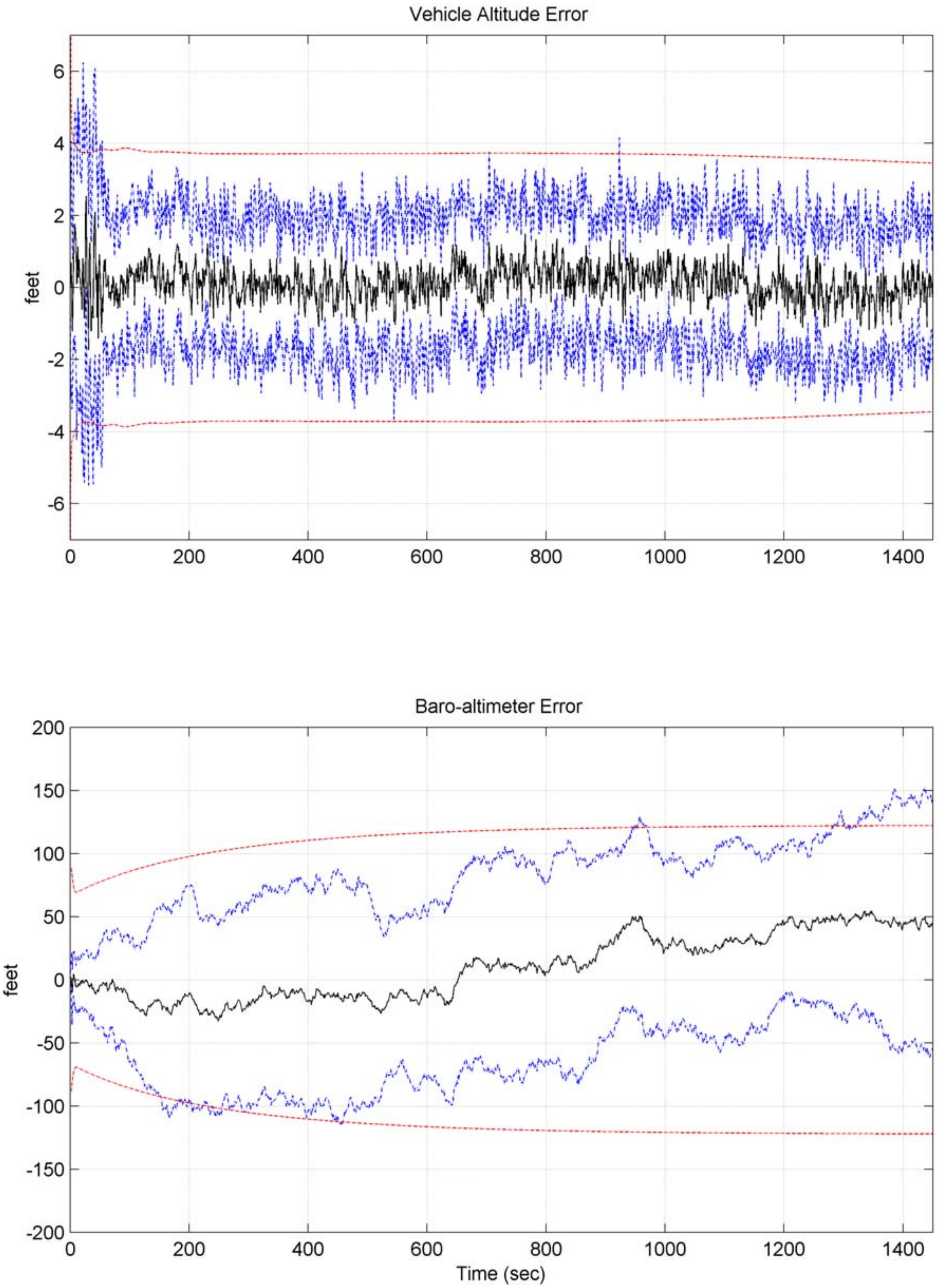
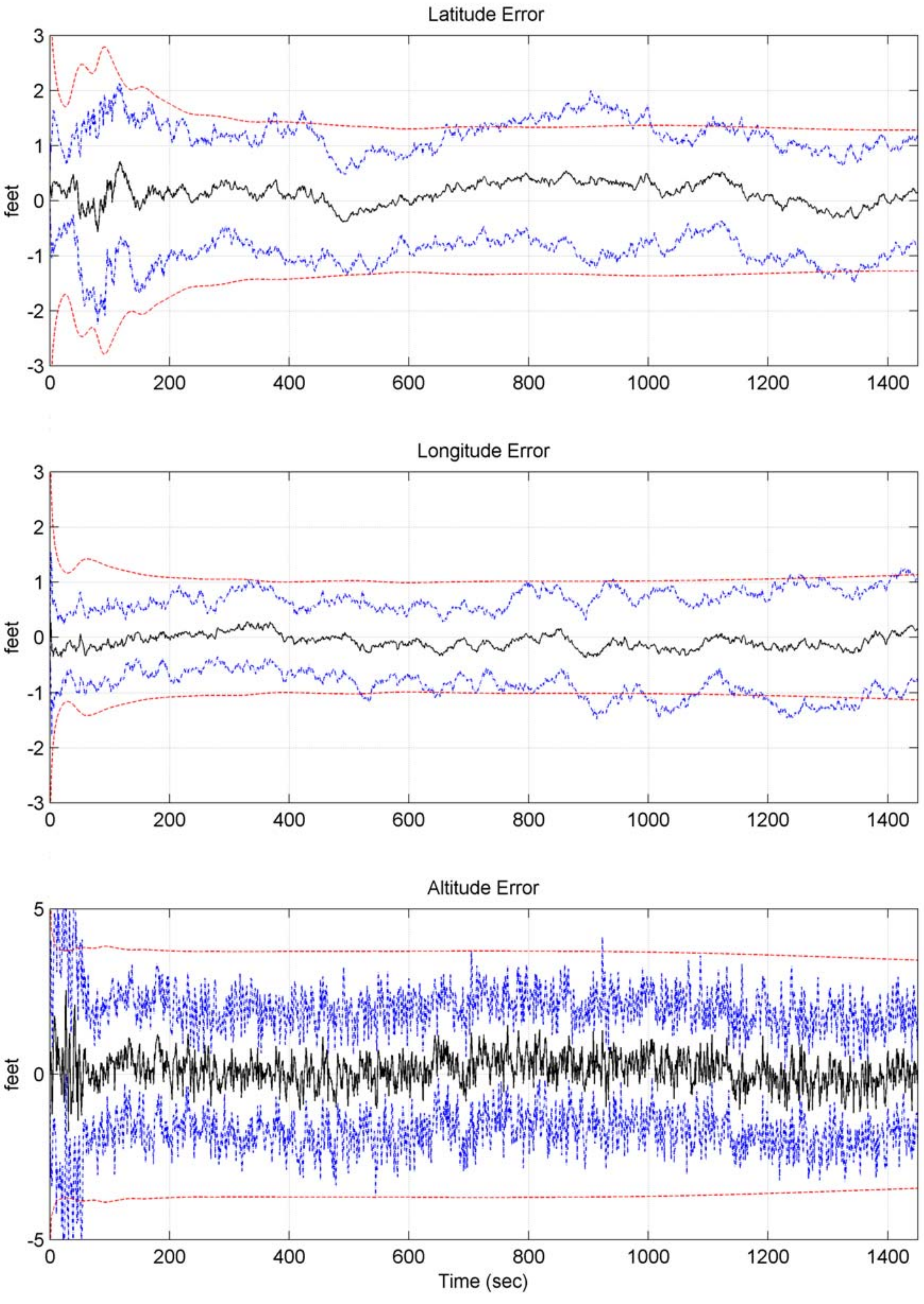


