Mars Entry Navigation Performance Analysis using Monte Carlo Techniques

by

Stephen C. Paschall II
B.S., Mechanical Engineering
Texas A&M University (2002)
Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2004

Author . . . .
Department of Aeronautics and Astronautics
May 14, 2004

Certified by . . . .
David K. Geller
Senior Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.
Technical Supervisor

Certified by . . . .
Nicholas Roy
Assistant Professor of Aeronautics and Astronautics
Thesis Advisor

Accepted by . . . .
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Mars Entry Navigation Performance Analysis using Monte Carlo Techniques

by

Stephen C. Paschall II

Submitted to the Department of Aeronautics and Astronautics
on May 14, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

An atmospheric entry and descent full-state navigation filter is developed and presented. Using this filter a navigation performance analysis is performed to examine the effects of various instrument packages and differing sensor scenarios for the entry and descent phase of the Mars Aerial Regional-scale Environmental Survey (ARES) mission.

Deterministic simulation in conjunction with Monte Carlo techniques is used to verify navigation performance of an extended Kalman filter. This analysis specifically compares the performance of four feasible instrument packages, examines navigation performance as it varies with changes to initial sensor activation altitude, and examines error sources and covariance trends for this entry and descent scenario.

The results from the analysis show that large attitude uncertainty resulting from the LN200 IMU bias causes a breakdown of the filter algorithm due to nonlinearities. The addition of a surface relative velocity measurement to the altimeter measurement provides only marginal position uncertainty improvement and significant velocity and attitude uncertainty improvement. Increasing the initial altitude for sensor activation provides slight improvements in position uncertainty, but large velocity and attitude uncertainty improvements. Finally, it is shown that initial state uncertainty dominates over all other error sources in this navigation analysis. Error growth within the principal states (position, velocity, and attitude) is predominantly a product of the near-constant attitude uncertainty as it transfers from the innocuous roll attitude channel into the more consequential pitch and yaw attitude channels.

Technical Supervisor: David K. Geller
Title: Senior Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.

Thesis Advisor: Nicholas Roy
Title: Assistant Professor of Aeronautics and Astronautics

Acknowledgments

I would like to take this opportunity to thank all those who have helped to make this thesis possible and my graduate studies at MIT and Draper Laboratory such a rewarding experience. ... To Draper Lab for supporting me through grad school and for being a pleasant and supportive working environment. ... To my technical supervisor, Dave Geller, for being such an incredible mentor and friend through my time here at Draper. Dave's unending patience and insightful intellect have made working with him a wonderful experience. ... To my thesis advisor, Nick Roy, for his help with this thesis and very supportive nature. ... To Drew for being a great officemate. Our incredibly diverse and pondering conversations over the years in our office have been both fascinating and an excellent sounding board for ideas. ... To my roommates Drew, Geoff, and Megan for providing such a great, happy home and putting up with my loss of housekeeping skills as thesis crunch time arrived. ... To all the Draper Fellows who made working on the fifth floor a fun place to be. ... To Melissa for helping to proof my thesis and providing an excellent, thorough, and insightful outside perspective on my work. ... And to my family for being the incredibly loving and supportive family they've always been.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under internal company sponsored research project 13177, GC DLF Support, Contract #IRD04-0-5043 and project 13529, Planetary Exploration, Contract #PT04-PROGRAMS-DFY04

Author's Signature Date

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Navigation is generally thought of as the determination of position and heading for use in moving from one place to another. Navigation as it applies to aerospace vehicles is the use of various sensor data and initial conditions to fundamentally estimate position, velocity, and attitude. Estimates of these primary states and information on their uncertainty are then used to guide the vehicle along a desired trajectory.

Navigation is essential if vehicle guidance is required. Without navigation, an unmanned vehicle could at best make open-loop guidance and control maneuvers assuming that the vehicle closely follows some nominal precomputed trajectory. This method does not allow for trajectory error that might arise due to uncertain initial conditions, vehicle parameters, and environmental conditions. Quantities such as vehicle mass, center of gravity, and aerodynamic coefficients constitute uncertain vehicle parameters for the case of atmospheric entry. Environmental uncertainties are unavoidable, but can be more severe at planets other than Earth where past observational data is limited and models are imprecise. Environmental effects can cause dispersions to a vehicle's trajectory from inaccurate modeling of atmospheric density, atmospheric temperature, and atmospheric wind.

The navigation system is a critical component in the design of a spacecraft. Navigation uncertainties limit the accuracy to which the vehicle can be guided and controlled. Guidance and control authority for correcting trajectory dispersions is only useful if the navigation system is precise enough to detect these dispersions in the trajectory.

The baseline configuration of an aerospace navigation system is the inertial navigation system. Inertial navigation without any external measurement sensors is termed “dead-reckoning” navigation. This involves incorporating observations from the inertial measurement unit (IMU) into propagation of the fundamental states forward in time. Different types of sensors can be added to this basic configuration to enhance the navigation system. These additional sensors are used to provide information updates to the vehicle states. Measurement sensors have differing regimes of usefulness during any mission. In interplanetary cruise and planetary orbit, sensors such as star cameras and sun sensors are useful. Altimeters, velocimeters, and radar/doppler systems are useful during atmospheric entry and flight.

1.1 Mission Description

The problem under examination for this analysis is the entry and descent phase of a Mars entry vehicle. The analysis is specifically applied to the proposed Mars Aerial Regional-scale Environmental Survey (ARES) Mission. The purpose of this mission is to deliver a small airplane to Mars to perform scientific measurements that fill a critical gap between the current ground-based and orbital-based Mars scientific observations. This analysis will therefore examine the atmospheric entry and descent phases of the vehicle containing this ARES airplane.

One of the significant engineering challenges for delivering payloads from Earth to Mars involves the entry and descent phases of the mission. Entry is defined as the period of the mission from atmospheric interface to parachute deployment, and descent is defined as the period of parachute descent until the airplane is deployed. Understanding the navigation issues during this phase is critically important. The entry vehicle is experiencing a tremendously dynamic environment for a short period of time. During this phase, any vehicle and environmental uncertainties can cause significant trajectory dispersions, jeopardizing the vehicle survival. In addition, vehicle delivery accuracy can be important for reaching the mission objectives. Science objectives necessitate precise delivery of a vehicle for maximum scientific return. Since this near-final mission phase is so short and subject to such large dynamic forces and potentially large trajectory dispersions, good navigation information is very important if the guidance system is expected to correct for these errors in the limited remaining flight time.

This analysis will therefore examine a variety of feasible instrumentation configurations for the ARES mission to determine their absolute and relative performance using a deterministic simulation in conjunction with Monte Carlo techniques. Additionally, the analysis will examine the sensitivity of the navigation performance to variations in the sensor activation time. This will help to answer questions concerning the level of technology required to reach differing levels of navigation precision.

1.2 Approach

This paper can be separated into two parts. The first part describes the algorithmic structure of the simulation used to perform the analysis. The second part describes the various navigation performance analyses and presents their results.

The simulation used within this analysis can be visualized as three parts: true vehicle dynamics and environment, sensors to measure information from the physical environment, and a navigation system to use the sensor information to estimate the true dynamics. Figure 1-1 shows the conceptual flow of the simulation.

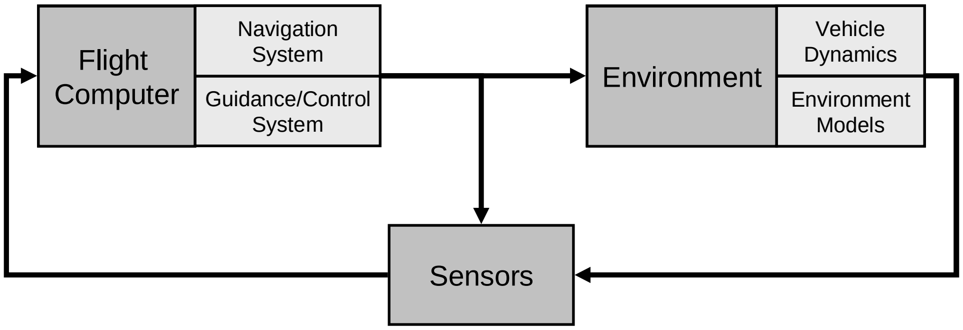


Figure 1-1: Simulation Conceptual Flow

The *Environment* block contains the quantitative description of the physical dynamics of the vehicle and the environment models of the physical world around the vehicle (planetary atmosphere, planetary gravity, and aerodynamic forces acting upon the vehicle). The vehicle dynamics are detected using onboard sensors. This is simulated within the *Sensors* block using models of the various instruments. The sensor measurements are fed into the *Flight Computer* block. The flight computer contains the navigation system to estimate the vehicle states and incorporate the current measurements. The guidance and control system uses this estimate of the states to issue sensor commands and actuator commands to affect the vehicle dynamics within the environment. The guidance and control system lies outside the scope of this analysis and will therefore not be addressed.

Chapter 2 describes the vehicle dynamics portion of the *Environment* block. These are essentially the rotational and translational vehicle dynamic equations. Chapter 3 discusses the environment models (atmospheric, aerodynamic, and gravity models) within the *Environment* block and the sensor models within the *Sensors* block. Finally, Chapter 4 provides the algorithmic background and configuration for the navigation system within the *Flight Computer* block.

The second portion of the analysis (Chapter 5) presents several key Mars entry navigation performance analyses and discusses their results. These analyses involve simulating the navigation system for the Mars ARES mission under a variety of realistic instrument configurations. The analysis is extended by investigating the sensitivity of the navigation system performance to variations in the sensor activation time. Conclusions are presented in Chapter 6.

1.3 Thesis Statement

The thesis for this navigation analysis is threefold. First, reduced sensors errors associated with higher precision inertial measurement units (IMUs) will produce significant performance improvements to the overall entry and descent navigation system. Second, providing measurements to the navigation system earlier in the trajectory will provide not only additional time with precise navigation, but improved overall state uncertainty. Last, the addition of a velocimeter to an altimeter sensor package will provide measurements of an additional filter state (velocity) and consequently create significant improvements to the navigation state uncertainty.

1.4 Notation

This analysis contains many analytical formulations to present concepts and algorithms. As a result, keeping track of what each equation is describing requires clear, thorough, and unencumbered notation. With this in mind, a notation convention is presented in Table 1.1.

Quantity	**Example**	**Description**
scalars	A, v, x	upper or lowercase variable
vector	$\boldsymbol{a}$, $\boldsymbol{v}$, $\boldsymbol{x}$	bold, lowercase variable
matrix	$\boldsymbol{A}$, $\boldsymbol{V}$, $\boldsymbol{X}$	bold, uppercase variable
true value	$\boldsymbol{A}$, v, $\boldsymbol{x}$	any regular variable
estimated value	$\hat{\boldsymbol{A}}$, $\hat{v}$, $\hat{\boldsymbol{x}}$	any variable with a hat
error value	$\tilde{\boldsymbol{A}}$, $\tilde{v}$, $\tilde{\boldsymbol{x}}$	any variable with a tilde
time derivative	$\dot{\boldsymbol{A}}$, $\dot{v}$, $\dot{\boldsymbol{x}}$	any variable with a dot

Table 1.1: Notation Conventions

In addition, subscripts on variables are used for any general, relevant notation. Superscripts on vectors are an indication of the coordinate frame from which they are measured. Two special variables are used frequently throughout the analysis. The first is the *direction cosine matrix* used for coordinate frame rotations from frame A to B. It is symbolized as $\boldsymbol{T}_{A\to B}$. The second is an equivalent method for describing a coordinate frame rotation from A to B. This is the four component quaternion vector. It is symbolized as $\boldsymbol{q}_{A\to B}$.

An often convenient representation of a vector cross-product is the cross product matrix formulation.

$$\boldsymbol{u} \times \boldsymbol{v} \;=\; \boldsymbol{U}_{\otimes}\, \boldsymbol{v} \;=\; \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \boldsymbol{v}$$

For this representation, a three component vector is converted into a three-by-three matrix. This is noted by a "$\otimes$" subscript.

The quaternion product operation is noted by the symbol "$\otimes$". For example, the rotation of a vector $\boldsymbol{v}$ from frame B to frame I is

$$\boldsymbol{v}^I = \boldsymbol{q}_{I \to B} \otimes \; \boldsymbol{v}^B \otimes \boldsymbol{q}_{B \to I}$$

1.5 Mission Parameters and Constants

In order to perform this analysis in a realistic manner, various planetary and mission specific parameters are required. The sections below provide the details of the parameters and constants used in this analysis.

Mars Constants

Accurate and up-to-date planetary constants are required for realistic atmospheric entry navigation analysis. The quantities are provided by the most current NASA specification on Mars planetary constants and models [17]. The relevant quantities are Mars gravity constants, geometry constants, and the rotation rate (Table 1.2).

Description	**Value**	**Units**
Zonal harmonic coefficient, J_2	$1.95639057 \times 10^{-3}$	–
Gravity constant, GM_{mars}	42828.376212	$[km^3/s^2]$
Reference radius, r_{mars}	3396.2	$[km]$
Equatorial radius, r_{eq}	3396.19	$[km]$
Polar radius, r_{pol}	3376.20	$[km]$
Rotation rate, $\dot{\omega}$	350.89198226	$[deg/day]$

Table 1.2: Mars Constants, [17]

Atmospheric Entry Interface Conditions

To properly simulate atmospheric entry for the ARES vehicle, entry interface (EI) initial conditions for this near-final mission phase are required. The necessary quantities are initial time, position, velocity, attitude, and angular velocity. Table 1.3 lists these values.

Description	**Value**	**Units**
Date	1-Sept-2008	–
Time	14:30:00	–
Radius	3522.2	$[km]$
Latitude	-44	$[deg]$
Longitude (east)	181.6	$[deg]$
Velocity	5.6	$[km/s]$
Flight path angle	-13	$[deg]$
Azimuth	34.2	$[deg]$
Roll Rate	1.6	$[RPM]$

Table 1.3: Entry Interface Initial Conditions, [13]

Vehicle Specifications

Specifications of the presumed ARES vehicle are used as a baseline for this analysis. These quantities are provided by the ARES Concept Report [13] or are approximated using data from similar past missions [3]. Table 1.4 lists these entry vehicle specifications.

Description	**Value**	**Units**
Aeroshell Geometry	sphere cone	–
Aeroshell Diameter	2.65	$[m]$
Aeroshell Reference Area	5.52	$[m^2]$
Vehicle Mass	580	$[kg]$
Ballistic Coefficient	65	$[kg/m^2]$
Lift/Drag	0	–
Inertia, I_{xx}	407	$[kg\, m^2]$
Inertia, $I_{yy} = I_{zz}$	264	$[kg\, m^2]$
Center of Mass	(-1 0 0)	$[m]$
Center of Pressure	(-3 0 0)	$[m]$

Table 1.4: ARES Entry Vehicle Specifications, [13]

Chapter 2

Mars Atmospheric Entry Dynamics

The environment encountered by a planetary entry vehicle ranges from the weightless vacuum of space to the intense temperatures and dynamic forces experienced in high velocity atmospheric deceleration. This makes the task of simulating atmospheric entry complicated. This chapter discusses the overall dynamics problem for planetary entry. Figure 2-1 highlights this section of the analysis as it fits into the complete simulation configuration.

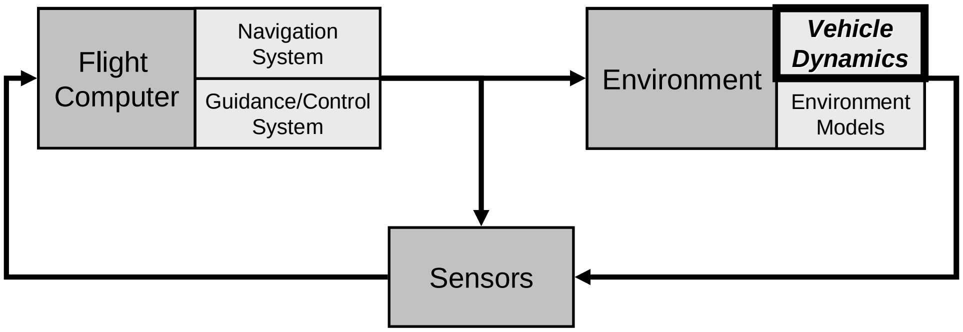


Figure 2-1: Simulation Conceptual Flow – Simulation Dynamics

The chapter is divided into three sections. First, the vehicle dynamic equations of motion are presented. Next, an approximate closed form analytical trajectory analysis is performed. Finally, a three degree-of-freedom (3-DOF) entry simulation is used to construct a baseline trajectory for this Mars entry scenario.

2.1 General Vehicle Dynamics Formulation

The analysis of this Mars entry problem is performed using a Mars centered, inertial coordinate frame. In order to determine the vehicle's position and attitude, the dynamic equations of motion are integrated using all the forces and torques that are applied to the vehicle.

Translational Dynamics

The translational dynamic equations of motion are

$$\begin{aligned} \dot{\boldsymbol{v}} &= \boldsymbol{a} \\ \dot{\boldsymbol{r}} &= \boldsymbol{v} \end{aligned} \tag{2.1}$$

where $\boldsymbol{a}$ is any acceleration applied to the vehicle. For this analysis, these accelerations are due to gravity forces, entry body aerodynamic forces, and parachute aerodynamic forces.

$$\boldsymbol{a} = \boldsymbol{a}_{gravity} + \boldsymbol{a}_{body\,aero} + \boldsymbol{a}_{chute\,aero}$$

The translational motion of the entry vehicle is determined using the position vector, velocity vector, and models of the relevant accelerations. Inertial position and velocity are found by integrating the differential equations in equation 2.1 using a fourth-order Runga-Kutta algorithm.

$$\begin{aligned} \boldsymbol{v}(t) &= \int_0^t \boldsymbol{a}(\tau)\ d\tau \\ \boldsymbol{r}(t) &= \int_0^t \boldsymbol{v}(\tau)\ d\tau \end{aligned} \tag{2.2}$$

Rotational Dynamics

The rotational dynamic equations of motion are

$$\begin{aligned}\dot{\boldsymbol{\omega}} &= \boldsymbol{I}^{-1}\left(\boldsymbol{T} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}\right) \\ \dot{\boldsymbol{q}} &= \frac{\boldsymbol{\omega} \otimes \boldsymbol{q}}{2}\end{aligned} \tag{2.3}$$

where $\boldsymbol{\omega}$ is the vehicle rotation rate, $\boldsymbol{I}$ is the vehicle moments of inertia matrix, $\boldsymbol{T}$ is any torques applied to the vehicle, and $\boldsymbol{q}$ is the vehicle attitude quaternion. For this analysis, any angular accelerations are due to entry body aerodynamic torques and parachute aerodynamic torques.

$$\boldsymbol{T} = \boldsymbol{T}_{body\ aero} + \boldsymbol{T}_{chute\ aero}$$

Using the initial attitude quaternion, initial angular velocity vector, vehicle moments of inertia, and models of the relevant torques, the rotational motion of the entry vehicle can be found. Vehicle attitude and angular velocity are found by integrating the differential equations in equation 2.3 using a fourth-order Runga-Kutta algorithm.

$$\begin{aligned}\boldsymbol{\omega}(t) &= \int_0^t \boldsymbol{I}^{-1}\left(\boldsymbol{T}(\tau) - \boldsymbol{\omega}(\tau) \times \boldsymbol{I}\boldsymbol{\omega}(\tau)\right)\, d\tau \\ \boldsymbol{q}(t) &= \int_0^t \frac{1}{2}\,\boldsymbol{\omega}(\tau) \otimes \boldsymbol{q}(\tau)\, d\tau\end{aligned} \tag{2.4}$$

2.2 3-DOF Trajectory Design

To begin the entry navigation analysis for this Mars ARES scenario mission, a nominal flight path trajectory is defined. A 3-DOF numerical simulation is used in order to establish this nominal trajectory. This simulation calculates the translational components of the entry vehicle, but ignores the rotational dynamics. No aerodynamic torques are calculated, and it is assumed that the vehicle's orientation is fixed in alignment with the current velocity vector. This method provides a reasonable approximation of the entry vehicle trajectory and a good reference for the higher-fidelity 6-DOF simulation used for the Mars entry navigation analysis. Figure 2-2 shows this nominal 3-DOF trajectory plotted over the planet fixed longitude/latitude grid.

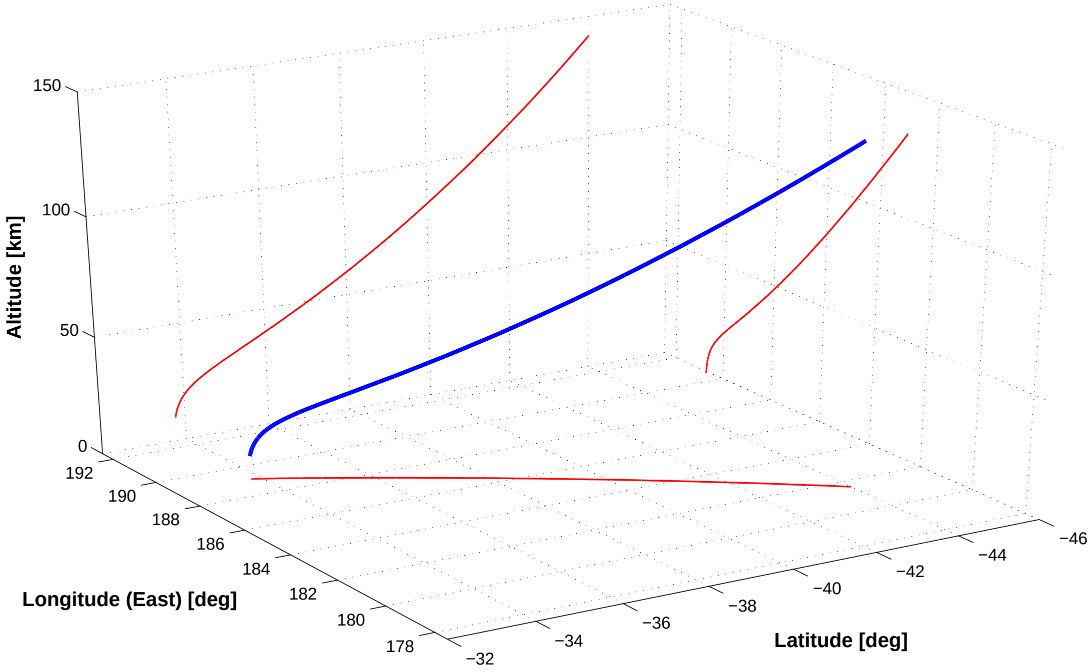


Figure 2-2: 3-DOF Vehicle Trajectory

The next set of figures present important vehicle properties calculated within this 3-DOF simulation as functions of the time of flight.

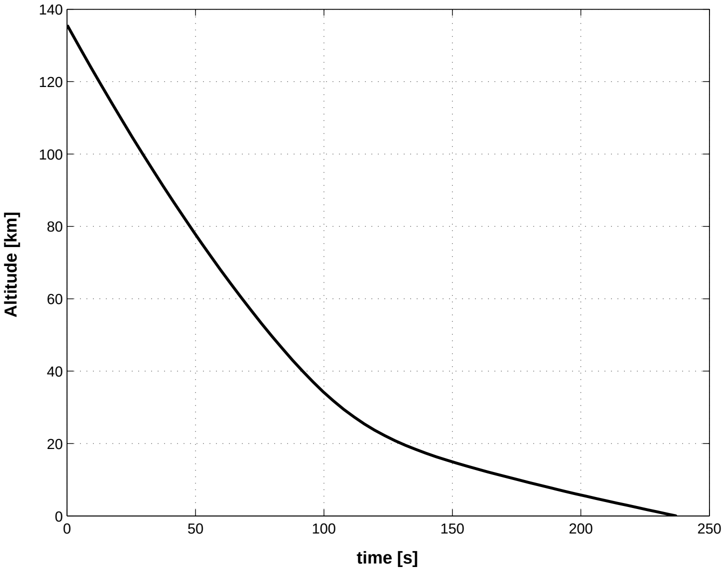


Figure 2-3: 3-DOF Vehicle Altitude

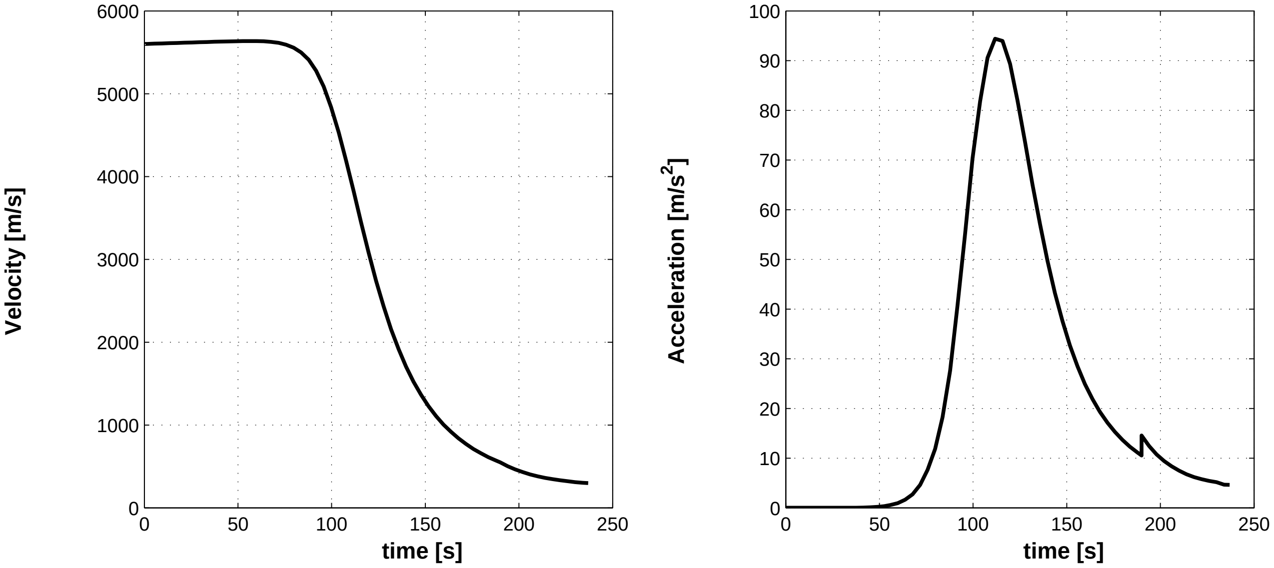
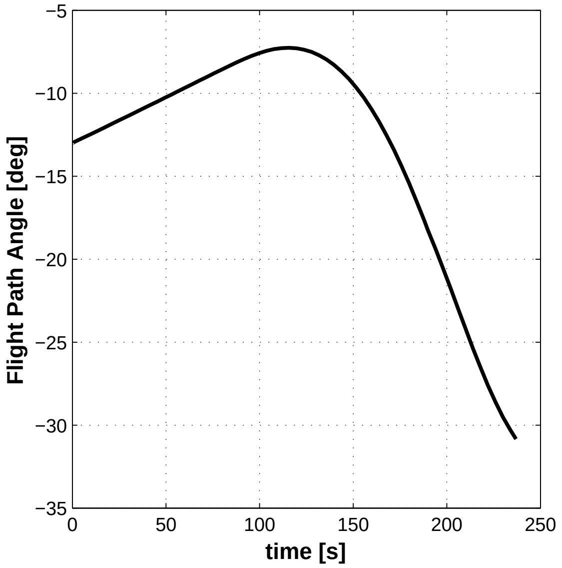


Figure 2-4: 3-DOF Vehicle Translational Dynamics



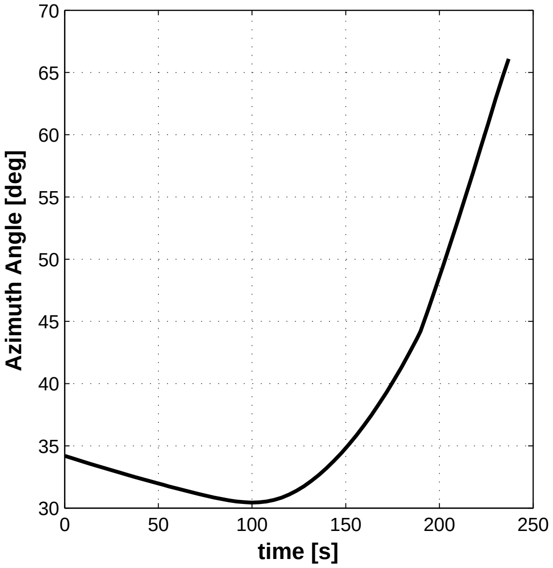
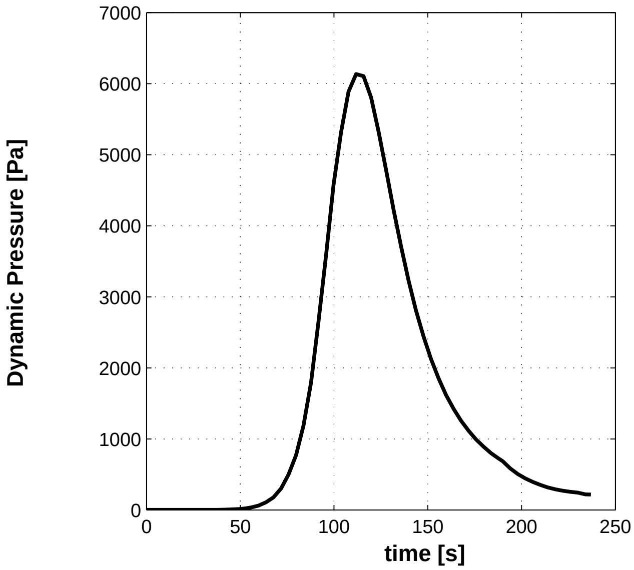


Figure 2-5: 3-DOF Vehicle Velocity Direction



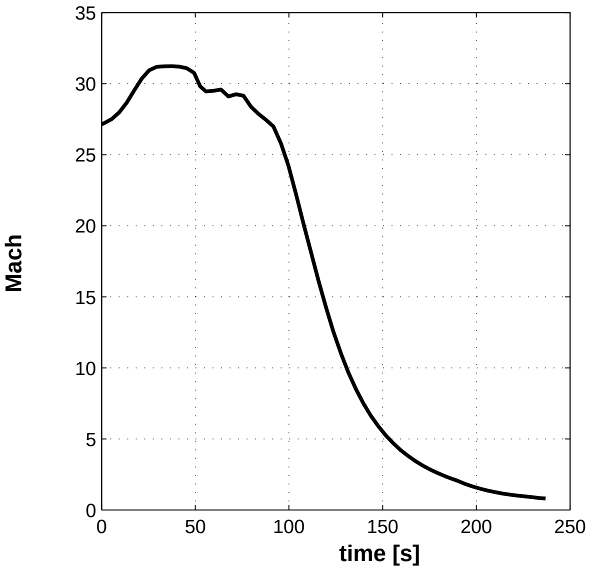


Figure 2-6: 3-DOF Vehicle Aerodynamic Properties

Chapter 3

6-DOF Atmospheric Entry Simulation

This chapter discusses the environmental and vehicle sensor models used within the simulation. These sections constitute major portions of the "true" simulated environment for use in the navigation analysis. Figure 3-1 highlights the these two sections within the total simulation configuration.

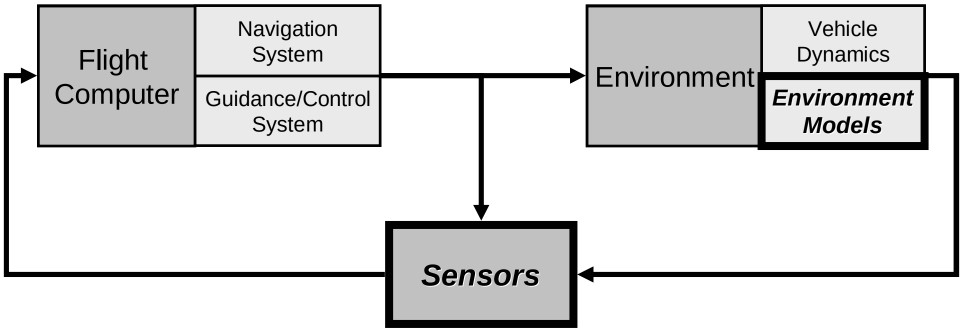


Figure 3-1: Simulation Conceptual Flow – Simulation Models

3.1 Environment Models

3.1.1 Atmosphere Model

Realistic representation of the Martian atmosphere is important for accurate entry navigation simulation. For the ARES mission, the atmospheric entry interface is defined as $3522.2\,km$ radius from the planet's center (approximately $126\,km$ altitude above the planet surface) [13]. Below this point the Mars Global Reference Atmospheric Model (Mars-GRAM 2001) [10] is used to simulate the Martian atmosphere. This model makes use of data table output generated from the NASA Ames Mars General Circulation Model (MGCM) and the University of Arizona Mars Thermospheric General Circulation Model (MTGCM) to create global, first-principal physics atmospheric data results [11]. Quantities such as temperature, pressure, density, and wind velocity are provided as a function of geographic location, altitude, time-of-day, and season. These atmospheric properties are used within the aerodynamic force and torque models described in Section 3.1.2 to properly calculate the atmospheric entry dynamics.

3.1.2 Entry Vehicle Aerodynamic Force and Torque Models

The entry vehicle aerodynamic model used in this analysis is a six degree-of-freedom (6-DOF) formulation using linearized stability derivatives. The aerodynamic forces and torques are found using the atmospheric properties (Section 3.1.1), the ARES entry vehicle specifications (Section 1.5), and the current position, velocity, and attitude of the vehicle (Section 2.1) [4, 14].

Aerodynamic Force The aerodynamic force vector is calculated in body coordinates as

$$\boldsymbol{f}^B = q\,S_{ref}\,\boldsymbol{c}_{xyz}(\alpha, \beta, M) \tag{3.1}$$

where q is the current dynamic pressure, S_{ref} is the entry body reference area (maximum frontal area), and $\boldsymbol{c}_{xyz}$ is the aerodynamic force coefficient vector. This vector is a function of angle-of-attack (α), side-slip (β), and Mach (M). It is defined as

$$\boldsymbol{c}_{xyz} = \begin{pmatrix} C_{x_0}(M) + \alpha^2\, C_{x_{\alpha 2}}(M) \\ \beta\, C_{yz_{\alpha\beta}}(M) \\ \alpha\, C_{yz_{\alpha\beta}}(M) \end{pmatrix}$$

The coefficients within the $\boldsymbol{c}_{xyz}$ vector are the zero angle-of-attack axial force coefficient (C_{x_0}), axial force versus α^2 coefficient ($C_{x_{\alpha 2}}$), and normal force derivative with α coefficient ($C_{yz_{\alpha\beta}}$). These coefficients are linearized functions of Mach number and are based on the Mars Smart Lander entry vehicle aerodynamics for the sphere cone aeroshell configuration.

Aerodynamic Torque The aerodynamic torque vector is calculated in body coordinates as

$$\begin{aligned} \boldsymbol{m}^B &= \boldsymbol{m}^B_{ref} \,-\, \boldsymbol{m}^B_{offset} \\ &= q\, S_{ref}\, L_{ref} \boldsymbol{c}_{lmn}(\alpha, \beta, M) \,-\, \left(\boldsymbol{r}_{cg} \,-\, \boldsymbol{r}_{ref}\right) \times \boldsymbol{f}^B \end{aligned} \tag{3.2}$$

where q is the current dynamic pressure, S_{ref} is the entry body reference area (maximum frontal area), L_{ref} is the reference length (diameter), $\boldsymbol{f}^B$ is the body frame aerodynamic force vector as calculated in the previous section, $\boldsymbol{r}_{cg}$ is the position vector of the entry vehicle center-of-gravity, and $\boldsymbol{r}_{ref}$ is the position vector to the aerodynamic coefficient reference point. Both position vectors are relative to the center nose of the entry vehicle aeroshell (Figure 3-2). The $\boldsymbol{c}_{lmn}$ term is the aerodynamic torque coefficient vector. This vector is a function of angle-of-attack (α), side-slip (β), and Mach (M). It is defined as

$$\boldsymbol{c}_{lmn} = \begin{pmatrix} \beta\, C_{L_\beta} \\ \alpha\, C_{m_\alpha}(M) \\ -\beta\, C_{m_\alpha}(M) \end{pmatrix}$$

The coefficients within the $\boldsymbol{c}_{lmn}$ vector are roll moment derivative with β coefficient (C_{L_β}) and pitch moment derivative with α coefficient (C_{m_α}). This last coefficient is a linearized function of Mach number. Both coefficients are based on the Mars Smart Lander entry vehicle aerodynamics.

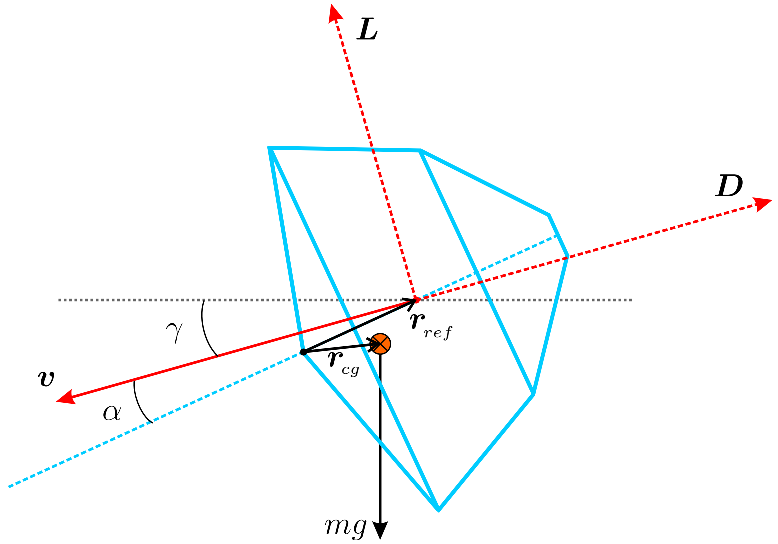


Figure 3-2: Entry Body Aerodynamic Free Body Diagram

Figure 3-2 shows a basic aerodynamic free-body diagram of the entry body. The quantities $\boldsymbol{L}$ and $\boldsymbol{D}$ represent the directions of Lift and Drag, respectively, as they correspond to the vehicle body frame (measured by α). The flight path angle (γ) represents the angle of the vehicle's velocity vector ($\boldsymbol{v}$) below the local horizontal plane.

3.1.3 Parachute Aerodynamic Force and Torque Models

The parachute aerodynamic model used in this analysis produces forces and torques that act on the vehicle. The aerodynamic forces and torques are found using the atmospheric properties (Section 3.1.1) and the current position and velocity of the vehicle (Section 2.1). This model is based on the parachutes designed for the Mars Pathfinder and Mars Smart Lander missions [8, 16].

Aerodynamic Force The aerodynamic force vector is calculated in inertial coordinates as

$$\boldsymbol{f}^I = -\,q\,S_{chute}\,C_{D_{chute}}(M)\,\hat{\boldsymbol{\imath}}_{v_{rel}} \tag{3.3}$$

where q is the current dynamic pressure, S_{chute} is the parachute reference area (maximum frontal area), $C_{D_{chute}}$ is the aerodynamic drag force coefficient, and $\hat{\boldsymbol{\imath}}_{v_{rel}}$ is the unit vector pointing in the vehicle's atmosphere-relative velocity direction. The drag coefficient is a function of Mach (M).

Aerodynamic Torque The aerodynamic torque vector is calculated in body coordinates as

$$\boldsymbol{m}^B = -\left(\boldsymbol{r}_{cg} - \boldsymbol{r}_{chute}\right) \times \boldsymbol{f}^B \tag{3.4}$$

where $\boldsymbol{r}_{cg}$ is the position vector of the entry vehicle center-of-gravity, $\boldsymbol{r}_{chute}$ is the position vector to the aerodynamic coefficient reference point, and $\boldsymbol{f}^B$ is the body frame aerodynamic force vector as calculated in equation 3.3. Both position vectors are relative to the center nose of the entry vehicle aeroshell as in the previous section.

3.1.4 Gravity Force Model

Gravity Force Model The other force of interest for atmospheric entry is gravity. The gravitational potential produced by a spherically symmetric planet is

$$\begin{aligned} \boldsymbol{u} &= \frac{-\,GM}{r}\,\hat{\boldsymbol{\imath}}_r \\ &= \frac{-\mu}{r}\,\hat{\boldsymbol{\imath}}_r \end{aligned} \tag{3.5}$$

where G is the gravity constant, M is the planet's mass, μ is the product of G and M and is known as the planet gravity constant, r is the radial distance from the center-of-gravity of the planet, and $\hat{\boldsymbol{\imath}}_r$ is the radial unit vector. Taking the gradient of this vector yields the gravity acceleration.

$$\begin{aligned} \boldsymbol{a}_g &= -\nabla \boldsymbol{u} \\ &= \frac{-\,\mu}{r^2}\,\hat{\boldsymbol{i}}_r \end{aligned} \tag{3.6}$$

Planets with an approximate spherically symmetric mass can have their gravity potential estimated with a perturbation to equation 3.5 [1, 19]

$$u = \frac{-\mu}{r} + b\left(r, \theta, \phi\right) \tag{3.7}$$

where the perturbation term is a function of radius (r), latitude (θ), and longitude (ϕ) and is defined as

$$\begin{aligned} b\left(r,\theta,\phi\right) = \; & \frac{\mu}{r}\left\{\sum_{n=2}^{\infty}\left[\left(\frac{r_{eq}}{r}\right)^n J_n\,P_{n0}\left(\cos\theta\right)\right.\right. \\ & \left.\left. + \sum_{m=1}^{n}\left(\frac{r_{eq}}{r}\right)^n \left(C_{nm}\cos m\,\phi + S_{nm}\sin m\,\phi\right)P_{nm}\left(\cos\theta\right)\right]\right\} \end{aligned} \tag{3.8}$$

where r_{eq} is the planet equatorial radius, J_n are the zonal harmonic coefficients, P_{nm} are the Legendre polynomials, and C_{nm} and S_{nm} are the tesseral and sectoral harmonic coefficients corresponding to $n \neq m$ and $n = m$, respectively [19]. The zonal harmonics correspond to planetary oblateness, while the tesseral harmonics describe longitudinal variation. If the planetary body is assumed axially symmetric[1], tesseral and sectoral harmonics become zero and simplify equation 3.7 into

$$u\left(r,\theta\right) = \frac{-\mu}{r}\left\{1 - \sum_{n=2}^{\infty}\left(\frac{r_{eq}}{r}\right)^n J_n\,P_{n0}\left(\cos\theta\right)\right\} \tag{3.9}$$

Taking the gradient of the perturbed gravity potential in equation 3.9 produces the

---

[1] Axial symmetry is often a valid assumption because longitudinal variations are both small and tend to average out for most orbital motion.

axially symmetric gravity acceleration equation.

$$\boldsymbol{a}_g\left(r,\theta\right) = \frac{-\mu}{r^2}\,\left\{\hat{\boldsymbol{i}}_r - \sum_{n=2}^{\infty}\left(\frac{r_{eq}}{r}\right)^n J_n\left[P'_{n+1}\left(\cos\theta\right)\hat{\boldsymbol{i}}_r - P'_n\left(\cos\theta\right)\hat{\boldsymbol{i}}_z\right]\right\} \tag{3.10}$$

For Mars the second order zonal harmonic (J_2) dominates the higher order terms. This allows further simplification of equation 3.10 into

$$\boldsymbol{a}_g\left(r,\theta\right) = \frac{-\mu}{r^2}\,\left\{\hat{\boldsymbol{i}}_r - \left(\frac{r_{eq}}{r}\right)^2 J_2\left[P'_3\left(\cos\theta\right)\hat{\boldsymbol{i}}_r - P'_2\left(\cos\theta\right)\hat{\boldsymbol{i}}_z\right]\right\} \tag{3.11}$$

According to Lear [12], this relationship for J_2 gravitational acceleration can be rewritten in cartesian X-Y-Z coordinates as

$$\boldsymbol{a}_g\left(x,y,z\right) = \frac{-(\,\mu_x\,x\,\hat{\boldsymbol{i}}_x + \mu_y\,y\,\hat{\boldsymbol{i}}_y + \mu_z\,z\,\hat{\boldsymbol{i}}_z)}{r^3} \tag{3.12}$$

where

$$\begin{aligned}
\mu_x &= \mu\left[1 + \frac{3}{2}\,J_2\,\frac{r_{eq}^2}{r^2}\left(1 - 5\frac{z^2}{r^2}\right)\right] \\
\mu_y &= \mu_x \\
\mu_z &= \mu\left[1 + \frac{3}{2}\,J_2\,\frac{r_{eq}^2}{r^2}\left(3 - 5\frac{z^2}{r^2}\right)\right] \;=\; \mu_x + 3\,J_2\,\frac{r_{eq}^2}{r^2}\,\mu \\
r &= \sqrt{x^2 + y^2 + z^2}
\end{aligned}$$

Lear's formulation in equation 3.12 is the gravity model used for this analysis. It allows for reasonably accurate evaluation of gravitational acceleration by modeling the spherical gravity field plus latitudinal perturbations.

3.2 Sensor Models

3.2.1 Accelerometers

Strapdown accelerometers are used for this analysis. This means that the unit is mounted to the vehicle body instead of to an inertial platform and therefore provides a measure of nongravitational accelerations in the vehicle body coordinate frame [2]. These sensed nongravitational accelerations are typically due to aerodynamic forces, parachute deployment forces, and reaction control system (RCS) forces. The accelerometer error model is formulated as [2, 6]

$$\boldsymbol{a}^B_m = (\boldsymbol{I} + \boldsymbol{\Gamma}_a)(\boldsymbol{I} + \boldsymbol{S}_a)(\boldsymbol{a}^B + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a) \tag{3.13}$$

which gives the measured accelerations ($\boldsymbol{a}^B_m$) in the body frame as corrupted values of the true accelerations ($\boldsymbol{a}^B$). This information is corrupted by input axis nonorthogonality and gyro-to-accelerometer misalignment errors ($\boldsymbol{\Gamma}_a$), accelerometer scale factor errors ($\boldsymbol{S}_a$), accelerometer biases ($\boldsymbol{b}_a$), and accelerometer white noise ($\boldsymbol{\epsilon}_a$). These error terms are defined as

$$\boldsymbol{S}_a = \begin{bmatrix} s_{a_x} & 0 & 0 \\ 0 & s_{a_y} & 0 \\ 0 & 0 & s_{a_z} \end{bmatrix} \qquad \boldsymbol{b}_a = \begin{pmatrix} b_{a_x} \\ b_{a_y} \\ b_{a_z} \end{pmatrix} \qquad \boldsymbol{\epsilon}_a = \begin{pmatrix} \boldsymbol{\epsilon}_{a_x} \\ \boldsymbol{\epsilon}_{a_y} \\ \boldsymbol{\epsilon}_{a_z} \end{pmatrix}$$

---

[2] The accelerometer unit is assumed to reside at the vehicle center-of-gravity for this analysis. Otherwise the measured accelerations would require an additional term to compensate for vehicle rotation.

$$
\begin{aligned}
\boldsymbol{\Gamma}_a &= \boldsymbol{\Gamma}_{a_{nonorth}} + \boldsymbol{\Gamma}_{a_{misalign}} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ -n_{a_{yz}} & 0 & 0 \\ n_{a_{zy}} & -n_{a_{zx}} & 0 \end{bmatrix} + \begin{bmatrix} 0 & m_{a_z} & -m_{a_y} \\ -m_{a_z} & 0 & m_{a_x} \\ m_{a_y} & -m_{a_x} & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & \gamma_{a_{xz}} & -\gamma_{a_{xy}} \\ -\gamma_{a_{yz}} & 0 & \gamma_{a_{yx}} \\ \gamma_{a_{zy}} & -\gamma_{a_{zx}} & 0 \end{bmatrix}
\end{aligned}
$$

The white process noise term ($\boldsymbol{\epsilon}_a$) has a covariance of

$$E\Big[\,\boldsymbol{\epsilon}_a\,\boldsymbol{\epsilon}_a^T\Big] = \boldsymbol{Q}_a\,\delta(t) \tag{3.14}$$

where $\boldsymbol{Q}_a$ is the process noise intensity and is defined as the square of the *random walk error* shown in Table 3.1. The $\delta(t)$ term is the Dirac delta function.

Neglecting second-order terms, the first two binomials of equation 3.13 can be approximated as

$$(\boldsymbol{I} + \boldsymbol{\Gamma}_a)\,(\boldsymbol{I} + \boldsymbol{S}_a) \approx (\boldsymbol{I} + \boldsymbol{\Gamma}_a + \boldsymbol{S}_a)$$

Using this approximation the first-order accelerometer error model becomes

$$
\begin{aligned}
\boldsymbol{a}^B_m &= (\boldsymbol{I} + \boldsymbol{\Gamma}_a + \boldsymbol{S}_a)\,(\boldsymbol{a}^B + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a) \\
&= \boldsymbol{a}^B + \boldsymbol{\Gamma}_a\boldsymbol{a}^B + \boldsymbol{S}_a\boldsymbol{a}^B + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a
\end{aligned}
\tag{3.15}
$$

where $\boldsymbol{a}^B_m$ is the acceleration vector output of the accelerometer sensor. This final formulation is the accelerometer error model used within the simulation sensor block in Figure 3-1.

The magnitude of the error sources are dependent on the configuration of the specific accelerometer in use. Accelerometers are combined with gyros in navigation systems to form an IMU. The two commercially available, spaceflight qualified IMU packages under consideration are the Honeywell Miniature Inertial Measurement Unit (MIMU) and the Litton LN200 Inertial Measurement Unit. The error specifications for the model shown above are listed in table 3.1. The accelerometer is simulated deterministically by randomly selecting error terms in equation 3.15 that are normally distributed according to the statistical specifications in the table.

Accelerometer Errors (1σ)	**MIMU**	**LN200**	**Units**
Bias, $\boldsymbol{b}_a$	100	300	$[\mu g]$
Scale Factor, $\boldsymbol{s}_a$	175	300	$[ppm]$
Nonorthogonality	15	20	$[arcsec]$
Misalignment	15	20	$[arcsec]$
Random Walk, $\boldsymbol{\epsilon}_a$	0.00015	0.00049	$[m/s/\sqrt{s}]$

Table 3.1: IMU Specifications – Accelerometer Errors

3.2.2 Gyros

The gyro unit captures angular velocities of the vehicle body frame relative to the inertial frame. The error model for the gyro unit is [2, 6]

$$\boldsymbol{\omega}_m^B = (\boldsymbol{I} + \boldsymbol{\Gamma}_g)(\boldsymbol{I} + \boldsymbol{S}_g)(\boldsymbol{\omega}^B + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g) \tag{3.16}$$

This gives the measured angular velocities ($\boldsymbol{\omega}_m^B$) in the body frame as corrupted values of the true angular velocities ($\boldsymbol{\omega}^B$). This information is corrupted by gyro input axis nonorthogonality and misalignment errors ($\boldsymbol{\Gamma}_g$), gyro scale factor errors ($\boldsymbol{S}_g$), gyro biases ($\boldsymbol{b}_g$), and gyro white noise ($\boldsymbol{\epsilon}_g$). These error terms are defined as

$$\boldsymbol{S}_g = \begin{bmatrix} s_{g_x} & 0 & 0 \\ 0 & s_{g_y} & 0 \\ 0 & 0 & s_{g_z} \end{bmatrix} \qquad \boldsymbol{b}_g = \begin{pmatrix} b_{g_x} \\ b_{g_y} \\ b_{g_z} \end{pmatrix} \qquad \boldsymbol{\epsilon}_g = \begin{pmatrix} \boldsymbol{\epsilon}_{g_x} \\ \boldsymbol{\epsilon}_{g_y} \\ \boldsymbol{\epsilon}_{g_z} \end{pmatrix}$$

$$
\begin{aligned}
\boldsymbol{\Gamma}_g &= \boldsymbol{\Gamma}_{g_{nonorth}} + \boldsymbol{\Gamma}_{g_{misalign}} \\
&= \begin{bmatrix} 0 & 0 & 0 \\ -n_{g_{yz}} & 0 & 0 \\ n_{g_{zy}} & -n_{g_{zx}} & 0 \end{bmatrix} + \begin{bmatrix} 0 & m_{g_z} & -m_{g_y} \\ -m_{g_z} & 0 & m_{g_x} \\ m_{g_y} & -m_{g_x} & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & \gamma_{g_{xz}} & -\gamma_{g_{xy}} \\ -\gamma_{g_{yz}} & 0 & \gamma_{g_{yx}} \\ \gamma_{g_{zy}} & -\gamma_{g_{zx}} & 0 \end{bmatrix}
\end{aligned}
$$

The white process noise term ($\boldsymbol{\epsilon}_g$) has a covariance of

$$
E\Big[\boldsymbol{\epsilon}_g \, \boldsymbol{\epsilon}_g^T\Big] = \boldsymbol{Q}_g \, \delta(t) \tag{3.17}
$$

where $\boldsymbol{Q}_g$ is the process noise intensity and is defined as the square of the *random walk error* shown in Table 3.2. The $\delta(t)$ term is the Dirac delta function.

Neglecting second-order terms, the first two binomials of equation 3.16 can be approximated as

$$
(\boldsymbol{I} + \boldsymbol{\Gamma}_g)\,(\boldsymbol{I} + \boldsymbol{S}_g) \approx (\boldsymbol{I} + \boldsymbol{\Gamma}_g + \boldsymbol{S}_g)
$$

Using this approximation the first-order gyro error model becomes

$$
\begin{aligned}
\boldsymbol{\omega}^B_m &= (\boldsymbol{I} + \boldsymbol{\Gamma}_g + \boldsymbol{S}_g)\,(\boldsymbol{\omega}^B + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g) \\
&= \boldsymbol{\omega}^B + \boldsymbol{\Gamma}_g\boldsymbol{\omega}^B + \boldsymbol{S}_g\boldsymbol{\omega}^B + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g
\end{aligned} \tag{3.18}
$$

where $\boldsymbol{\omega}^B_m$ is the rotation rate vector output by the gyro sensors. This final formulation is the gyro error model used within the simulation sensor block in Figure 3-1.

The error sources are dependent on the configuration of the specific gyro in use. Gyros are combined with accelerometers in navigation systems to form an IMU. The two commercially available, spaceflight qualified IMU packages under consideration are the Honeywell Miniature Inertial Measurement Unit (MIMU) and the Litton

LN200 Inertial Measurement Unit. The error specifications for the model shown above are listed in table 3.2. The gyro is simulated deterministically by randomly selecting error terms in equation 3.18 that are normally distributed according to the statistical specifications in the table. The gyro lacks a misalignment error (as compared to the accelerometer) because its alignment to the vehicle defines the body coordinate frame.

Gyro Errors (1σ)	**MIMU**	**LN200**	**Units**
Bias, $\boldsymbol{b}_g$	0.05	1	$[\,deg/hr\,]$
Scale Factor, $\boldsymbol{s}_g$	5	100	$[\,ppm\,]$
Nonorthogonality, $\boldsymbol{\gamma}_g$	25	20	$[\,arcsec\,]$
Random Walk, $\boldsymbol{\epsilon}_g$	.0001	.0012	$[\,deg/\sqrt{s}\,]$

Table 3.2: IMU Specifications – Gyro Errors

3.2.3 Radar Altimeter

The radar altimeter is a vehicle sensor that provides range measurements to the planet surface. This type of instrument typically is concealed within the entry body and can only begin collecting data upon aeroshell jettison. The instrument is fixed to the vehicle body frame and therefore measures distance to the surface as a function of vehicle attitude. The sensor essentially takes the true distance being measured and corrupts it with various error sources. The altimeter model with these error terms is formulated as

$$h_m = h\,(1 + k_a) + \nu_a \tag{3.19}$$

where h_m is the measured distance, h is the true distance, k_a is the scale factor, and ν_a is the sensor noise. The error sources are dependent on the configuration of the specific altimeter in use. The two sensors under consideration are the Mars Surveyor Program (MSP) and Mars Exploration Rover (MER) -type altimeters. The error specifications for the model shown above are listed in Table 3.3. The altimeter is simulated deterministically by randomly selecting error terms in equation 3.19 that are normally distributed according to the statistical specifications in the table.

Altimeter Errors (1σ)	**MSP**	**MER**	**Units**
Scale Factor, k_a	1.66	0.33	[% of altitude]
Noise, ν_a	1	0.2	[m]

Table 3.3: Altimeter Specifications

The true altimeter distance can be defined geometrically in terms of vehicle position and attitude. Figure 3-3 shows the important quantities for defining this altimeter measurement.

To find the value of h, a geometric relationship can be formed through vector addition.

$$\left|\,\boldsymbol{r}^{\,I} + (\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B)\,h\,\right|^2 = R_s^2 \tag{3.20}$$

where $\boldsymbol{r}^{\,I}$ is the vehicle position vector, $\boldsymbol{T}_{B\to I}$ is the body to inertial transformation, $\boldsymbol{i}_a^B$ is the altimeter pointing direction unit vector in body coordinates, and R_s is the radius of the planet surface at the point of altimeter signal contact. R_s is a function of the current vehicle position and attitude above the planet and is determined using data from the Mars Orbiter Laser Altimeter (MOLA) aboard the Mars Global Surveyor (MGS) spacecraft. Expanding equation 3.20 gives

$$\left|\boldsymbol{r}^{\,I}\right|^2 + 2\,\boldsymbol{r}^{\,I}\cdot(\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B)\,h + h^2 = R_s^2 \tag{3.21}$$

This polynomial can then be solved for the true altimeter distance, h.

$$h = -\left[\,\boldsymbol{r}^{\,I}\cdot\left(\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\right)\right] \;-\; \tfrac{1}{2}\sqrt{4\left[\,\boldsymbol{r}^{\,I}\cdot\left(\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\right)\right]^2 \;-\; 4\left|\boldsymbol{r}^{\,I}\right|^2 \;+\; 4R_s^2} \tag{3.22}$$

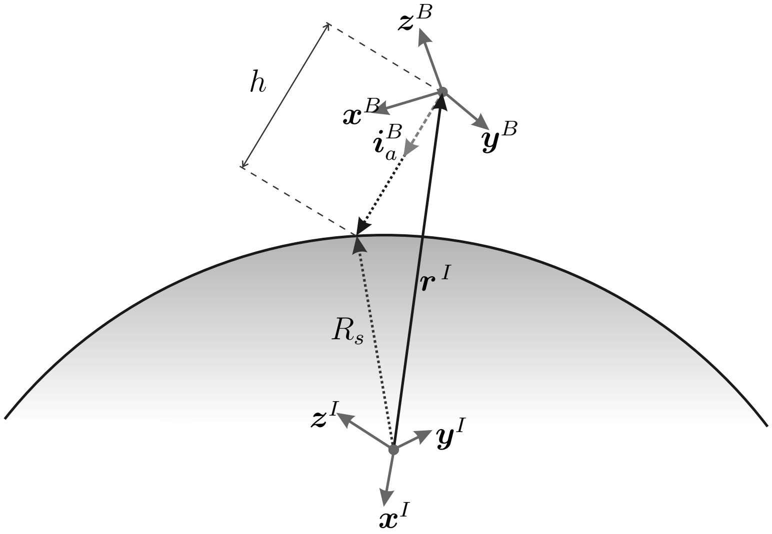


Figure 3-3: Altimeter Measurement

3.2.4 Velocimeter

The velocimeter is an instrument that uses beams to determine the surface relative velocity of the vehicle. The model for this instrument is formulated as

$$\boldsymbol{v}_m^B = \boldsymbol{T}_{I\to B}\left(\boldsymbol{v}^I - \boldsymbol{v}_s\right)(1 + k_v) + \boldsymbol{\nu}_v \tag{3.23}$$

where $\boldsymbol{v}_m^B$ is the surface relative measured velocity vector in body coordinates, $\boldsymbol{v}^I$ is the true inertial vehicle velocity, $\boldsymbol{v}_s$ is the planet surface velocity, k_v is the scale factor, and $\boldsymbol{\nu}_v$ is the sensor noise. The surface velocity is defined as

$$\boldsymbol{v}_s = \boldsymbol{\omega} \times \boldsymbol{R}_s \tag{3.24}$$

where $\boldsymbol{\omega}$ is the rotational rate of the planet (see Table 1.2, page 19) and $\boldsymbol{R}_s$ is a vector to the point where the radar beam intercepts the planet surface. It is defined as

$$\boldsymbol{R}_s = \boldsymbol{r}^I + h\left(\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\right) \tag{3.25}$$

The error sources are dependent on the configuration of the specific velocimeter in use. The sensor under consideration is the Mars Surveyor Program (MSP) -type velocimeter. The error specifications for the model shown above are listed in table 3.4. The velocimeter is simulated deterministically by randomly selecting error terms in equation 3.23 that are normally distributed according to the statistical specifications in the table. Figure 3-4 shows the important quantities for defining this velocimeter measurement.

Velocimeter Errors (1σ)	**MSP**	**Units**
Scale Factor, k_v	1.33	[% of velocity]
Noise, $\boldsymbol{\nu}_v$	0.1	$[\, m/s \,]$

Table 3.4: Velocimeter Specifications

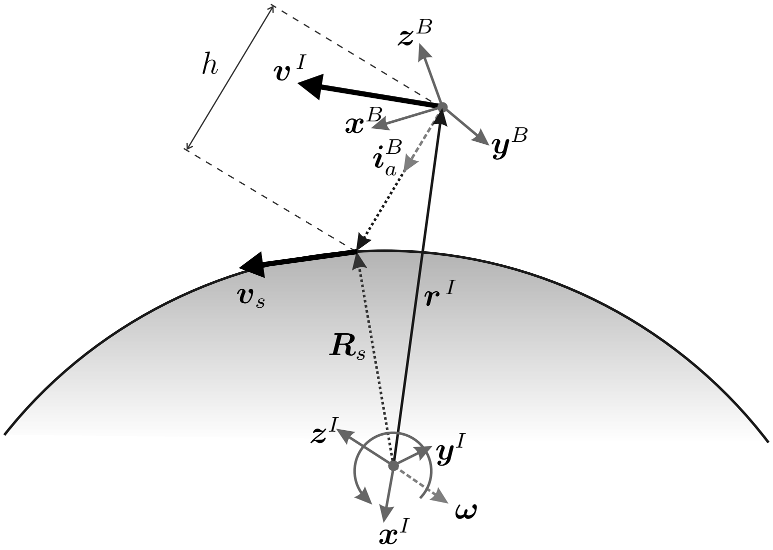


Figure 3-4: Velocimeter Measurement

Chapter 4

Navigation System and Algorithms

This chapter provides the detailed information on the configuration and estimation algorithms used within the navigation system implemented for this analysis. Figure 4-1 highlights the navigation system within the total simulation configuration.

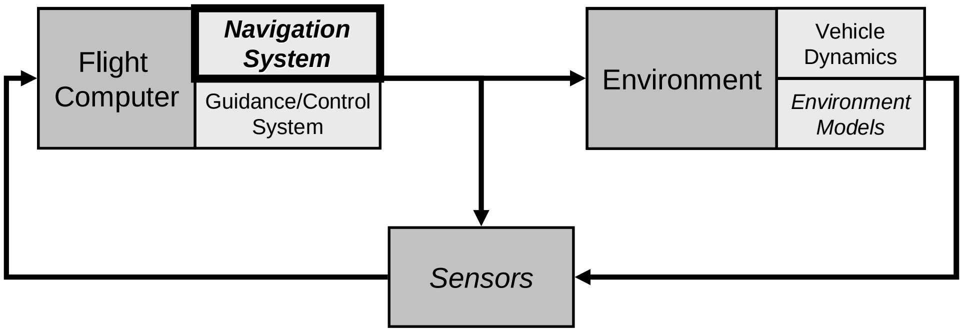


Figure 4-1: Simulation Conceptual Flow – Navigation System

The fundamental basis of this navigation system is the Kalman filter estimation algorithm. Estimation can trace its history back to Gauss (circa 1800) [9] in his least-squares estimation method. Since that time numerous varieties of data processing and estimation methodologies have come about. The Kalman filter was a significant step forward in the world of estimation because it is a state-space, time domain recursive estimation filter. It is optimal in the least-squares sense, making it very much like a recursive version of Gauss's original least-squares problem. This filter is also nonstationary by using dynamic models of the system to predict state changes over time. The Kalman filter estimation algorithm is well suited for digital computer implementation because of its recursive nature. This explains its early use in spacecraft navigation where the recursive estimation required very little computer memory for the real-time calculations.

The conceptual state vector used for this implementation of Kalman filter is seen below. The components of the state vector are chosen so that the filter tracks the principle vehicle states (position, velocity, and attitude) and any sensor error terms and Mars topography/terrain model errors. The estimates of sensor error terms are used to improve the estimates of the primary states. The primary states are then potentially used by the guidance system to make corrective maneuvers to mitigate any dispersions from the vehicle's nominal trajectory.

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{position} \\ \boldsymbol{velocity} \\ \boldsymbol{attitude} \\ \boldsymbol{gyro\ errors} \\ \boldsymbol{accelerometer\ errors} \\ \boldsymbol{altimeter\ errors} \\ \boldsymbol{velocimeter\ errors} \end{pmatrix}$$

The state vector separates into nine subvectors and 3 scalars. This nominal state vector is defined to be

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{r}^I & \boldsymbol{v}^I & \boldsymbol{q}_{_{I\to B}} & \boldsymbol{b}_g & \boldsymbol{s}_g & \boldsymbol{\gamma}_g & \boldsymbol{b}_a & \boldsymbol{s}_a & \boldsymbol{\gamma}_a & x_t & k_a & k_v \end{pmatrix}^T$$

The first three subvectors are the principle states of the vehicle: position ($\boldsymbol{r}^I$) specified in inertial coordinates, velocity ($\boldsymbol{v}^I$) specified in inertial coordinates, and attitude specified as an inertial to body quaternion ($\boldsymbol{q}_{_{I\to B}}$)[1]. The remaining state vector terms correspond to sensor errors as defined in Section 3.2. Of these, the first six terms are the gyro bias ($\boldsymbol{b}_g$), gyro scale factor ($\boldsymbol{s}_g$), gyro misalignment ($\boldsymbol{\gamma}_g$), accelerometer bias ($\boldsymbol{b}_a$), accelerometer scale factor ($\boldsymbol{s}_a$), and accelerometer misalignment ($\boldsymbol{\gamma}_a$). Further

---

[1] Definition and discussion of the quaternion is found in Section 4.3

discussion on these terms is provided in Section 4.2. The three remaining terms are scalars representing the surface topography error (x_t), altimeter scale factor (k_a), and velocimeter scale factor (k_v).

This chapter is organized in the following way. First, a description and derivation is provided of the basic, discrete Kalman filter algorithm. Next, the sensor models used within the navigation filter are described. Finally, detailed descriptions of the implemented Kalman filter components are provided.

4.1 Discrete Linear Kalman Filter Formulation

The Kalman filter is well described as an "optimal recursive data processing algorithm" or more commonly known as the "optimal linear estimator" [15]. Optimal can be defined in many ways. With the assumptions of the system being described by a linear model and the system and measurement noise being white and Gaussian, Maybeck says the

> filter is optimal with respect to virtually any criterion that makes sense. One aspect of this optimality is that the Kalman filter incorporates all information that can be provided to it. It processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest, with use of (1) knowledge of the system and measurement device dynamics, (2) the statistical description of the system noises, measurement errors, and uncertainty in the dynamics models, and (3) any available information about initial conditions of the variables of interest. [15]

This section describes the discrete formulation of the Kalman filter as described by Gelb, Lear, Maybeck, and VanderVelde [9, 12, 15, 18]. First, the state dynamics and measurement models are developed in Section 4.1.1. Next, the Kalman filter is formulated as a two-stage process: the propagation of the estimated state vector, $\hat{\boldsymbol{x}}$, and state error covariance matrix, $\boldsymbol{P}$, forward in time (Section 4.1.2); and the measurement update of the state and state error covariance (Section 4.1.3). Finally, a summary is given in Section 4.1.4. Figure 4-2 provides a basic description of the Kalman filter loop.

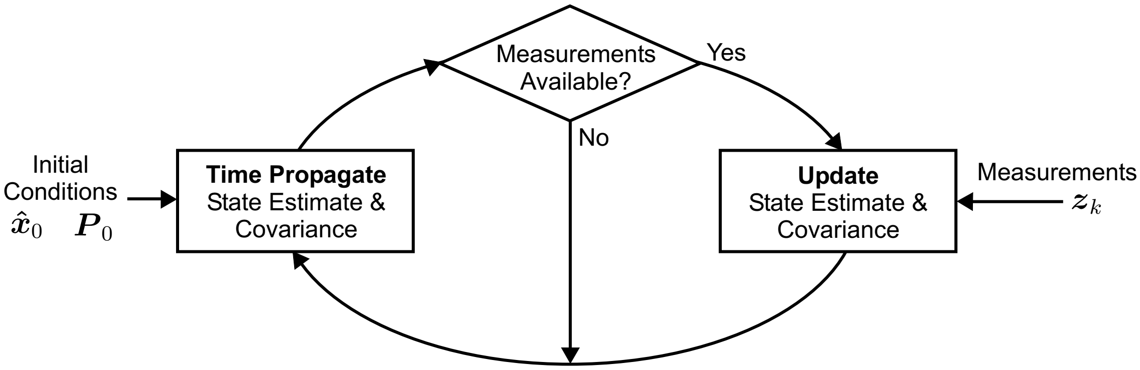


Figure 4-2: Discrete Kalman Filter Algorithm Flow

4.1.1 State Dynamics and Measurement Models

For discrete dynamics, time-variant quantities possess the subscript "k" corresponding to the time step index. The linear dynamic system is of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \, \boldsymbol{x}_k + \boldsymbol{w}_k \tag{4.1}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k \, \boldsymbol{x}_k + \boldsymbol{v}_k \tag{4.2}$$

where equation 4.1 is the discrete representation of the state dynamics with $\boldsymbol{\Phi}_k$ as the *state transformation matrix* and $\boldsymbol{w}_k$ as the white *process noise vector* with zero mean and covariance $\boldsymbol{Q}_k$.

$$E\Big[\, \boldsymbol{w}_k \,\Big] = 0 \tag{4.3}$$

$$E\Big[\, \boldsymbol{w}_j \, \boldsymbol{w}_k^T \,\Big] = \left\{ \begin{array}{ll} \boldsymbol{Q}_k & j = k \\ 0 & j \neq k \end{array} \right\} \tag{4.4}$$

Equation 4.2 is the linear discrete form of the measurement vector where $\boldsymbol{H}_k$ is the *measurement sensitivity matrix* (also known as the *observation matrix*) and is used to describe the linear combination of state variables which form $\boldsymbol{z}_k$ in the absence of noise [9]. The term $\boldsymbol{v}_k$ is the white *measurement noise vector* with zero mean and covariance $\boldsymbol{R}_k$.

$$E\Big[\,\boldsymbol{v}_k\,\Big] = 0 \tag{4.5}$$

$$E\Big[\,\boldsymbol{v}_j\,\boldsymbol{v}_k^T\,\Big] = \left\{\begin{array}{ll} \boldsymbol{R}_k & j = k \\ 0 & j \neq k \end{array}\right\} \tag{4.6}$$

The process noise and measurement noise are assumed to be uncorrelated. This is shown by

$$E\Big[\,\boldsymbol{v}_k\,\boldsymbol{w}_k^T\,\Big] = E\Big[\,\boldsymbol{w}_k\,\boldsymbol{v}_k^T\,\Big] = 0$$

The state estimate vector, $\hat{\boldsymbol{x}}_k$, is defined as

$$\hat{\boldsymbol{x}}_k = E\Big[\,\boldsymbol{x}_k\,\Big] \tag{4.7}$$

The state error vector, $\tilde{\boldsymbol{x}}_k$, is defined as the difference between the state estimate, $\hat{\boldsymbol{x}}_k$, and the true state, $\boldsymbol{x}_k$.

$$\tilde{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k \tag{4.8}$$

The state error covariance matrix, $\boldsymbol{P}_k$, is then defined as

$$E\Big[\,\tilde{\boldsymbol{x}}_k\,\tilde{\boldsymbol{x}}_k^T\,\Big] = \boldsymbol{P}_k \tag{4.9}$$

4.1.2 Time Propagation Equations

State Propagation

During the estimation process, the estimated state vector is propagated forward in time between the period of available measurements. This vector, $\hat{\boldsymbol{x}}_k$, is propagated from time (k) to $(k+1)$ using

$$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{\Phi}_k \, \hat{\boldsymbol{x}}_k \tag{4.10}$$

where $\boldsymbol{\Phi}_k$ is the state transformation matrix.