Figure J-2. Rocket Altitude and Baro-Altimeter Errors


Latitude Error

Longitude Error

Altitude Error

Figure J-3. Rocket Latitude, Longitude, and Altitude Errors

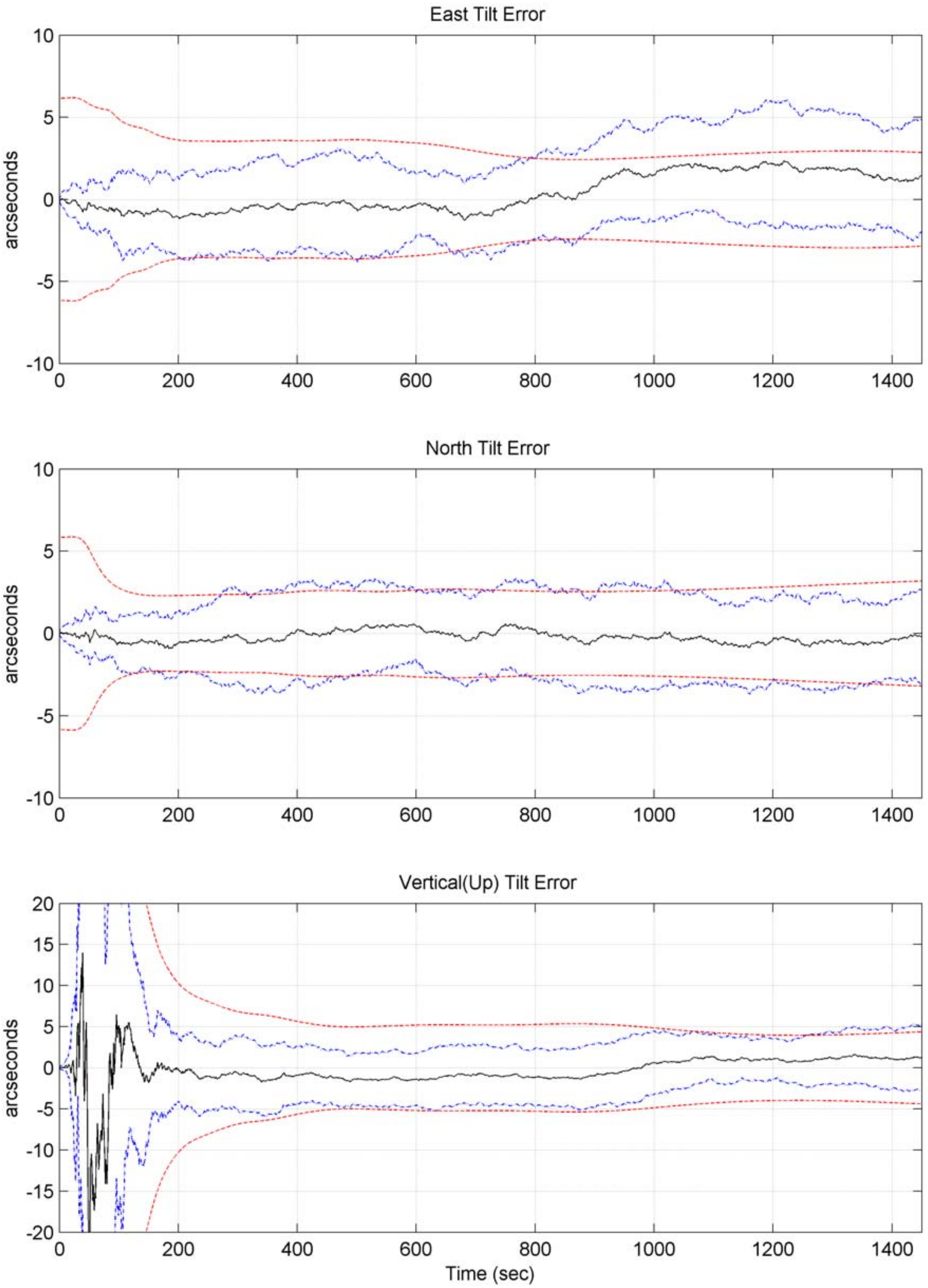


Figure J-4. East, North, and Vertical Tilt Errors

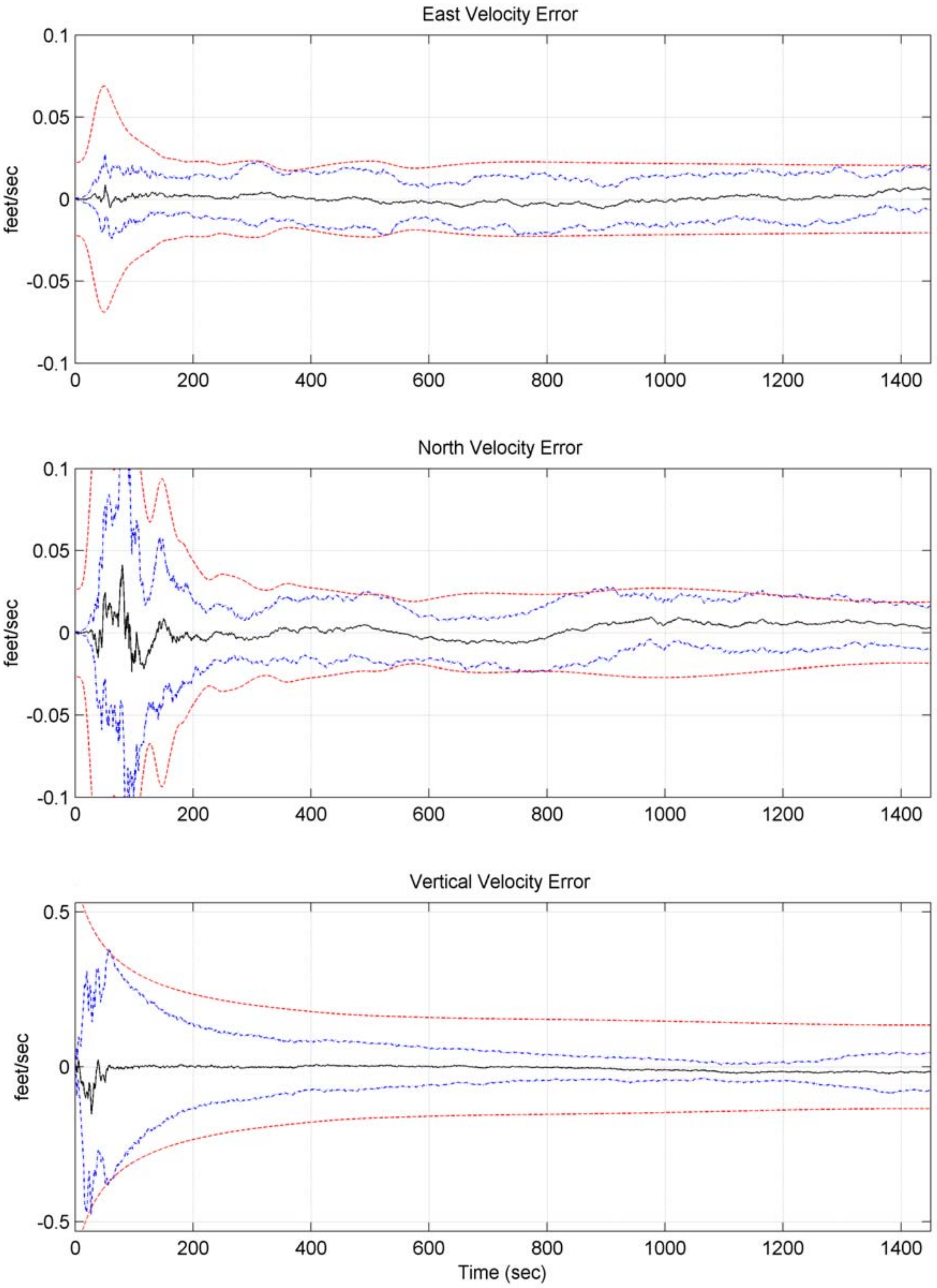


Figure J-5. East, North, and Vertical Velocity Errors

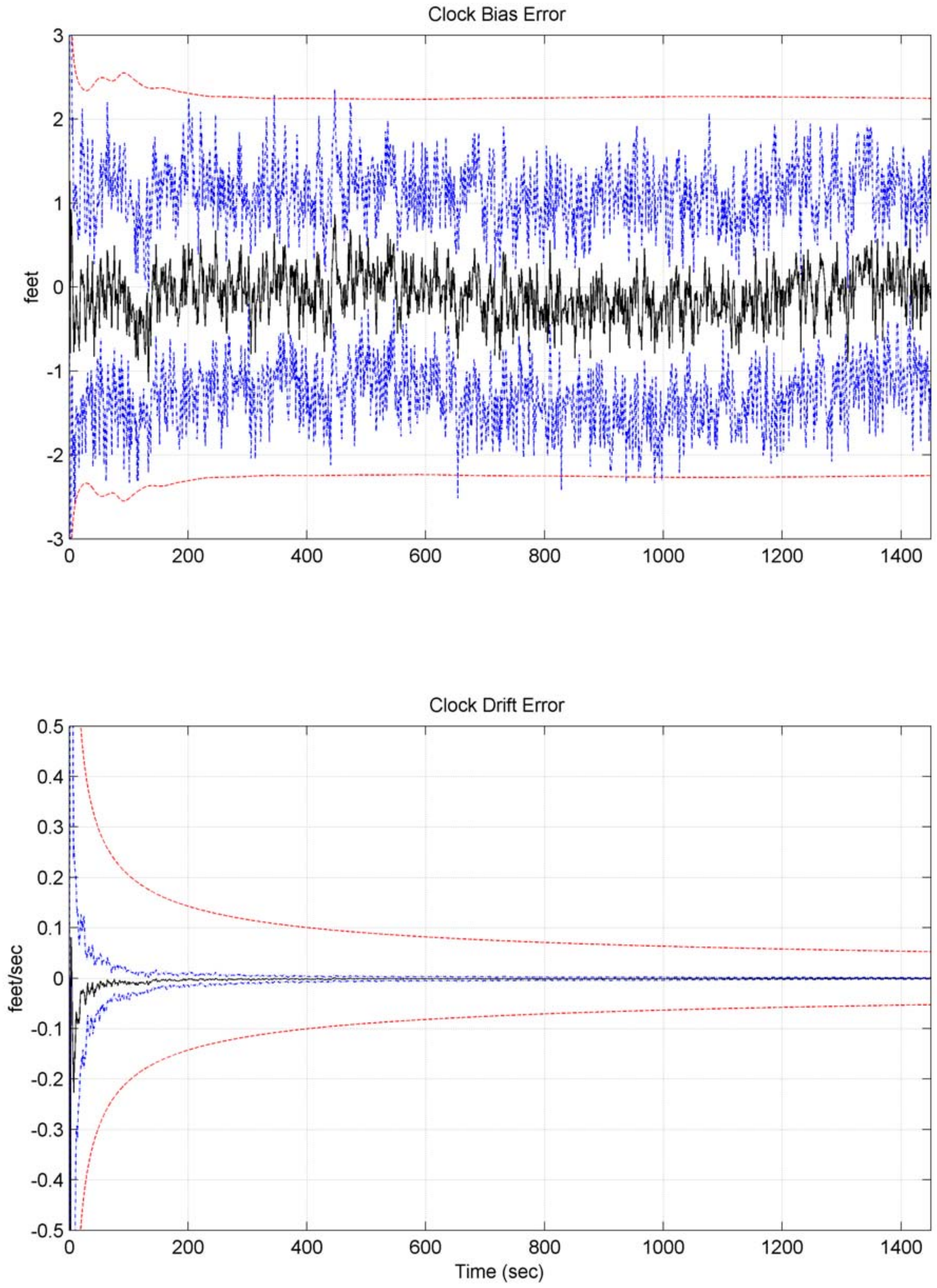


Figure J-6. DGPS User Clock Bias and Clock Drift Errors

Appendix K

Plots of Case III

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)

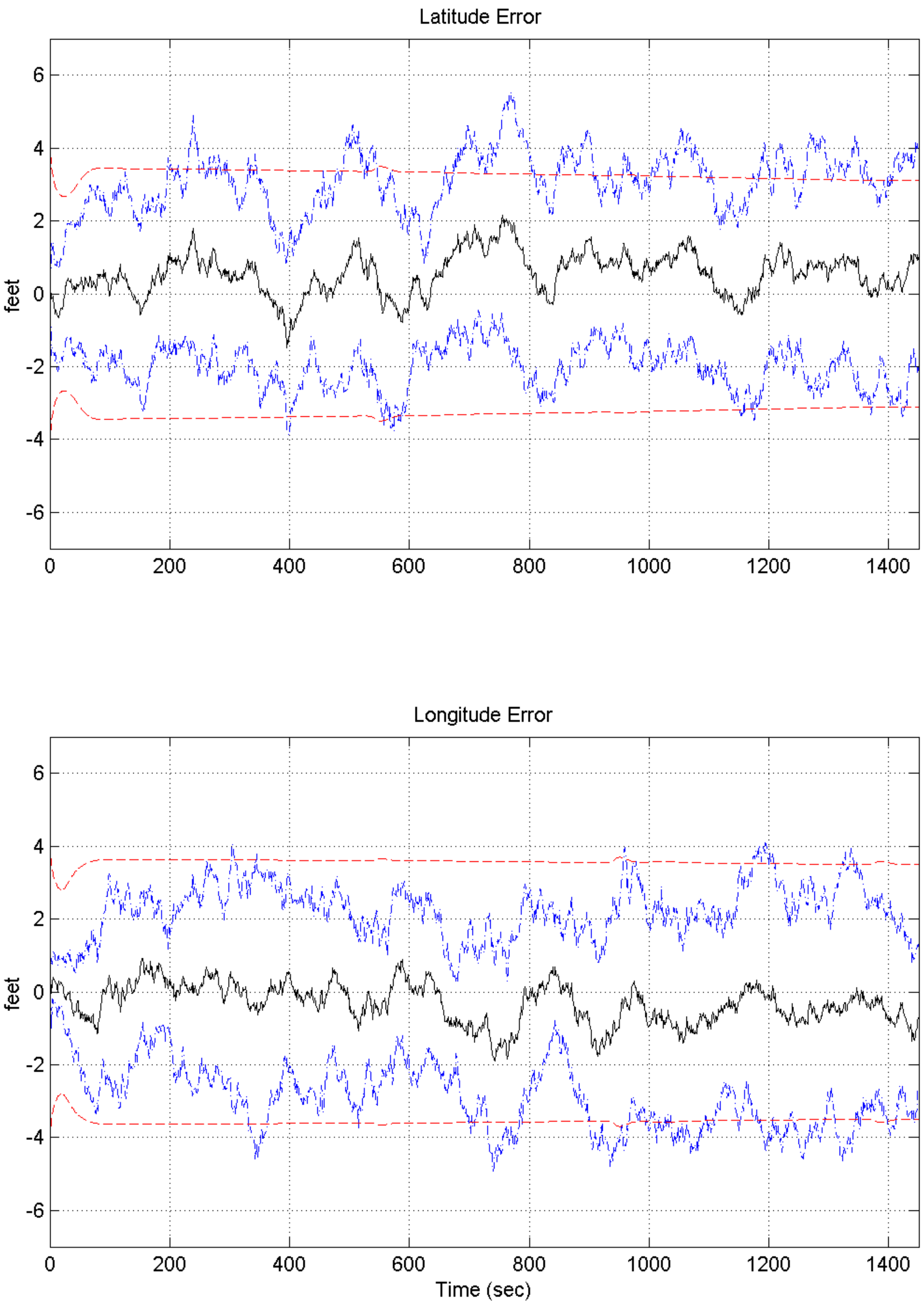


Figure K-1. Rocket Latitude and Longitude Errors

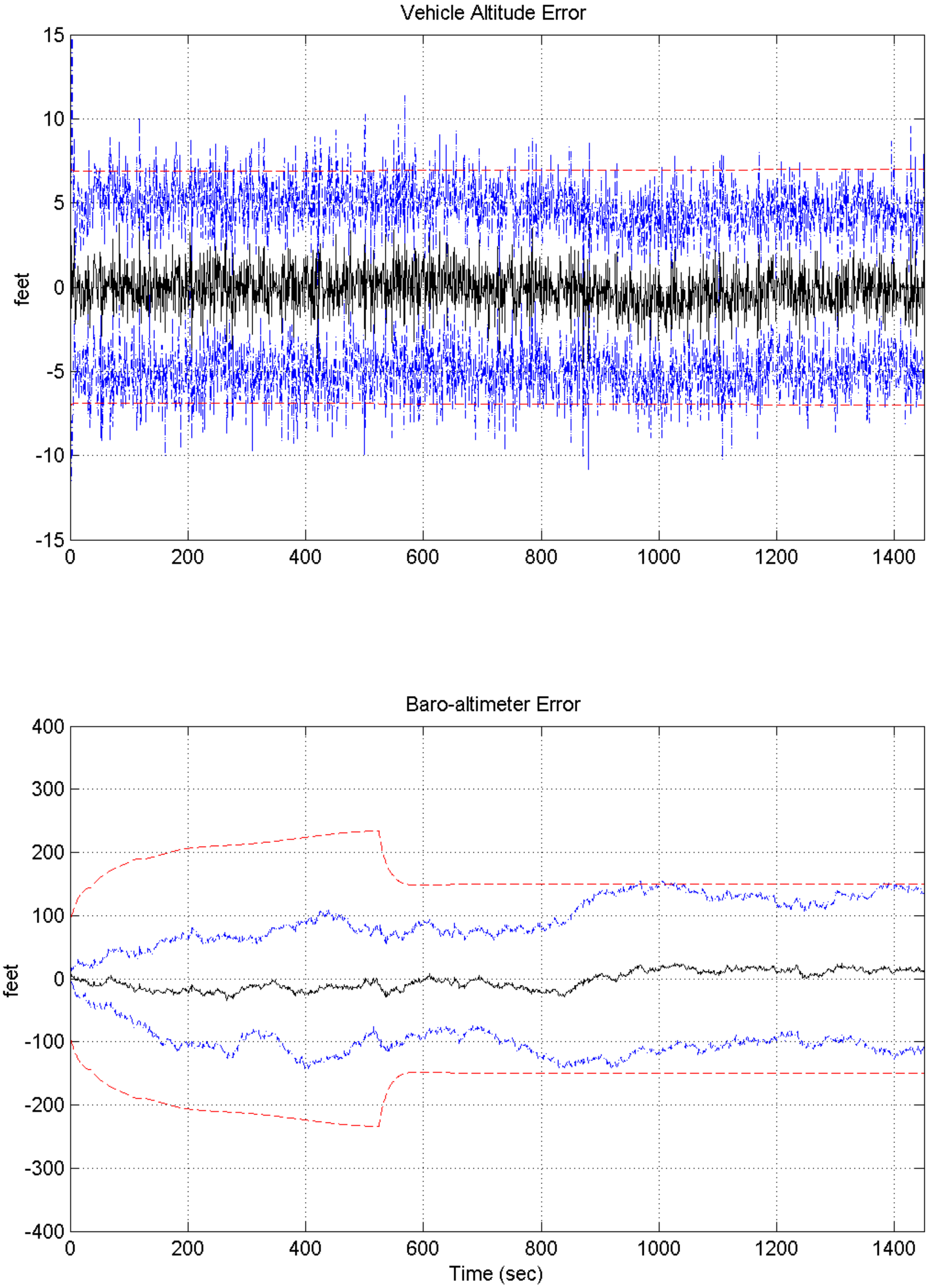


Figure K-2. Rocket Altitude and Baro-Altimeter Errors

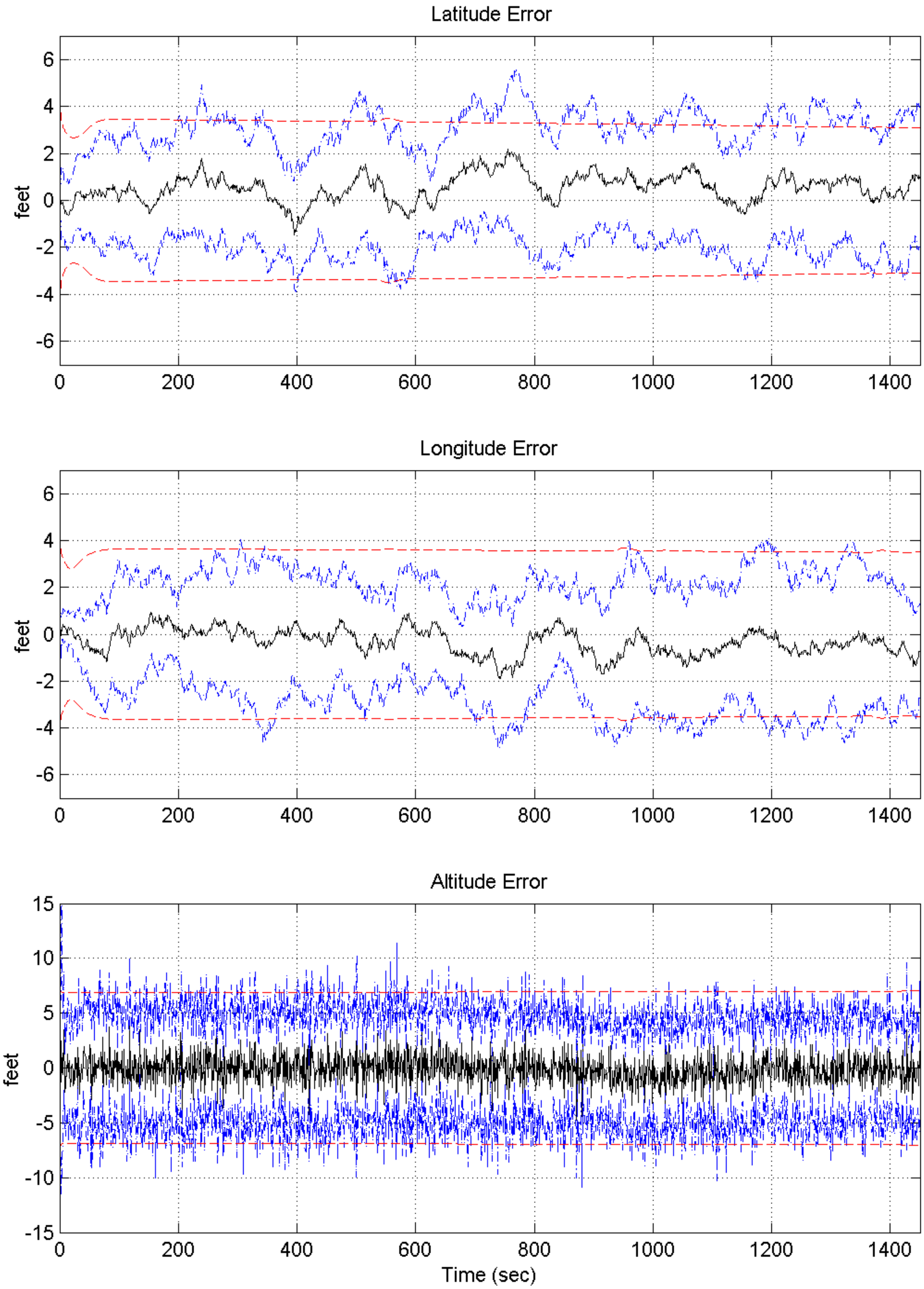
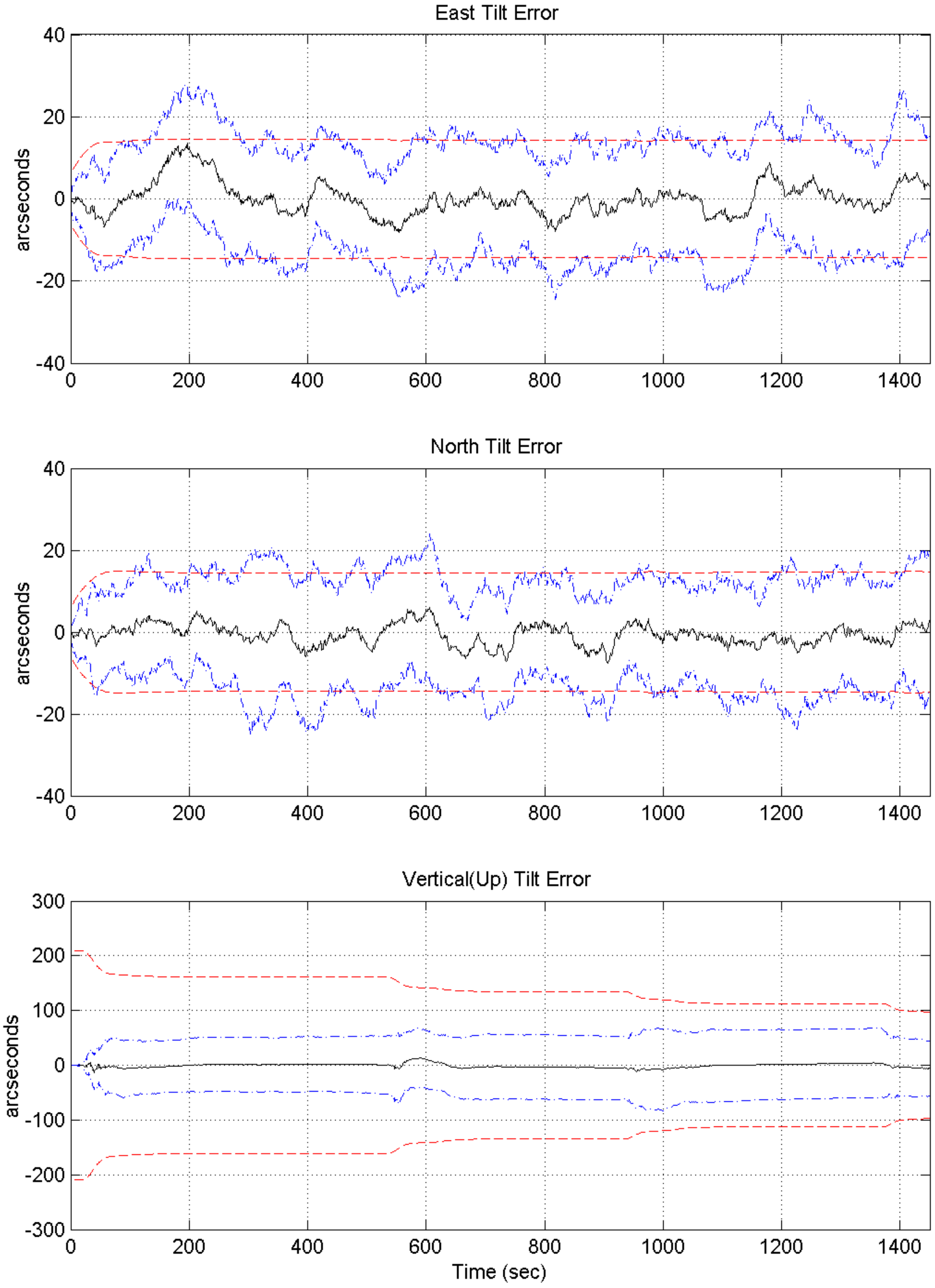