Covariance Propagation

The state error covariance matrix as defined in equation 4.9 must also be propagated in time between the measurements. Substituting equations 4.1 and 4.10 into the state error definition

$$\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}$$

gives the form

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1} &= (\boldsymbol{\Phi}_k \, \hat{\boldsymbol{x}}_k) - (\boldsymbol{\Phi}_k \, \boldsymbol{x}_k + \boldsymbol{w}_k) \\ &= \boldsymbol{\Phi}_k \, \tilde{\boldsymbol{x}}_k - \boldsymbol{w}_k \end{aligned} \tag{4.11}$$

Substituting equation 4.11 into the state error covariance definition (equation 4.9) gives

$$\begin{aligned} \boldsymbol{P}_{k+1} &= E\Big[\, \tilde{\boldsymbol{x}}_{k+1} \, \tilde{\boldsymbol{x}}_{k+1}^T \Big] \\ &= E\Big[\Big(\boldsymbol{\Phi}_k \, \tilde{\boldsymbol{x}}_k - \boldsymbol{w}_k\Big)\Big(\boldsymbol{\Phi}_k \, \tilde{\boldsymbol{x}}_k - \boldsymbol{w}_k\Big)^T \Big] \\ &= E\Big[\, \boldsymbol{\Phi}_k \, \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^T \boldsymbol{\Phi}_k^T - \boldsymbol{w}_k \, \tilde{\boldsymbol{x}}_k^T \boldsymbol{\Phi}_k^T - \boldsymbol{\Phi}_k \, \tilde{\boldsymbol{x}}_k \, \boldsymbol{w}_k^T + \boldsymbol{w}_k \, \boldsymbol{w}_k^T \Big] \end{aligned} \tag{4.12}$$

Since there is no cross-correlation between the state error and process noise, the two inner terms of the expectation go to zero leaving

$$\boldsymbol{P}_{k+1} = \boldsymbol{\Phi}_k \, \boldsymbol{P}_k \, \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \tag{4.13}$$

as the state error covariance matrix time propagation formulation.

4.1.3 Measurement Update Equations

State Update

The best state estimate update after a discrete measurement is some combination of the previous state and the new measurement

$$\hat{\boldsymbol{x}}_{k_+} = \boldsymbol{K}_k' \, \hat{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k \, \boldsymbol{z}_k \tag{4.14}$$

In this formulation $\boldsymbol{K}_k'$ and $\boldsymbol{K}_k$ are some unknown weighting factors, $\hat{\boldsymbol{x}}_{k_-}$ and $\hat{\boldsymbol{x}}_{k_+}$ are the states before and after the measurement update, respectively, and $\boldsymbol{z}_k$ is the current discrete measurement. Substituting the state error definition (equation 4.8) and measurement (equation 4.2) into equation 4.14 yields

$$\boldsymbol{x}_k + \tilde{\boldsymbol{x}}_{k_+} = \boldsymbol{K}_k' \Big(\boldsymbol{x}_k + \tilde{\boldsymbol{x}}_{k_-} \Big) + \boldsymbol{K}_k \Big(\boldsymbol{H}_k \, \boldsymbol{x}_k + \boldsymbol{v}_k \Big) \tag{4.15}$$

Rearranging equation 4.15 for the post-update state error, $\tilde{\boldsymbol{x}}_{k_+}$, gives

$$\tilde{\boldsymbol{x}}_{k_+} = \Big(\boldsymbol{K}_k' + \boldsymbol{K}_k \, \boldsymbol{H}_k - \boldsymbol{I} \Big) \boldsymbol{x}_k + \boldsymbol{K}_k' \, \tilde{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k \, \boldsymbol{v}_k \tag{4.16}$$

For an unbiased estimator, the expectation of the state error is zero. Therefore, the expectation of equation 4.16 requires that

$$\boldsymbol{K}_k' + \boldsymbol{K}_k \, \boldsymbol{H}_k - \boldsymbol{I} = 0 \tag{4.17}$$

Using this relationship between the two unknown weighting factors and substituting it into equation 4.14 gives the state error measurement update formulation as

$$\begin{aligned}\hat{\boldsymbol{x}}_{k_+} &= \Big(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\Big)\hat{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k\,\boldsymbol{z}_k \\ &= \hat{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k\Big(\boldsymbol{z}_k - \boldsymbol{H}_k\,\hat{\boldsymbol{x}}_{k_-}\Big)\end{aligned} \tag{4.18}$$

Covariance Update

The process of updating the state error covariance matrix after a discrete measurement begins by substituting equations 4.2 and 4.8 into equation 4.18.

$$\tilde{\boldsymbol{x}}_{k_+} = \Big(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\Big)\tilde{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k\,\boldsymbol{v}_k \tag{4.19}$$

From the state error covariance definition (equation 4.9), the pre- and post-update covariance matrices are defined as

$$\boldsymbol{P}_{k_+} = E\Big[\,\tilde{\boldsymbol{x}}_{k_+}\tilde{\boldsymbol{x}}_{k_+}^T\,\Big] \tag{4.20}$$

$$\boldsymbol{P}_{k_-} = E\Big[\,\tilde{\boldsymbol{x}}_{k_-}\tilde{\boldsymbol{x}}_{k_-}^T\,\Big] \tag{4.21}$$

Substituting equation 4.19 into 4.20 gives

$$\boldsymbol{P}_{k_+} = E\Big[\,\Big((\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k)\tilde{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k\,\boldsymbol{v}_k\Big)\Big(\tilde{\boldsymbol{x}}_{k_-}^T\,(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k)^T + \boldsymbol{v}_k^T\,\boldsymbol{K}_k^T\,\Big)\Big] \tag{4.22}$$

Expanding equation 4.22 gives

$$\begin{aligned}\boldsymbol{P}_{k_+} = E\Big[\,&\Big(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\Big)\,\tilde{\boldsymbol{x}}_{k_-}\,\Big(\tilde{\boldsymbol{x}}_{k_-}^T\,(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k)^T + \boldsymbol{v}_k^T\,\boldsymbol{K}_k^T\,\Big) \\ &+ \boldsymbol{K}_k\;\boldsymbol{v}_k\;\Big(\tilde{\boldsymbol{x}}_{k_-}^T\,(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k)^T + \boldsymbol{v}_k^T\,\boldsymbol{K}_k^T\,\Big)\Big]\end{aligned} \tag{4.23}$$

$$\begin{aligned}\boldsymbol{P}_{k_+} = E\Big[&\Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big) \tilde{\boldsymbol{x}}_{k_-} \tilde{\boldsymbol{x}}_{k_-}^T \Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big)^T \\ &+ \Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big) \tilde{\boldsymbol{x}}_{k_-} \boldsymbol{v}_k^T \boldsymbol{K}_k^T \\ &\qquad + \boldsymbol{K}_k \boldsymbol{v}_k \tilde{\boldsymbol{x}}_{k_-}^T \Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big)^T \\ &\qquad + \boldsymbol{K}_k \boldsymbol{v}_k \boldsymbol{v}_k^T \boldsymbol{K}_k^T \Big]\end{aligned} \tag{4.24}$$

Using 4.6, 4.21, and the lack of cross-correlation between the state error and measurement noise

$$E\Big[\tilde{\boldsymbol{x}}_{k_-} \boldsymbol{v}_k^T \Big] = E\Big[\boldsymbol{v}_k \tilde{\boldsymbol{x}}_{k_-}^T \Big] = 0$$

allows equation 4.24 to be simplified into the state error covariance matrix equation used for measurement updates

$$\boldsymbol{P}_{k_+} = \Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big) \boldsymbol{P}_{k_-} \Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T \tag{4.25}$$

Optimal Gain

The gain, $\boldsymbol{K}_k$, is sought to minimize the sum of the mean-square state errors. To perform this minimization, we first define the cost function, J_k, as the trace of the error covariance matrix.

$$J_k = tr\Big[\boldsymbol{P}_{k_+} \Big]$$

The minimum is found by taking the partial derivative of the cost function with respect to the gain and setting this to zero.

$$\frac{\partial}{\partial \boldsymbol{K}_k} \Big(tr\Big[\boldsymbol{P}_{k_+} \Big] \Big) = 0 \tag{4.26}$$

Substituting equation 4.25 into 4.26 produces

$$\frac{\partial}{\partial \boldsymbol{K}_k} \Big(tr\Big[\Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big) \boldsymbol{P}_{k_-} \Big(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\Big)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T \Big] \Big) = 0 \tag{4.27}$$

Expanding the state error covariance matrix equation within equation 4.27 gives

$$\frac{\partial}{\partial \boldsymbol{K}_k}\left(tr\Big[\boldsymbol{P}_{k\_} - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k\_} - \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T \boldsymbol{K}_k^T \right.$$
$$\left. + \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T \boldsymbol{K}_k^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T \Big] \right) = 0 \tag{4.28}$$

Equation 4.28 can be further simplified using the following *trace* identities:

$$tr\Big[\boldsymbol{A} \Big] = tr\Big[\boldsymbol{A}^T \Big] \tag{4.29a}$$

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{A}} \left(tr\Big[\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}^T \Big] \right) &= \boldsymbol{A}(\boldsymbol{B} + \boldsymbol{B}^T) \\ &= 2\boldsymbol{A}\boldsymbol{B} \qquad (\text{for symmetric } \boldsymbol{B}) \end{aligned} \tag{4.29b}$$

$$\frac{\partial}{\partial \boldsymbol{A}} \left(tr\Big[\boldsymbol{I}\boldsymbol{A}\boldsymbol{C} \Big] \right) = \boldsymbol{C}^T \tag{4.29c}$$

Equation 4.29a allows the two middle terms of equation 4.28 to collapse, producing

$$\frac{\partial}{\partial \boldsymbol{K}_k}\left(tr\Big[\boldsymbol{P}_{k\_} - 2\,\boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_{k\_} + \Big(\boldsymbol{K}_k \Big[\boldsymbol{H}_k \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T + \boldsymbol{R}_k \Big] \boldsymbol{K}_k^T \Big) \Big] \right) = 0 \tag{4.30}$$

Application of the remaining trace identities (equations 4.29b and 4.29c) allows for the evaluation of the partial derivative of the trace function.

$$-2 \Big(\boldsymbol{H}_k \boldsymbol{P}_{k\_} \Big)^T + 2\,\boldsymbol{K}_k \Big(\boldsymbol{H}_k \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T + \boldsymbol{R}_k \Big) = 0 \tag{4.31}$$

$$\boldsymbol{K}_k \Big(\boldsymbol{H}_k \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T + \boldsymbol{R}_k \Big) = \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T \tag{4.32}$$

Rearrangement of equation 4.32 provides the optimal Kalman gain matrix.

$$\boldsymbol{K}_k = \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T \Big(\boldsymbol{H}_k \boldsymbol{P}_{k\_} \boldsymbol{H}_k^T + \boldsymbol{R}_k \Big)^{-1} \tag{4.33}$$

4.1.4 Summary

For reference, Table 4.1 lists the fundamental discrete linear Kalman filter time propagation and measurement update equations derived in the previous sections.

Propagate	$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{\Phi}_k \, \hat{\boldsymbol{x}}_k$	(4.10)
	$\boldsymbol{P}_{k+1} = \boldsymbol{\Phi}_k \, \boldsymbol{P}_k \, \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k$	(4.13)
Update	$\boldsymbol{K}_k = \boldsymbol{P}_{k_-} \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \, \boldsymbol{P}_{k_-} \boldsymbol{H}_k^T + \boldsymbol{R}_k \right)^{-1}$	(4.33)
	$\hat{\boldsymbol{x}}_{k_+} = \hat{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k \left(\boldsymbol{z}_k - \boldsymbol{H}_k \, \hat{\boldsymbol{x}}_{k_-} \right)$	(4.18)
	$\boldsymbol{P}_{k_+} = \left(\boldsymbol{I} - \boldsymbol{K}_k \, \boldsymbol{H}_k \right) \boldsymbol{P}_{k_-} \left(\boldsymbol{I} - \boldsymbol{K}_k \, \boldsymbol{H}_k \right)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T$	(4.25)

Table 4.1: Discrete Kalman Filter Equation Summary

The time propagation equations estimate the state vector and state error covariance matrix through periods without measurements. To account for process noise within the state dynamics, the error covariance matrix is made to grow faster by incorporating the process noise covariance matrix, $\boldsymbol{Q}_k$.

For the update equations, the state vector is estimated by combining information of the previous state with information from the current discrete measurement. The weighting between these two quantities is determined by the Kalman gain matrix, $\boldsymbol{K}_k$. This gain matrix can be thought of as a ratio between the state estimate uncertainty and the measurement uncertainty. As confidence in the measurements increase, the Kalman gain increases giving measurements more weight in the state update equation. The converse of this situation also holds true. The state error covariance matrix is updated by incorporating information from the measurement partial derivative matrix and by including measurement noise to cause covariance growth.

4.2 Sensors and Instruments

4.2.1 Inertial Measurement Unit

The inertial measurement unit (IMU) is a standard instrument package used within space navigation systems. A strapdown IMU is used for this analysis. It consists of accelerometers and gyros arranged to measure accelerations and rotation rates in the three body-frame coordinate axes. The two following sections will discuss the derivation of the accelerometer and gyro models as used within the navigation system.

Accelerometers

The model for the accelerometer measurements is derived in Section 3.2.1 and is shown here for reference.

$$\boldsymbol{a}_m^B = \boldsymbol{a}^B + \boldsymbol{\Gamma}_a \boldsymbol{a}^B + \boldsymbol{S}_a \boldsymbol{a}^B + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a \tag{3.15}$$

Within the navigation system, the accelerometer data are used in the time propagation of the vehicle states between periods of non-IMU measurements. Since the navigation filter is estimating the accelerometer error terms, these quantities are used in conjunction with the measured accelerometer acceleration to produce the best possible estimate of true acceleration, $\hat{\boldsymbol{a}}^B$.

First, equation 3.15 must be solved for the true acceleration, $\boldsymbol{a}^B$. To solve for this, the misalignment, nonorthogonality, and scale factors are grouped into the term Δ_a.

$$\begin{aligned} \Delta_a &= \boldsymbol{\Gamma}_a + \boldsymbol{S}_a \\ \boldsymbol{a}_m^B &= (\boldsymbol{I} + \Delta_a)\boldsymbol{a}^B + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a \end{aligned}$$

For small Δ_a the inverse of the $(\boldsymbol{I} + \Delta_a)$ is

$$(\boldsymbol{I} + \Delta_a)^{-1} \approx (\boldsymbol{I} - \Delta_a)$$

This is shown by the following brief derivation where second order terms have been neglected.

$$\begin{aligned}(\boldsymbol{I}+\Delta_a)^{-1} &= (\boldsymbol{I}+\boldsymbol{D})\\ (\boldsymbol{I}+\Delta_a)(\boldsymbol{I}+\boldsymbol{D}) &= \boldsymbol{I}\\ \boldsymbol{I}+\Delta_a+\boldsymbol{D} &\approx \boldsymbol{I}\\ \boldsymbol{D} &\approx -\Delta_a\end{aligned}$$

Therefore, true acceleration is found to be

$$\begin{aligned}\boldsymbol{a}^B &= (\boldsymbol{I}+\Delta_a)^{-1}(\boldsymbol{a}_m^B-\boldsymbol{b}_a-\boldsymbol{\epsilon}_a)\\ &= (\boldsymbol{I}-\Delta_a)\,(\boldsymbol{a}_m^B-\boldsymbol{b}_a-\boldsymbol{\epsilon}_a)\\ &= (\boldsymbol{I}-\Delta_a)\,\boldsymbol{a}_m^B-\boldsymbol{b}_a-\boldsymbol{\epsilon}_a\\ &= \boldsymbol{a}_m^B-\boldsymbol{\Gamma}_a\boldsymbol{a}_m^B-\boldsymbol{S}_a\boldsymbol{a}_m^B-\boldsymbol{b}_a-\boldsymbol{\epsilon}_a\end{aligned} \tag{4.34}$$

again neglecting second order terms. For the navigation filter to estimate the error terms within the accelerometer model, they all must be represented as vectors. In this case, equation 4.34 can then be rewritten as

$$\boldsymbol{a}^B = \boldsymbol{a}_m^B-\boldsymbol{F}_a\boldsymbol{\gamma}_a-\boldsymbol{D}_a\boldsymbol{s}_a-\boldsymbol{b}_a-\boldsymbol{\epsilon}_a \tag{4.35}$$

where the alignment and scale factor matrices ($\boldsymbol{\Gamma}_a$ and $\boldsymbol{S}_a$) are converted into vectors ($\boldsymbol{\gamma}_a$ and $\boldsymbol{s}_a$), and the measured acceleration ($\boldsymbol{a}_m^B$) is represented as matrices $\boldsymbol{F}_a$ and $\boldsymbol{D}_a$.

$$\begin{aligned}\boldsymbol{\gamma}_a &= \left(\gamma_{a_{xy}}\,\gamma_{a_{xz}}\,\gamma_{a_{yx}}\,\gamma_{a_{yz}}\,\gamma_{a_{zx}}\,\gamma_{a_{zy}}\right)^T\\ \boldsymbol{s}_a &= \left(s_{a_x}\,s_{a_y}\,s_{a_z}\right)^T\end{aligned}$$

$$\boldsymbol{F}_a = \boldsymbol{F}(\boldsymbol{a}^B_m) = \begin{bmatrix} -a_{m_z} & a_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{m_z} & -a_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{m_y} & a_{m_x} \end{bmatrix}$$

$$\boldsymbol{D}_a = \boldsymbol{D}(\boldsymbol{a}^B_m) = \begin{bmatrix} a_{m_x} & 0 & 0 \\ 0 & a_{m_y} & 0 \\ 0 & 0 & a_{m_z} \end{bmatrix}$$

The best possible estimate of true acceleration, $\boldsymbol{a}^B$, is then obtained by taking the expectation of equation 4.35. Since the expectation of the noise term is zero ($E(\boldsymbol{\epsilon}_a) = 0$), the best estimate of the acceleration within the navigation filter is

$$\hat{\boldsymbol{a}}^B = \boldsymbol{a}^B_m - \boldsymbol{F}_a\hat{\boldsymbol{\gamma}}_a - \boldsymbol{D}_a\hat{\boldsymbol{s}}_a - \hat{\boldsymbol{b}}_a \tag{4.36}$$

Gyros

The model for the gyro measurements is derived in Section 3.2.2 and shown here for reference.

$$\boldsymbol{\omega}^B_m = \boldsymbol{\omega}^B + \boldsymbol{\Gamma}_g\boldsymbol{\omega}^B + \boldsymbol{S}_g\boldsymbol{\omega}^B + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g \tag{3.18}$$

Within the navigation system, the gyro data are used in the time propagation of the vehicle states between periods of non-IMU measurements. Since the navigation filter is estimating the gyro error terms, these quantities are used in conjunction with the measured gyro rotation rates to produce the best possible estimate of true rotation rates, $\hat{\boldsymbol{\omega}}^B$.

First, equation 3.18 must be solved for the true rotation rate, $\boldsymbol{\omega}^B$. To solve for this, the nonorthogonality and scale factors are grouped into the term Δ_g.

$$\begin{aligned}
\Delta_g &= \boldsymbol{\Gamma}_g + \boldsymbol{S}_g \\
\boldsymbol{\omega}^B_m &= (\boldsymbol{I} + \Delta_g)\boldsymbol{\omega}^B + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g
\end{aligned}$$

For small Δ_g the inverse of the $(\boldsymbol{I} + \Delta_g)$ is

$$(\boldsymbol{I} + \Delta_g)^{-1} \approx (\boldsymbol{I} - \Delta_g)$$

Therefore, true rotation rate is found to be

$$\begin{aligned}
\boldsymbol{\omega}^B &= (\boldsymbol{I} + \Delta_g)^{-1}(\boldsymbol{\omega}^B_m - \boldsymbol{b}_g - \boldsymbol{\epsilon}_g) \\
&= (\boldsymbol{I} - \Delta_g)\,(\boldsymbol{\omega}^B_m - \boldsymbol{b}_g - \boldsymbol{\epsilon}_g) \\
&= (\boldsymbol{I} - \Delta_g)\,\boldsymbol{\omega}^B_m - \boldsymbol{b}_g - \boldsymbol{\epsilon}_g \\
&= \boldsymbol{\omega}^B_m - \boldsymbol{\Gamma}_g\boldsymbol{\omega}^B_m - \boldsymbol{S}_g\boldsymbol{\omega}^B_m - \boldsymbol{b}_g - \boldsymbol{\epsilon}_g
\end{aligned} \tag{4.37}$$

by neglecting second order terms. For the navigation filter to estimate the error terms within the gyro model, they all must be represented as vectors. In this case equation 4.37 can then be rewritten as

$$\boldsymbol{\omega}^B = \boldsymbol{\omega}^B_m - \boldsymbol{F}_g\boldsymbol{\gamma}_g - \boldsymbol{D}_g\boldsymbol{s}_g - \boldsymbol{b}_g - \boldsymbol{\epsilon}_g \tag{4.38}$$

where the nonorthogonality and scale factor matrices ($\boldsymbol{\Gamma}_g$ and $\boldsymbol{S}_g$) are converted into vectors ($\boldsymbol{\gamma}_g$ and $\boldsymbol{s}_g$), and the measured rotation rate ($\boldsymbol{\omega}^B_m$) is represented as matrices $\boldsymbol{F}_g$ and $\boldsymbol{D}_g$.

$$\begin{aligned}
\boldsymbol{\gamma}_g &= \begin{pmatrix}\gamma_{g_{xy}}\,\gamma_{g_{xz}}\,\gamma_{g_{yx}}\,\gamma_{g_{yz}}\,\gamma_{g_{zx}}\,\gamma_{g_{zy}}\end{pmatrix}^T \\
\boldsymbol{s}_g &= \begin{pmatrix}s_{g_x}\,s_{g_y}\,s_{g_z}\end{pmatrix}^T
\end{aligned}$$

$$\boldsymbol{F}_g = \boldsymbol{F}(\boldsymbol{\omega}^B_m) = \begin{bmatrix} -\omega_{m_z} & \omega_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{m_z} & -\omega_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_{m_y} & \omega_{m_x} \end{bmatrix}$$

$$\boldsymbol{D}_g = \boldsymbol{D}(\boldsymbol{\omega}^B_m) = \begin{bmatrix} \omega_{m_x} & 0 & 0 \\ 0 & \omega_{m_y} & 0 \\ 0 & 0 & \omega_{m_z} \end{bmatrix}$$

The best possible estimate of true rotation rate, $\hat{\boldsymbol{\omega}}^B$, is then obtained by taking the expectation of equation 4.38. Since the expectation of the noise term is zero ($E(\boldsymbol{\epsilon}_g) = 0$), the best estimate of the rotation rate within the navigation filter is

$$\hat{\boldsymbol{\omega}}^B = \boldsymbol{\omega}^B_m - \boldsymbol{F}_g \hat{\boldsymbol{\gamma}}_g - \boldsymbol{D}_g \hat{\boldsymbol{s}}_g - \hat{\boldsymbol{b}}_g \tag{4.39}$$

4.2.2 Altimeter

The radar altimeter provides range measurements from the vehicle down to the planet surface. The model for this sensor measurement is derived in Section 3.2.3 and the resulting model equation is shown here for reference.

$$h_m = h\,(1 + k_a) + \nu_a \tag{3.19}$$

For the true altimeter measurement, h varies as a function of vehicle position, vehicle attitude, and planet topography.

$$h = -\left[\boldsymbol{r}^I \cdot \left(\boldsymbol{T}_{B\to I}\boldsymbol{i}^B_a\right)\right] - \sqrt{\left[\boldsymbol{r}^I \cdot \left(\boldsymbol{T}_{B\to I}\boldsymbol{i}^B_a\right)\right]^2 - |\boldsymbol{r}^I|^2 + R_s^2} \tag{3.22}$$

The planet radius, R_s, varies in some fashion according to the local topography, but it is assumed that the navigation filter does not have access to topography data.

To compensate for this difference, the estimated planet radius ($\hat{R}_s$) is treated as a constant and an additional error term ($\hat{x}_t$) is added to the estimated altimeter measurement model. The noise term, ν_a is dropped because its expected value is zero. The estimated altimeter measurement model is then

$$\hat{h}_m = \hat{h}\left(1 + \hat{k}_a\right) + \hat{x}_t \tag{4.40}$$

where the estimate of the true altitude, $\hat{h}$, is

$$\hat{h} = -\left[\hat{\boldsymbol{r}}^I \cdot \left(\hat{\boldsymbol{T}}_{\hat{B}\to I}\boldsymbol{i}_a^B\right)\right] - \sqrt{\left[\hat{\boldsymbol{r}}^I \cdot \left(\hat{\boldsymbol{T}}_{\hat{B}\to I}\boldsymbol{i}_a^B\right)\right]^2 - |\hat{\boldsymbol{r}}^I|^2 + \hat{R}_s^2} \tag{4.41}$$

This new error term, $\hat{x}_t$, is meant to simulate topographic variations and is therefore chosen to be a Markov process. The discrete Markov process dynamics equation is

$$x_{t\,i+1} = e^{\frac{-\Delta t}{\tau}} x_{t\,i} + w_i \tag{4.42}$$

where w_i is the driving white noise term. This bias has a variance based on the statistical altitude variation of the topography and a time constant defined as

$$\tau = \frac{d}{v_{rel}}$$

where d is a reference length of major topographic features and v_{rel} is the surface relative velocity of the vehicle. See Appendix C (page 141) for details on this Markov process and the associated topography statistics.

4.2.3 Velocimeter

The velocimeter provides measurements of the vehicle's velocity by bouncing radar signals off the planet's surface. The model for this sensor measurement is derived in Section 3.2.4 and the resulting model equation is shown here for reference.

$$\boldsymbol{v}_m^B = \boldsymbol{T}_{I\to B}\left(\boldsymbol{v}^I - \boldsymbol{v}_s\right)(1 + k_v) + \boldsymbol{\nu}_v \tag{3.23}$$

The estimated velocimeter measurement is formed by taking the expectation of equation 3.23. Since the expectation of the noise term is zero ($E(\boldsymbol{\nu}_v) = 0$), the best estimate of the surface relative velocity is

$$\hat{\boldsymbol{v}}_m^B = \hat{\boldsymbol{T}}_{I \to \hat{B}} \left(\hat{\boldsymbol{v}}^I - \hat{\boldsymbol{v}}_s\right) \left(1 + \hat{k}_v\right) \tag{4.43}$$

The estimated surface velocity is defined as

$$\hat{\boldsymbol{v}}_s = \boldsymbol{\omega} \times \hat{\boldsymbol{R}}_s \tag{4.44}$$

where $\boldsymbol{\omega}$ is the rotational rate of the planet (see Table 1.2, page 19) and $\hat{\boldsymbol{R}}_s$ is the estimated radius vector of the planet defined as

$$\hat{\boldsymbol{R}}_s = \hat{\boldsymbol{r}}^I + \hat{h} \left(\hat{\boldsymbol{T}}_{\hat{B} \to I} \boldsymbol{i}_a^B\right) \tag{4.45}$$

where $\hat{h}$ is the estimated altitude measurement described in the previous section.

4.3 Filter Design

The filter algorithm used for the ARES entry scenario is an Extended Kalman filter. The extended filter follows the discrete linear algorithm outlined in section 4.1, but models system dynamics for state propagation with non-linear equations and calculates the measurement partial derivative matrices in real-time as functions of the current estimated state.

Generally, the size of the filter state vector and the error state vector used for the covariance are equal in size. For this implementation, there is a vector size difference of one element. This discrepancy resides within the attitude portion of the state. Attitude is generally thought of as three rotation angles. For the error state, this holds true, but for the filter state these three angles are replaced with a four component attitude quaternion. A quaternion still only contains the information of three independent angles, but has the added benefit of avoiding numerical singularities and using less memory for coordinate transformations. Several styles of quaternions exist. The version in use within this analysis is a "right-handed" scalar-last quaternion. A rotation from coordinate frame a to b is defined as

$$\boldsymbol{q}_a^b = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}\sin(\phi/2) \\ \cos(\phi/2) \end{bmatrix} \tag{4.46}$$

where ϕ is the rotation angle from coordinate frame a to b and $\boldsymbol{u}$ is the unit vector defining the plane of rotation [19].

This section provides the specific design details of the Extended Kalman Filter used for this Mars entry navigation problem. First, the non-linear time propagation of the states is derived for all the state sub-vectors. Next, the discrete, linearized state transition matrix and process noise covariance are derived for time propagation of the state error covariance. The design description is concluded with explanations of the measurement update portion of the filter. The last part of this section discusses numerical problems that arise in digital implementation of the Kalman filter and how to correct for them.

4.3.1 State Propagation

One key feature of the Kalman filter algorithm is the propagation of the state vector forward in time. This is accomplished by using reasonable descriptions of the state dynamics to calculate the new value at some later time. The state vector, $\boldsymbol{x}$, contains the principal vehicle states (ie. position, velocity, and attitude vectors) and sensor error states. The state vector is defined to be

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{r}^I & \boldsymbol{v}^I & \boldsymbol{q}_{_{I\rightarrow B}} & \boldsymbol{b}_g & \boldsymbol{s}_g & \boldsymbol{\gamma}_g & \boldsymbol{b}_a & \boldsymbol{s}_a & \boldsymbol{\gamma}_a & x_t & k_a & k_v \end{pmatrix}^T \tag{4.47}$$

where the bold terms within this state vector are subvectors. These subvectors are the position ($\boldsymbol{r}^I$) specified in inertial coordinates, velocity ($\boldsymbol{v}^I$) specified in inertial coordinates, attitude specified as an inertial to body quaternion ($\boldsymbol{q}_{_{I\rightarrow B}}$), gyro error states ($\boldsymbol{b}_g, \boldsymbol{s}_g, \boldsymbol{\gamma}_g$), and accelerometer error states ($\boldsymbol{b}_a, \boldsymbol{s}_a, \boldsymbol{\gamma}_a$). The three remaining terms are scalars representing the altimeter topography error (x_t), altimeter scale factor (k_a), and velocimeter scale factor (k_v). Details of these sensor error states can be found in Sections 3.2 and 4.2.

The state propagation routine involves propagating the best estimate of the state ($\hat{\boldsymbol{x}}$) forward in time. This is defined to be

$$\hat{\boldsymbol{x}} = \begin{pmatrix} \hat{\boldsymbol{r}}^I & \hat{\boldsymbol{v}}^I & \hat{\boldsymbol{q}}_{_{I\rightarrow \hat{B}}} & \hat{\boldsymbol{b}}_g & \hat{\boldsymbol{s}}_g & \hat{\boldsymbol{\gamma}}_g & \hat{\boldsymbol{b}}_a & \hat{\boldsymbol{s}}_a & \hat{\boldsymbol{\gamma}}_a & \hat{x}_t & \hat{k}_a & \hat{k}_v \end{pmatrix}^T$$

Principal State Propagation

The principal states consist of position, velocity, and attitude vectors. The time propagation of these states is accomplished by integrating the vehicle dynamic equations (2.1) and incorporating measured acceleration and rotation rates from the IMU (4.2.1) to predict the state values at some later time. The integration of the dynamics equations is accomplished using the fourth order Runge-Kutta integrator. This algorithm is shown below [7]

$$x_{i+1} = x_i + \tfrac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)\,\Delta t$$

where the "k_i" terms are the time derivatives of the function "x" in the interval and are defined as [2]

$$\begin{aligned}
k_1 &= f(x_i) \\
k_2 &= f\left(\tfrac{1}{2}\Delta t,\ x_i + \tfrac{1}{2}k_1\Delta t\right) \\
k_3 &= f\left(\tfrac{1}{2}\Delta t,\ x_i + \tfrac{1}{2}k_2\Delta t\right) \\
k_4 &= f(\Delta t,\ x_i + k_3 h)
\end{aligned}$$

The time derivatives (slopes) are necessary for this time propagation integration algorithm. The following sections will describe the time derivatives of the position, velocity, and attitude subvectors within the full state.

Position and Velocity Derivative Position ($\hat{\boldsymbol{r}}^{I}$) and velocity ($\hat{\boldsymbol{v}}^{I}$) estimates are propagated using the non-linear gravitational acceleration equations that incorporate the J_2 gravity model ($\boldsymbol{a}_g^I$) and the best estimate non-gravitational accelerations sensed by the accelerometers ($\hat{\boldsymbol{a}}^B$). The time derivative for the position and velocity states are

$$\begin{aligned}
\dot{\hat{\boldsymbol{r}}}^{I} &= \hat{\boldsymbol{v}}^{I} \\
\dot{\hat{\boldsymbol{v}}}^{I} &= \boldsymbol{a}_g^I(x,y,z) \ + \ \hat{\boldsymbol{q}}_{I\rightarrow\hat{B}} \otimes \left[\hat{\boldsymbol{a}}^B(\boldsymbol{a}_m^B, \hat{\boldsymbol{\gamma}}_a, \hat{\boldsymbol{s}}_a, \hat{\boldsymbol{b}}_a)\right] \otimes \hat{\boldsymbol{q}}_{\hat{B}\rightarrow I}
\end{aligned} \tag{4.48}$$

where $\hat{\boldsymbol{q}}_{I\rightarrow\hat{B}}$ is the estimated attitude quaternion state variable used to transform the non-gravitational accelerations into the inertial frame, and "$\otimes$" is the quaternion product operator. These non-gravitational accelerations are functions of the directly measured acceleration ($\boldsymbol{a}_m^B$) and the estimated accelerometer errors ($\hat{\boldsymbol{\gamma}}_a, \hat{\boldsymbol{s}}_a, \hat{\boldsymbol{b}}_a$) tracked within the state vector (See equation 4.36).

---

[2] Sensor information is not available at intermediary time intervals. This results in less accurate integration of the state.