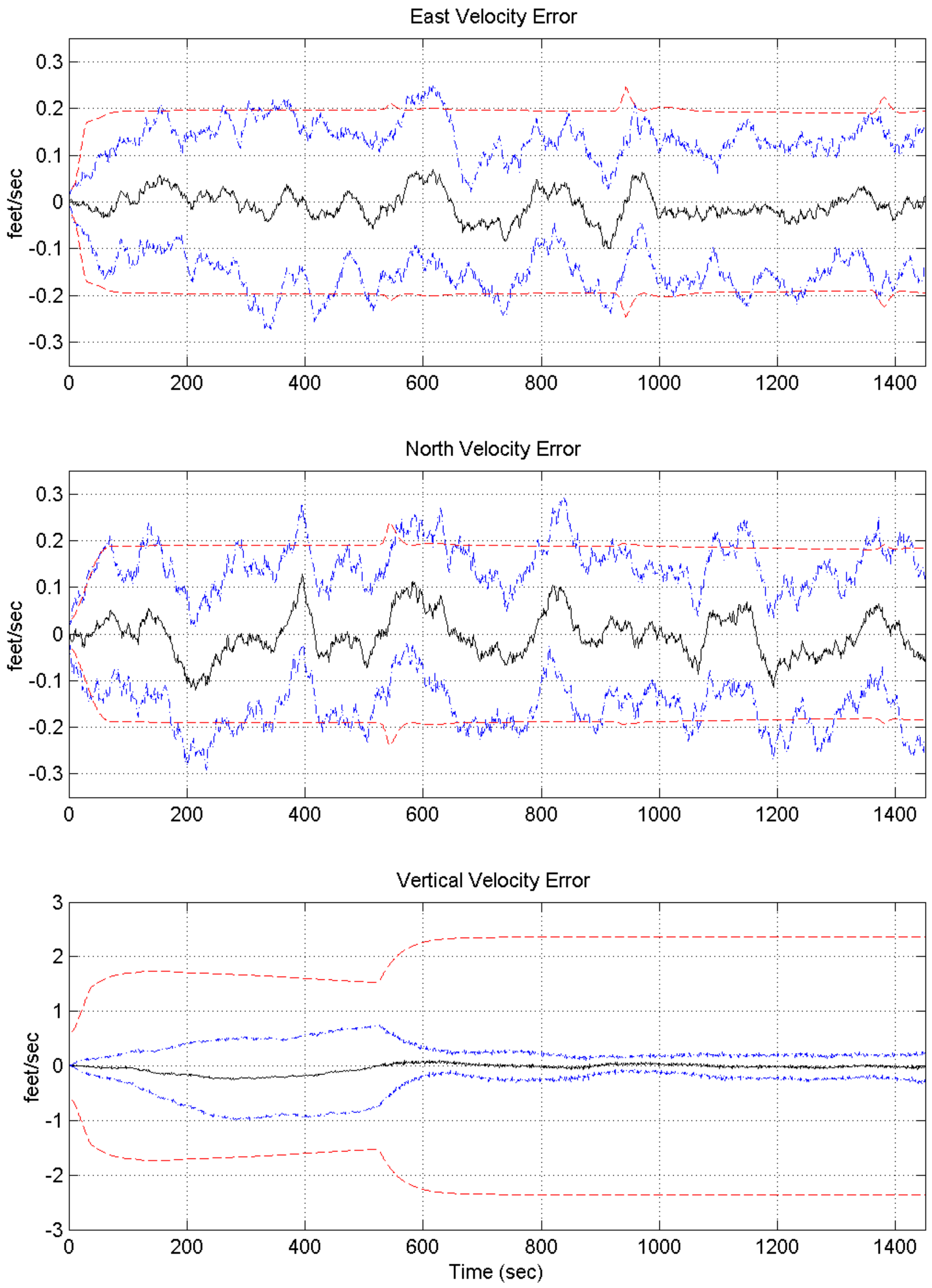
Figure K-3. Rocket Latitude, Longitude, and Altitude Errors


East Tilt Error

North Tilt Error

Vertical(Up) Tilt Error

Figure K-4. East, North, and Vertical Tilt Errors

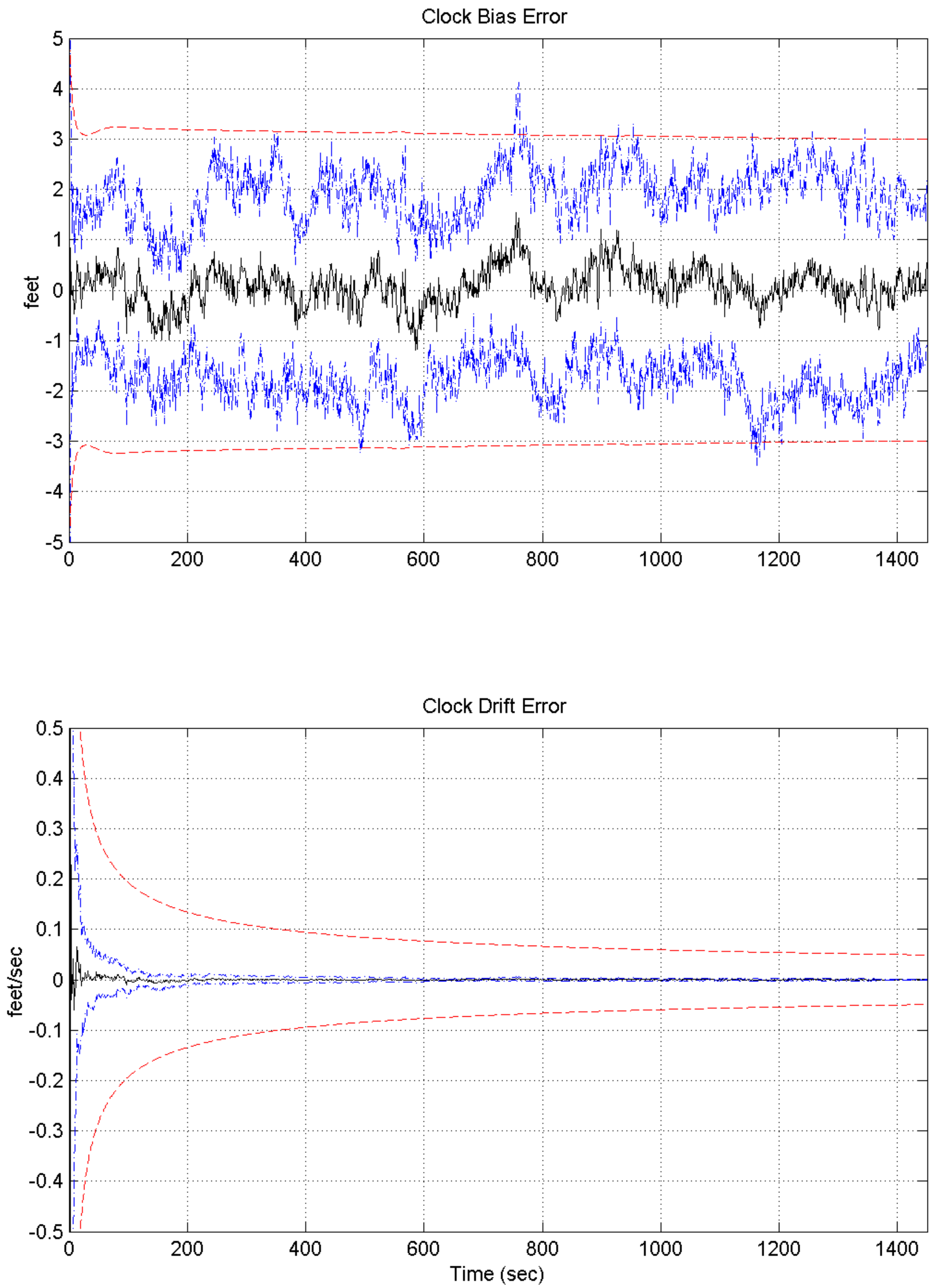


East Velocity Error

North Velocity Error

Vertical Velocity Error

Figure K-5. East, North, and Vertical Velocity Errors



Clock Bias Error

Clock Drift Error

Figure K-6. GPS User Clock Bias and Clock Drift Errors

Appendix L

Plots of Case IV

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)