Attitude Derivative For the Runge-Kutta algorithm, the quaternion derivative is calculated using the best estimate attitude rotation rates ($\hat{\boldsymbol{\omega}}^B$) provided by the gyros. These values are the rotation rates of the body frame with respect to the inertial frame and are specified in body coordinates. The derivative needed for the inertial to body quaternion propagation is [19]

$$\begin{aligned}
\dot{\hat{\boldsymbol{q}}}_{I\to\hat{B}} &= \tfrac{1}{2}\,\hat{\boldsymbol{q}}_{I\to\hat{B}} \otimes \hat{\boldsymbol{\omega}}^B \\
&= \tfrac{1}{2}\,\Omega(\hat{\boldsymbol{\omega}}^B)\,\hat{\boldsymbol{q}}_{I\to\hat{B}} \\
&= \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\omega}}^B_{\otimes} & \boldsymbol{\omega}^B \\ (-\boldsymbol{\omega}^B)^T & 0 \end{bmatrix}\hat{\boldsymbol{q}}_{I\to\hat{B}} \\
&= \frac{1}{2}\begin{bmatrix} 0 & \hat{\omega}_z & -\hat{\omega}_y & \hat{\omega}_x \\ -\hat{\omega}_z & 0 & \hat{\omega}_x & \hat{\omega}_y \\ \hat{\omega}_y & -\hat{\omega}_x & 0 & \hat{\omega}_z \\ -\hat{\omega}_x & -\hat{\omega}_y & -\hat{\omega}_z & 0 \end{bmatrix}\hat{\boldsymbol{q}}_{I\to\hat{B}}
\end{aligned} \tag{4.49}$$

where the subscript "$\otimes$" represents the cross product matrix form of a three component vector as shown below.

$$\boldsymbol{u} \times \boldsymbol{v} \;=\; \boldsymbol{U}_{\otimes}\,\boldsymbol{v} \;=\; \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}\boldsymbol{v}$$

The estimated attitude rotation rates are a function of the rates directly measured by the gyros ($\boldsymbol{\omega}^B_m$) and the estimated gyro errors ($\hat{\boldsymbol{\gamma}}_g, \hat{\boldsymbol{s}}_g, \hat{\boldsymbol{b}}_g$) tracked within the state vector (See equation 4.39).

Propagation of Sensor Error States

The remaining state subvectors are the sensor error states. Of these states, all but the altimeter topography error ($\hat{x}_t$) are modeled as constants. The states that are treated as constant over the scenario duration do not require any adjustment during the state time propagation. The altimeter topography error is treated as a first order Markov process. It is propagated forward in time using the estimated version of the relations shown in Section 4.2.2.

$$\hat{x}_{t\,i+1} = e^{\frac{-\Delta t}{\tau}} \hat{x}_{t\,i} \tag{4.42}$$

This bias has a variance based on the statistical altitude variation of the topography and a time constant defined as

$$\tau = \frac{d}{\hat{v}_{rel}}$$

where d is a reference length of major topographic features and $\hat{v}_{rel}$ is the estimated surface relative velocity of the vehicle. See Appendix C (page 141) for details on the this Markov process and the associated topography statistics.

4.3.2 Covariance Propagation

The error covariance matrix ($\boldsymbol{P}_k$) provides a description of the state vector uncertainty over time. The definition is provided here again for reference.

$$\boldsymbol{P}_k = E\left[\delta\boldsymbol{x}_k \, \delta\boldsymbol{x}_k^T \right]$$

This is the covariance of the state error vector. The state error vector ($\delta\boldsymbol{x}$) is defined as

$$\delta\boldsymbol{x} = \hat{\boldsymbol{x}} - \boldsymbol{x} \tag{4.50}$$

$$\delta\boldsymbol{x} = \begin{pmatrix} \delta\boldsymbol{r}^I & \delta\boldsymbol{v}^I & \delta\boldsymbol{\theta}^B & \delta\boldsymbol{b}_g & \delta\boldsymbol{s}_g & \delta\boldsymbol{\gamma}_g & \delta\boldsymbol{b}_a & \delta\boldsymbol{s}_a & \delta\boldsymbol{\gamma}_a & \delta x_t & \delta k_a & \delta k_v \end{pmatrix}^T \tag{4.51}$$

Notice that this state error vector is consistent with the state vector (equation 4.47) in all terms but the third subvector, $\delta\boldsymbol{\theta}^B$. As mentioned previously, this is the attitude error specified using three rotation angles instead of a quaternion. The attitude error is defined as rotation angles from the vehicle body-frame to the estimated body-frame. This vector discrepancy results in a one element size difference between the state vector and state error vector.

Using the dynamic descriptions of the states, the covariance is propagated forward in time. The basic time propagation relation derived in Section 4.1.2 is shown below.

$$\boldsymbol{P}_{k+1} = \boldsymbol{\Phi}_k \, \boldsymbol{P}_k \, \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \tag{4.13}$$

The navigation filter uses this relationship to properly calculate the covariance matrix for the next time step. This process depends on the state transition matrix, $\boldsymbol{\Phi}_k$, and the process noise covariance, $\boldsymbol{Q}_k$. The following sections describe how these two quantities are derived.

State Transition Matrix

The first step in calculating the state transition matrix ($\boldsymbol{\Phi}_k$) is to determine the linearized state error differential equation

$$\delta\dot{\boldsymbol{x}} = \dot{\hat{\boldsymbol{x}}} - \dot{\boldsymbol{x}} \tag{4.52}$$

where $\dot{\hat{\boldsymbol{x}}}$ is the estimated state time derivative and $\dot{\boldsymbol{x}}$ is the true state time derivative. The individual states within that state error vector derivative are shown below.

$$\delta\dot{\boldsymbol{x}} = \begin{pmatrix} \delta\dot{\boldsymbol{r}}^I & \delta\dot{\boldsymbol{v}}^I & \delta\dot{\boldsymbol{\theta}}^B & \delta\dot{\boldsymbol{b}}_g & \delta\dot{\boldsymbol{s}}_g & \delta\dot{\boldsymbol{\gamma}}_g & \delta\dot{\boldsymbol{b}}_a & \delta\dot{\boldsymbol{s}}_a & \delta\dot{\boldsymbol{\gamma}}_a & \delta\dot{x}_t & \delta\dot{k}_a & \delta\dot{k}_v \end{pmatrix}^T$$

The sensor errors ($\delta\boldsymbol{b}_g, \delta\boldsymbol{s}_g, \delta\boldsymbol{\gamma}_g, \delta\boldsymbol{b}_a, \delta\boldsymbol{s}_a, \delta\boldsymbol{\gamma}_a, \delta k_a, \delta k_v$) are assumed to be constant over time, and therefore the associated rates ($\delta\dot{\boldsymbol{b}}_g$, $\delta\dot{\boldsymbol{s}}_g$, $\delta\dot{\boldsymbol{\gamma}}_g$, $\delta\dot{\boldsymbol{b}}_a$, $\delta\dot{\boldsymbol{s}}_a$, $\delta\dot{\boldsymbol{\gamma}}_a$, $\delta\dot{k}_a$, $\delta\dot{k}_v$) are zero. The altimeter topography error (δx_t) is treated as a Markov process and therefore varies in time.

In order to form the state transition matrix, the state error derivatives must be formulated as linear functions of error states. The next sections show the derivation of the state error derivative for these nonconstant states.

Position Error Derivative The linearized error derivative for the position is

$$\begin{aligned} \delta\dot{\boldsymbol{r}}^{I} &= \dot{\hat{\boldsymbol{r}}}^{I} - \dot{\boldsymbol{r}}^{I} \\ &= \hat{\boldsymbol{v}}^{I} - \boldsymbol{v}^{I} \\ &= \delta\boldsymbol{v}^{I} \end{aligned} \tag{4.53}$$

Velocity Error Derivative The linearized error derivative for the velocity is

$$\delta\dot{\boldsymbol{v}}^{I} = \dot{\hat{\boldsymbol{v}}}^{I} - \dot{\boldsymbol{v}}^{I} \tag{4.54}$$

where $\dot{\hat{\boldsymbol{v}}}^{I}$ is the estimated inertial velocity time derivative (similar to equation 4.48) and is defined as

$$\dot{\hat{\boldsymbol{v}}}^{I} = \boldsymbol{g}(\hat{\boldsymbol{r}}^{I}) + \Big[\boldsymbol{T}(\hat{\boldsymbol{q}}_{\hat{B}\to I})\Big]\ \hat{\boldsymbol{a}}^{B}(\boldsymbol{a}_{m}^{B}, \hat{\boldsymbol{s}}_{a}, \hat{\boldsymbol{\gamma}}_{a}, \hat{\boldsymbol{b}}_{a}) \tag{4.55}$$

This equation assumes a point-mass gravity model evaluated at the estimated inertial position plus the estimated non-gravitational acceleration rotated from the body-frame into the inertial-frame using the estimated body-to-inertial quaternion. The true inertial velocity time derivative, $\dot{\boldsymbol{v}}^{I}$, is modeled as

$$\begin{aligned} \dot{\boldsymbol{v}}^{I} &= \boldsymbol{g}^{I} + \boldsymbol{a}^{I} + \boldsymbol{\epsilon}_{accel} \\ &= \boldsymbol{g}(\boldsymbol{r}^{I}) + \Big[\boldsymbol{T}(\boldsymbol{q}_{B\to I})\Big]\ \boldsymbol{a}^{B}(\boldsymbol{a}_{m}^{B}, \boldsymbol{s}_{a}, \boldsymbol{\gamma}_{a}, \boldsymbol{b}_{a}) + \boldsymbol{\epsilon}_{accel} \end{aligned} \tag{4.56}$$

which also uses a point-mass gravity model evaluated at the true inertial position vector plus the true non-gravitational acceleration rotated from the body-frame to the inertial-frame using the true body-to-inertial quaternion. The term $\boldsymbol{\epsilon}_{accel}$ is a white noise term added to compensate for any unmodeled accelerations (ie. higher order gravity dynamics and aerodynamics). Its covariance is defined as

$$E\Big[\boldsymbol{\epsilon}_{accel}\ \boldsymbol{\epsilon}_{accel}^{T}\Big] = \boldsymbol{Q}_{accel}\,\delta(t) \tag{4.57}$$

where $\boldsymbol{Q}_{accel}$ is the process noise intensity for the unmodeled accelerations. The $\delta(t)$ term is the Dirac delta function.

Substituting equation 4.55 and 4.56 into 4.54 results in

$$\delta \dot{\boldsymbol{v}}^I = \left[\boldsymbol{g}(\hat{\boldsymbol{r}}^I) - \boldsymbol{g}(\boldsymbol{r}^I)\right] + \left[\hat{\boldsymbol{T}}_{\hat{B}\to I}\hat{\boldsymbol{a}}^B - \boldsymbol{T}_{B\to I}\,\boldsymbol{a}^B\right] - \boldsymbol{\epsilon}_{accel} \tag{4.58}$$

The function notation has been removed in the transformation matrix and acceleration vector for clarity. Using the definition of the position vector error, ($\delta\boldsymbol{r}^I = \hat{\boldsymbol{r}}^I - \boldsymbol{r}^I$), the gravity vector can be simplified using a first-order approximation.

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{r}^I) &= \boldsymbol{g}(\hat{\boldsymbol{r}}^I - \delta\boldsymbol{r}^I) \\ &= \boldsymbol{g}(\hat{\boldsymbol{r}}^I) - \left(\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}^I}\right|_{\hat{\boldsymbol{r}}^I}\right)\delta\boldsymbol{r}^I \end{aligned} \tag{4.59}$$

$$\boldsymbol{g}(\hat{\boldsymbol{r}}^I) - \boldsymbol{g}(\boldsymbol{r}^I) = \left(\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}^I}\right|_{\hat{\boldsymbol{r}}^I}\right)\delta\boldsymbol{r}^I \tag{4.60}$$

Substituting equation 4.60 and the equations for the estimated and true accelerometer model (equations 3.15 and 4.36) into equation 4.58 gives

$$\begin{aligned} \delta\dot{\boldsymbol{v}}^I = \left(\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}^I}\right|_{\hat{\boldsymbol{r}}^I}\right)\delta\boldsymbol{r}^I &+ \hat{\boldsymbol{T}}_{\hat{B}\to I}\left[\boldsymbol{a}^B_m - \boldsymbol{D}_a\hat{\boldsymbol{s}}_a - \boldsymbol{F}_a\hat{\boldsymbol{\gamma}}_a - \hat{\boldsymbol{b}}_a\right] \\ &- \boldsymbol{T}_{B\to I}\left[\boldsymbol{a}^B_m - \boldsymbol{D}_a\boldsymbol{s}_a - \boldsymbol{F}_a\boldsymbol{\gamma}_a - \boldsymbol{b}_a - \boldsymbol{\epsilon}_a\right] \\ &- \boldsymbol{\epsilon}_{accel} \end{aligned} \tag{4.61}$$

To further simplify this equation and render it useful for filter implementation, the right-hand-side of equation 4.61 must be written in terms of error states ($\delta\boldsymbol{r}^I$, $\delta\boldsymbol{v}^I$, $\delta\boldsymbol{\theta}^B$, etc). Noting that

$$
\begin{aligned}
\boldsymbol{T}_{B\to I} &= \hat{\boldsymbol{T}}_{\hat{B}\to I}\boldsymbol{T}_{B\to\hat{B}} \\
&= \hat{\boldsymbol{T}}_{\hat{B}\to I}\begin{bmatrix} 1 & \delta\theta_z & -\delta\theta_y \\ -\delta\theta_z & 1 & \delta\theta_x \\ \delta\theta_y & -\delta\theta_x & 1 \end{bmatrix} \\
&= \hat{\boldsymbol{T}}_{\hat{B}\to I}\left(\boldsymbol{I} - \delta\boldsymbol{\theta}_{\otimes}\right)
\end{aligned}
$$

where $\delta\boldsymbol{\theta}$ is a small angle rotation, and $\delta\boldsymbol{\theta}_{\otimes}$ is the associated cross-product matrix, allows equation 4.61 to be written as

$$
\begin{aligned}
\delta\dot{\boldsymbol{v}}^{I} = & \left(\left.\frac{\partial\boldsymbol{g}}{\partial\boldsymbol{r}^{I}}\right|_{\hat{\boldsymbol{r}}^{I}}\right)\delta\boldsymbol{r}^{I} \\
& + \hat{\boldsymbol{T}}_{\hat{B}\to I}\left[\boldsymbol{D}_a(\boldsymbol{s}_a - \hat{\boldsymbol{s}}_a) + \boldsymbol{F}_a(\boldsymbol{\gamma}_a - \hat{\boldsymbol{\gamma}}_a) + (\boldsymbol{b}_a - \hat{\boldsymbol{b}}_a) + \boldsymbol{\epsilon}_a\right] \\
& + \hat{\boldsymbol{T}}_{\hat{B}\to I}\delta\boldsymbol{\theta}_{\otimes}\boldsymbol{a}^{B}_{m} - \boldsymbol{\epsilon}_{accel}
\end{aligned}
\tag{4.62}
$$

Where all second-order terms have been discarded. Rewriting equation 4.62 in terms of state errors produces

$$
\begin{aligned}
\delta\dot{\boldsymbol{v}}^{I} = & \left(\left.\frac{\partial\boldsymbol{g}}{\partial\boldsymbol{r}^{I}}\right|_{\hat{\boldsymbol{r}}^{I}}\right)\delta\boldsymbol{r}^{I} \\
& - \hat{\boldsymbol{T}}_{\hat{B}\to I}\left[\boldsymbol{D}_a\delta\boldsymbol{s}_a + \boldsymbol{F}_a\delta\boldsymbol{\gamma}_a + \delta\boldsymbol{b}_a - \boldsymbol{\epsilon}_a\right] \\
& - \hat{\boldsymbol{T}}_{\hat{B}\to I}(\boldsymbol{a}^{B}_{m})_{\otimes}\,\delta\boldsymbol{\theta} - \boldsymbol{\epsilon}_{accel}
\end{aligned}
\tag{4.63}
$$

Attitude Error Derivative The time derivative of the body-frame attitude error vector is provided.

$$
\delta\dot{\boldsymbol{\theta}}^{B} = \delta\boldsymbol{\omega}^{B} - \boldsymbol{\omega}^{B}\times\delta\boldsymbol{\theta}^{B} \tag{4.64}
$$

In this equation, $\delta\boldsymbol{\omega}^{B}$ are the unknown gyro rate errors, $\boldsymbol{\omega}^{B}$ are the true angular attitude rates, and $\delta\boldsymbol{\theta}^{B}$ are the attitude angle errors. Appendix B (page 137) provides a detailed derivation of this equation. Substituting equation 3.18 for $\boldsymbol{\omega}^{B}$ and canceling higher-order terms results in

$$\delta\dot{\boldsymbol{\theta}}^B = \delta\boldsymbol{\omega}^B - \underset{m}{\boldsymbol{\omega}}^B \times \delta\boldsymbol{\theta}^B \tag{4.65}$$

where $\underset{m}{\boldsymbol{\omega}}^B$ is the measured angular attitude rate vector provided by the gyros. The gyro rate errors ($\delta\boldsymbol{\omega}^B$) are found by differencing the estimated and true gyro model equations (equations 4.39 and 4.38, respectively).

$$\begin{aligned}
\delta\boldsymbol{\omega}^B &= \hat{\boldsymbol{\omega}}^B - \boldsymbol{\omega}^B \\
&= \left(\underset{m}{\boldsymbol{\omega}}^B - \boldsymbol{D}_g\hat{\boldsymbol{s}}_g - \boldsymbol{F}_g\hat{\boldsymbol{\gamma}}_g - \hat{\boldsymbol{b}}_g\right) \\
&\qquad - \left(\underset{m}{\boldsymbol{\omega}}^B - \boldsymbol{D}_g\boldsymbol{s}_g - \boldsymbol{F}_g\boldsymbol{\gamma}_g - \boldsymbol{b}_g - \boldsymbol{\epsilon}_g\right) \\
&= \boldsymbol{D}_g(\boldsymbol{s}_g - \hat{\boldsymbol{s}}_g) + \boldsymbol{F}_g(\boldsymbol{\gamma}_g - \hat{\boldsymbol{\gamma}}_g) + (\boldsymbol{b}_g - \hat{\boldsymbol{b}}_g) + \boldsymbol{\epsilon}_g \\
&= -\boldsymbol{D}_g\,\delta\boldsymbol{s}_g - \boldsymbol{F}_g\,\delta\boldsymbol{\gamma}_g - \delta\boldsymbol{b}_g + \boldsymbol{\epsilon}_g
\end{aligned} \tag{4.66}$$

Substituting equation 4.66 into equation 4.65 provides the final attitude error derivative needed for the state transition matrix calculation.

$$\delta\dot{\boldsymbol{\theta}}^B = -\boldsymbol{D}_g\,\delta\boldsymbol{s}_g - \boldsymbol{F}_g\,\delta\boldsymbol{\gamma}_g - \delta\boldsymbol{b}_g + \boldsymbol{\epsilon}_g - (\underset{m}{\boldsymbol{\omega}}^B)_{\otimes}\,\delta\boldsymbol{\theta}^B \tag{4.67}$$

Altitude Topography Error Derivative The estimated altimeter model includes a term to account for the planet surface topography variations. This term is described as a Markov process. The error rate is defined as

$$\delta\dot{x}_t = -\tfrac{1}{\tau}\,\delta x_t + w \tag{4.68}$$

The white process noise term (w) has a variance of

$$\begin{aligned}
E\Big[\,w^2\,\Big] &= Q_{alt}\,\delta(t) \\
&= \frac{2\sigma_s^2}{\tau}\,\delta(t)
\end{aligned} \tag{4.69}$$

where Q_{alt} is the process noise intensity, σ_s is the topography standard deviation as described in appendix C, τ is the Markov process time constant, and $\delta(t)$ is the Dirac delta function.

State Dynamics Equation Using the individual state error derivatives previous derived, the full state error derivative vector can now be filled in. To summarize, the entire error vector rate ($\delta\dot{\boldsymbol{x}}$) is defined as

$$\delta\dot{\boldsymbol{x}} = \begin{pmatrix} \delta\dot{\boldsymbol{r}}^{I} & \delta\dot{\boldsymbol{v}}^{I} & \delta\dot{\boldsymbol{\theta}}^{B} & \delta\dot{\boldsymbol{b}}_g & \delta\dot{\boldsymbol{s}}_g & \delta\dot{\boldsymbol{\gamma}}_g & \delta\dot{\boldsymbol{b}}_a & \delta\dot{\boldsymbol{s}}_a & \delta\dot{\boldsymbol{\gamma}}_a & \delta\dot{x}_t & \delta\dot{k}_a & \delta\dot{k}_v \end{pmatrix}^T$$

where the terms within the vector are derived to be

$$\begin{aligned}
\delta\dot{\boldsymbol{r}}^{I} &= \delta\boldsymbol{v}^{I} \\
\delta\dot{\boldsymbol{v}}^{I} &= \left(\left. \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}^{I}} \right|_{\hat{\boldsymbol{r}}^{I}} \right) \delta\boldsymbol{r}^{I} \\
&\quad - \hat{\boldsymbol{T}}_{\hat{B}\to I} \left[\boldsymbol{D}_a \delta\boldsymbol{s}_a + \boldsymbol{F}_a \delta\boldsymbol{\gamma}_a + \delta\boldsymbol{b}_a - \boldsymbol{\epsilon}_a \right] \\
&\quad - \hat{\boldsymbol{T}}_{\hat{B}\to I} (\boldsymbol{a}_m^B)_{\otimes}\, \delta\boldsymbol{\theta} - \boldsymbol{\epsilon}_{accel} \\
\delta\dot{\boldsymbol{\theta}}^{B} &= -\boldsymbol{D}_g\, \delta\boldsymbol{s}_g - \boldsymbol{F}_g\, \delta\boldsymbol{\gamma}_g - \delta\boldsymbol{b}_g + \boldsymbol{\epsilon}_g - (\boldsymbol{\omega}_m^B)_{\otimes}\, \delta\boldsymbol{\theta}^{B} \\
\delta\dot{x}_t &= -\tfrac{1}{\tau}\, \hat{x}_t + w \\
\delta\dot{\boldsymbol{b}}_g &= \delta\dot{\boldsymbol{s}}_g = \delta\dot{\boldsymbol{\gamma}}_g = \delta\dot{\boldsymbol{b}}_a = \delta\dot{\boldsymbol{s}}_a = \delta\dot{\boldsymbol{\gamma}}_a = \delta\dot{k}_a = \delta\dot{k}_v = 0
\end{aligned}$$

The state error vector and the noise terms can be factored out of this state error rate equation allowing it to be rewritten as

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{A}\,\delta\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{\epsilon} \tag{4.70}$$

where $\boldsymbol{A}$ is the state dynamics matrix and $\boldsymbol{B}$ is the coefficient matrix for the white noise vector $\boldsymbol{\epsilon}$. The white noise vector consists of the noise terms from the accelerometer, gyro, unmodeled acceleration, and Markov process and is defined as

$$\boldsymbol{\epsilon} = \begin{pmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_g \\ \boldsymbol{\epsilon}_{accel} \\ w \end{pmatrix} \tag{4.71}$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are shown in Appendix D due to their size.

The state transition matrix is defined as the matrix exponential of the state dynamics matrix [5]

$$\boldsymbol{\Phi}_k \;=\; e^{\boldsymbol{A}\Delta t} \tag{4.72}$$

where Δt is the navigation filter time step. A second-order approximation of the state transition matrix using Taylors series expansion of the matrix exponential produces

$$\boldsymbol{\Phi}_k \;\cong\; \boldsymbol{I} + \boldsymbol{A}\Delta t + \boldsymbol{A}^2\tfrac{\Delta t^2}{2} \tag{4.73}$$

Process Noise

The process noise covariance ($\boldsymbol{Q}_k$) is defined in section 4.1.1 as

$$E\Big[\,\boldsymbol{w}_j\,\boldsymbol{w}_k^T\,\Big] = \left\{\begin{array}{ll} \boldsymbol{Q}_k & j = k \\ 0 & j \neq k \end{array}\right\} \tag{4.4}$$

where $\boldsymbol{w}$ is the state process noise vector. One way to derive the details of the discrete process noise covariance is from the continuous system process noise [6].

$$\boldsymbol{w}(t) = \int_0^{\Delta t} \boldsymbol{\Phi}(\Delta t - \tau)\;\boldsymbol{B}\;\boldsymbol{\epsilon}(\tau)\;d\tau \tag{4.74}$$

where $\boldsymbol{\Phi}$, $\boldsymbol{B}$, and $\boldsymbol{\epsilon}$ are defined above. Using the definition in equation 4.4, the process noise covariance is found to be

$$\begin{aligned}
\boldsymbol{Q}(t) &= E\Big[\,\boldsymbol{w}(t)\,\boldsymbol{w}(t)^T\,\Big] \\
&= E\left[\int_0^{\Delta t}\!\!\int_0^{\Delta t} \boldsymbol{\Phi}(\Delta t-\tau)\;\boldsymbol{B}(\tau)\;\boldsymbol{\epsilon}(\tau)\;\boldsymbol{\epsilon}^T(\tau')\;\boldsymbol{B}^T(\tau')\boldsymbol{\Phi}^T(\Delta t-\tau')\;d\tau\,d\tau'\right] \\
&= \int_0^{\Delta t}\!\!\int_0^{\Delta t} \boldsymbol{\Phi}(\Delta t-\tau)\;\boldsymbol{B}(\tau)\;\;E\Big[\boldsymbol{\epsilon}(\tau)\;\boldsymbol{\epsilon}^T(\tau')\Big]\;\boldsymbol{B}^T(\tau')\boldsymbol{\Phi}^T(\Delta t-\tau')\;d\tau\,d\tau'
\end{aligned} \tag{4.75}$$

Noting that the expectation of the white noise vectors is defined as

$$E\Big[\,\boldsymbol{\epsilon}(\tau)\;\boldsymbol{\epsilon}^T(\tau')\,\Big] = \boldsymbol{Q}_{\boldsymbol{\epsilon}}\;\delta(\tau-\tau') \tag{4.76}$$

allows equation 4.75 to simplify into

$$\boldsymbol{Q}(t) = \int_0^{\Delta t} \boldsymbol{\Phi}(\Delta t-\tau)\boldsymbol{B}(\tau)\;\boldsymbol{Q}_{\boldsymbol{\epsilon}}\,\boldsymbol{B}^T(\tau)\boldsymbol{\Phi}^T(\Delta t-\tau)\;d\tau \tag{4.77}$$

Using a first order approximation of the state transition matrix in equation 4.77

$$\boldsymbol{\Phi}_k\;(\Delta t-\tau) \cong\; \boldsymbol{I} + \boldsymbol{A}(\Delta t-\tau)$$

produces

$$\boldsymbol{Q}(t) = \int_0^{\Delta t} \big[\boldsymbol{I} + \boldsymbol{A}(\Delta t-\tau)\big]\boldsymbol{B}(\tau)\;\boldsymbol{Q}_{\boldsymbol{\epsilon}}\;\boldsymbol{B}^T(\tau)\big[\boldsymbol{I} + \boldsymbol{A}(\Delta t-\tau)\big]^T d\tau \tag{4.78}$$

Expanding the terms in equation 4.78 and evaluating the integral produces the process noise covariance used for this analysis.

$$\begin{aligned}\boldsymbol{Q}_k &= \left(\boldsymbol{B}_k\boldsymbol{Q}_{\boldsymbol{\epsilon}}\boldsymbol{B}_k^T\right)\Delta t \\ &\quad+\left(\boldsymbol{A}_k\boldsymbol{B}_k\boldsymbol{Q}_{\boldsymbol{\epsilon}}\boldsymbol{B}_k^T\right)\frac{\Delta t^2}{2} \\ &\quad+\left(\boldsymbol{B}_k\boldsymbol{Q}_{\boldsymbol{\epsilon}}\boldsymbol{B}_k^T\boldsymbol{A}_k^T\right)\frac{\Delta t^2}{2} \\ &\quad+\left(\boldsymbol{A}_k\boldsymbol{B}_k\boldsymbol{Q}_{\boldsymbol{\epsilon}}\boldsymbol{B}_k^T\boldsymbol{A}_k^T\right)\frac{\Delta t^3}{3}\end{aligned} \tag{4.79}$$

The term $\boldsymbol{Q}_{\boldsymbol{\epsilon}}$ is the process noise intensity and is defined as

$$\boldsymbol{Q}_{\boldsymbol{\epsilon}} = \begin{bmatrix} \boldsymbol{Q}_a & 0 & 0 & 0 \\ 0 & \boldsymbol{Q}_g & 0 & 0 \\ 0 & 0 & \boldsymbol{Q}_{accel} & 0 \\ 0 & 0 & 0 & Q_{alt} \end{bmatrix} \tag{4.80}$$

where $\boldsymbol{Q}_a$ is accelerometer noise variance (Equation 3.14), $\boldsymbol{Q}_g$ is gyro noise variance (Equation 3.17), $\boldsymbol{Q}_{accel}$ is the variance of unmodeled acceleration (Equation 4.57), and Q_{alt} is the variation of terrain uncertainty (Equation 4.69).

4.3.3 Measurement Updates

The power of the recursive Kalman filter derives from the optimal estimation of the state from past data and current sensor measurements. This section discusses the measurement update to the state vector and error covariance matrix. The state vector and error covariance matrix are updated within the Kalman filter algorithm using relationships similar to those as derived in Section 4.1.3. The standard extended Kalman filter version of these equations are

$$\hat{\boldsymbol{x}}_{k_+} = \hat{\boldsymbol{x}}_{k_-} + \boldsymbol{K}_k\left(\boldsymbol{z}_m - \hat{\boldsymbol{z}}_m\right) \tag{4.81}$$

$$\boldsymbol{P}_{k_+} = \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-}\left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k\right)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \tag{4.82}$$

The term $\boldsymbol{K}_k$ is the Kalman gain matrix, $\boldsymbol{z}_m$ is the sensor measurement vector, $\hat{\boldsymbol{z}}_m$ is the estimated sensor measurement vector, $\boldsymbol{H}_k$ is the measurement sensitivity matrix, and $\boldsymbol{R}_k$ is the measurement noise intensity matrix. The $\boldsymbol{K}_k\,(\boldsymbol{z}_m - \hat{\boldsymbol{z}}_m)$ term represents the correction to the state. However, for attitude corrections this is represented by a small angle vector that needs to be applied to the estimated attitude quaternion, $\hat{\boldsymbol{q}}_{I\to\hat{B}}$. This is accomplished using quaternion multiplication.

$$\hat{\boldsymbol{q}}^{+}_{I\to\hat{B}} = \hat{\boldsymbol{q}}^{-}_{I\to\hat{B}} \otimes \boldsymbol{\theta}_{\hat{B}\to B} \tag{4.83}$$

The Kalman gain is used in equation 4.81 to optimally combine the previous state $(\hat{\boldsymbol{x}}_{k_-})$ and the current measurement residual $(\boldsymbol{z}_m - \hat{\boldsymbol{z}}_m)$. The gain is defined as

$$\boldsymbol{K}_k = \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\,\boldsymbol{P}_{k_-}\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)^{-1} \tag{4.33}$$

Altimeter and velocimeter measurements and their estimated values are defined in Section 4.2.

$$\boldsymbol{z}_m = \begin{bmatrix} h_m \\ \boldsymbol{v}_m^B \end{bmatrix} \quad \hat{\boldsymbol{z}}_m = \begin{bmatrix} \hat{h}_m \\ \hat{\boldsymbol{v}}_m^B \end{bmatrix}$$

The measurement sensitivity matrix $(\boldsymbol{H}_k)$ is derived for the specific set of instruments in use by the navigation filter. It is defined as the partial derivative of the measurement value with respect to the state vector. Each row of this matrix corresponds to the partial derivative of a particular instrument measurement, and each column represents the partial derivative with respect to a particular state. The sensitivity matrix is evaluated at each time step requiring a measurement update. The resulting matrix for this implementation of the Kalman filter is

$$\boldsymbol{H} = \begin{bmatrix} \dfrac{\partial \hat{h}_m}{\partial \boldsymbol{x}} \\[2ex] \dfrac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{x}} \end{bmatrix} \tag{4.84}$$

where the partial derivative in respect to the altimeter and velocimeter are derived to be

$$
\begin{aligned}
\frac{\partial \hat{h}_m}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \hat{h}_m}{\partial \boldsymbol{r}^I} & \frac{\partial \hat{h}_m}{\partial \boldsymbol{v}^I} & \frac{\partial \hat{h}_m}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{h}_m}{\partial \boldsymbol{b}_g, \partial \boldsymbol{s}_g, \partial \boldsymbol{\gamma}_g} & \frac{\partial \hat{h}_m}{\partial \boldsymbol{b}_a, \partial \boldsymbol{s}_a, \partial \boldsymbol{\gamma}_a} & \frac{\partial \hat{h}_m}{\partial x_t} & \frac{\partial \hat{h}_m}{\partial k_a} & \frac{\partial \hat{h}_m}{\partial k_v} \end{bmatrix} \\
&= \begin{bmatrix} (1+k_a)\frac{\partial h}{\partial \boldsymbol{r}^I} & 0 & (1+k_a)\frac{\partial h}{\partial \boldsymbol{\theta}} & 0 & 0 & 1 & \hat{h} & 0 \end{bmatrix}
\end{aligned}
\tag{4.85}
$$

$$
\begin{aligned}
\frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{r}^I} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{v}^I} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{b}_g, \partial \boldsymbol{s}_g, \partial \boldsymbol{\gamma}_g} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{b}_a, \partial \boldsymbol{s}_a, \partial \boldsymbol{\gamma}_a} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \hat{x}_t} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial k_a} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial k_v} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{r}^I} & (1+k_v)\boldsymbol{T}_{I\to B} & \frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{\theta}} & 0 & 0 & 0 & 0 & \boldsymbol{T}_{I\to B}(\boldsymbol{v}^I - \boldsymbol{v}_s) \end{bmatrix}
\end{aligned}
\tag{4.86}
$$

and where

$$
\frac{\partial h}{\partial \boldsymbol{r}^I} = \frac{-\left[\,(\boldsymbol{r}^I)^T + \, h\,(\boldsymbol{i}_a^B)^T \boldsymbol{T}_{I\to B}\right]}{(\boldsymbol{r}^I)^T \boldsymbol{T}_{B\to I} \boldsymbol{i}_a^B + \, h}
$$

$$
\frac{\partial h}{\partial \boldsymbol{\theta}} = \frac{-\left[\, h\,(\boldsymbol{r}^I)^T \hat{\boldsymbol{T}}_{\hat{B}\to I} \left(\boldsymbol{i}_a^B\right)_{\otimes}\right]}{(\boldsymbol{r}^I)^T\, \boldsymbol{T}_{B\to I} \boldsymbol{i}_a^B + \, h}
$$

$$
\frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{r}^I} = -(1+k_v)\, \hat{\boldsymbol{T}}_{I\to \hat{B}} \boldsymbol{\omega}_{\otimes} \left[\boldsymbol{I} + \hat{\boldsymbol{T}}_{\hat{B}\to I} \boldsymbol{i}_a^B \frac{\partial \hat{h}}{\partial \boldsymbol{r}^I}\right]
$$

$$
\frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{\theta}} = (1+k_v) \left[\left(\hat{\boldsymbol{T}}_{I\to \hat{B}} \hat{\boldsymbol{v}}_s\right)_{\otimes} - \, \hat{\boldsymbol{T}}_{I\to \hat{B}} \boldsymbol{\omega}_{\otimes} \left(\boldsymbol{T}_{B\to I} \boldsymbol{i}_a^B \, \frac{\partial \hat{h}}{\partial \boldsymbol{\theta}} + \hat{h}\, \hat{\boldsymbol{T}}_{\hat{B}\to I} (\boldsymbol{i}_a^B)_{\otimes}\right)\right]
$$

The derivation of these measurement partial derivatives is provided in Appendix G.

The measurement noise intensity matrix ($\boldsymbol{R}_k$) is defined by the statistical noise behavior of the sensors providing measurements to the navigation filter. It is a diagonal matrix containing the white noise variances for each measurement sensor. The values for the specific altimeter and velocimeter being used are found in Sections 3.2.3

and 3.2.4.

$$\boldsymbol{R} = \begin{bmatrix} \sigma_{z_1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{z_2}^2 & 0 & 0 \\ 0 & 0 & ... & ... \\ 0 & 0 & ... & \sigma_{z_n}^2 \end{bmatrix} \tag{4.87}$$

4.3.4 Filter Numerical Issues

An important numerical issue that often arises in the implementation of the Kalman filter is matrix symmetry. Stable performance of the filter requires that the covariance matrix and process noise matrix remain symmetric throughout the filter operation. Asymmetry can arise from errors caused by numerical accuracy limitations of the computer. To prevent this potential problem, the process noise matrix, post-time-propagation error covariance matrix, and post-update error covariance matrix are symmetrized using

$$\boldsymbol{A} = \frac{\boldsymbol{A} + \boldsymbol{A}^T}{2}$$

Chapter 5

Navigation System Performance Analysis

5.1 Introduction

As the entry vehicle approaches Mars, final navigation measurements are used to initialize the onboard navigation filter. This update is performed using the Deep Space Network (DSN) earth-based antenna network for position and velocity and a cruise stage mounted star camera for attitude. This occurs approximately twelve hours prior to atmospheric entry interface (EI). From this point, the entry vehicle separates from the cruise stage and begins its coast to EI. The entry vehicle navigation filter uses this initial information (ie. position, velocity, and attitude) to propagate the state information through time. No additional external measurements or navigation aids are available during this final cruise phase. The IMU gyros are used during this "dead-reckoning" orbital cruise period to assist in the state estimation. For this analysis, the primary focus is on the atmospheric entry and descent phases of the mission. To produce a realistic entry navigation analysis, though, the change in the state covariance must be understood over this twelve hour cruise. A state and covariance propagation is performed for the twelve hour cruise using only gyros data from the IMU. Accelerometer data is ignored because no significant non-gravitational accelerations are experienced during this period. Processing of the accelerometer data would unnecessarily add noise from this instrument to the state estimation. This 12 hour cruise analysis provides the appropriate navigation information to be used as initial conditions for the atmospheric entry analysis. This pre-entry analysis is presented in Appendix E.

The analysis begins by first presenting the 6-DOF trajectory for the ARES mission (Figure 5-1) and the vehicle dynamics experienced during entry and descent. Next the mission scenario is presented with discussion of the navigation configuration under consideration and the specific analysis approach. Next the analysis for each sensor configuration is presented with observations for that specific scenario. Finally, an overall discussion of the results is presented.

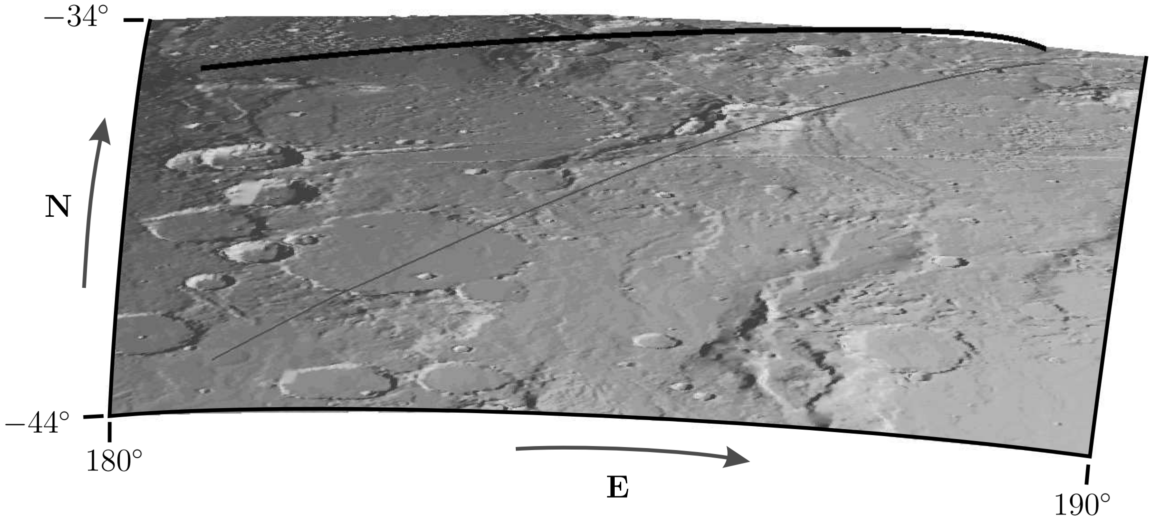


Figure 5-1: ARES Entry & Descent Trajectory

5.2 6-DOF Vehicle Dynamics

5.2.1 Translational Dynamics

The nominal mission scenario begins with the vehicle in its aerodynamically stable orientation spinning about the atmosphere relative velocity vector. Spinning is induced in order to mitigate any aerodynamic lift that might result from uncertainty in the vehicle's center of gravity. Spinning also has the additional benefit of containing attitude uncertainty in two axes orthogonal to the spin axis. The vehicle descends through the atmosphere decelerating with aerodynamic drag. Figure 5-2 shows the inertial velocity and non-gravitational acceleration magnitude profiles for the trajectory. This large rise in acceleration is the aerodynamic drag that results from the vehicle's high velocity and the atmosphere's rapid density increase as the vehicle descends. This effect is shown more directly by the dynamic pressure profile in Figure 5-3.

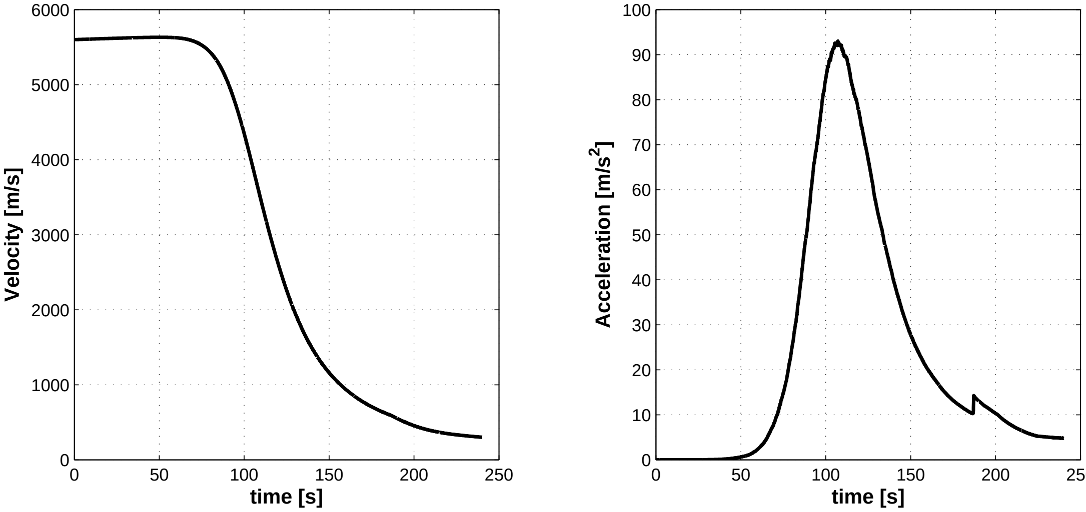


Figure 5-2: Vehicle Translational Dynamics

To further decelerate the vehicle, a supersonic parachute is deployed. This parachute has a nominal, safe deployment condition specified as a dynamic pressure of 540 Pa and 2.1 $Mach$ [13]. This condition is met for the trajectory around 10.2 km altitude. The parachute event is visible as a rapid jump in acceleration in Figure 5-2 near 185 seconds into the atmospheric entry.

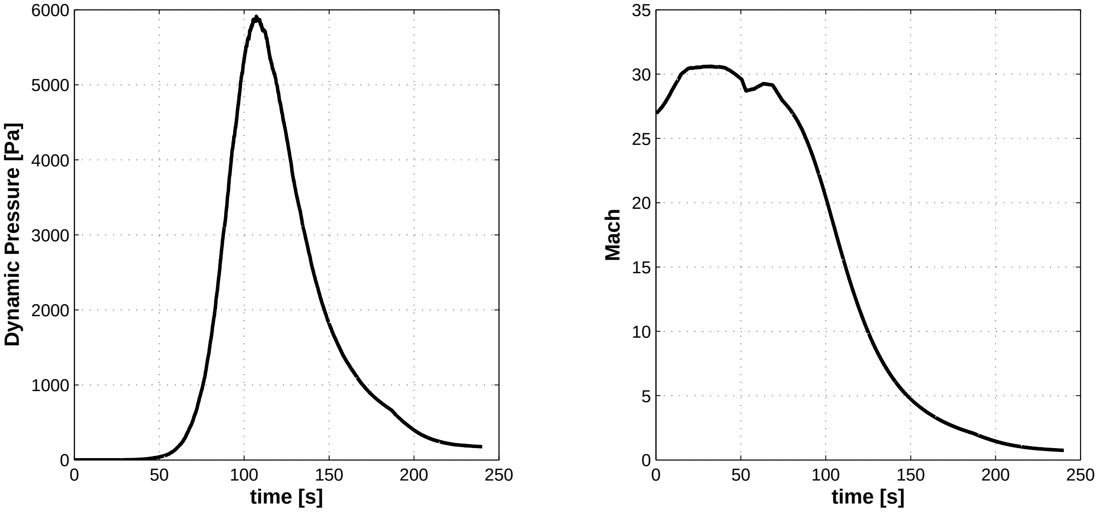


Figure 5-3: Vehicle Aerodynamics

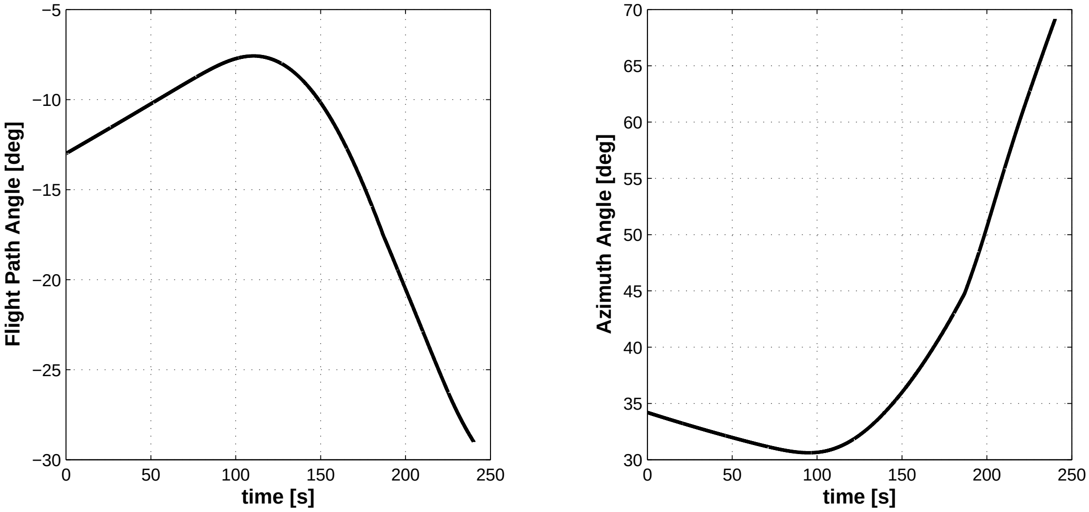


Figure 5-4: Vehicle Velocity Direction

Figure 5-4 shows the vehicle flight path angle and azimuth over time from atmospheric entry interface. Flight path angle is the angle measuring from the local horizontal, up to the vehicle's velocity vector. The ARES vehicle begins with a -13 degree flight path angle, meaning that its velocity is pointed in toward the planet by 13 degrees. This angle initially decreases resulting in a more level flight of the entry vehicle. This is because this first portion of the trajectory is still largely orbital motion uninhibited by the Martian atmosphere. Without any atmospheric interference, this flight path angle would decrease in magnitude toward zero at the orbital perigee. The angle begins to increase in magnitude as atmospheric effects dominates the trajectory and causes rapid deceleration of the vehicle. The vehicle effectively falls into the planet at increasingly steeper angles as it slows down. The azimuth plot shows the direction of the horizontal component of the velocity vector with respect to north. The trend is once again the orbital mechanics dominating the problem early, followed by atmospheric effects later. A slope discontinuity is visible near 185 seconds. This is an effect of parachute deployment.

The vehicle continues its descent until the Mars airplane is released for flight. The nominal airplane flight is at an altitude of 1.5 km. This is chosen as the stop condition for this analysis. Dynamics of the airplane deployment and flight are not considered. Figure 5-5 shows the entry vehicle trajectory plotted as altitude versus the planetary fixed longitude/latitude coordinates. The faint curves in this figure are constant latitude and longitude trajectory projections. This shows that the vehicle is traveling north-east, up from the southern pole.

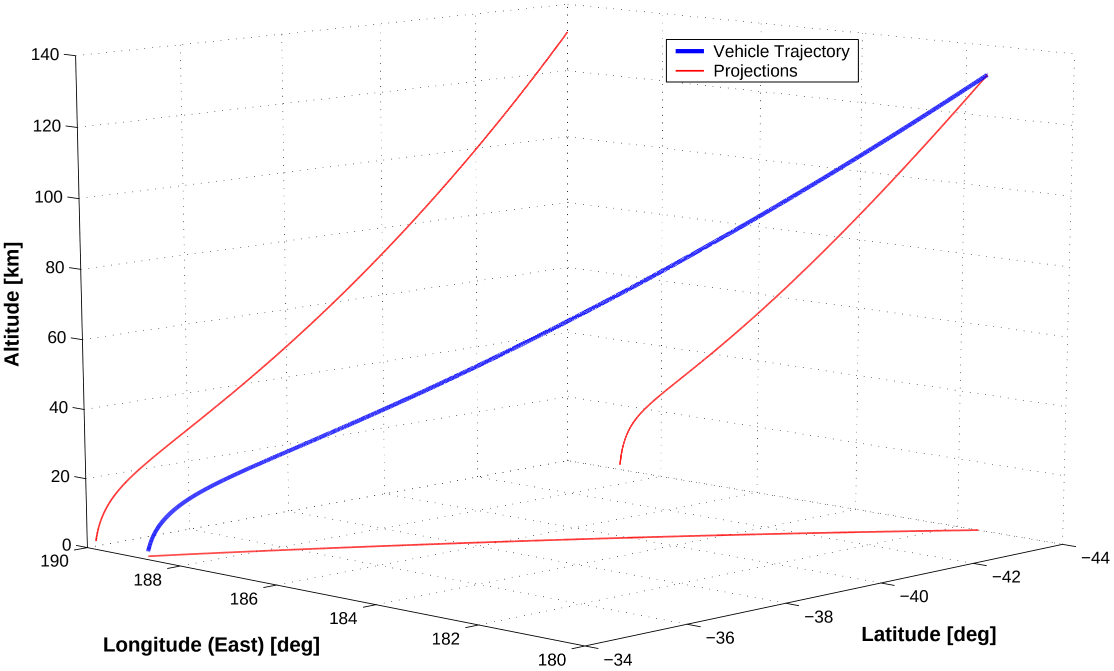


Figure 5-5: Vehicle Trajectory

5.2.2 Rotational Dynamics

The simulation also evaluates the vehicle attitude dynamics through the trajectory. Figure 5-6 shows the vehicle body frame attitude dynamics. Angle of attack represents the angle of the vehicle's body axis from the velocity vector around the pitch axis. Side slip is the same type of measurement, but around the yaw axis. Total angle of attack is the total angle measured from the vehicle velocity vector. Angle of attack and side slip are seen to oscillate around zero, meaning the vehicle is stable and consequently produces no net lift. The oscillations are caused by the aerodynamic torques on the vehicle. They are seen to increase in frequency, but decrease in magnitude at the peak of dynamic pressure. Past this point, the aerodynamic forces and torques decrease, and the oscillations increase in magnitude and decrease in frequency. The rapid angle change is due to parachute deployment. Figure 5-7 shows the body frame rotation rates. The first plot shows that the vehicle is spinning about the roll axis at 1.6 RPM. The trends of the pitch and yaw axis rotation rates match the previously discussed attitude plots prior to parachute deploy. After the parachute is deployed, the rotation rates increase because of the increased drag and lengthened aerodynamic moment arm created by the trailing parachute.

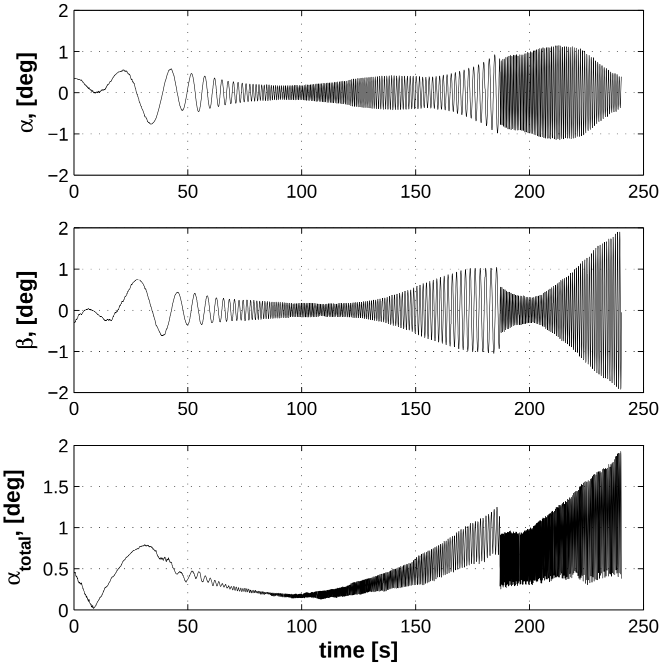


Figure 5-6: Vehicle Attitude

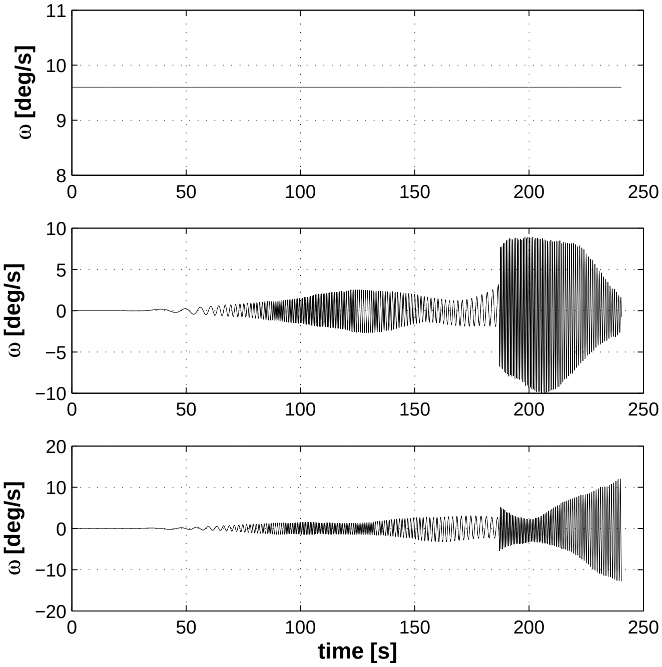


Figure 5-7: Vehicle Angular Rates

5.3 Mission Scenario

The entry and descent phase of this mission begins at entry interface. The initial conditions for the translational and rotational dynamics are presented in table 1.3. Using this, the environment portion of the simulation calculates the vehicle's trajectory over time by incorporating gravity dynamics, atmosphere dynamics, and aerodynamics. To estimate the navigation states, the entry vehicle has use of an IMU and sensor package for measurements.

The nominal navigation instrument scenario consists of the IMU providing acceleration and angular rate data throughout the trajectory, while other sensors provide measurement updates at appropriate times. This analysis examines the performance of two commercially available, spacecraft qualified IMUs used by the navigation filter for state and covariance propagation. These are the Honeywell MIMU and Litton LN200. Performance specifications for these IMUs are found in Tables 3.1 and 3.2. Two commercially available, spacecraft qualified sensor packages also under consideration are those used for the Mars Exploration Rover (MER) and Mars Surveyor Program (MSP) NASA missions. The MER-type sensor is a radar altimeter and the MSP-type sensor is a radar altimeter packaged with a radio Doppler velocimeter. Performance specifications for these sensors are found in Tables 3.3 and 3.4. The four combinations of these instruments (an IMU paired with a sensor) are examined in the analysis.

Monte Carlo analysis is the primary technique used for this navigation performance analysis. For the nominal altimeter/velocimeter activation of $8.75km$ [13], a Monte Carlo analysis is performed by executing the simulation 400 times with the initial principal vehicle states being randomly selected from a normal distribution according to the statistics contained within the initial error covariance. Atmospheric properties are also randomly perturbed for each run. The details of the initial state error covariance for the particular instrument configuration under consideration are provided in Appendix E.

The navigation filter uses these initial conditions to estimate the state and state error covariance through time as the vehicle traverses its entry and descent trajectory. The vehicle states are estimated within the propagation routine of the Extended Kalman filter for the first 200 seconds using only inertial dead reckoning navigation. After 200 seconds, the entry vehicle jettisons the heat shield revealing the altimeter. At this point the altimeter/velocimeter is activated and begins providing measurements to the navigation filter. The simulation concludes near 240 seconds, representing the end of the entry and descent phases of the mission. In addition to this nominal case, the sensors are activated over a range of altitudes to observe the effect this has on the navigation state uncertainty.

The results of the analysis are plots of the principal state estimation errors for each case run and the corresponding state error covariance. The state error covariance is represented with ± 3-σ curves. When the 400 trial runs are bounded between these two curves, this gives confidence that the Kalman filter is tuned properly and that the various assumptions of linearity throughout its design are valid. The position and velocity data is presented in the local vertical-local horizontal coordinate system (LVLH). This coordinate system is oriented with one axis pointing in the planet radial direction (altitude), one axis pointed in the $\vec{r} \times \vec{v}$ direction (cross track), and the final axis completing the triad (downrange). These final two axes define a plane tangent to the planet (local horizontal plane) with the downrange direction in line with the velocity vector and the cross track normal to the velocity vector.

5.4 Scenario 1: MIMU/MER Sensor Suite

For the first scenario, the MIMU is used with the MER-type altimeter. These two instruments are the best possible combination of IMU and altimeter under consideration. Details of the initial state error covariance for this instrument configuration are provided in Appendix E. Figures 5-8 through 5-10 present the principal state errors for 400 cases and the corresponding 3-σ uncertainty level (heavy bounding line) as predicted by the covariance matrix within the Kalman filter. Numerous trends are apparent in these state error/covariance plots. The trends result from both physical dynamics experienced by the entry vehicle and errors within the onboard instruments.

Attitude Errors

For attitude errors (Figure 5-8), the vehicle begins EI with a large roll 3-σ uncertainty of approximately 6.5 degrees, but small pitch and yaw errors of approximately 0.065 degrees. This ”cigar”-shaped initial attitude covariance ellipsoid is due to the IMU gyro errors being incorporated into the error covariance during the 12 hours cruise period. Spinning the vehicle largely inhibits covariance growth in pitch and yaw since the dominant error source, gyro bias, integrates out to zero after each revolution. The small growth from the star camera update at EI-12 hours is a result of gyro noise. The roll axis covariance, though, experiences growth due to gyro bias, scale factors, and noise. The magnitude of the attitude covariance grows very little during atmospheric entry and descent (approximately 250 seconds) since the uncertainty growth is due to the same factors during the 12 hour cruise.

The apparent error growth is primarily due to an increase in the vehicle's flight path angle midway through atmospheric entry flight. This results in the vehicle rotating with respect to the attitude ellipsoid, causing uncertainty within the roll covariance channel to transfer into the pitch and yaw channels. This is seen by the tapering of the roll 3-σ plot and the corresponding growth of the pitch and yaw plot. The oscillating behavior of the pitch and yaw covariance is due to the spin rate of the vehicle. They oscillate between the min and max axes of the attitude

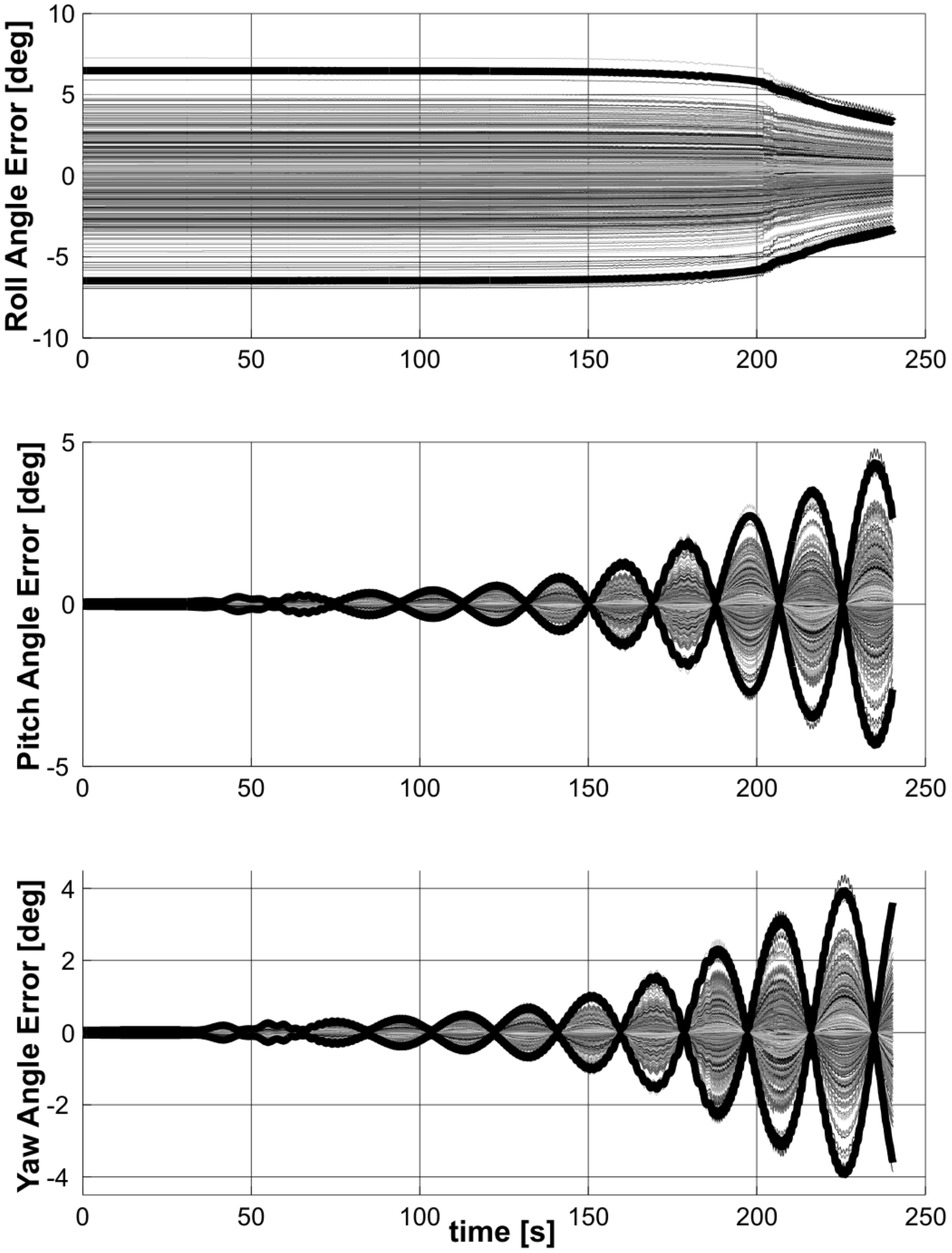


Figure 5-8: MIMU\MER Scenario: Attitude Error Monte Carlo Analysis

error ellipse projected onto the body y-z axes. The pitch and yaw axes spin with respect to this covariance ellipse creating oscillating trends 90 degrees out of phase with one another. Activation of the altimeter at 200 seconds results in small attitude knowledge improvement for the roll axes. The final data trend to notice is the small, high frequency oscillations within the covariance plots. This is the effect of the high frequency attitude vehicle dynamics (Figures 5-6 and 5-7).

Velocity Errors

The velocity errors (Figure 5-9) begin with those estimated from the 12 hour cruise to EI. Velocity is estimated by integration of the acceleration data from the IMU. Attitude estimates must be used to properly interpret the direction of the measured accelerations for the velocity estimation. This creates a coupling of attitude errors with the velocity errors. Growth in the state uncertainty before the altimeter measurement is due to this attitude error coupling and accelerometer errors. The cross track direction of velocity covariance grows rapidly beginning around 100 seconds. This is due to the growth of attitude error with respect to the yaw direction. Activation of the altimeter creates a strong knowledge improvement within the altitude rate channel due to its strong correlation with the altitude position. Downrange and cross track rates are not as correlated and therefore have a slower knowledge improvement from this measurement.

Position Errors

The positions errors in Figure 5-10 begin with those estimated from the 12 hour cruise to EI. Position is estimated by integrating the estimated velocity data. This creates a coupling of the velocity and attitude errors with the position errors. Early in the simulation, the entry vehicle's trajectory is dominated by orbital mechanics. Position error covariance follows distinctive trends for orbital motion. Downrange errors oscillate, but grow with time. This is due to an orbital period error or more fundamentally an orbital energy error. Altitude and cross track errors oscillate around a constant point. This is due to an altitude and orbit inclination error, respectively.

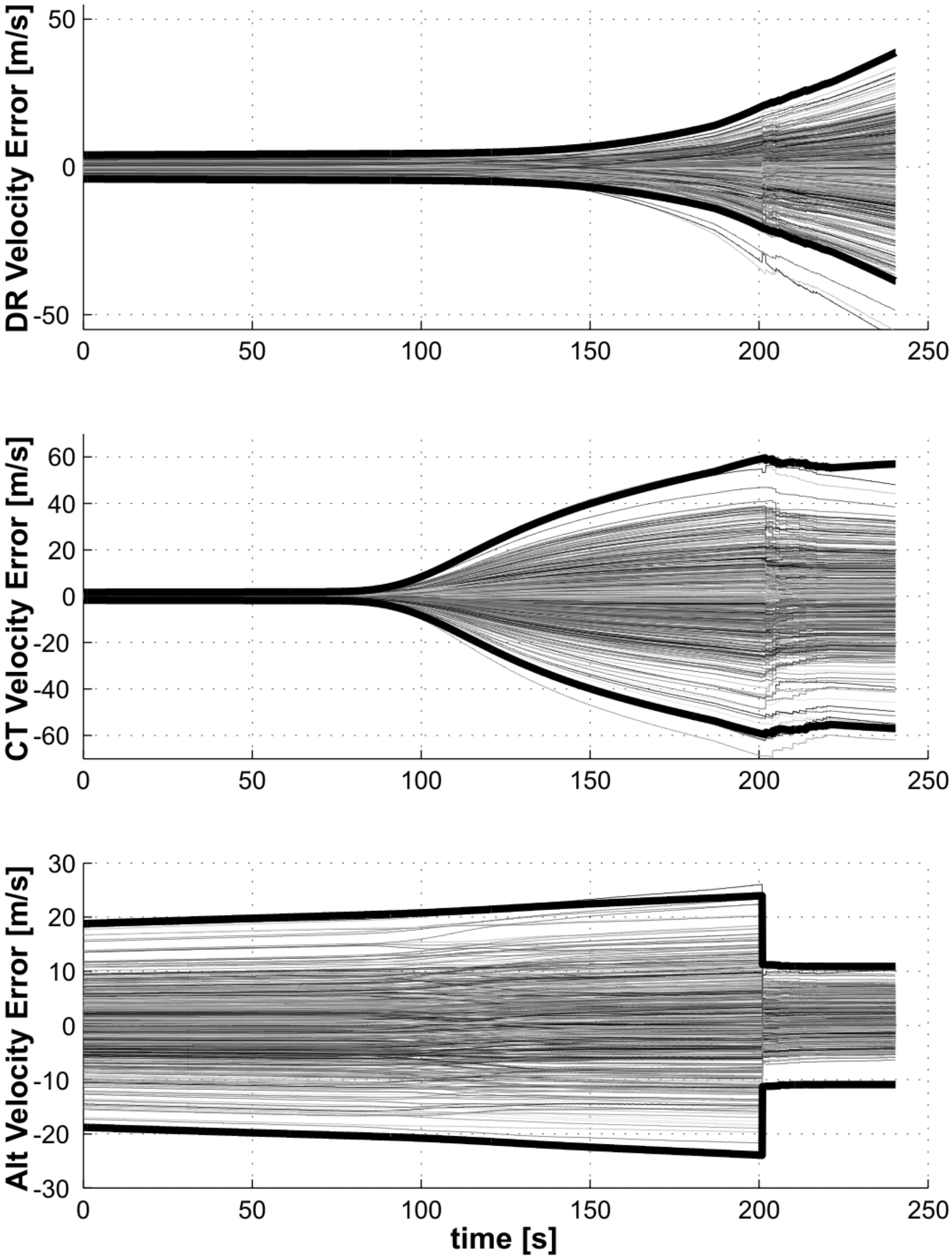


Figure 5-9: MIMU\MER Scenario: Velocity Error Monte Carlo Analysis

Upon activation of the altimeter, the measurement update provides information very strongly into the altitude and downrange state error channels. This sharing of information is due to correlations between altitude and downrange channels. Cross track position error grows predominantly as a result of the cross track velocity error growth. Altimeter measurement information is used to correct other state errors throughout the filter through correlation with downrange and altitude errors.

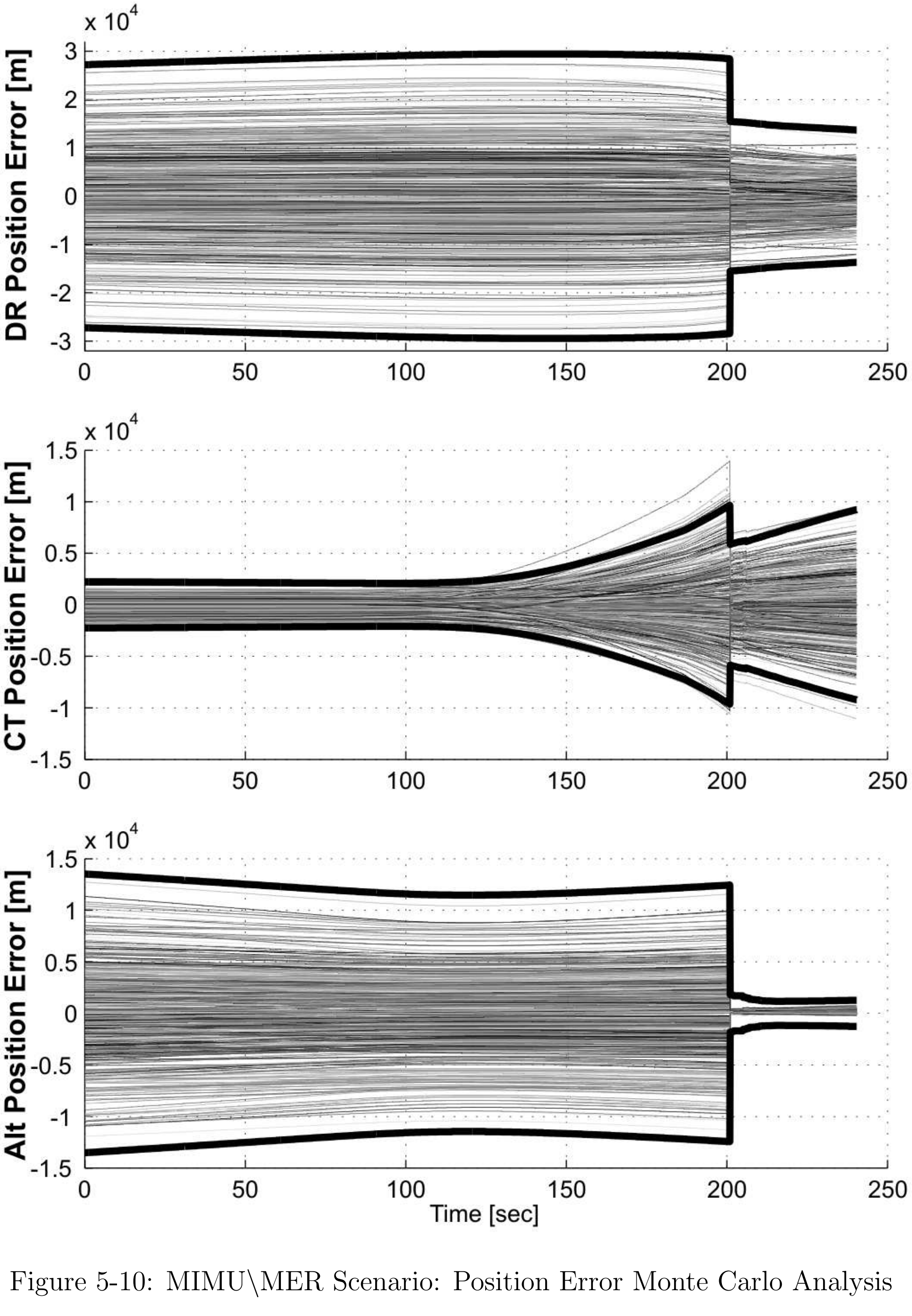


Figure 5-10: MIMU\MER Scenario: Position Error Monte Carlo Analysis

5.4.1 Altitude Activation Study

Improving navigation uncertainty early in the trajectory through measurements can be critical for entry guidance. Also, earlier measurement updates generally result in smaller final landing or delivery errors and less fuel expenditure. To examine the effect of varied measurement activation, the navigation simulation is run for several scenarios where the altimeter is activated at different altitudes. These altitudes are: $16km$, $10.2km$, $8.75km$, and $4km$. The case with no measurement updates (only inertial navigation) is included for comparison. The $8.75km$ case is the nominal scenario for the ARES mission. $10.2km$ corresponds to supersonic parachute deployment. The remaining two altitudes are chosen as extreme high and low cases.