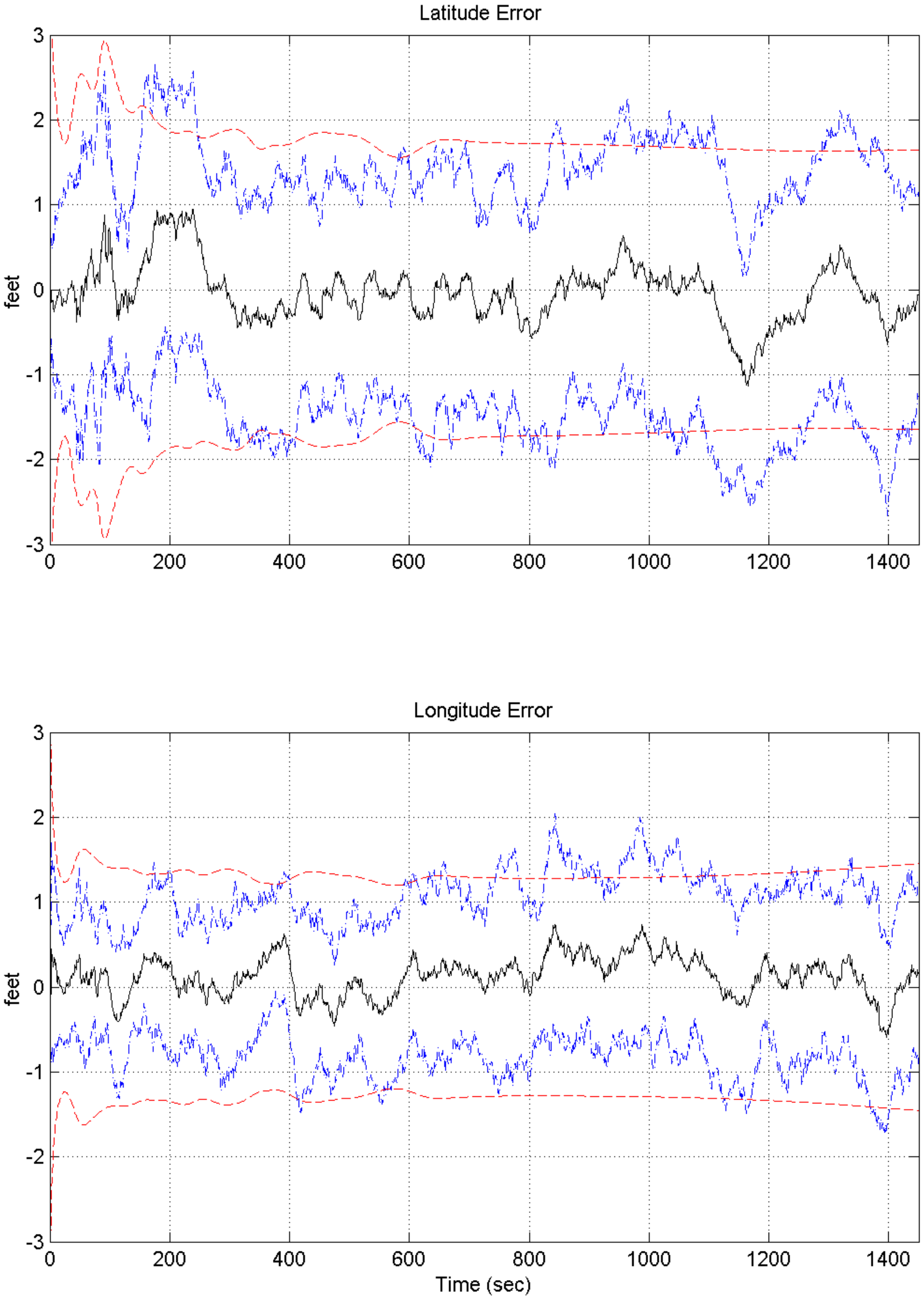


Figure L-1. Rocket Latitude and Longitude Errors

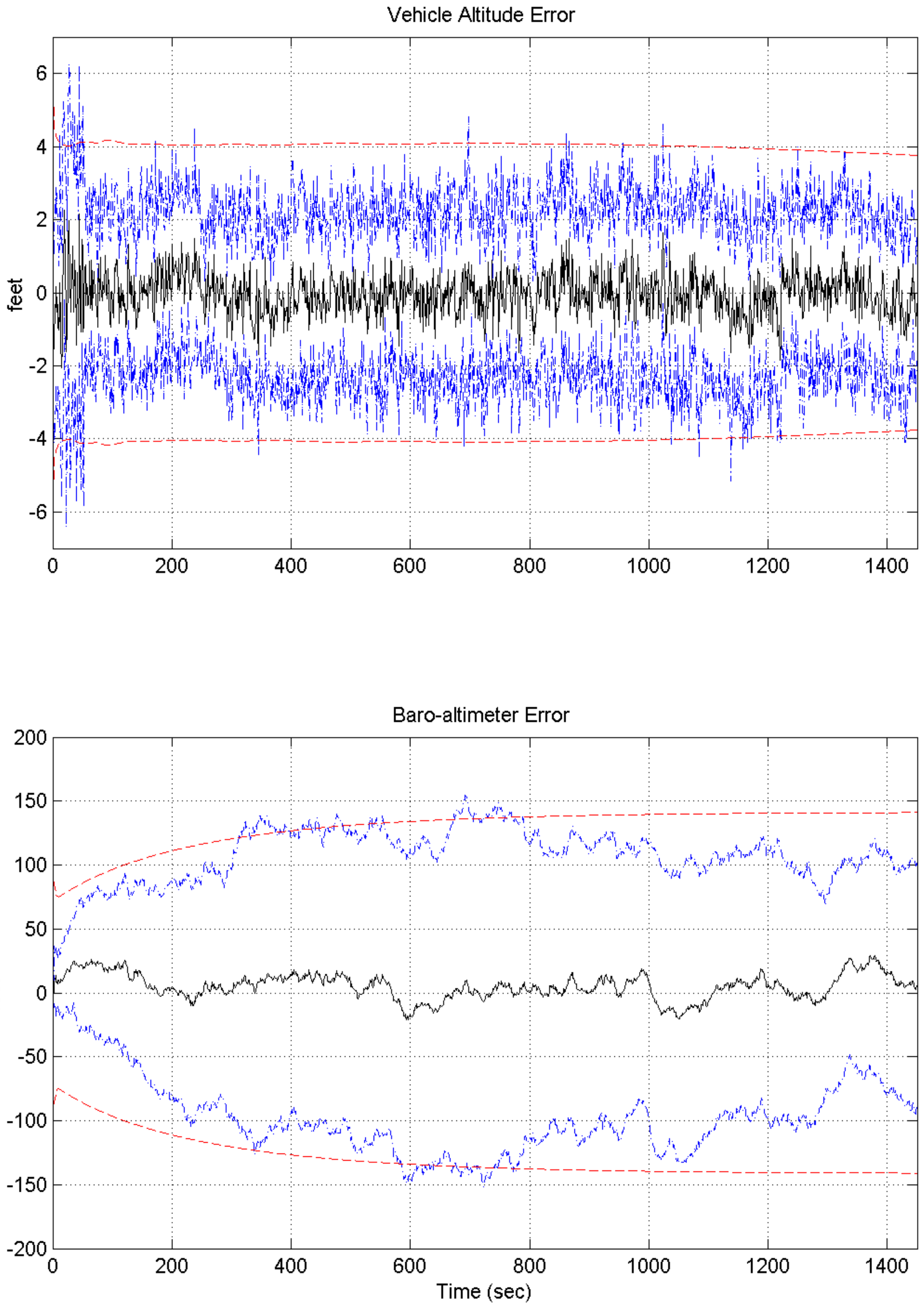


Figure L-2. Rocket Altitude and Baro-Altimeter Errors

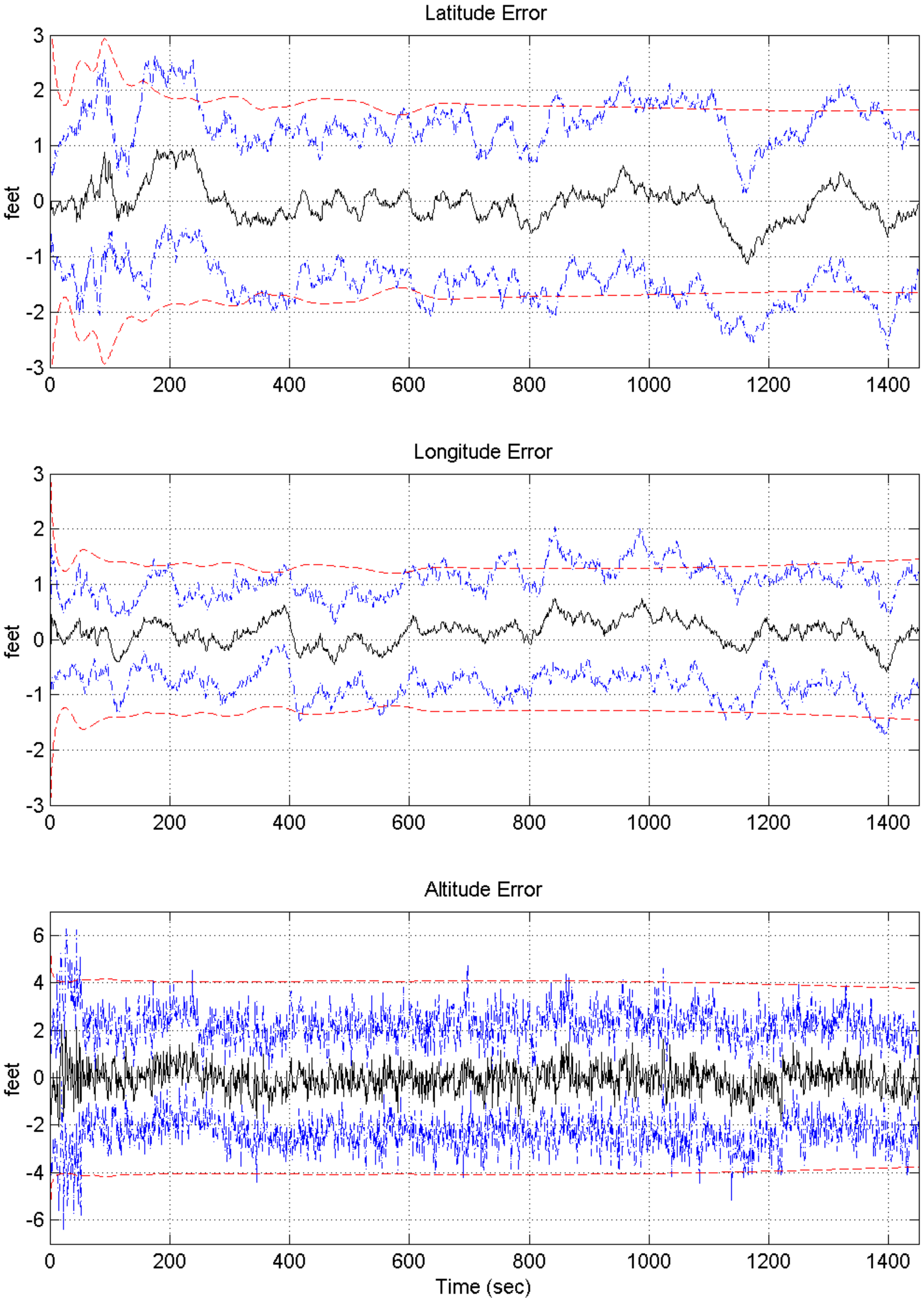


Figure L-3. Rocket Latitude, Longitude, and Altitude Errors

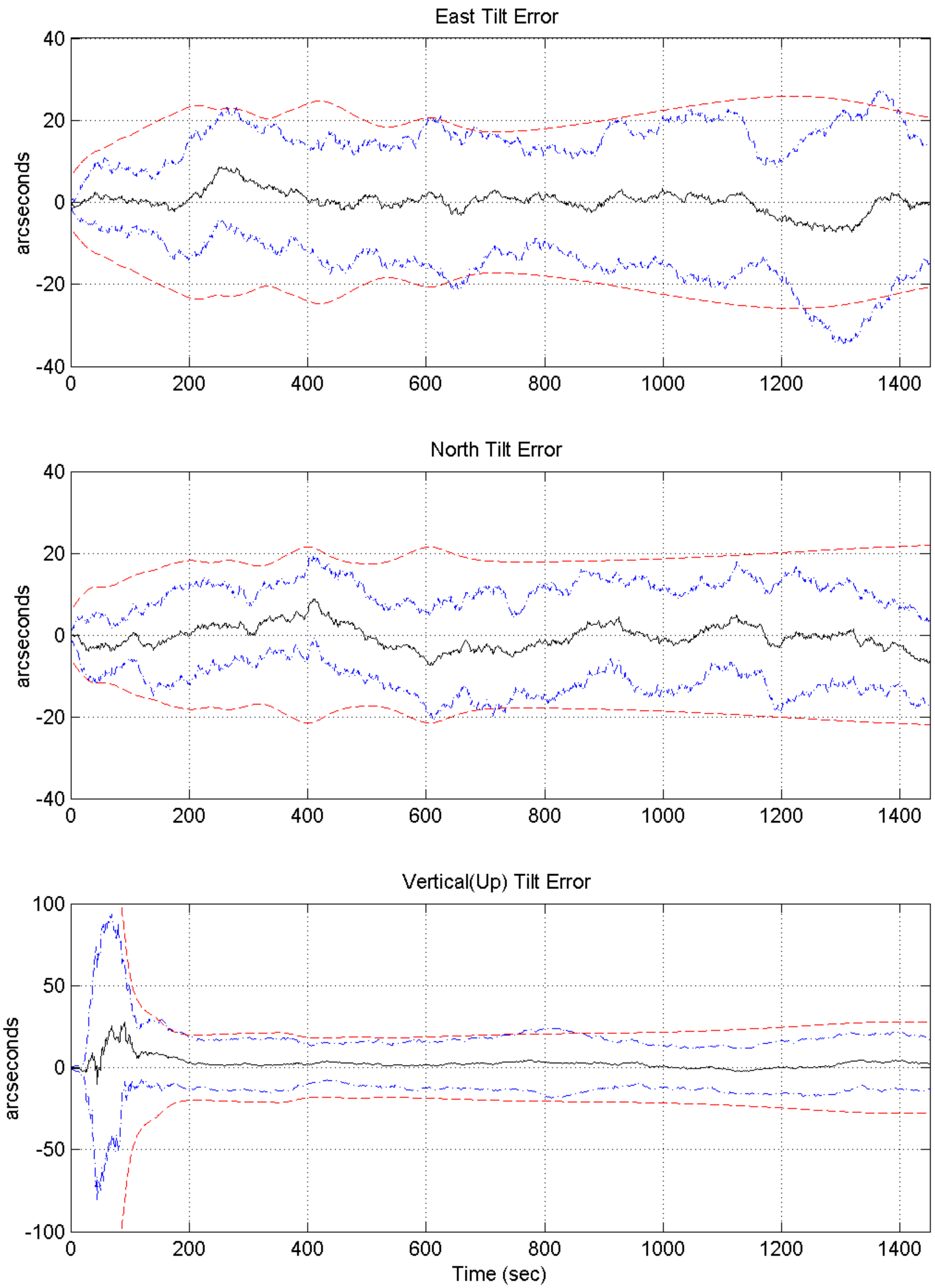


Figure L-4. East, North, and Vertical Tilt Errors

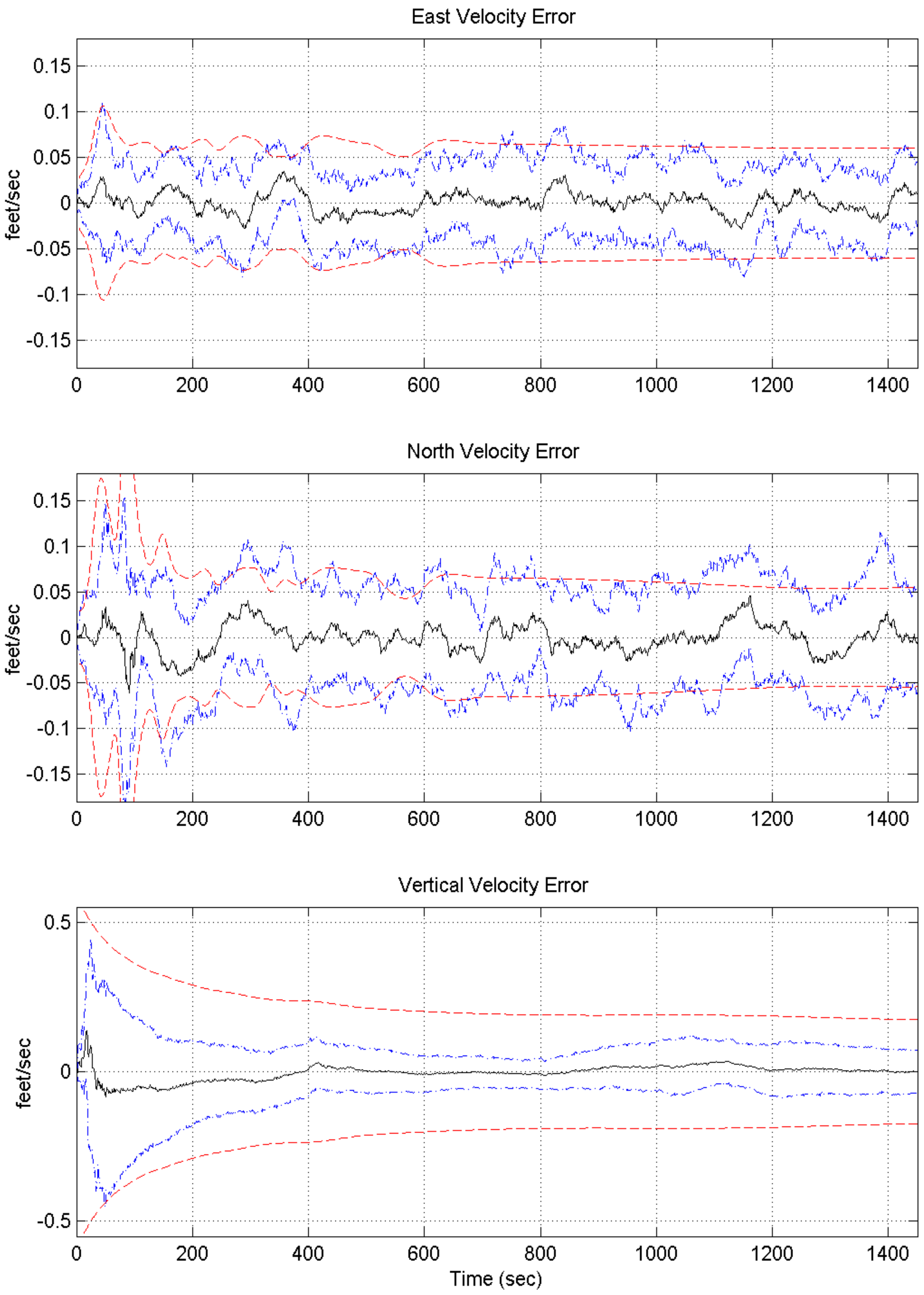


Figure L-5. East, North, and Vertical Velocity Errors

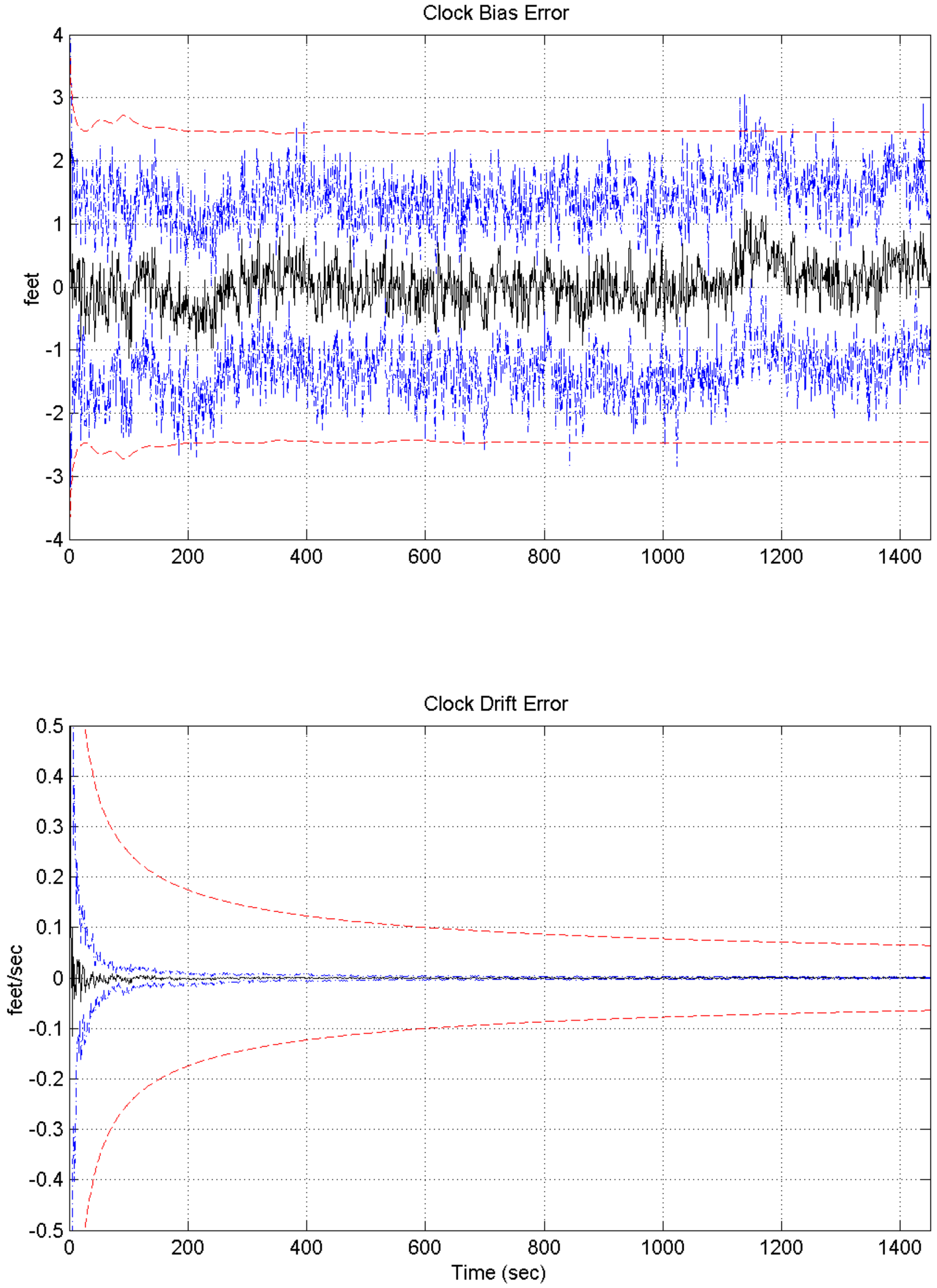


Figure L-6. DGPS User Clock Bias and Clock Drift Errors

Appendix M.

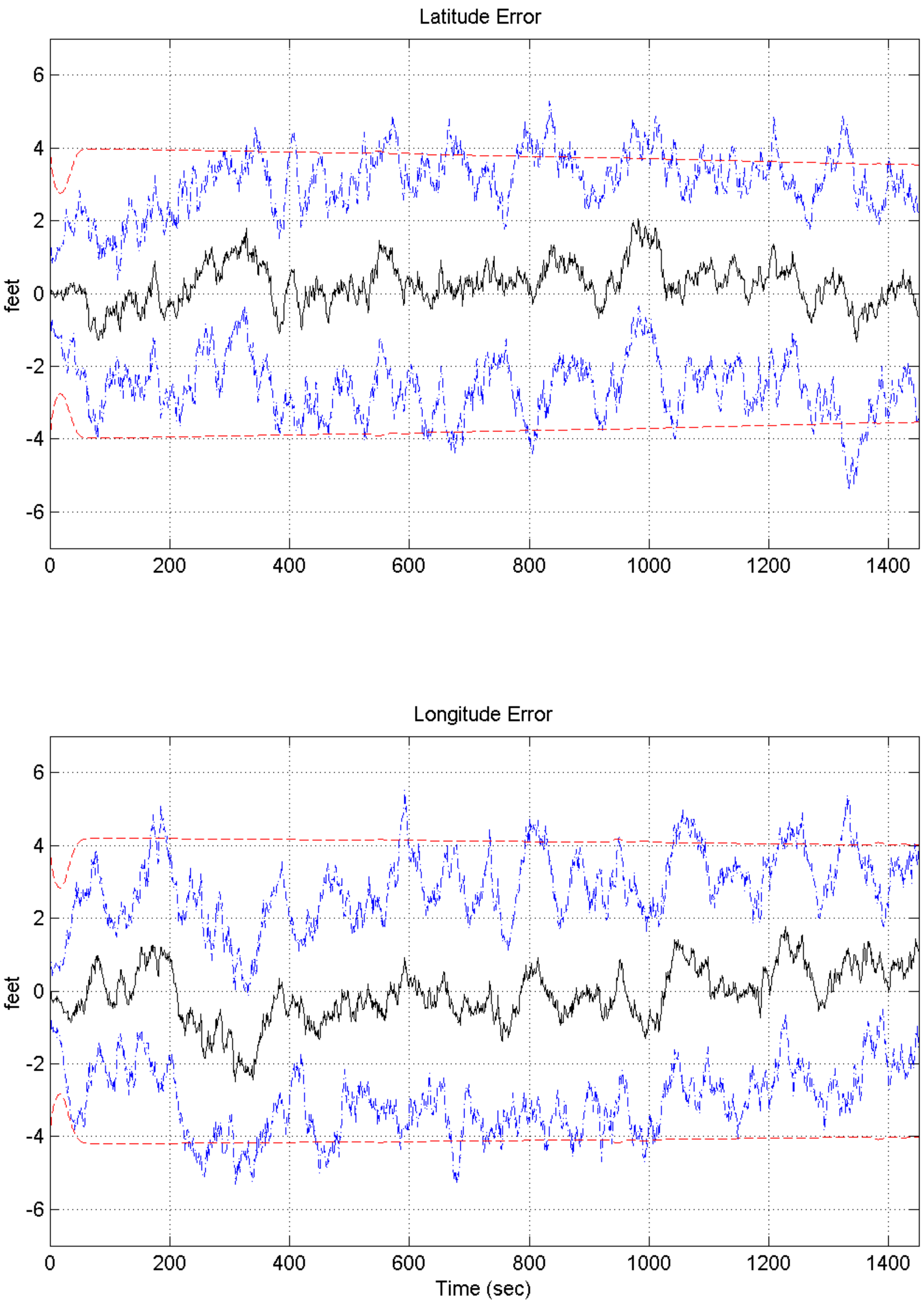
Plots of Case V

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)