A general explanation of why earlier measurements improve navigation uncertainty is that all the various white noise sources degrade correlations between the states over time. The earlier measurements begin, the more these correlations can be used to estimate navigation filter state errors. For this scenario, increasing the activation point of the altimeter produces relatively small improvements in the final position uncertainty (Figure 5-11). The altitude uncertainty for the low altimeter activation has the largest discrepancy, having approximately twice the uncertainty of the other altitudes.

The velocity error (Figure 5-12) shows much more differentiation among the different altitude cases. These uncertainties decrease rapidly and level off with near-zero or gradually sloped lines. The highest activation altitude case gives the best improvement for downrange and cross track rates and provides this improvement for the remainder of the trajectory. In contrast, the low altitude case provides the least downrange and cross track improvement, but good altitude improvement.

Attitude exhibits similar error covariance improvements to those seen in the velocity state. Early measurements provide significant uncertainty improvement, and late measurements provide very little improvement.

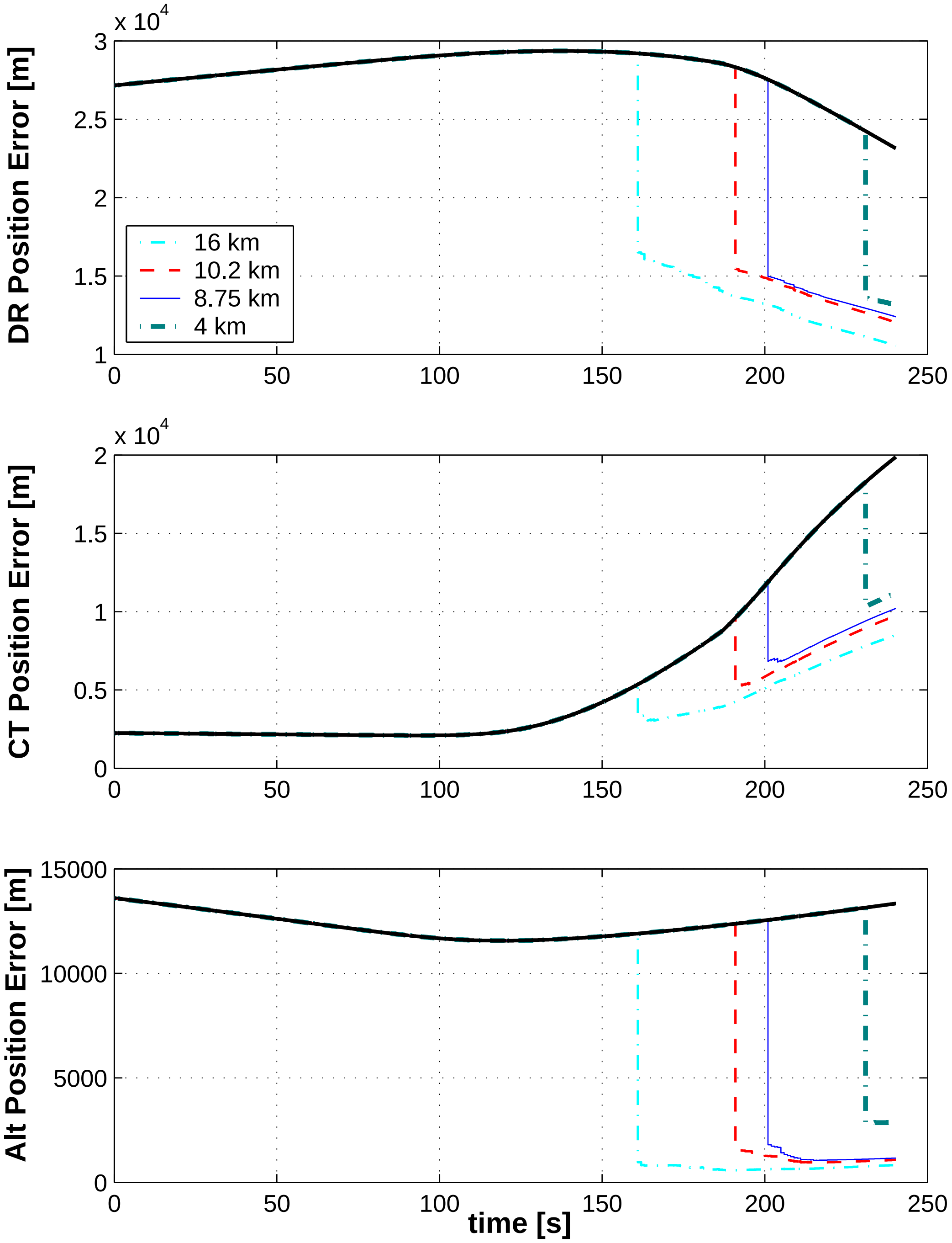


Figure 5-11: MIMU\MER Altitude Study – Position Errors

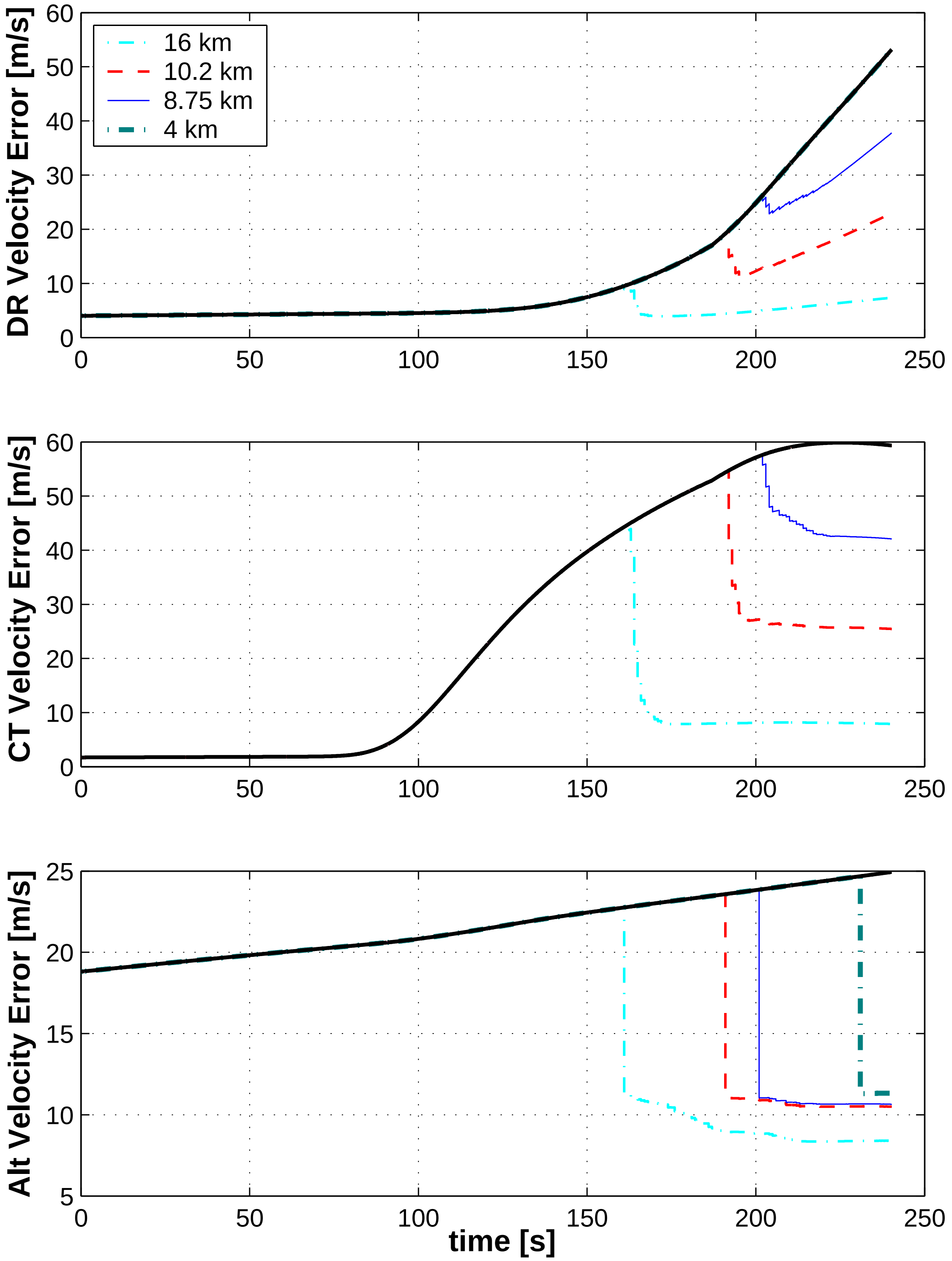


Figure 5-12: MIMU\MER Altitude Study – Velocity Errors

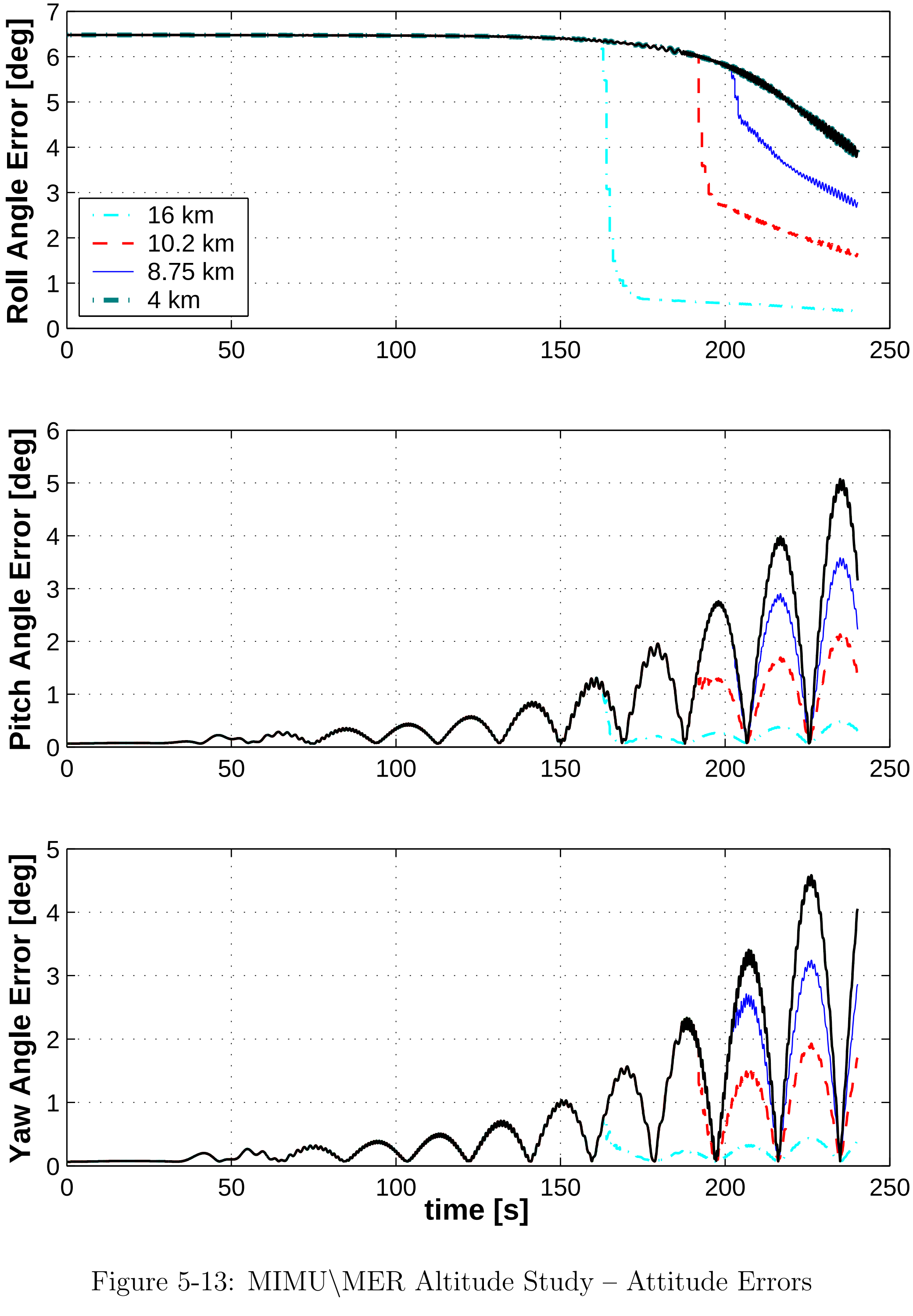


Figure 5-13: MIMU\MER Altitude Study – Attitude Errors

5.5 Scenario 2: MIMU/MSP Sensor Suite

The next scenario under consideration incorporates the MIMU with the MSP-type instrument package. The MSP package includes an altimeter and a velocimeter. This altimeter has a lower level of performance than the one in the previous scenario, but is supplemented with this velocimeter to provide measurement information into the velocity states. Details of the initial state error covariance for this instrument configuration are provided in Appendix E.

Figures 5-14 through 5-16 present the principal state errors for 400 cases and the corresponding 3-σ uncertainty level (heavy bounding line) as predicted by the covariance matrix within the Kalman filter. Because of the large discrepancy between the initial velocity errors and the high precision of the velocimeter, the filter for this scenario is tuned by down-weighting measurements slightly with increased measurement noise. This helps avoid instability that might result from non-linearities in the problem or integration errors present in the filter. Numerous trends are apparent in these state error/covariance plots. The trends result from both physical dynamics experienced by the entry vehicle and errors within the onboard instruments.

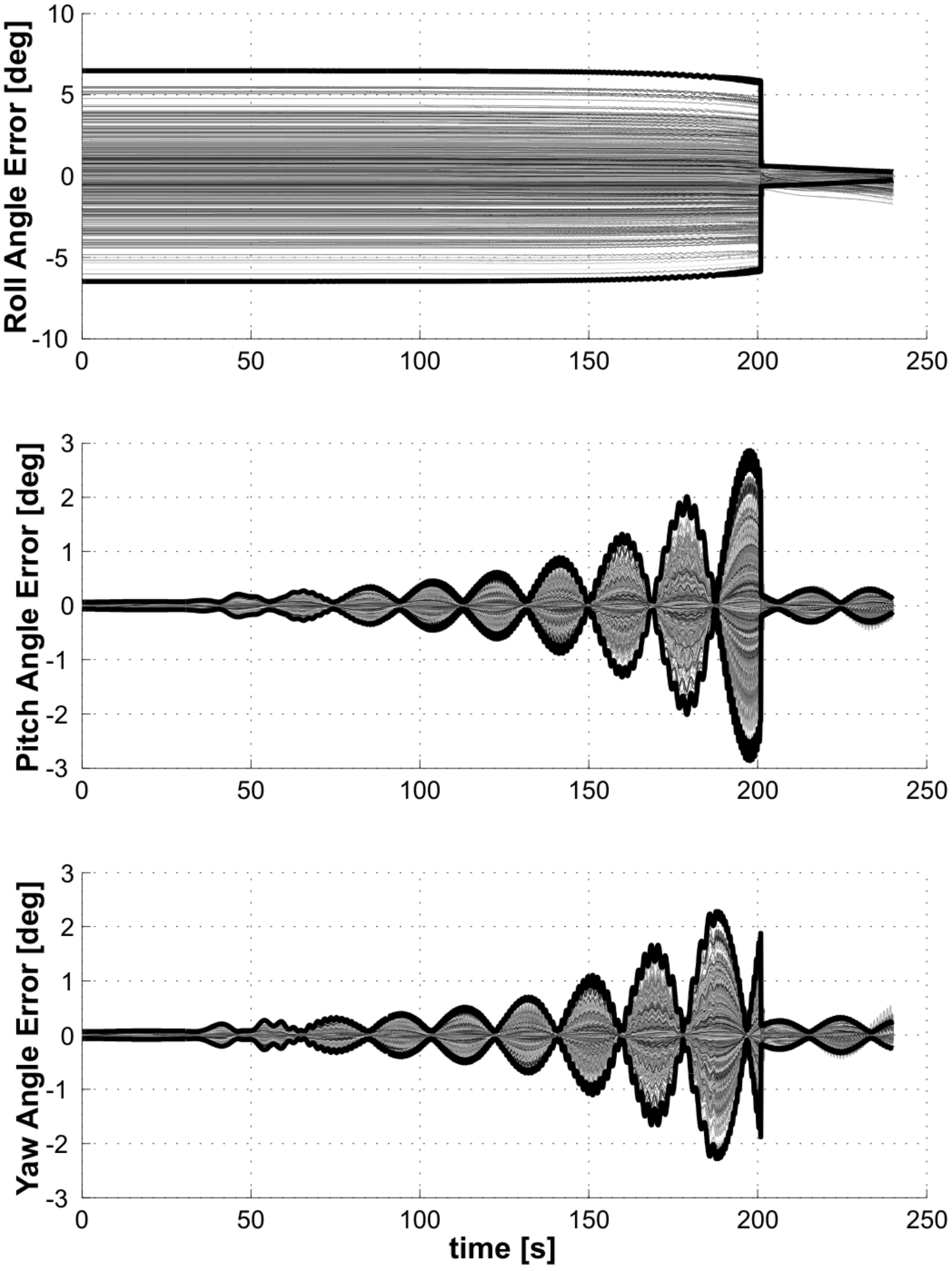


Figure 5-14: MIMU\MSP Scenario: Attitude Error Monte Carlo Analysis

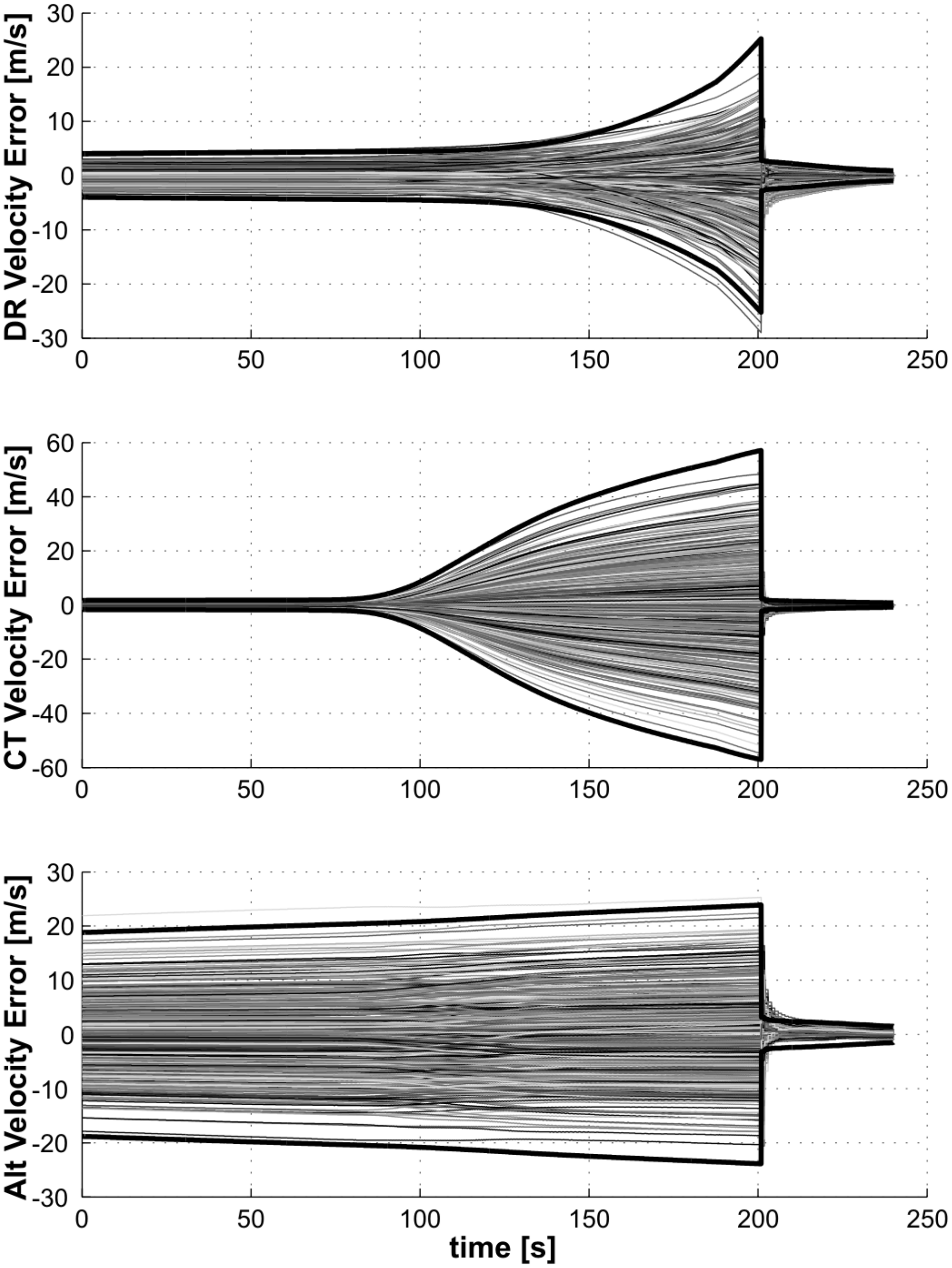


Figure 5-15: MIMU\MSP Scenario: Velocity Error Monte Carlo Analysis

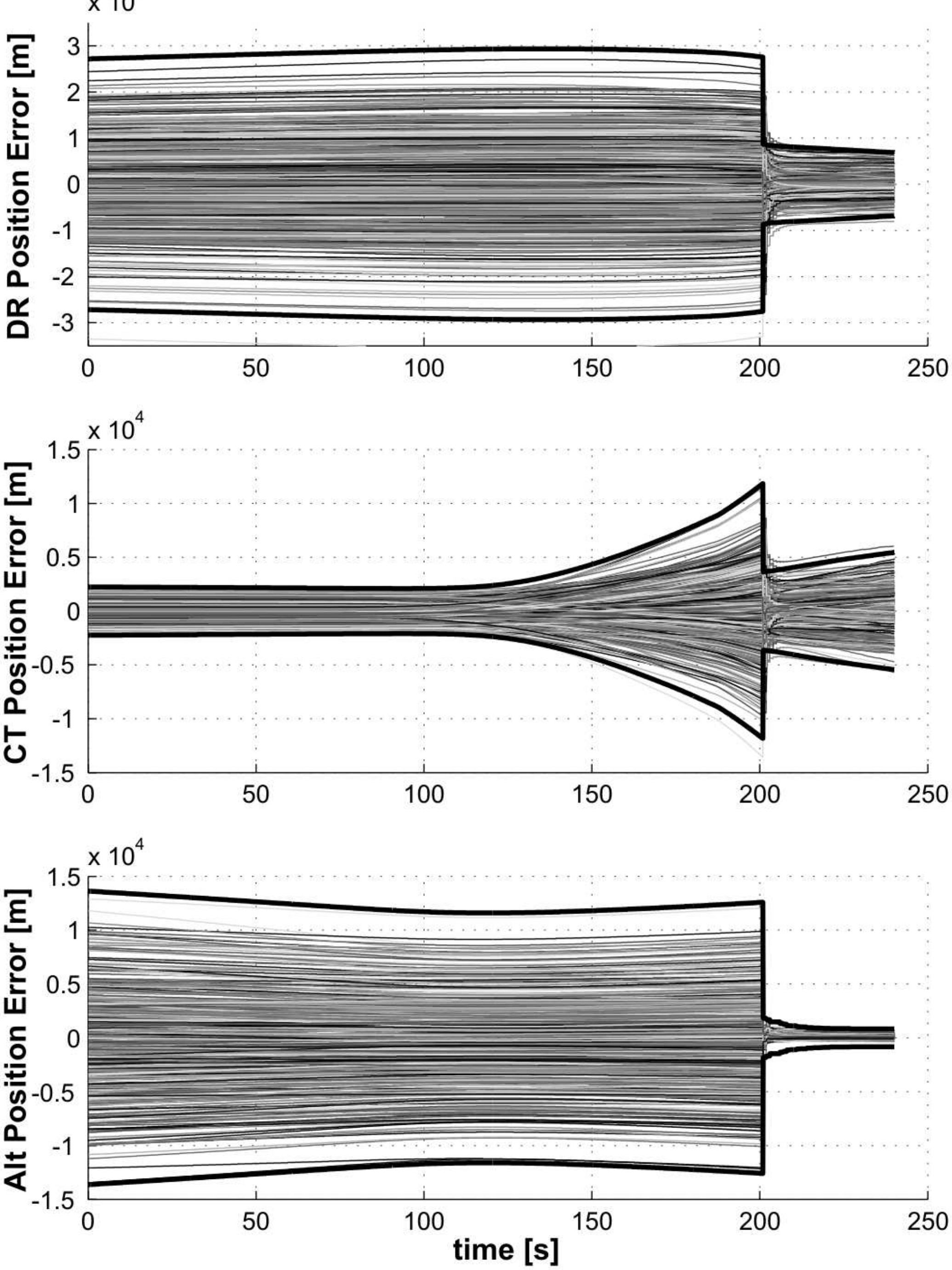


Figure 5-16: MIMU\MSP Scenario: Position Error Monte Carlo Analysis

Attitude Errors

Attitude errors for this sensor configuration follows similar trends to those of the MIMU/MER scenario prior to measurement updates. Once the altimeter and velocimeter updates begin, the attitude errors decrease dramatically. This is explained by the strong correlation between attitude errors, velocity errors, and the measurement of the velocity error by the velocimeter.

Velocity Errors

Velocity error also follow the same trends as seen in the MIMU/MER sensor configuration prior to measurement update. The very strong improvement in velocity uncertainty is explained primarily by the velocimeter measurements providing information directly into this estimated state. Altimeter measurements provide additional improvement through position/velocity correlations.

Position Errors

Position state errors match closely with the MIMU/MER scenario errors. Performance is better in all channels despite the use of a lower accuracy altimeter. This is because velocimeter measurements provide information on velocity errors which are strongly correlated with position errors.

5.5.1 Altitude Activation Study

The same altitude study discussed in Section 5.4.1 is performed with the MIMU/MSP sensor configuration. The initial measurement altitudes under examination are: $16km$, $10.2km$, $8.75km$, and $4km$. The case with no measurement updates (only inertial navigation) is included for comparison. The $8.75km$ case is the nominal scenario for the ARES mission. $10.2km$ corresponds to supersonic parachute deployment. The remaining two altitudes are chosen as extreme high and low cases.

For this scenario, increasing the activation point of the altimeter produces relatively small improvements in the final position uncertainty (Figure 5-17). The altitude uncertainty for the low altimeter activation has the largest discrepancy, having approximately twice the uncertainty of the other altitudes. The downrange and cross track position errors show consistent improvement over the MIMU/MER scenario in the previous section.

The velocity error (Figure 5-18) also shows very little differentiation among the different altitude cases. All cases produce very large velocity error improvements shortly after measurements begin.

The attitude errors in Figure 5-19 exhibit similar covariance improvements to those seen in the velocity error covariance. All altitude cases produce large and fast improvements in the attitude uncertainty.

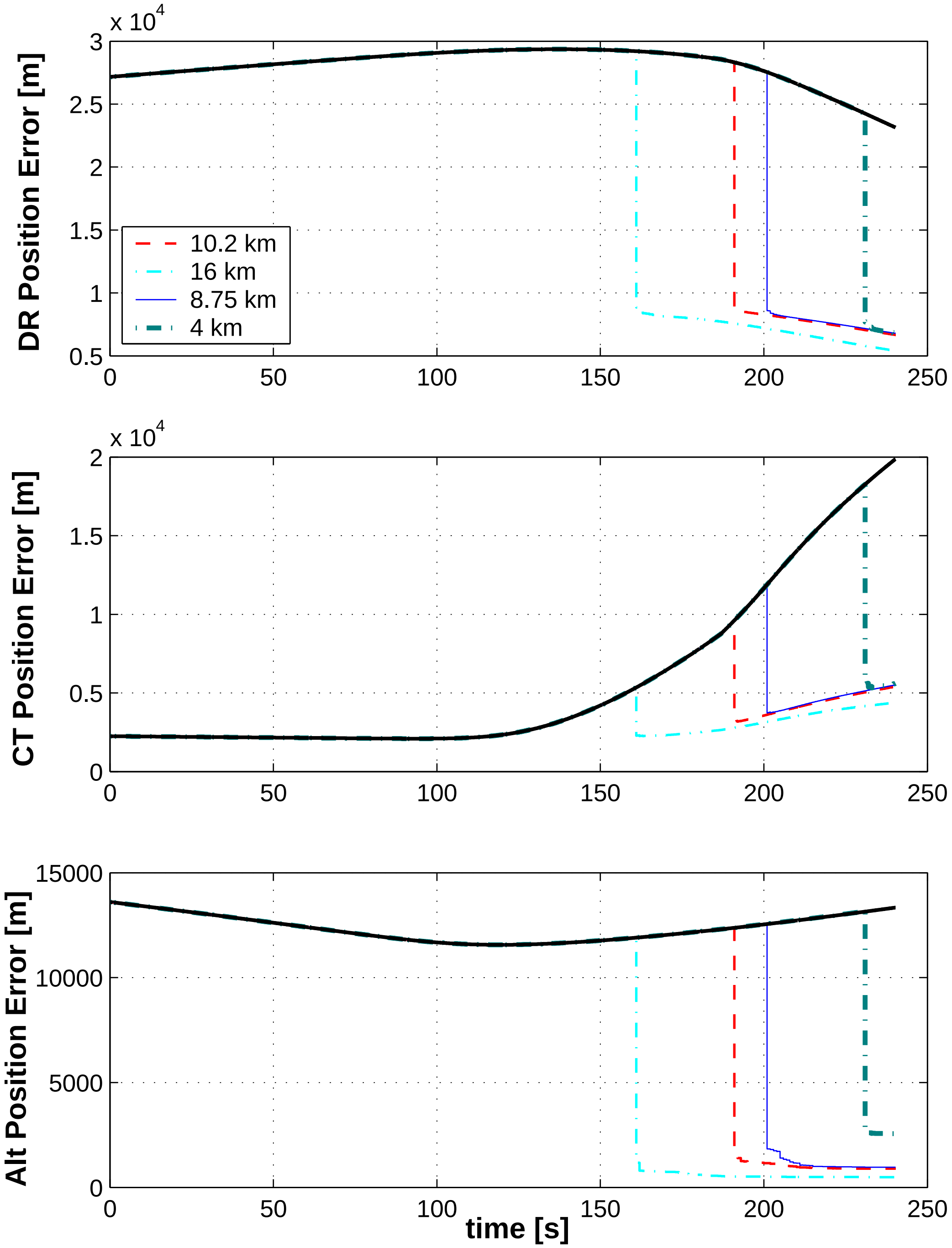


Figure 5-17: MIMU\MSP Altitude Study – Position Errors

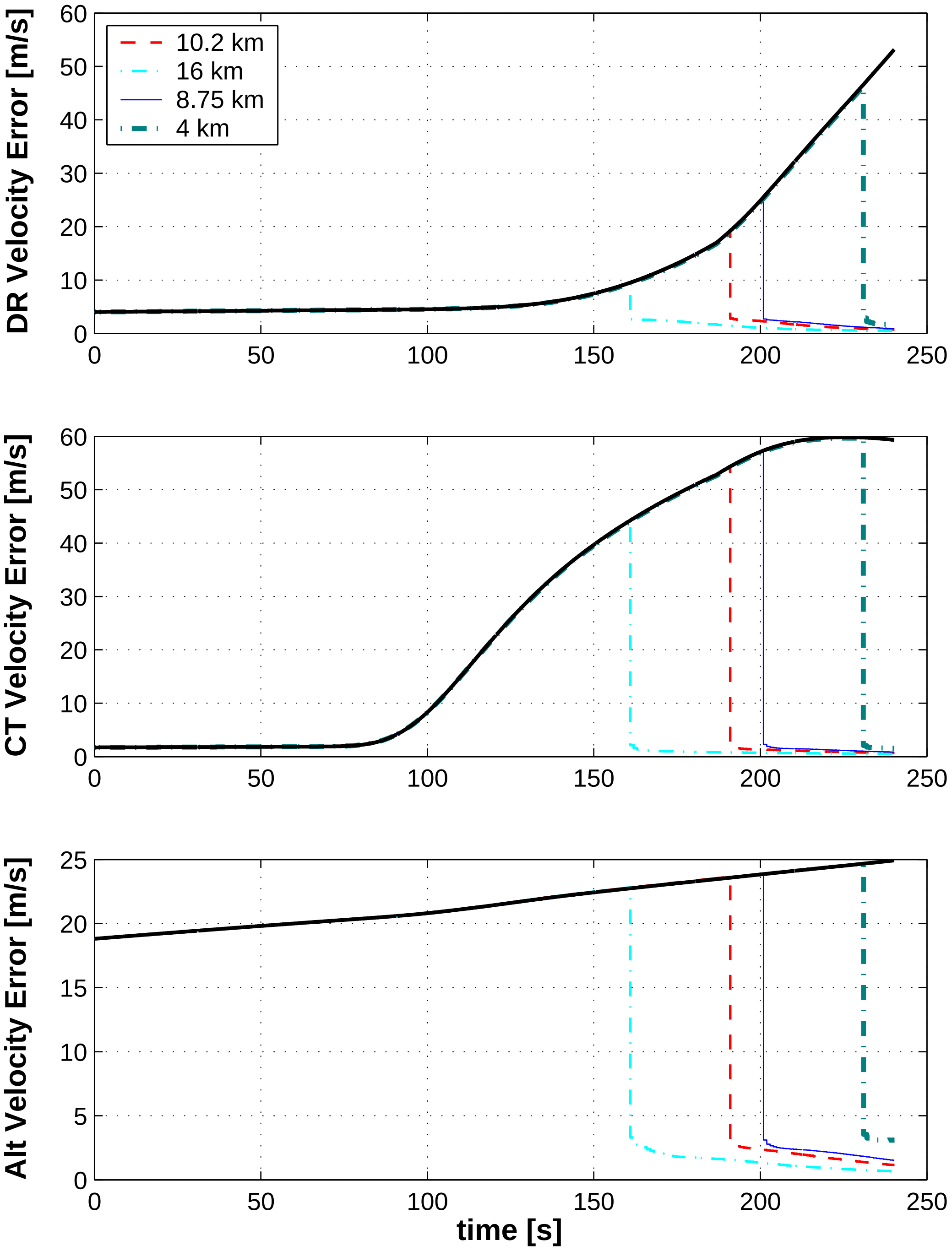


Figure 5-18: MIMU\MSP Altitude Study – Velocity Errors

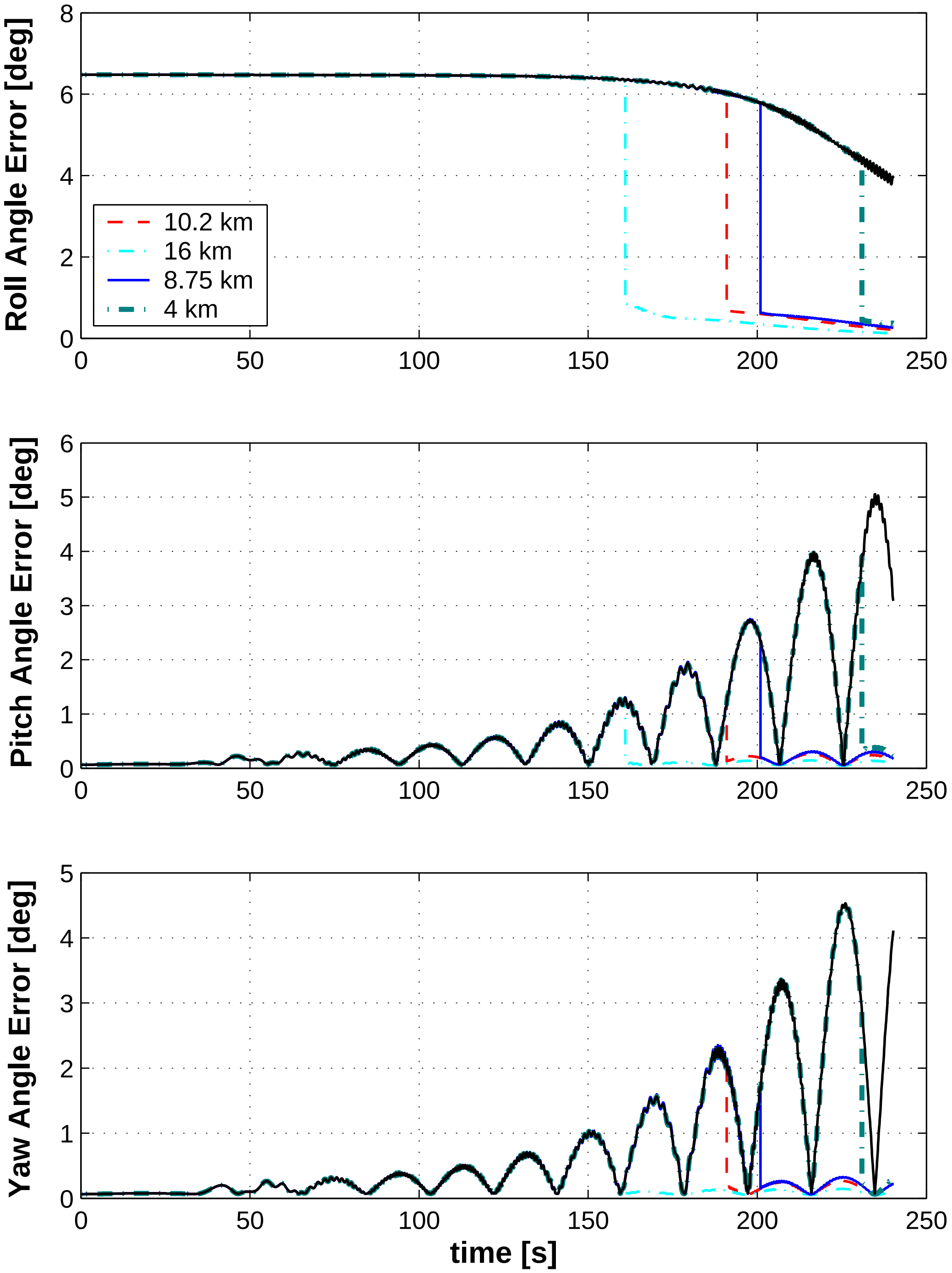


Figure 5-19: MIMU\MSP Altitude Study – Attitude Errors

5.6 Scenario 3: LN200/MER Sensor Suite

This scenario consists of the LN200 IMU with the MER-type altimeter. The LN200 provides comparable accelerometer performance to that of the MIMU system, but much lower gyro performance. Details of the initial state error covariance for this instrument configuration are provided in Appendix E.

A limited Monte Carlo analysis shows a failure of the navigation filter upon measurement update. The reason for this filter problem is the low performance of the LN200 gyros, which allows the attitude covariance to grow to large angle uncertainties of nearly 180° in roll after 12 hours of coast to EI. The measurement partial matrix within the extended Kalman filter is linearized assuming small angles. For the large attitude errors seen using the LN200, this assumption breaks down and causes the filter to fail due to nonlinearities.

Figures 5-20 through 5-22 present the principal state errors for this limited Monte Carlo analysis. Before measurements are processed, the error covariance is growing in many of the states to very large uncertainties. The 3-σ uncertainty level (heavy bounding line) fails to provide good state error predictions upon processing of the altimeter measurements. This breakdown causes large, sudden velocity and position state estimate errors.

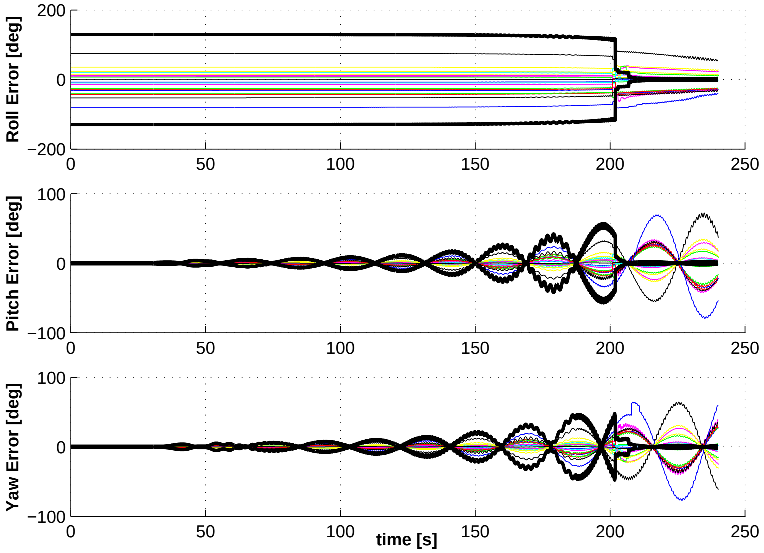


Figure 5-20: LN200\MER Scenario: Attitude Error Monte Carlo Analysis

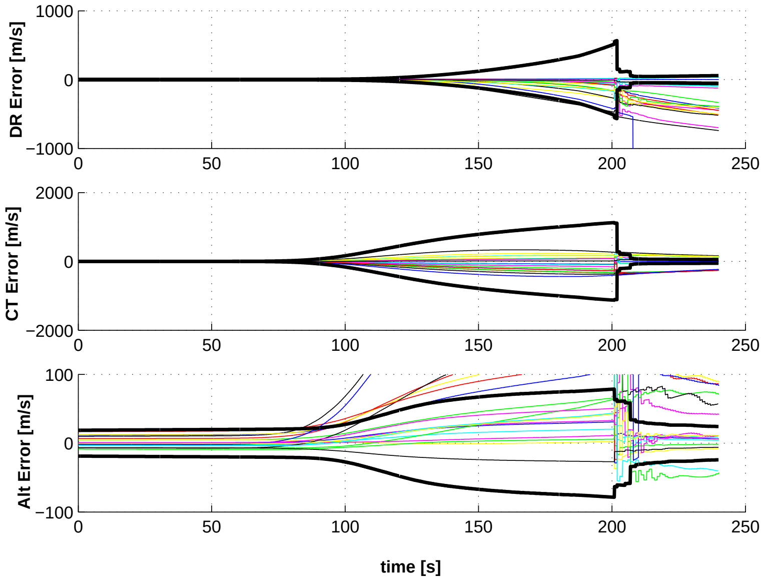


Figure 5-21: LN200\MER Scenario: Velocity Error Monte Carlo Analysis

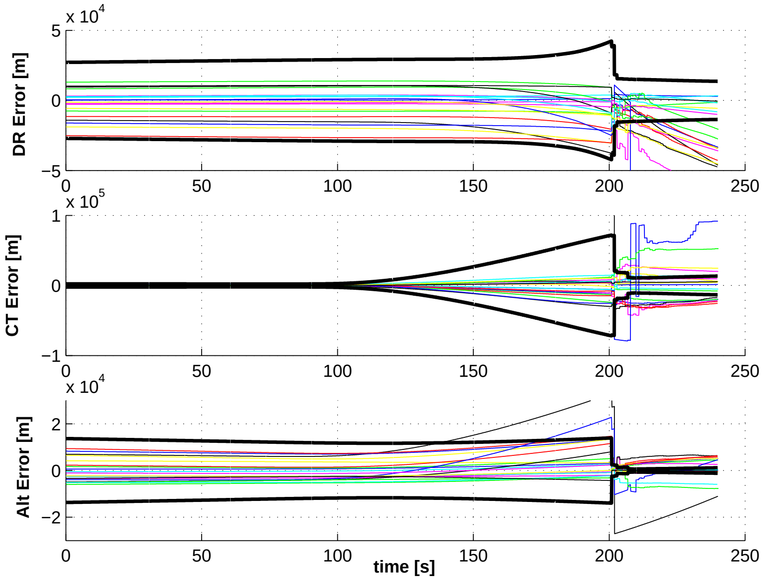


Figure 5-22: LN200\MER Scenario: Position Error Monte Carlo Analysis

5.6.1 Potential Improvements

At least two reasonable solutions for this problem exist. One is to include a nonlinear filter in the navigation system to process sensor measurements in the situation with very large attitude uncertainty. Another option is to supplement the attitude estimate process with additional instrumentation.

The current approach to space vehicle design is to incorporate off-the-shelf, space-qualified components instead of designing new systems for enhanced performance or implementing advanced ad-hoc algorithms. With this in mind, a reasonable solution for this navigation problem is to add additional sensors to mitigate the large attitude errors stemming from the LN200 gyro performance. This additional attitude sensing could come in the form of space-based sensing from star cameras or atmosphere-based sensing from sun sensors, for example.

The approach taken for this analysis is to assume that the entry vehicle has access to a star camera instrument up to one hour prior to entry interface. This would allow the vehicle to perform an attitude measurement update and then propagate this new initial state error covariance forward in time. Details of this new EI-1 hour state error covariance for this instrument configuration are provided in Appendix E.

A limited Monte Carlo analysis for this proposed configuration shows that the filter performs reasonably well for this augmented entry scenario. Error covariance trends follow closely those seen in the MIMU/MER scenario. The Monte Carlo results for this case are presented in Appendix F.

5.6.2 Altitude Activation Study

The altitude study is performed on this augmented LN200/MER sensor configuration. The initial measurement altitudes under examination are: $16km$, $10.2km$, $8.75km$, and $4km$. The case with no measurement updates (only inertial navigation) is included for comparison. The $8.75km$ case is the nominal scenario for the ARES mission. $10.2km$ corresponds to supersonic parachute deployment. The remaining two altitudes are chosen as extreme high and low cases.

For this scenario, increasing the activation point of the altimeter produces very little improvement in the final position uncertainty (Figure 5-23). Comparing this to the MIMU/MER scenario (Figure 5-11) shows that this LN200 case could have more improvement with increasing initial measurement altitudes if the attitude errors could be reduced to a lower level. The larger growth of the position cross track covariance is due to the larger cross track velocity uncertainty, which is a consequence of the large and growing yaw channel uncertainty.

The velocity errors shown in Figure 5-24 exhibit similar trends to those of the MIMU/MER scenario (Figure 5-12). The major differences are that this LN200 case generally has larger errors throughout the trajectory, and the altitude rate channel shows more differentiation according to altitude activation.

The attitude errors in Figure 5-25 exhibit similar covariance improvements to those seen in the velocity error covariance. The performance is similar in character to that of the MIMU/MER scenario (Figure 5-13), with the exception of the attitude errors being larger in magnitude.

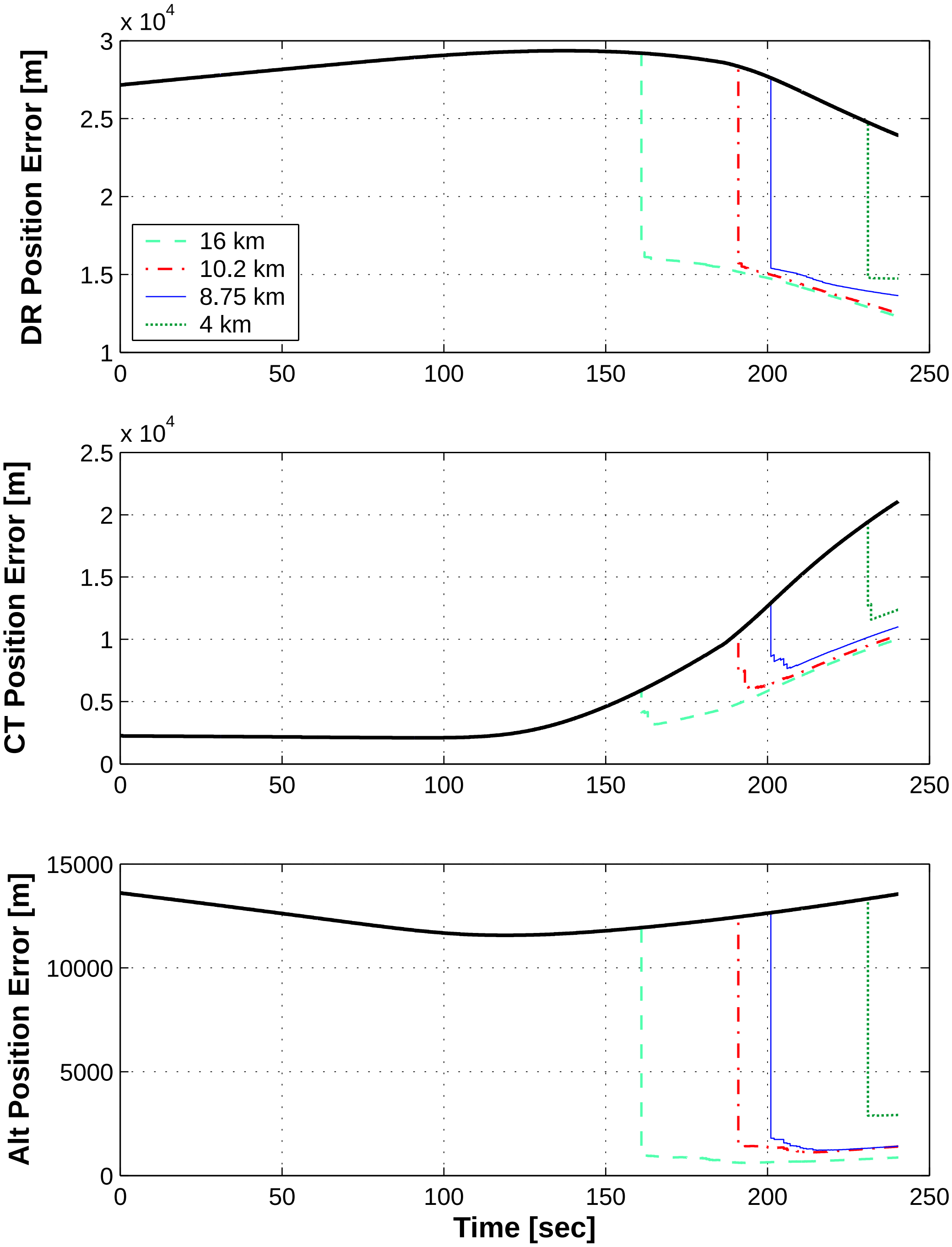


Figure 5-23: Augmented LN200\MER Altitude Study – Position Errors

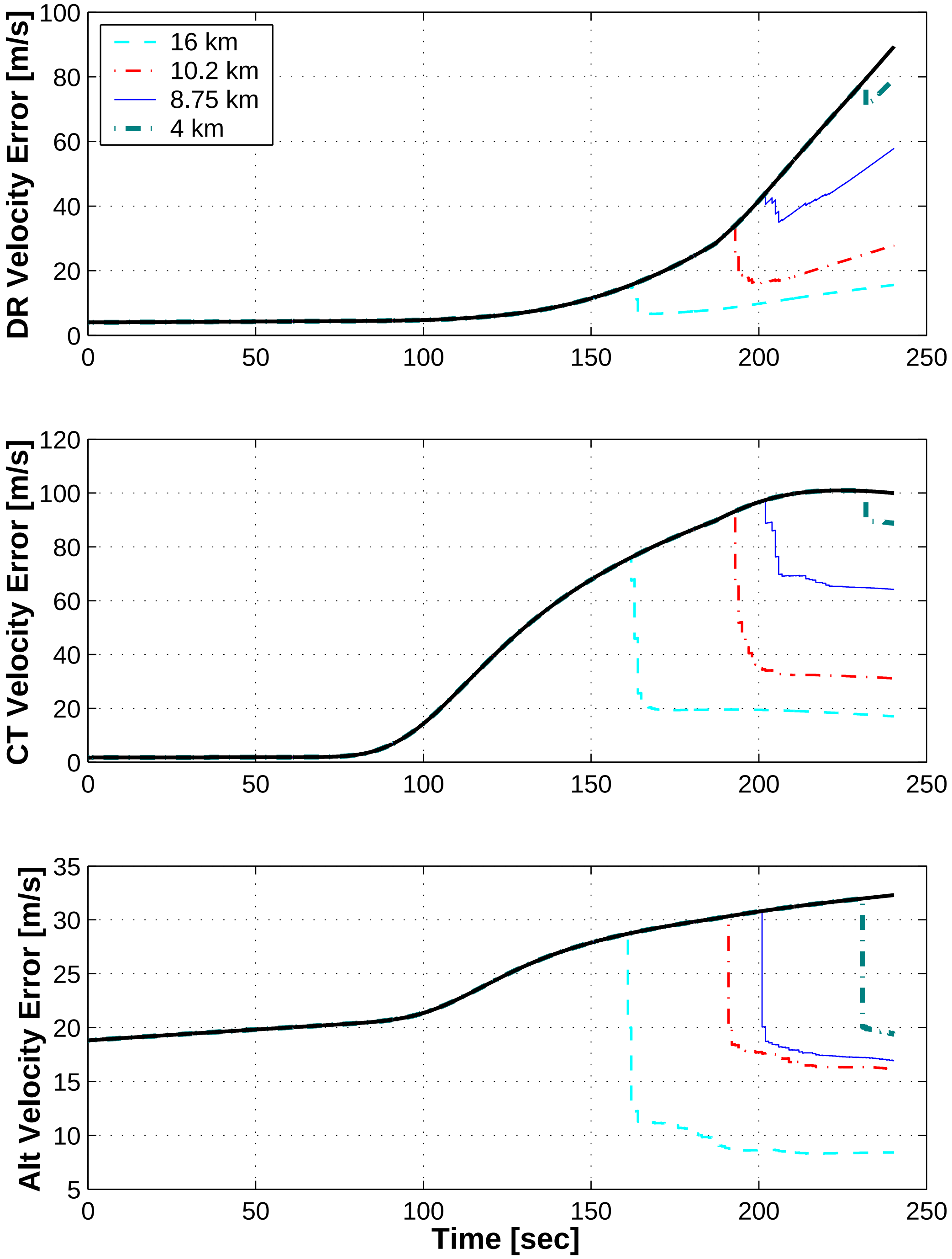


Figure 5-24: Augmented LN200\MER Altitude Study – Velocity Errors

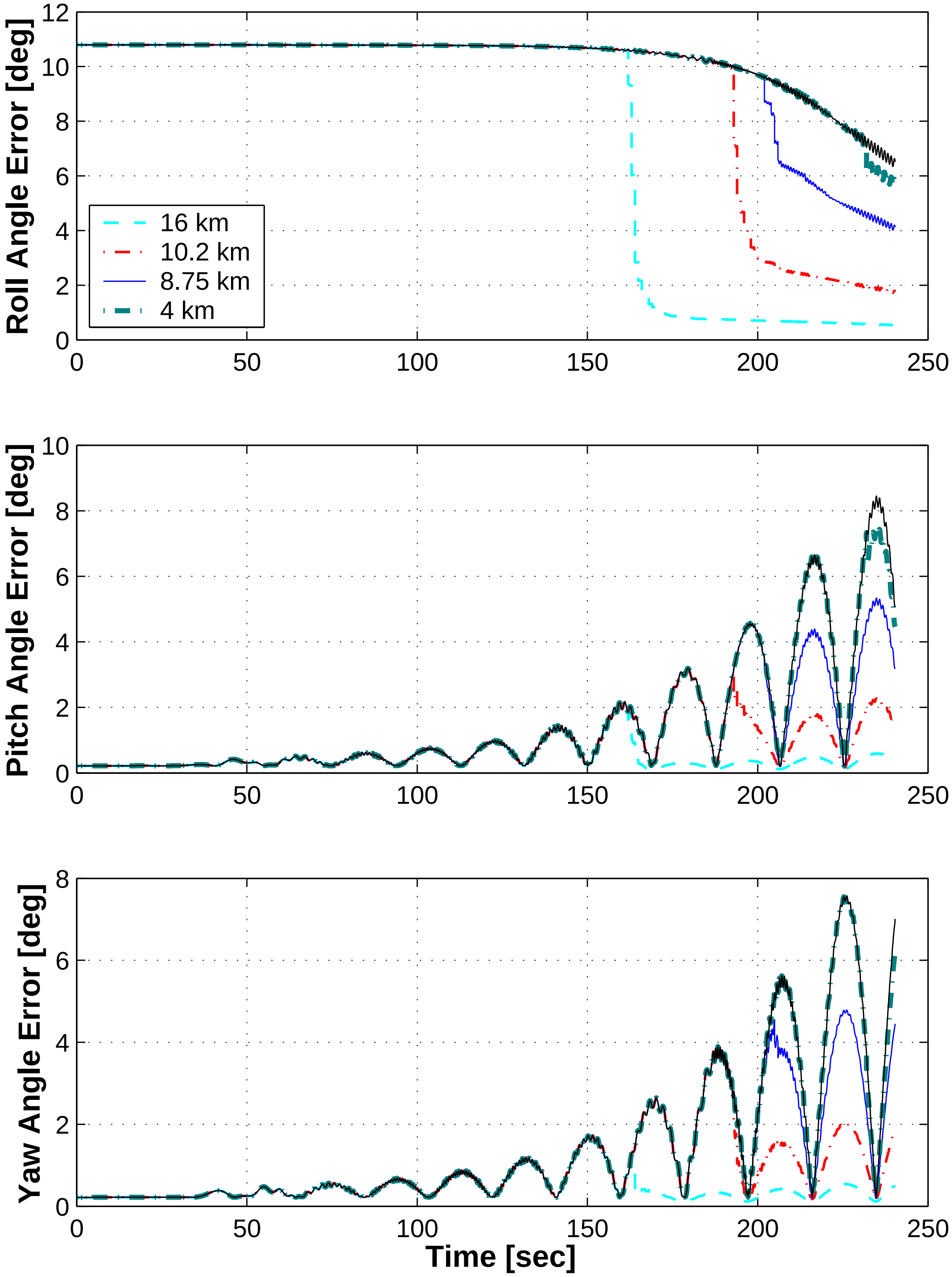


Figure 5-25: Augmented LN200\MER Altitude Study – Attitude Errors

5.7 Scenario 4: LN200/MSP Sensor Suite

This scenario consists of the LN200 IMU with the MSP-type altimeter and velocimeter. The LN200 provides comparable accelerometer performance to that of the MIMU system, but much lower gyro performance. Details of the initial state error covariance for this instrument configuration are provided in Appendix E.

A limited Monte Carlo analysis shows a failure of the navigation filter upon measurement update. This problem with the filter results from the low performance of the LN200 gyros, which allows the attitude covariance to grow to large angle uncertainties of nearly 180° in roll after 12 hours of coast to EI. The measurement partial matrix within the extended Kalman filter is linearized assuming small angles. For the large attitude errors seen using the LN200, this assumption breaks down and causes the filter to fail due to nonlinearities.

Figures 5-26 through 5-28 present the principal state errors for this limited Monte Carlo analysis. Before measurements are processed, the error covariance is growing in many of the states to very large uncertainties. The 3-σ uncertainty level (heavy bounding line) fails to provide good state error predictions upon processing of the altimeter measurements. This breakdown causes large, sudden velocity and position state estimate errors.

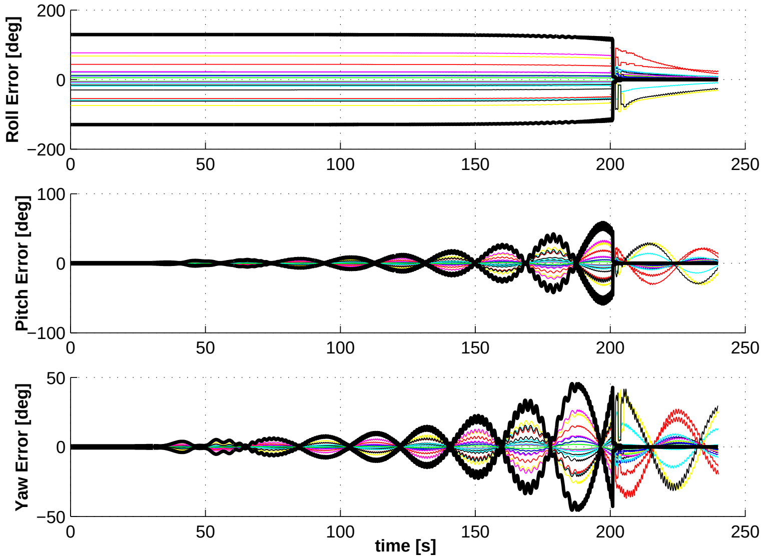


Figure 5-26: LN200\MSP Scenario: Attitude Error Monte Carlo Analysis

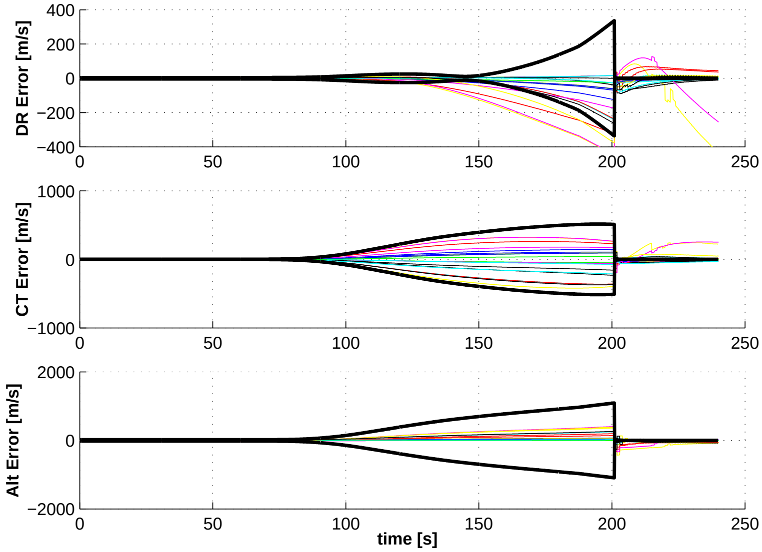


Figure 5-27: LN200\MSP Scenario: Velocity Error Monte Carlo Analysis

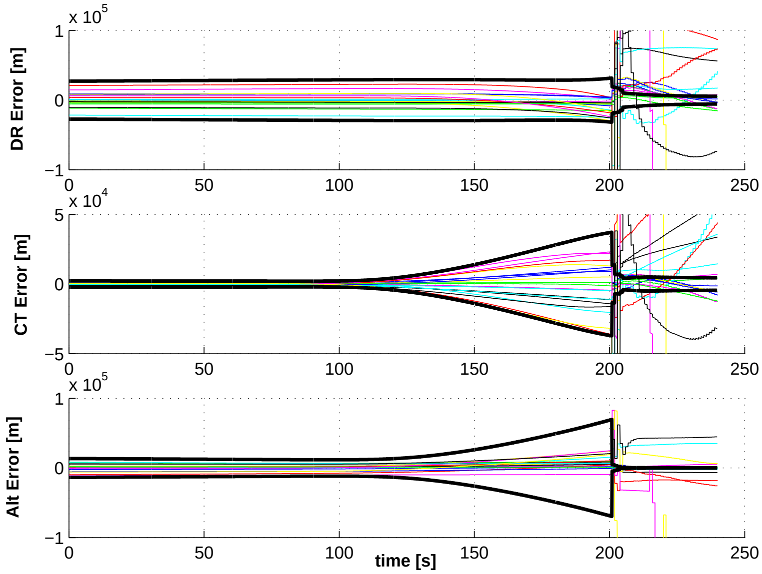


Figure 5-28: LN200\MSP Scenario: Position Error Monte Carlo Analysis

5.7.1 Potential Improvements

As in the previous LN200 navigation scenario, this problem can be addressed with a nonlinear filter in the navigation system to process sensor measurements or a supplement to the attitude estimation process with additional instrumentation.

Again, the solution examined in this analysis is to assume that the entry vehicle has access to a star camera instrument up to one hour prior to entry interface. This would allow the vehicle to perform an attitude measurement update and then propagate this new initial state error covariance forward in time. Details of this new EI-1 hour state error covariance for this instrument configuration are provided in Appendix E.

A limited Monte Carlo analysis for this proposed configuration shows that the filter performs reasonably well for this augmented entry scenario. Error covariance trends follow closely those seen in the MIMU/MSP scenario. The Monte Carlo results for this case are presented in Appendix F.

5.7.2 Altitude Activation Study

The altitude study is performed on this augmented LN200/MSP sensor configuration. The initial measurement altitudes under examination are: $16km$, $10.2km$, $8.75km$, and $4km$. The case with no measurement updates (only inertial navigation) is included for comparison. The $8.75km$ case is the nominal scenario for the ARES mission. $10.2km$ corresponds to supersonic parachute deployment. The remaining two altitudes are chosen as extreme high and low cases.

For this scenario, increasing the activation point of the altimeter produces very little improvement in the final position uncertainty (Figure 5-29). The highest altitude sensor activation shows some improvement over the other cases for the downrange and cross track channels. Comparing this to the MIMU/MSP scenario (Figure 5-17) shows that this LN200 case has larger downrange and cross track errors, but similar altitude performance.

The velocity errors shown in Figure 5-30 exhibit similar trends to those of the

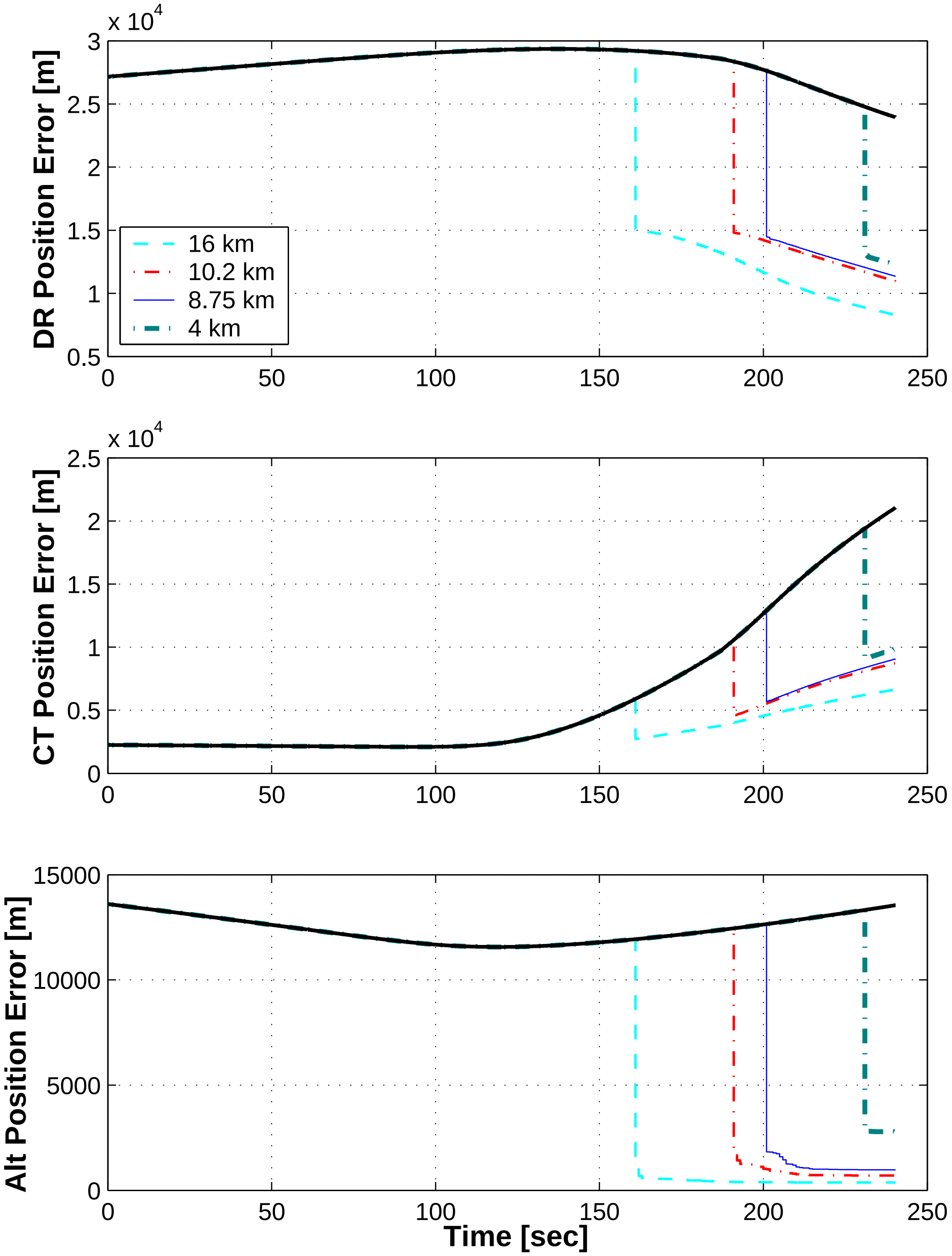


Figure 5-29: Augmented LN200\MSP Altitude Study – Position Errors

MIMU/MSP scenario (Figure 5-18). The strength in which the improvement occurs is due to the very precise velocimeter in conjunction with the altimeter measurements.

The attitude errors in Figure 5-31 exhibit similar improvements to those seen in the velocity error covariance. The performance is similar in character to that of the MIMU/MSP scenario (Figure 5-19), with the exception of the attitude errors being larger in magnitude.