Latitude Error

Longitude Error

Figure M-1. Rocket Latitude and Longitude Errors

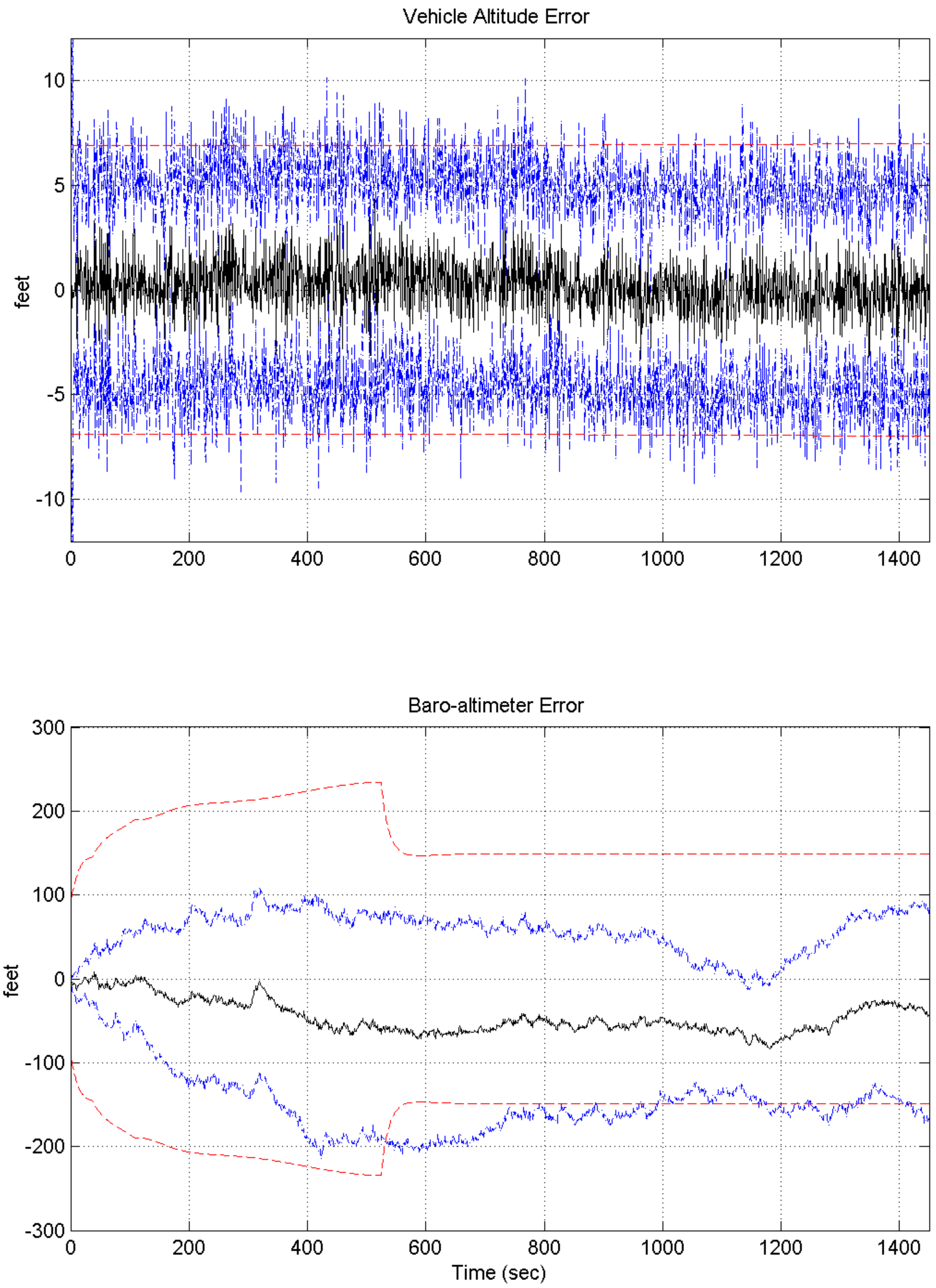


Figure M-2. Rocket Altitude and Baro-Altimeter Errors

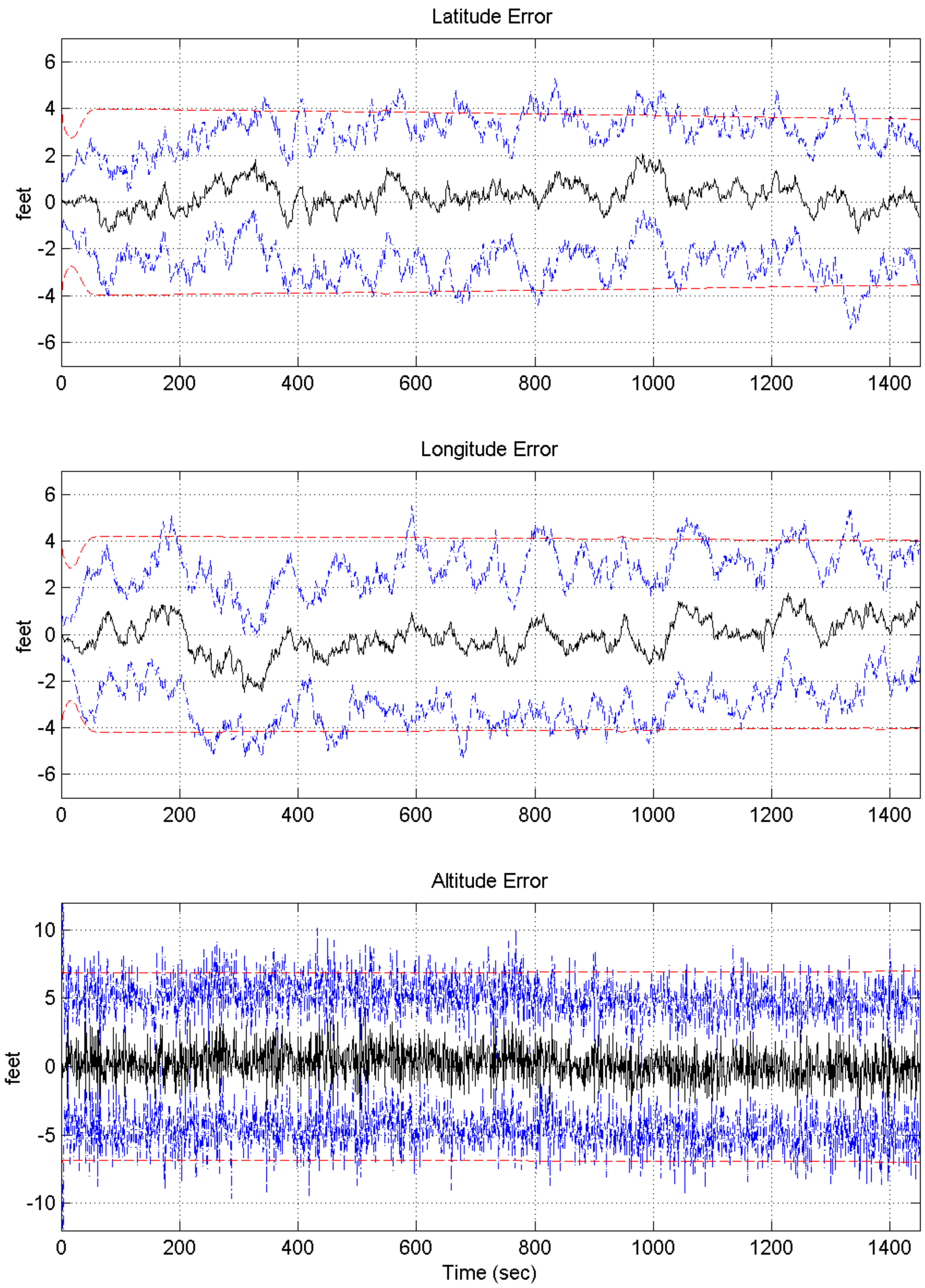


Figure M-3. Rocket Latitude, Longitude, and Altitude Errors

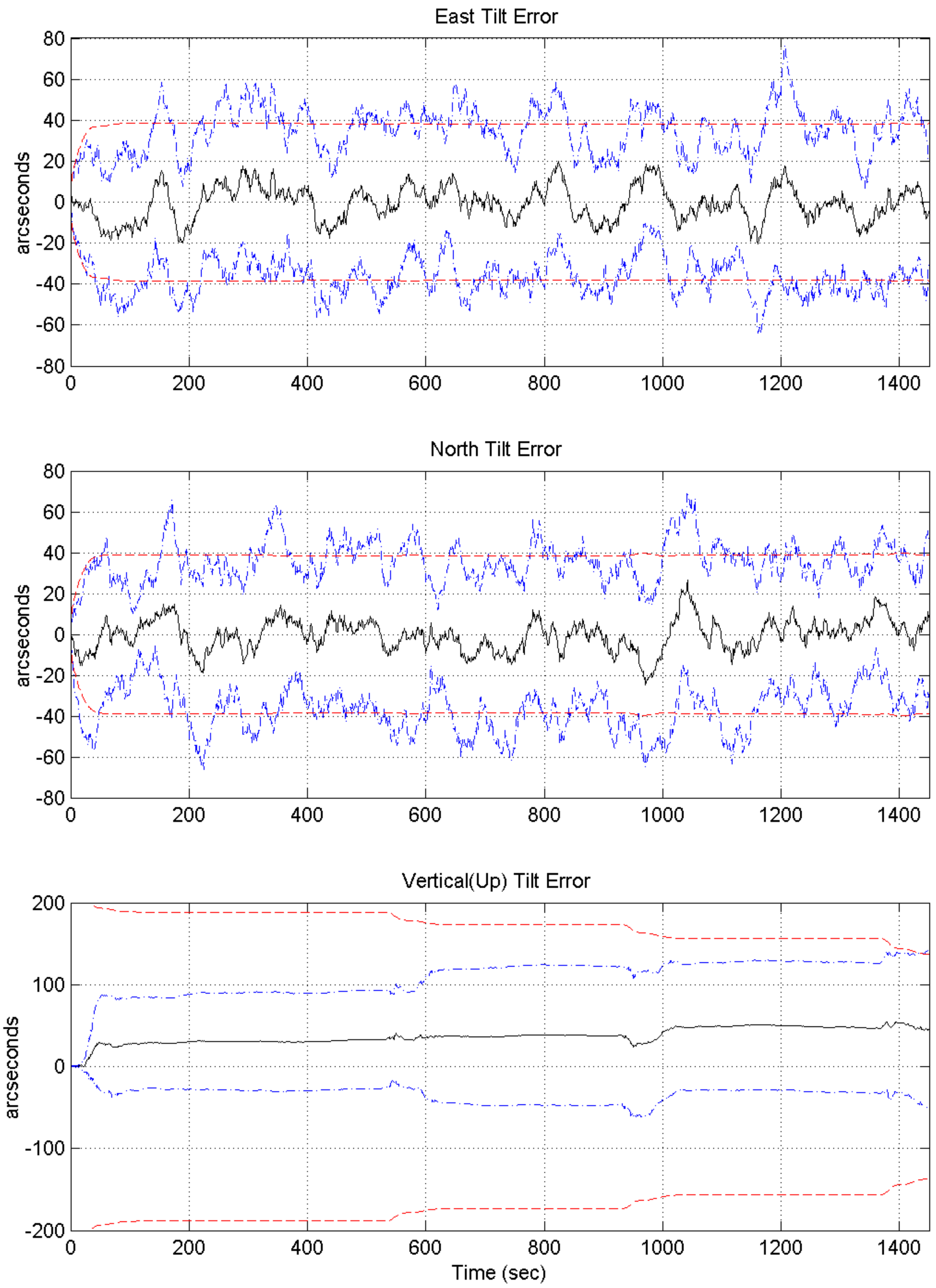
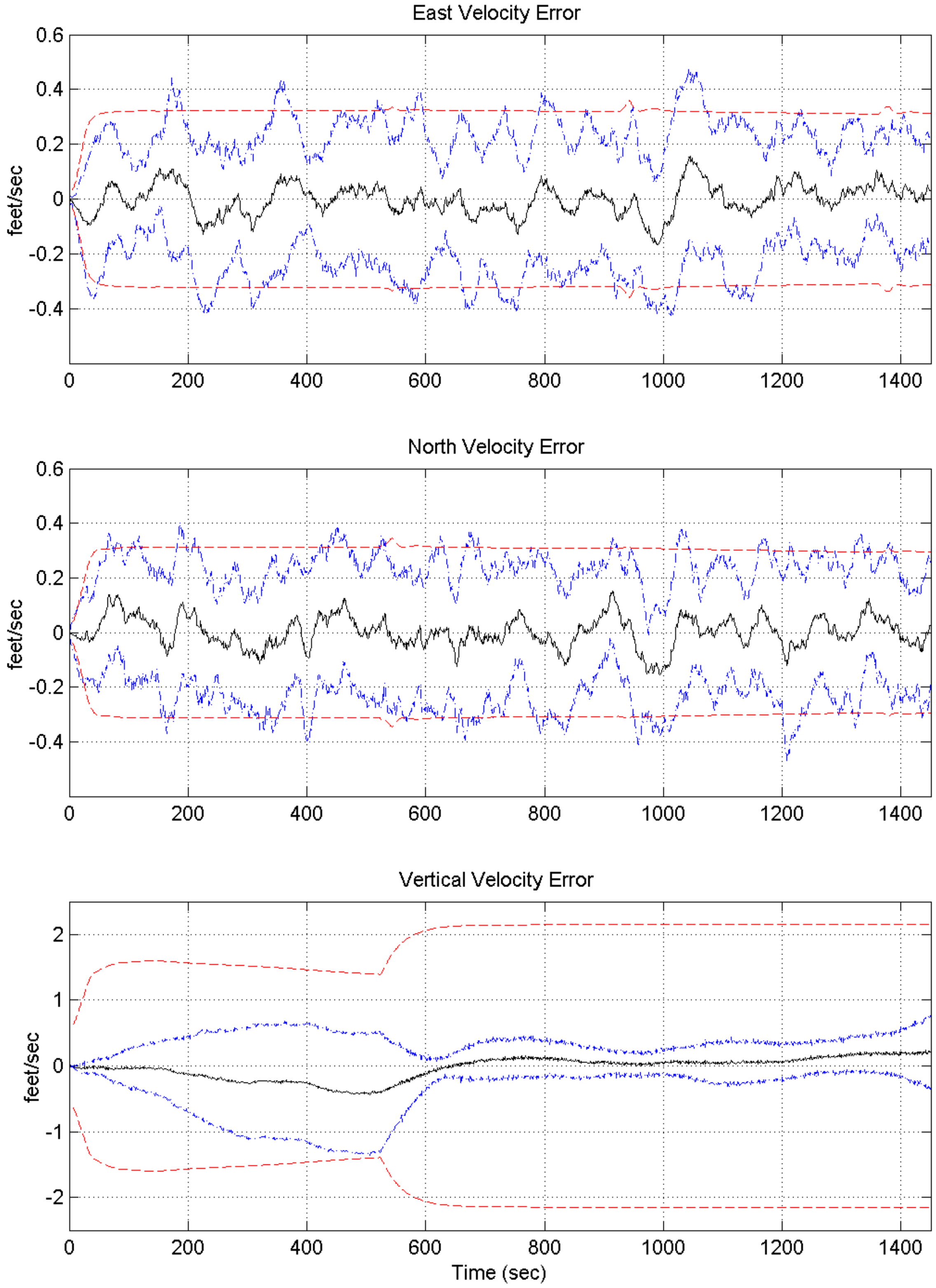


Figure M-4. East, North, and Vertical Tilt Errors



East Velocity Error

North Velocity Error

Vertical Velocity Error

Figure M-5. East, North, and Vertical Velocity Errors

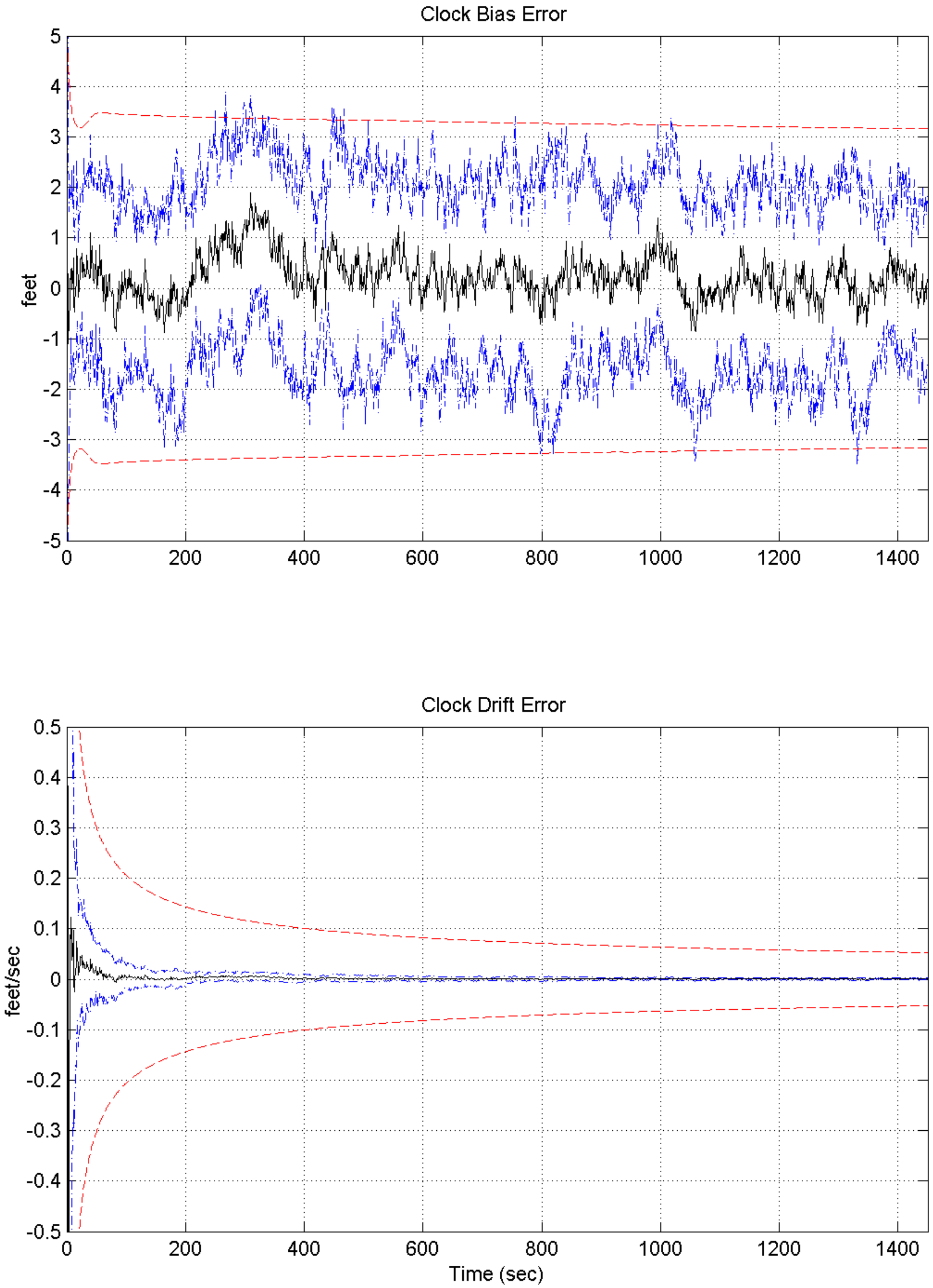


Figure M-6. GPS User Clock Bias and Clock Drift Errors

Appendix N

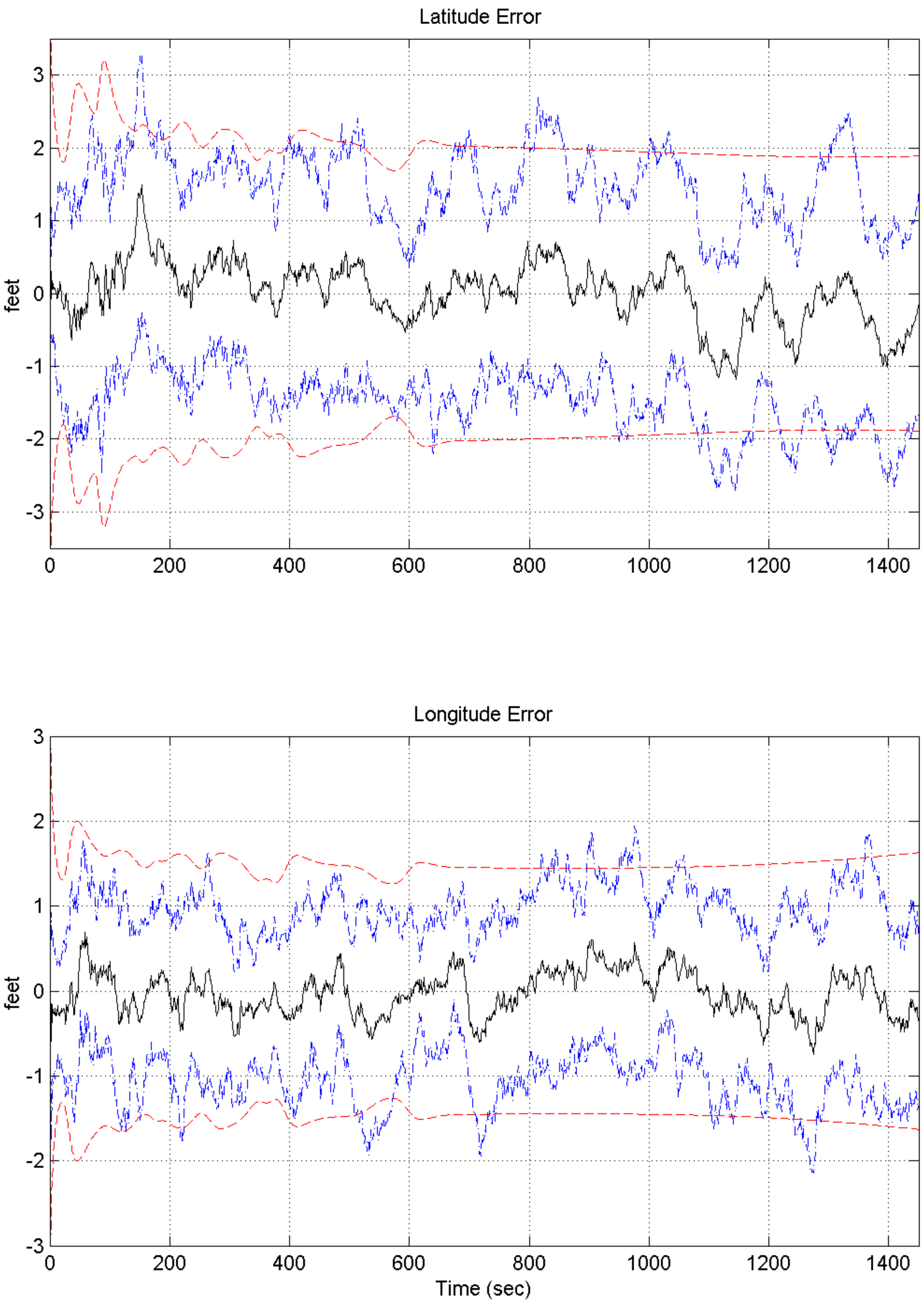
Plots of Case VI

Plot legend:

—— sample mean error

…… true error (sample mean error $\pm\sigma_{true}$)

------ filter-predicted error ($0\pm\sigma_{filter}$)



Latitude Error

Longitude Error

Figure N-1. Rocket Latitude and Longitude Errors

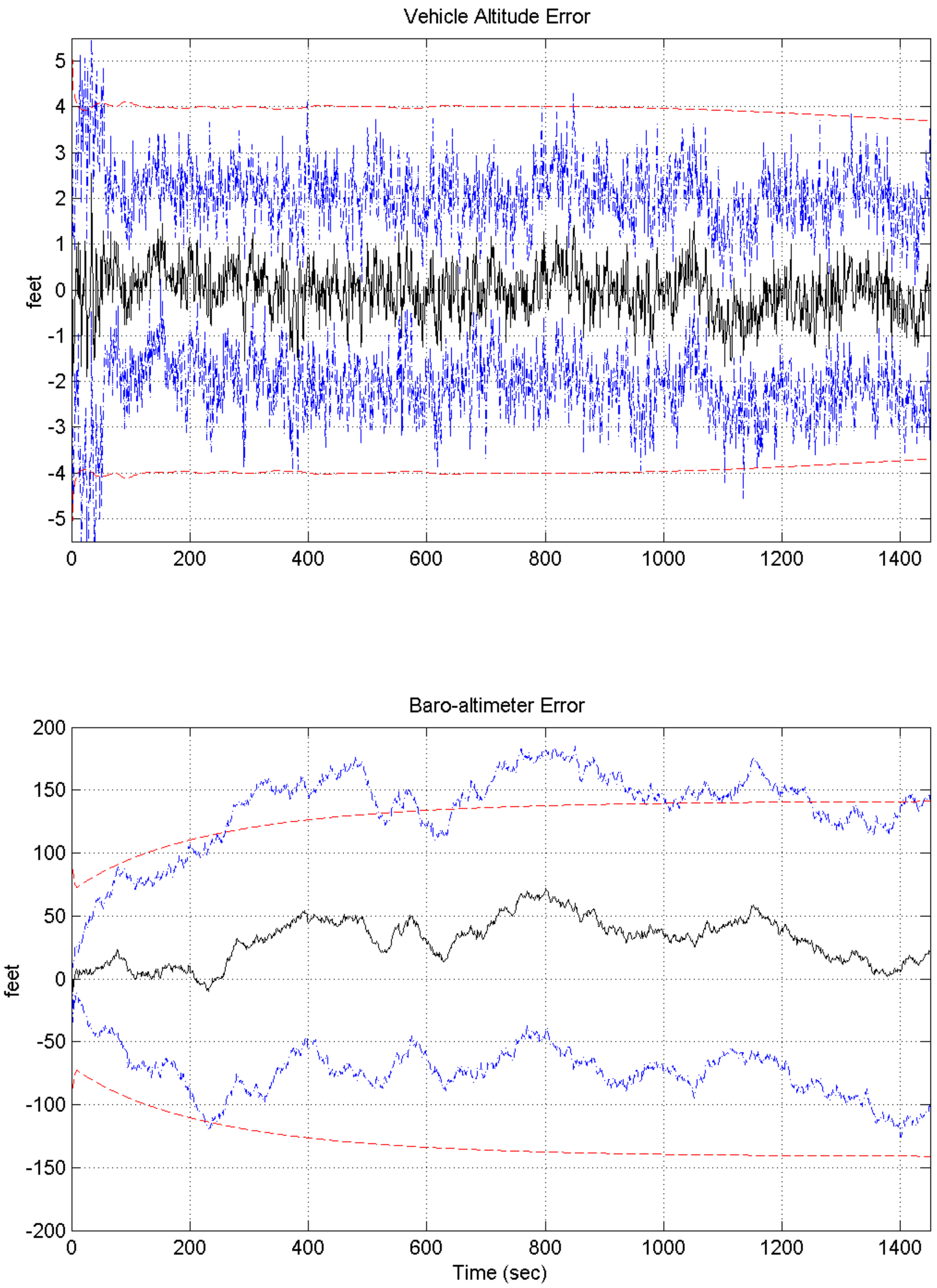


Figure N-2. Rocket Altitude and Baro-Altimeter Errors

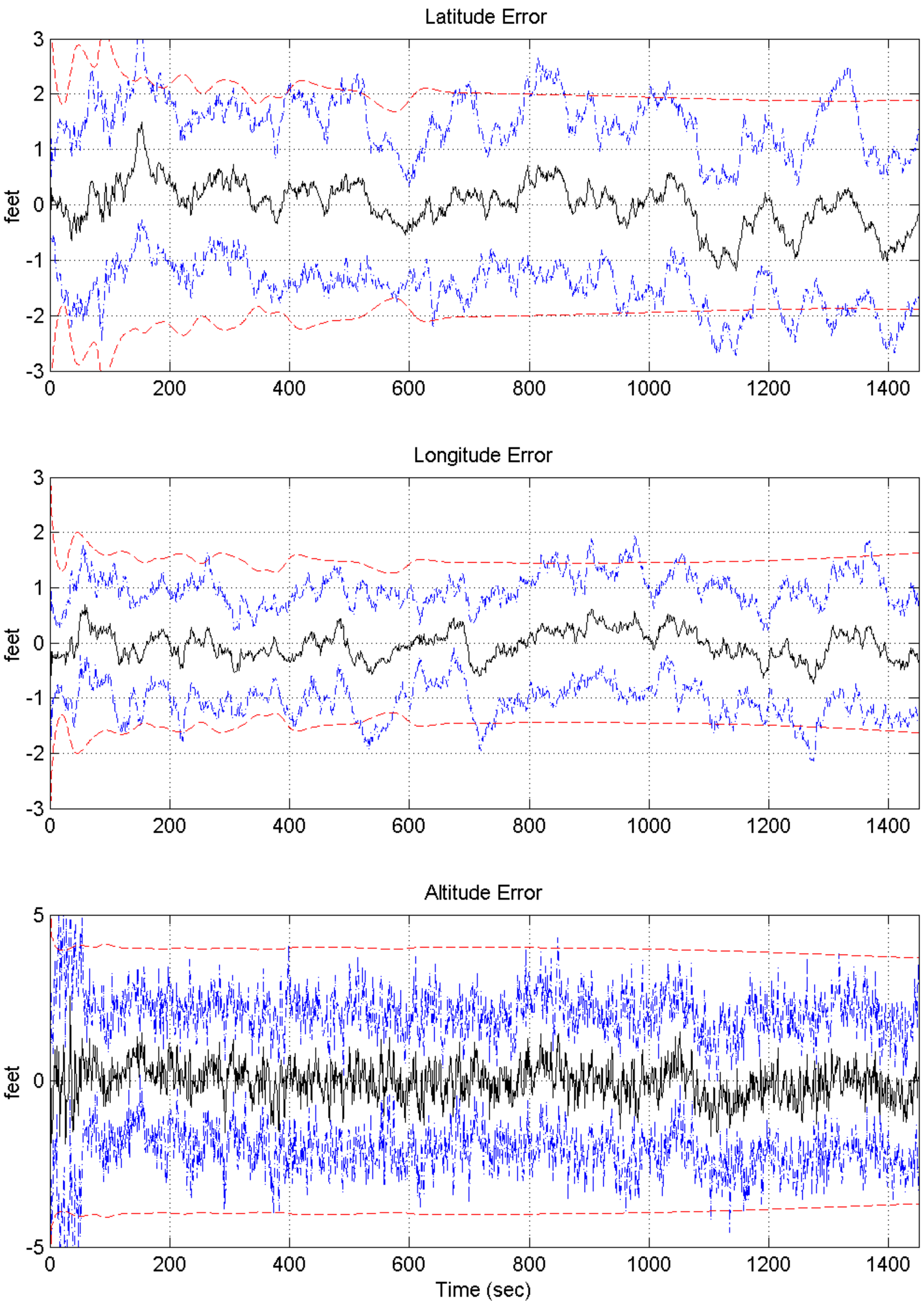


Figure N-3. Rocket Latitude, Longitude, and Altitude Errors

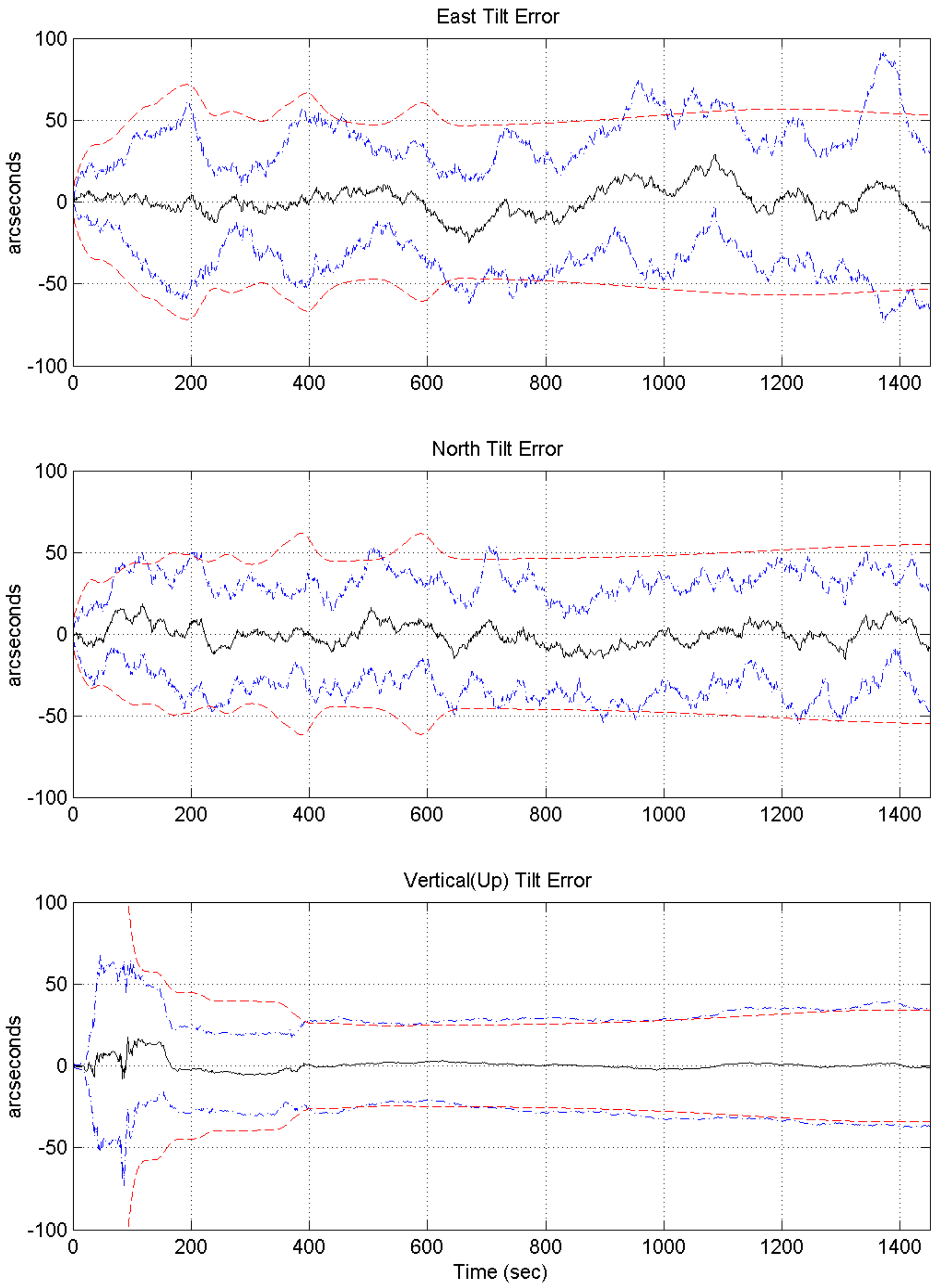


Figure N-4. East, North, and Vertical Tilt Errors

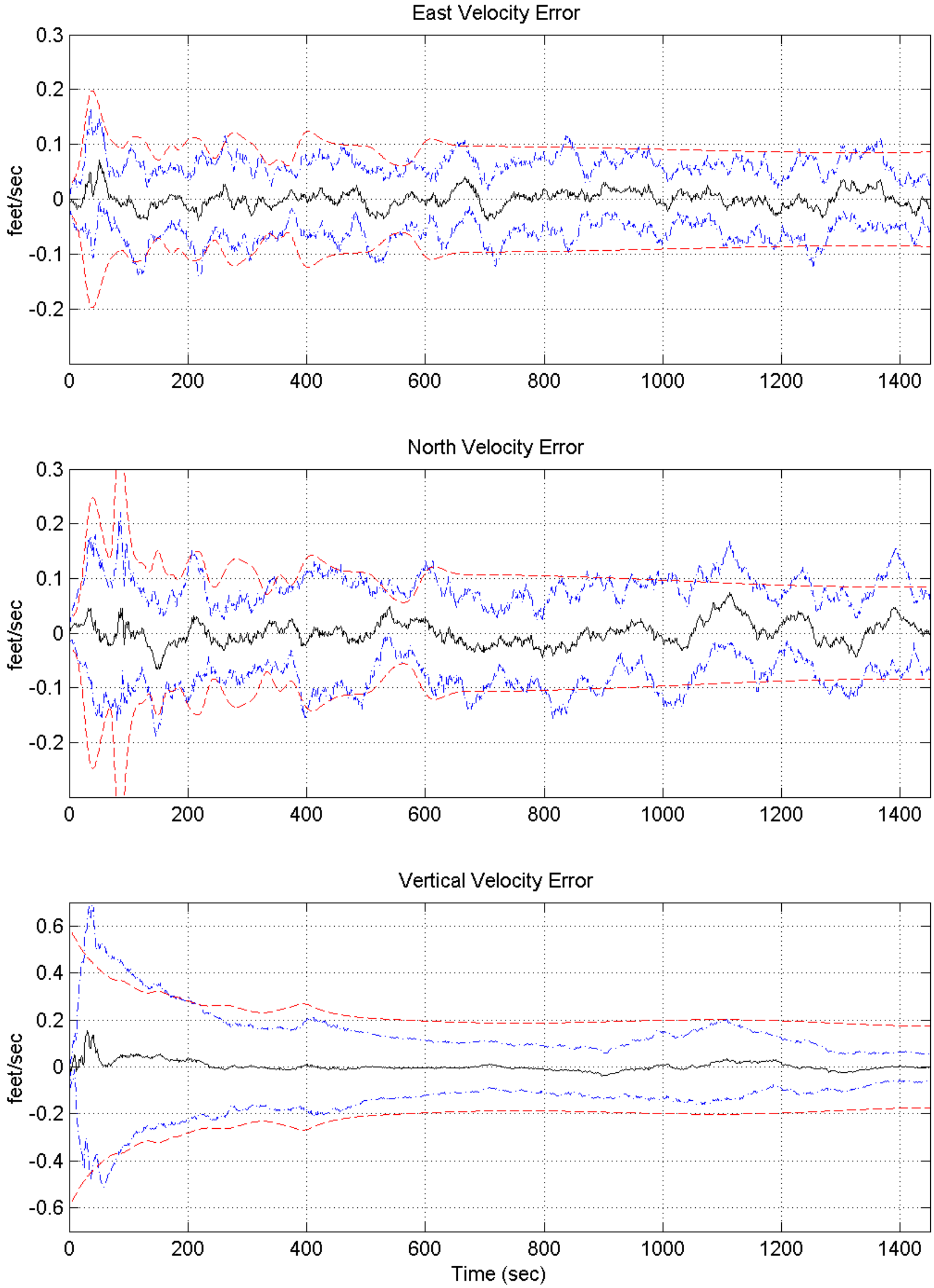
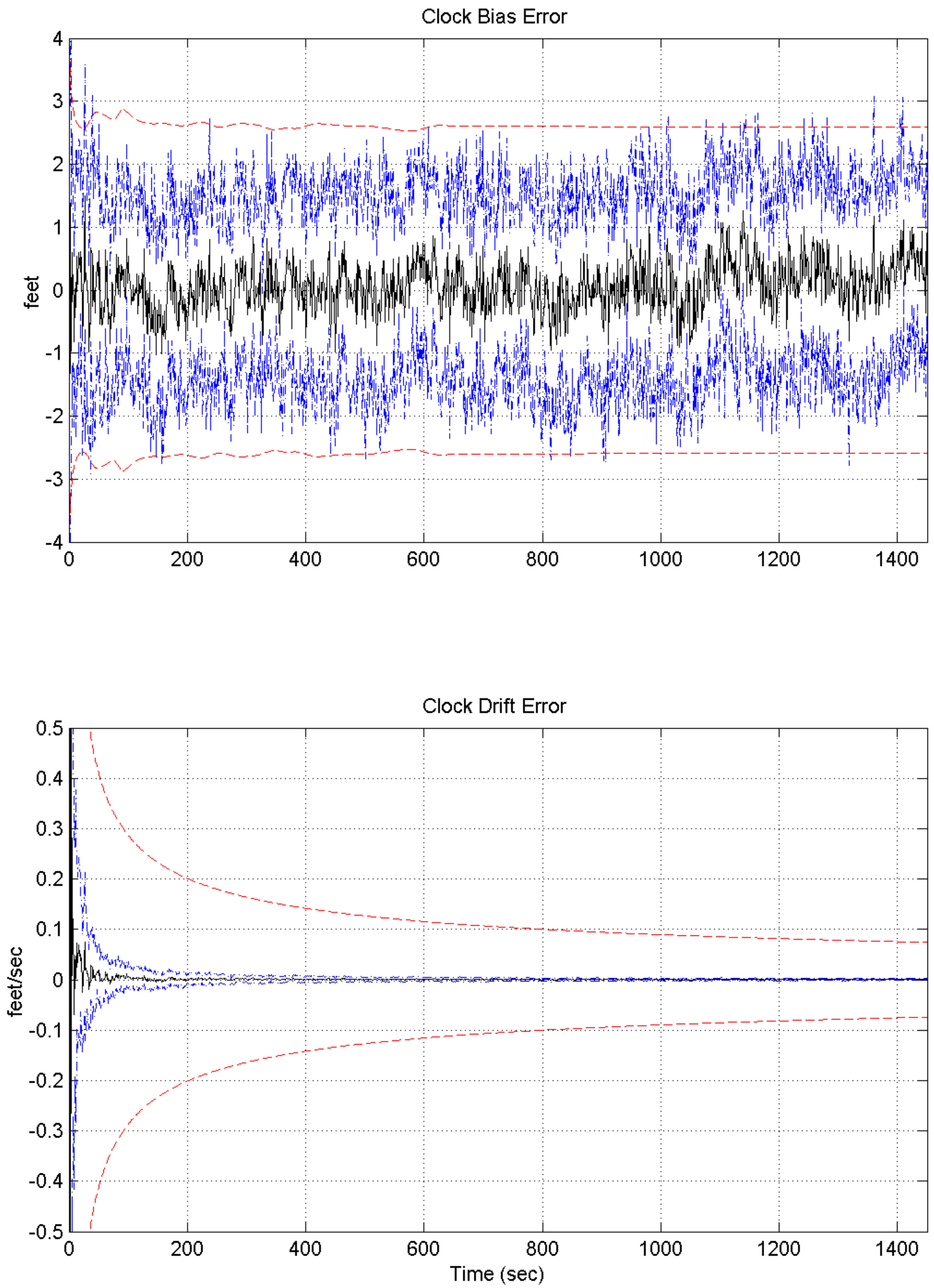


Figure N-5. East, North, and Vertical Velocity Errors



Clock Bias Error

Clock Drift Error

Figure N-6. DGPS User Clock Bias and Clock Drift Errors

Bibliography

1 Altmayer, Christian and Sven Martin. "Autonomous Onboard Orbit and Attitude Control of Geostationary Satellites Using Pseudolites," *Proceedings of the11th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Nashville, Tennessee, 1565-1578, September 15-18, 1998.

2 ARINC, Inc. "Global Positioning System, Single-Location User System Effectiveness Model Demonstration Software for Windows (WSEM), Version 3.6.4.", Reference Manual, San Diego, CA, January 2000.
http://www.arinc.com/
http;//www.arinc.com/Products_Services/gpsstat.html

3 Bagley, Daniel T. *GPS/INS Integration for Improved Aircraft Attitude Estimates*. MS Thesis, AFIT/GE/ENG/91D-04. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, December 1992 (AD-A243947).

4 Braden, Kevin, et al. "Integrated Inertial Navigation System/Global Positioning System (INS/GPS) for Automatic Space Return Vehicle," *IEEE/AIAA/NASA Digital Avionics Systems Conference, 9th,"* Virginia Beach VA, 409-414, October 15-18, 1990.

5 Britting, Kenneth R. *Inertial Navigation Systems Analysis*. New York: Wiley-Interscience, 1971.

6 Britton, Ryan, L. *A Differential GPS-Aided INS for Aircraft Landings.* MS Thesis, AFIT/GE/ENG/95D-O3, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1995 (AD-A306008).

7 Carvalho, H., et al. "Optimal Nonlinear Filtering in GPS/INS Integration*,*" *IEEE Transactions on Aerospace and Electronics Systems*, 33: 835-850 (1997).

8 Clark, Fred D. and Ann Christofferson. *Applicability of Relative GPS to Automated Rendezvous Between the Space Shuttle and Space Station.* NASA Technical Reports, Jan 01,1991, DID No. 19930013052 N (93N22241).

9 Department of the Air Force. *NAVSTAR GPS USER EQUIPMMENT.* MZI0298.001, US Air Force Space Systems Division, NAVSTAR-GPS Joint Program Office, Los Angeles, CA. February 1991.

10 Gray, Robert A. *An Integrated GPS/INS/BARO and Radar Altimeter System for Aircraft Precision Approach Landings.* MS Thesis, AFIT/GE/ENG/94D-13. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1994 (AD-A289280).

11 Griffen, Gordon, C. Evans, R. Gray, B.J. Bohenek, *The Creation and Validation of a Matlab Tool as a Possible Alternative to MSOFE-Final Report*, AFIT Course EENG-735, Air Force Institution of Technology (AFIT), Air Education and Training Command, WPAFB, Ohio, 3 June 1994.

12 Harms, Pamela L. *MatSOFE Modifications*. Sverdrup Technologies, Inc., TEAS Group. Task Order Number: YH-941531. TEAS Reference Number: 9500202-60U. Eglin AFB, FL, January 1995.

13 Hein, G. W., and M. M. Ertel *High-Precision Aircraft Navigation Using DGPS/INS Integration*, Institute of Astronomical and Physical Geodesy (IAPG), University FAF Munich, Neubiberg, 1993.

14 Institute of Electrical and Electronics Engineers. *IEEE Standard Dictionary of Electrical and Electronics Terms.* IEEE Std. 100-1972. New York, NY, 1972

15 Isakowitz, J. Steven, et al. *INTERNATIONAL REFRENCE GUIDE TO SPACE LAUNCH SYSTEMS* (3rd Edition). Reston, Virginia: AIAA Technical Publications, 1999.

16 Kalman R. E. "A New Approach to Linear Filtering and Prediction Problems," *Transactions ASME, Series D: Journal of Basic Engineering,* Vol. 82, 35-45 (March 1960).

17 Kim, Jinwon, et al. "A Complete GPS/INS Integration Technique Using GPS Carrier Phase Measurement," *IEEE Position Location and Navigation Symposium.* 526-533 (April 1998), IEEE Cat. No. 98CH36153.

18 Knudsen, L. *Performance Accuracy (Truth Model/Error Budget) Analysis for the LN-93 Inertial Navigation Unit.* Technical Report, Litton Guidance and Control Systems, Woodland Hills, CA: January 1985. DID No. DI-S-21433 B/T: CDRL No. 1002.

19 Lockheed Martin Corporation web site. "Launch vehicles – Atlas".
January 2002.
http://www.lockheedmartin.com/factsheets/product383.html.
http://www.ast.lmco.com/launch_atlas.shtml

20 Maybeck, Peter S. *Stochastic Models, Estimation and Control.* Volume 1. New York: Academic Press. Inc., 1979. Republished, Navtech, Alexandria, Virginia, 1994.

21 Maybeck, Peter S. *Stochastic Models, Estimation and Control.* Volume 2. New York: Academic Press. Inc., 1982. Republished, Navtech, Alexandria, Virginia, 1994.

22 Maybeck, Peter S. *Stochastic Models, Estimation and Control.* Volume 3. New York: Academic Press. Inc., 1982. Republished, Navtech, Alexandria, Virginia, 2001.

23 Miller, Mikel M., Course handouts, EENG 735, Inertial Navigation Systems Analysis and Integration. School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, Spring Quarter 2001.

24 Milliken, R.J., and C.J. Zoller. "Principle of Operation of NAVSTAR and System Characteristics," *Global Positioning System*, The Institute of Navigation, Vol. I Alexandria, VA, 1980.

25 Misra, Pratap and Per Enge., *GLOBAL POSITIONING SYSTEMS Signals, Measurements, and Performance.* Lincoln, Massachusetts: Ganga-Jamuna Press, 2001.

26 Moreau, Michael C., and Penina Axelrad. “GPS Receiver Architecture and Expected Performance for Autonomous Navigation in High Earth Orbits”, *Journal of The Institute of Navigation*, Vol.47, No.3, 191-204, Fall 2000.

27 Mosle, William B. *Detection, Isolation, and Recovery of Failures in an Integrated Navigation System*. MS Thesis, AFIT/GE/ENG/93D-28. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, December 1993 (AD-A274056).

28 Musick, Stanton H., “*PROFGEN – A Computer Program for Generating Flight Profiles”,* Technical Report, Air Force Avionics Laboratory, WPAFB, OH, November 1976. AFAL-TR-76-247, DTIC AD-A034993.

29 Musick, Stanton H., “User’s Guide for PROFGEN (A Flight Profile Generator)”, AFRL Technical Memorandum, WPAFB, Ohio, June 2001

30 Musick, Stanton H., and Neil Carlson. *User's Manual for a Multimode Simulation for Optimal Filter Evaluation (MSOFE).* AFWAL-TR-88-1138, Wright-Patterson AFB OH: A.F. Avionics Laboratory, AFWAL/AARN-2, April 1990.

31 Musick, Stanton H. Electronics Engineer, WL/AAAS-3, Air Force Avionics Laboratory, WPAFB, OH. Personal telephone conversations. Winter 2002

32 National Geodetic Survey(NGS)’s Continuously Operating Reference Stations (CORS) web page. “Global Positioning System (GPS) carrier phase and code range measurements in support of 3-dimensional positioning activities throughout the United States and its territories”. Winter 2001. http://www.ngs.noaa.gov/CORS/

http://www.ngs.noaa.gov/CORS/download1/
http://www.ngs.noaa.gov/CORS/download2/

33 Negast, William Joseph. *Incorporation of Differential Global Positioning System Measurements Using an Extended Kalman Filter for Improved Reference System Performance*. MS Thesis, AFIT/GE/ENG/91D-41. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, December 1991 (AD-A243742).

34 Raquet, John F. Course handouts, EENG 533, *Navigation Using The GPS.* School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, Spring Quarter 2001.

35 Sabatini, Roberto (Italian Air Force). *High Precision DGS and DGPS / INS Positioning for Flight Testing.* NASA Technical Reports, 6th Saint Petersburg International Conference on Integrated Navigation Systems, 18-1 – 18-17, Oct 01, 1999, DID No. 2000001219.

36 Stacey, Richard D. *A Navigation Reference System (NRS) Using Global Positioning System (GPS) and Transponder Aiding.* MS Thesis, AFIT/GE/ENG/ 91M-04. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, March 1991 (AD-A238890).

37 The MathWorks, Inc. "MATLAB, The Language of Technical Computing", Version 6.0, Winter 2001.
http://www.mathworks.com/

38 The United States Coast Guard web site. "Navigation Center (navcen)"
http://www.navcen.uscg.mil/gps/almanacs/default.htm.

39 Tragesser, Steven G. Assistant Professor of Aerospace Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH. Personal interview. 18 Jan 2002.

40 Upadhyay, Triveni N., et al. *Autonomous Integrated GPS / INS Navigation Experiment for OMV: Phase 1 Feasibility Study,* NASA Technical Reports, NASA- CR- 4267, Jan 01, 1990, DID No. 19900011653 N (90N20969), Contract No. NAS8-38931.

41 Upadhyay, T. N. "Autonomous GPS/INS Navigation Experiment for Space Transfer Vehicle," *IEEE Transactions on Aerospace and Electronics Systems,* 29: 772-785 (1993).

42 Varner, Christopher and Elizabeth M. Cannon. "The Effects of Altitude on GPS Dilution of Precision and Coverage". *Proceedings of the 51st Annual ION Meeting*, Colorado Springs, Colorado,119-127, June 5-7, 1995.

43 Vasquez, Juan R. *Detection of Spoofing, Jamming, or Failure of a Global Positioning System (GPS).* MS Thesis, AFIT/GE/ENG/92D-37. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, December 1992 (AD-A259023).

44 White, Nathan Alan. MMAE Detection of Interference/Jamming and Spoofing in a DGPS-Aided Inertial System. MS Thesis, AFIT/GE/ENG/96D-21, School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 1996 (AD-A320882).

45 Williams, W., et al. "A Comparison of State Space, Range Space, and Carrier Phase Differential GPS/INS Relative Navigation," *Proceedings of the 2000 American Control Conference.* Volume 4, 2932-2938, Chicago: IEEE Press, 2000, IEEE Cat. No. 00CH36334.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)* 01-03-2002	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* Jun 2001 – Mar 2002

4. TITLE AND SUBTITLE PERFORMANCE TRADEOFF STUDY OF A GPS-AIDED INS FOR A ROCKET TRAJECTORY	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Muhittin Istanbulluoglu, 1st Lt, TuAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GE/ENG/02M-11
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/SNAR Attn: Dr James Leonard Avionics circle Bldg 620, Rm N3-X19 DSN: 785-5668 ext. 4137 WPAFB OH 45433-7765 e-mail: James.Leonard@wpafb.af.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Turkish Air Force (TuAF) has started a project to launch a satellite using only Turkish resources. Primary motivation behind this research is to assist TuAF's project by keeping up with these innovations in the navigational arena. The basic challenge in navigation system design is to decide which navigation system (or systems) and implementation techniques will be used, depending on accuracy requirements. The two primary navigation systems that will be integrated in this research are the Inertial Navigation System (INS), and the Global Positioning System (GPS). The Kalman filter algorithm is used to integrate INS and GPS. The rocket (Atlas IIAS launch vehicle) flight profile is generated by using PROFGEN and simulated "true" GPS ephemeris data is incorporated into system as GPS measurements. The "modified" alternative system performance analysis tool, MatSOFE, is utilized in this research study. Standard and differential GPS are compared, as are three different grades of INS, in a tradeoff performance analysis.

15. SUBJECT TERMS

Meeting Management, Delphi Technique, Malcolm Baldridge, Business Meetings, Management, Leadership Training, Organizational Meetings, Quality, Meeting Guide, Meeting Training, Management Training, Organizational Theory, Group Dynamics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Peter S. Maybeck, Civ., AFIT/ENG
a. REPORT	b. ABSTRACT	c. THIS PAGE			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	232	(937) 785-3636, ext 4639; e-mail: Peter.Maybeck@afit.edu