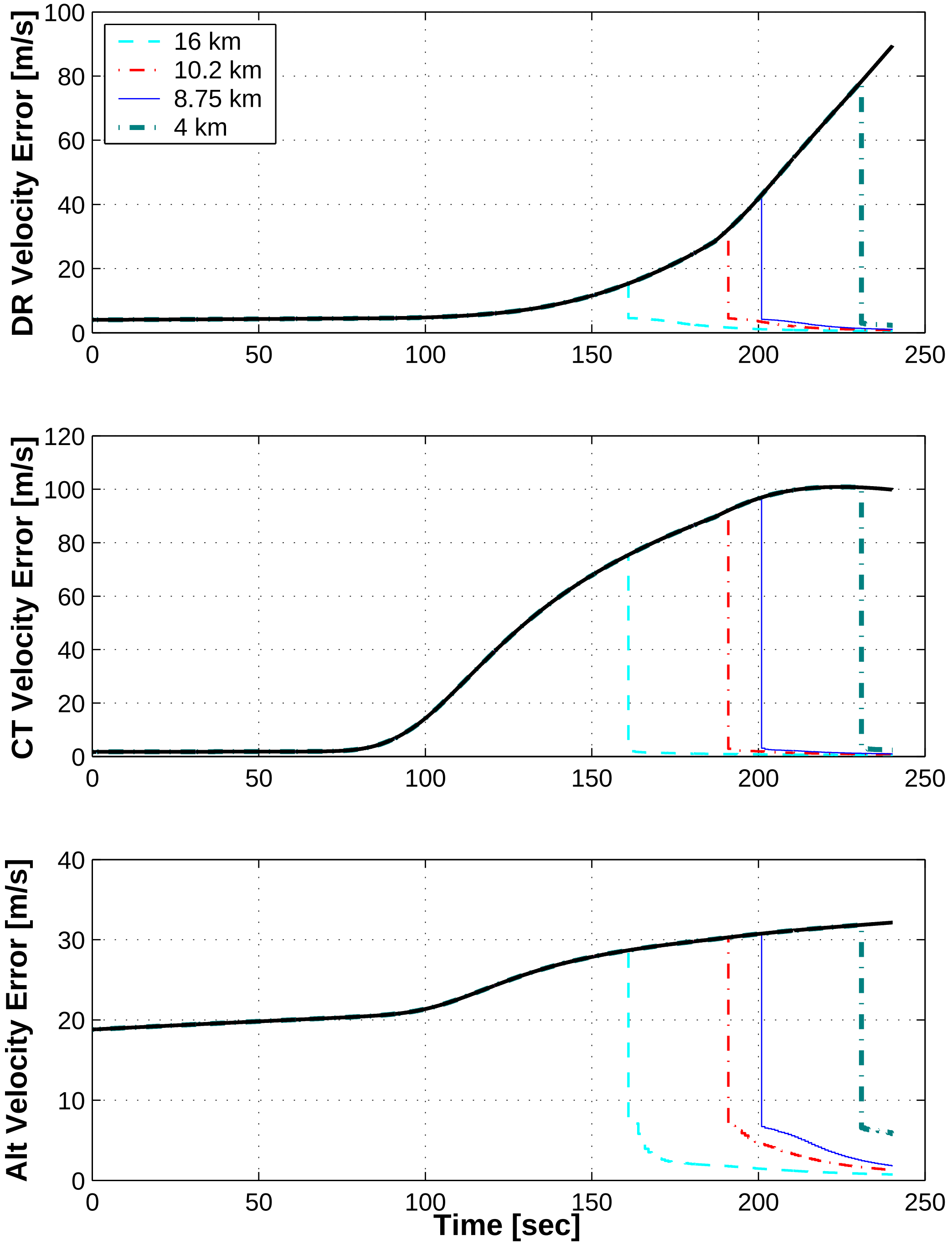


Figure 5-30: Augmented LN200\MSP Altitude Study – Velocity Errors

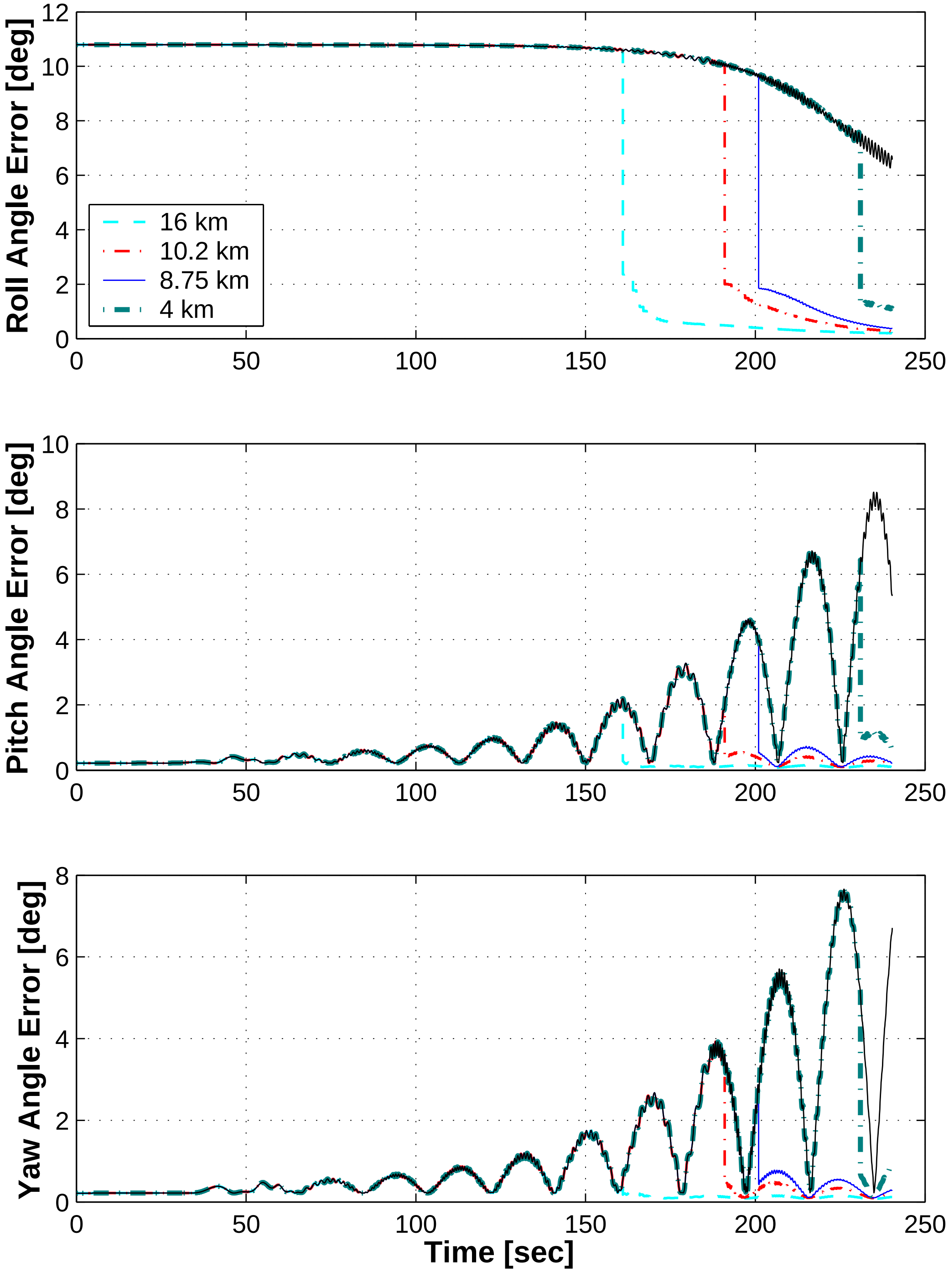


Figure 5-31: Augmented LN200\MSP Altitude Study – Attitude Errors

5.8 Discussion of Results

This navigation performance analysis shows that for the nominal ARES entry and descent, the MIMU IMU is the appropriate inertial instrument package for use with the extended Kalman filter navigation system. The LN200 lacks the attitude precision to maintain the estimated states within the linear operating range of the filter. This effectively eliminates the LN200 from consideration for this mission under the nominal use scenario. The LN200 IMU is a cheaper and less massive instrument package. In scenarios where this is a crucial factor, it would be possible to supplement the LN200 with additional attitude measurements to allow the navigation system to operate effectively. To examine the navigation performance of the LN200 in this situation, an additional star camera attitude measurement update is provided to the navigation filter one hour prior to EI. Since this augmented LN200 is a departure from the nominal ARES scenario, presentation of its compiled performance comparison is provided in Appendix H.

An important fact to notice is that the position uncertainties are extremely large until sensor measurements are taken. This is a product of initial state uncertainty that originates with the Deep Space Network (DSN) update 12 hours prior to EI. The large initial state uncertainty for this scenario makes using the navigation filter state for guidance maneuvers or parachute deployment prior to measurement update unreasonable.

MIMU Performance Summary

Comparing the MIMU/MER and MIMU/MSP sensor configurations show the distinct advantage of using two instruments to provide measurement information into multiple states of the filter. Figures 5-32 through 5-34 compare the navigation uncertainty for these two instrument packages being flown on the nominal ($8.75km$ instrument activation) scenario.

Altitude errors over the trajectory duration show similar results between the two sensor packages. This is because the two altimeters have relatively similar performance specifications, and altimeter performance dominates this channel of position error. In downrange and cross track errors, though, the MSP altimeter/velocimeter package is able to reduce uncertainty by about 5 km as compared to the MER altimeter. Velocity errors exhibit significant improvement when using the MSP package over the MER altimeter. This is explained by the velocimeter providing measurement information directly into the velocity state, while the other case must improve velocity errors through correlations with the measurements entering the position state. Attitude errors exhibit a similar dramatic improvement. The strong coupling of attitude states with the velocity states receiving the measurement information also explains this.

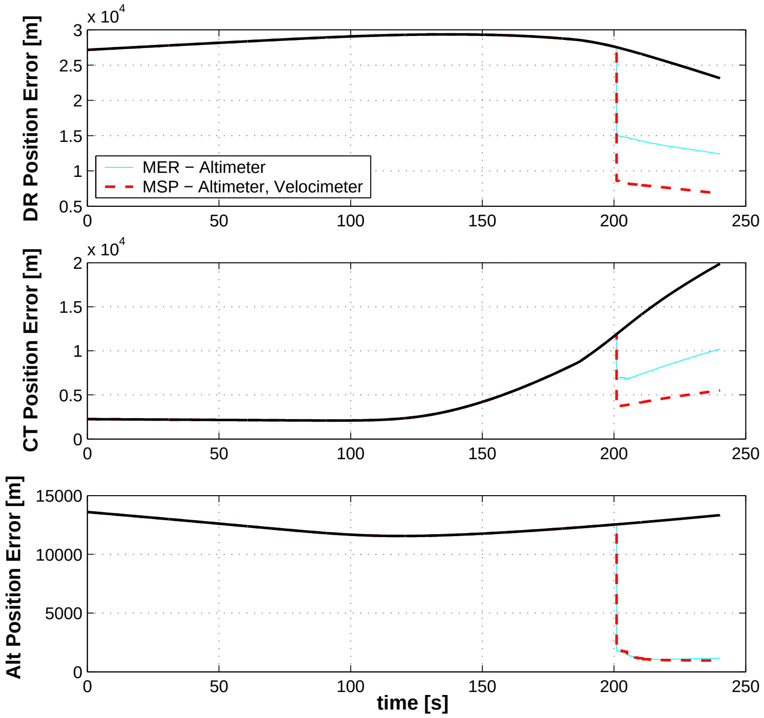


Figure 5-32: Nominal MIMU Scenario: Position Error Comparison

Tables 5.1 through 5.3 present the final primary state 3-σ uncertainties for the nominal entry and descent scenario. The data provides the performance of the MIMU/MER and MIMU/MSP sensor configurations and shows the improvement achieved through use of the MSP sensor package. Again, the data shows that the addition of a velocimeter measurement provides moderate uncertainty improvements

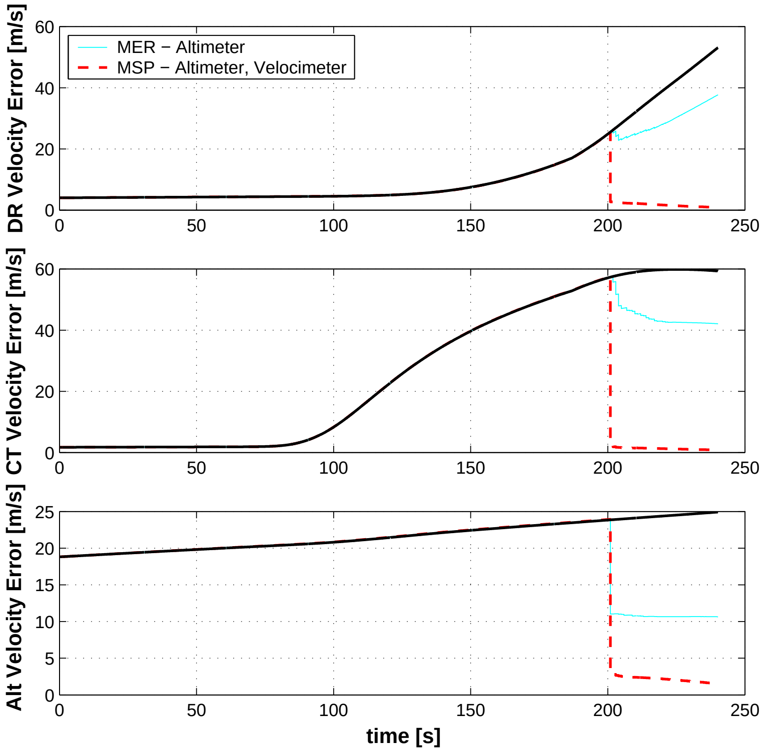


Figure 5-33: Nominal MIMU Scenario: Velocity Error Comparison

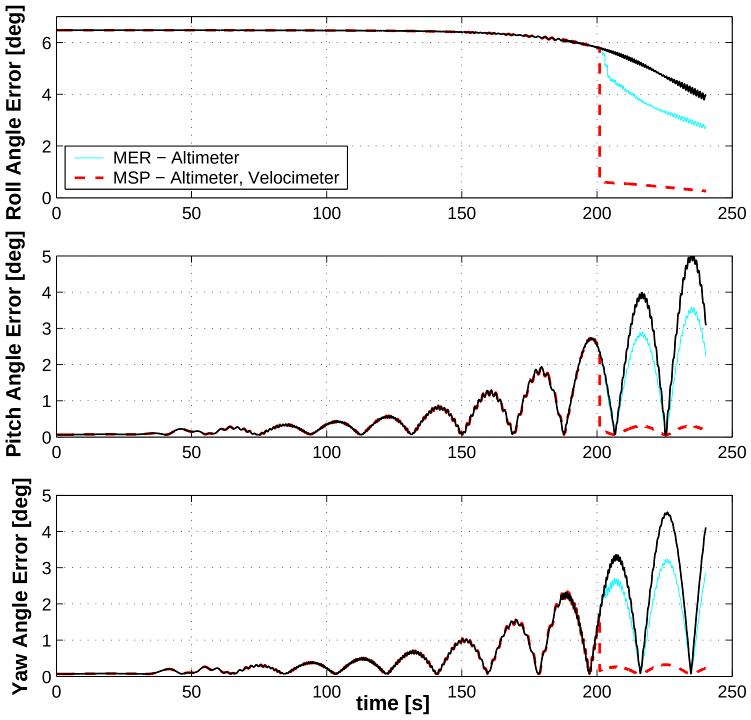


Figure 5-34: Nominal MIMU Scenario: Attitude Error Comparison

within the position state, but dramatic improvements within the velocity and attitude states.

	Downrange	**Cross track**	**Altitude**
MIMU/MER	12.40 $[km]$	10.21 $[km]$	1.16 $[km]$
MIMU/MSP	6.79 $[km]$	5.51 $[km]$	0.96 $[km]$
MSP improvement	58%	60%	19%

Table 5.1: Final Position Uncertainty (3-σ)

	Downrange	**Cross track**	**Altitude**
MIMU/MER	37.80 $[m/s]$	42.08 $[m/s]$	10.65 $[m/s]$
MIMU/MSP	0.90 $[m/s]$	0.83 $[m/s]$	1.52 $[m/s]$
MSP improvement	191%	192%	150%

Table 5.2: Final Velocity Uncertainty (3-σ)

	Roll	**Pitch**	**Yaw**
MIMU/MER	2.78 $[deg]$	2.23 $[deg]$	2.86 $[deg]$
MIMU/MSP	0.27 $[deg]$	0.17 $[deg]$	0.22 $[deg]$
MSP improvement	165%	172%	171%

Table 5.3: Final Attitude Uncertainty (3-σ)

Figures 5-35 through 5-37 show only the terminal principal state errors as a function of initial measurement altitude. Using the MSP altimeter/velocimeter consistently reduces the final state uncertainties. Selection of one sensor package over another, though, depends on the specific mission requirements.

If final position knowledge is of greatest importance, the MSP package performs better, but only marginally. Figure 5-35 shows that there is little improvement in final position error by increasing the altitude of initial measurements. This is a significant finding because an additional cost, mass, and technology development is likely required to move the initial altitude of measurements above the nominal ($8.75km$) location. One reason is that the entry vehicle heat shield hides the instrumentation up to this point. Activation before this point requires early jettison of the heat shield or a new approach to sensor design in order to allow it to sense through or around this shield. Sensor activation at a higher altitude also requires a more powerful sensor package (due to signal attenuation), which generally drives up cost and mass of the entry vehicle. If the mission requires precise final velocity or attitude knowledge, then the MSP package shows significant performance improvement. Performance differences between the MSP and MER packages range almost up to two orders of magnitude for velocity uncertainty and one order of magnitude for attitude uncertainty. An interesting trend of these results is that increased activation altitude reduces the performance gap between the MER and MSP sensor packages for velocity and attitude uncertainty.

For guided entry vehicles, such as the Mars Science Lab (MSL), precise navigation information is important early in the entry and descent phase. In this case, early sensor measurements provide the advantage of precise navigation sooner in the vehicle trajectory. The earlier the sensor activation occurs, the more time the guidance system has to make corrective trajectory maneuvers. In addition, the potential range of the entry vehicle is extended by gaining precise navigation (and consequently control authority) as early as possible in the entry and descent. Developing methods of altimeter/velocimeter sensing at higher altitudes is crucial if precise navigation is needed throughout a significant portion of the entry and descent trajectory. This navigation knowledge improvement is even more significant if measurements can occur before parachute deployment. This is a consequence of the entry vehicle being more maneuverable and having more control authority during entry compared to the vehicle under parachute descent. However, issues associated with getting measurements through the heat shield, as previously mentioned, may be a problem.

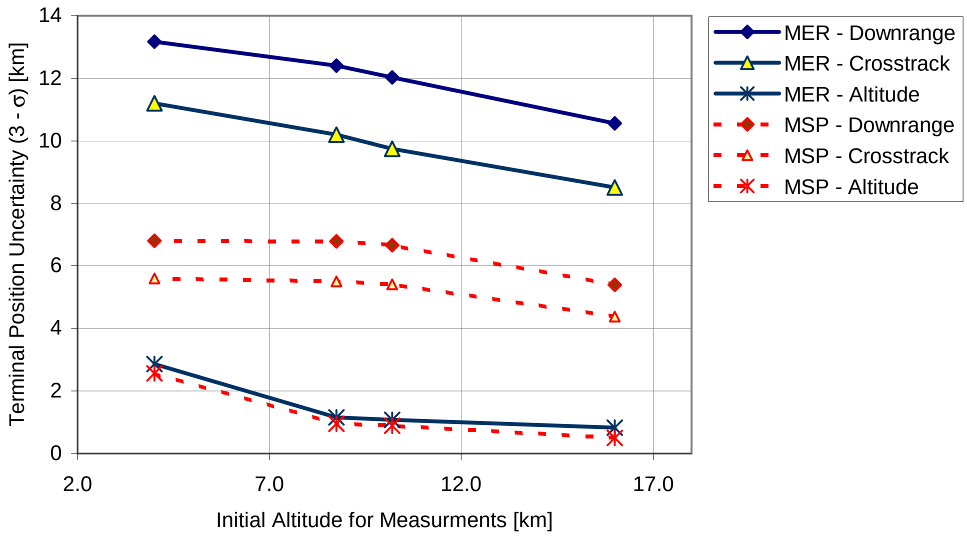


Figure 5-35: MIMU Scenario: Terminal Position Error Comparison

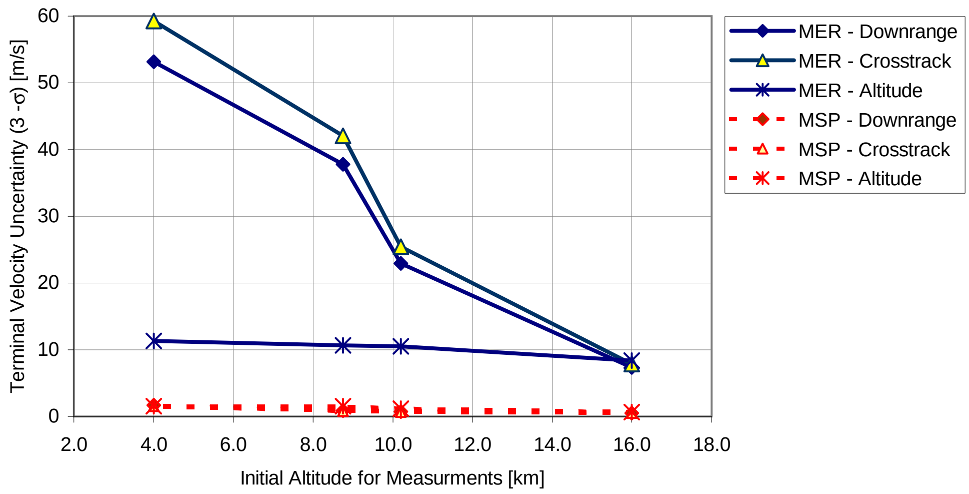


Figure 5-36: MIMU Scenario: Terminal Velocity Error Comparison

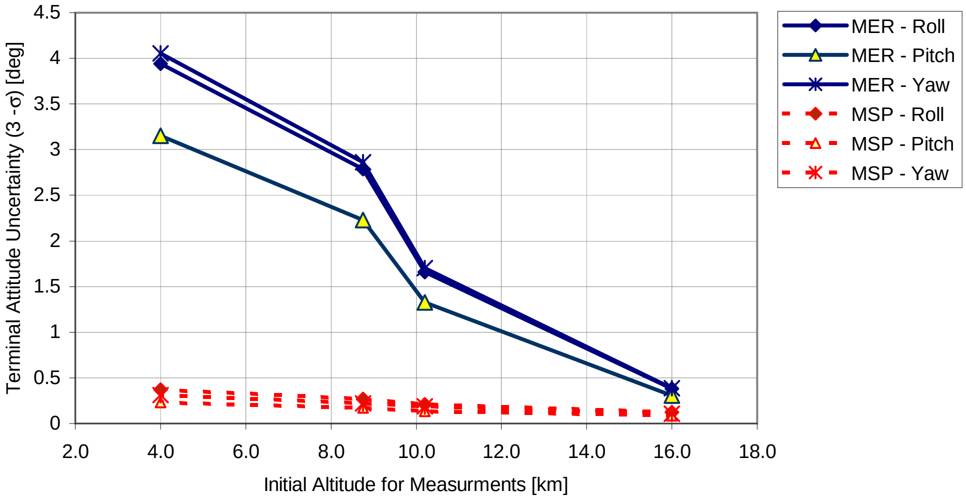


Figure 5-37: MIMU Scenario: Terminal Attitude Error Comparison

In general, initial errors dominate this entry and descent navigation problem. Attitude errors in particular explain much of the state error growth seen. Error growth within the attitude channels due to gyro noise, scale factor, and bias is negligible over the duration of the entry and descent phase of the mission. The dominant source is the initial attitude error at EI. This error is a function of gyro errors that accumulate over the 12 hour cruise prior to atmosphere interface. During the cruise the vehicle spins about its roll axis. This contains pitch and yaw uncertainty, but drives roll uncertainty to grow relatively large. As the vehicle follows its entry and descent trajectory, atmospheric and gravity forces cause the vehicle to rotate with respect to the initial attitude covariance ellipsoid. This rotation transfers the large attitude roll uncertainty into the pitch and yaw attitude channels. This transfer of roll attitude error into other channels has a cascade effect through the other principle states via improper integration of the accelerometer data. This effect is a primary driver for uncertainty growth in the position and velocity states through the entry and descent phase. Velocity error growth due to accelerometer noise, scale factor, and bias is a secondary effect contributing first to the velocity state error growth and second to the position state error growth.

Chapter 6

Conclusions

The navigation system is a critical component in the design of a spacecraft. Without good navigation information, the guidance system is unable to correct for any trajectory dispersions which can jeopardize the mission objectives and possibly the vehicle survival.

This analysis first develops an atmospheric entry and descent full-state filter. This filter is used to examine the scenario of Mars entry and descent for the ARES mission entry vehicle. Under examination are the error covariance trends within the navigation filter during dead-reckoning navigation and the subsequent change in state uncertainty as sensor measurements are taken later in the trajectory. The analysis compares four IMU/altimeter/velocimeter combinations for the nominal ARES entry and descent profile and then extends the discussion by investigating the effect of the altitude at which the sensor become activated.

Analysis of the entry and descent navigation scenario using the LN200 IMU shows that its poor gyro accuracy builds up very large attitude errors after the 12 hour cruise to atmospheric entry interface. This causes failure of the extended Kalman filter due to nonlinearities upon processing sensor measurements. This problem can be overcome with additional attitude measurements before or during the entry and descent trajectory. To examine the performance characteristics of the LN200 IMU, it is assumed that the entry vehicle has access to a star camera instrument up to one hour prior to entry interface. This augmented entry scenario provides improved EI initial conditions that allow for reasonably filter performances. Since this augmented LN200 is a departure from the nominal ARES scenario, results are considered separately from the MIMU scenario results.

The addition of a surface relative velocity measurement to the altimeter measurement was expected to provide significant improvements to the navigation state uncertainty. The analysis shows that this additional sensor provides only marginal position uncertainty improvement and significant improvement in the velocity and attitude uncertainty. The addition of this velocimeter to the baseline altimeter adds cost and complexity to the entry vehicle, but depending on the mission requirements this additional position, velocity, and attitude knowledge might be crucial.

Increasing the altitude for initial altimeter/velocimeter activation provides more time for the sensors to supply measurements to the navigation filter. This was expected to provide additional precision to the overall estimated state. For the scenarios under consideration, this produces only marginal improvements in position uncertainty, but relatively large improvements in velocity and attitude uncertainty. In situations where only final position knowledge is important, there is little advantage to early sensor activation. If good velocity and attitude information is an important factor then there is an advantage to early measurements. Examples of this situation might be a requirement for precise attitude knowledge for science imaging instruments or precise final velocity knowledge for proper engine throttling for a powered engine descent. A distinct advantage of early sensor activation is that this provides improved navigation states for a longer portion of the entry and descent trajectory. This situation is important if the vehicle is being actively guided along a desired trajectory. The specific requirements of the mission under consideration determine what approach should be taken in regard to initial sensor activation altitude. Increasing the altitude of initial measurement above the nominal point of heat shield jettison likely requires additional cost, mass, and technology development. Improvements derived from this approach must overcome technological hurdles, and possibly at a significant cost.

Higher precision IMUs were expected to produce significant performance improvements to the overall entry and descent navigation system due to the assumed dominant role of IMU errors in covariance predictions. An important finding of this analysis is that the initial state uncertainty dominates the entry and descent navigation problem. For this scenario, the initial state uncertainty originates with the Deep Space Network (DSN) update 12 hours prior to EI. This state uncertainty is so large that using the navigation filter for guidance maneuvers or parachute deployment prior to measurement update is unreasonable. State uncertainty growth due to IMU gyro and accelerometer errors over the course of the short final trajectory is small. Integration of accelerometer data under the influence of attitude errors causes significant growth in velocity and position uncertainty. This effect is driven by the near-constant attitude errors and the effect of vehicle rotation with respect to the attitude covariance ellipsoid during the trajectory. This has the effect of redistributing attitude errors from the innocuous attitude roll channel, into the more consequential pitch and yaw attitude channels. This apparent growth in yaw and pitch uncertainty cascades through the filter states, causing growth in velocity and position uncertainty.

This analysis provides insight into how initial atmospheric entry interface errors and sensor performance affect the entry and descent navigation problem. In addition, limitations of the extended Kalman filter for this application are explored. Extensions of this analysis could include investigation into methods for achieving "pinpoint" (sub-kilometer) navigation position accuracy. This might include methods for reducing state uncertainty at atmosphere interface and examination of additional or higher accuracy measurement sensors. Improvements to initial conditions at entry interface could focus on reducing roll error so that error transfer into pitch and yaw attitude channels is less significant. It could also focus on achieving better position/velocity errors at EI via better Mars approach navigation (eg. improved DSN updates, landmark tracking or optical measurements of the Martian moons). Enhancements to the sensor package during entry, descent, and landing could include improved altimeters that have access to surface terrain models or terrain imaging systems that provide highly accurate surface relative position measurements. This could also include range/Doppler measurements to surface and/or orbiting beacons. Since the

IMU errors (gyro and accelerometer bias, scale factor, and misalignment) are not significant performance drivers, a reduced-state filter excluding these terms might be realizable without reduced navigation performance. Additional future work could include tuning of the extended Kalman filter for this application to reduce the number of states tracked to improve computation performance.

Appendix A

Simplified Covariance Measurement Update Formulation

Expanding equation 4.25 and substituting in equation 4.33 in the following way allows for further simplification of the error covariance matrix update equation [18].

$$
\begin{aligned}
\boldsymbol{P}_{k_+} &= \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-} - \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-} - \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T + \boldsymbol{K}_k\,\boldsymbol{H}_k\boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-} - \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T + \boldsymbol{K}_k\left(\boldsymbol{H}_k\boldsymbol{P}_{k_-}\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)\boldsymbol{K}_k^T \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-} - \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T + \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\left(\boldsymbol{H}_k\,\boldsymbol{P}_{k_-}\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)^{-1}\left(\boldsymbol{H}_k\boldsymbol{P}_{k_-}\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)\boldsymbol{K}_k^T \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-} - \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T + \boldsymbol{P}_{k_-}\boldsymbol{H}_k^T\,\boldsymbol{K}_k^T \\
&= \left(\boldsymbol{I} - \boldsymbol{K}_k\,\boldsymbol{H}_k\right)\boldsymbol{P}_{k_-}
\end{aligned}
\tag{A.1}
$$

Appendix B

Derivation of the Body-Frame Attitude Error Time Derivative

For information on the vehicle attitude, the navigation filter tracks the estimated inertial-to-body quaternion ($\hat{\boldsymbol{q}}_{I\to\hat{B}}$). The state error covariance tracks the error of this quaternion to the true inertial-to-body quaternion ($\boldsymbol{q}_{I\to B}$), but in terms of three small angle rotations ($\delta\boldsymbol{\theta}^B$) about the body X-Y-Z axis. This error is defined as the small angle rotation from the true body frame (B) to the estimated body frame ($\hat{B}$). In order to formulate the proper state error covariance, the time derivative of these three attitude angles is required. The error quaternion is given by

$$\hat{\boldsymbol{q}}_{B\to\hat{B}} = \boldsymbol{q}_{B\to I} \otimes \hat{\boldsymbol{q}}_{I\to\hat{B}} \approx \begin{pmatrix} \delta\boldsymbol{\theta}^B/\,2 \\ 1 \end{pmatrix} \tag{B.1}$$

The time derivative of this quaternion error is

$$\dot{\hat{\boldsymbol{q}}}_{B\to\hat{B}} = \dot{\boldsymbol{q}}_{B\to I} \otimes \hat{\boldsymbol{q}}_{I\to\hat{B}} + \boldsymbol{q}_{B\to I} \otimes \dot{\hat{\boldsymbol{q}}}_{I\to\hat{B}} \approx \begin{pmatrix} \delta\dot{\boldsymbol{\theta}}^B/\,2 \\ 0 \end{pmatrix} \tag{B.2}$$

where

$$\dot{\boldsymbol{q}}_{B\to I} = \tfrac{1}{2}\, \boldsymbol{q}_{B\to I} \otimes \boldsymbol{\omega}^I_{I/B} \tag{B.3}$$

$$\begin{aligned}\dot{\hat{\boldsymbol{q}}}_{I\to\hat{B}} &= \tfrac{1}{2}\,\hat{\boldsymbol{q}}_{I\to\hat{B}}\otimes\,\hat{\boldsymbol{\omega}}^{B}_{B/I}\\ &= \tfrac{1}{2}\,\hat{\boldsymbol{q}}_{I\to\hat{B}}\otimes\left[\boldsymbol{\omega}^{B}_{B/I}+\delta\boldsymbol{\omega}^{B}_{B/I}\right]\end{aligned}\tag{B.4}$$

The term $\boldsymbol{\omega}^{I}_{I/B}$ is the angular rate of the inertial frame with respect to the body frame, coordinitized in the inertial coordinate system, and $\hat{\boldsymbol{\omega}}^{B}_{B/I}$ is the angular rate of the body frame with respect to the inertial frame, coordinitized in the body coordinate system. Substituting equations B.3 and B.4 into equation B.2 produces

$$\dot{\hat{\boldsymbol{q}}}_{B\to\hat{B}} = \left(\tfrac{1}{2}\,\boldsymbol{q}_{B\to I}\otimes\,\boldsymbol{\omega}^{I}_{I/B}\otimes\,\hat{\boldsymbol{q}}_{I\to\hat{B}}\right)+\left(\tfrac{1}{2}\,\boldsymbol{q}_{B\to I}\otimes\,\hat{\boldsymbol{q}}_{I\to\hat{B}}\otimes\left[\boldsymbol{\omega}^{B}_{B/I}+\delta\boldsymbol{\omega}^{B}_{B/I}\right]\right)\tag{B.5}$$

Replacing both occurrences of $\hat{\boldsymbol{q}}_{I\to\hat{B}}$ in equation B.5 with

$$\hat{\boldsymbol{q}}_{I\to\hat{B}} = \boldsymbol{q}_{I\to B}\otimes\,\hat{\boldsymbol{q}}_{B\to\hat{B}}$$

and using the relationships

$$\boldsymbol{\omega}^{B}_{I/B} = \boldsymbol{q}_{B\to I}\otimes\,\boldsymbol{\omega}^{I}_{I/B}\otimes\,\boldsymbol{q}_{I\to B}$$

and

$$\boldsymbol{q}_{B\to I}\otimes\,\boldsymbol{q}_{I\to B} = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \{Identity\}$$

allows the equation to simplify into

$$\dot{\hat{\boldsymbol{q}}}_{B\to\hat{B}} = \tfrac{1}{2}\,\boldsymbol{\omega}^{B}_{I/B}\otimes\,\hat{\boldsymbol{q}}_{B\to\hat{B}}\;+\;\tfrac{1}{2}\,\hat{\boldsymbol{q}}_{B\to\hat{B}}\otimes\left[\boldsymbol{\omega}^{B}_{B/I}+\delta\boldsymbol{\omega}^{B}_{B/I}\right]\tag{B.6}$$

Equation B.6 can be rewritten as

$$\begin{aligned}\dot{\hat{\boldsymbol{q}}}_{B\to\hat{B}} &= -\frac{1}{2}\begin{pmatrix}\boldsymbol{\omega}^{B}_{B/I}\\0\end{pmatrix}\otimes\begin{pmatrix}\delta\boldsymbol{\theta}^{B}/\,2\\1\end{pmatrix}\;+\;\frac{1}{2}\begin{pmatrix}\delta\boldsymbol{\theta}^{B}/\,2\\1\end{pmatrix}\otimes\begin{pmatrix}\boldsymbol{\omega}^{B}_{B/I}+\delta\boldsymbol{\omega}^{B}_{B/I}\\0\end{pmatrix}\\ &= -\frac{1}{4}\begin{pmatrix}\boldsymbol{\omega}^{B}_{B/I}\\0\end{pmatrix}\otimes\begin{pmatrix}\delta\boldsymbol{\theta}^{B}\\2\end{pmatrix}\;+\;\frac{1}{4}\begin{pmatrix}\delta\boldsymbol{\theta}^{B}\\2\end{pmatrix}\otimes\begin{pmatrix}\boldsymbol{\omega}^{B}_{B/I}+\delta\boldsymbol{\omega}^{B}_{B/I}\\0\end{pmatrix}\end{aligned}\tag{B.7}$$

Converting equation B.7 into quaternion matrix multiplication and dropping the $\boldsymbol{\omega}^B$ "body frame with respect to inertial frame" subscript for clarity results in

$$\begin{aligned} \dot{\hat{\boldsymbol{q}}}_{B\rightarrow\hat{B}} =& -\frac{1}{4}\begin{bmatrix} \boldsymbol{\omega}^B_{\otimes} & \boldsymbol{\omega}^B \\ (-\boldsymbol{\omega}^B)^T & 0 \end{bmatrix}\begin{pmatrix} \delta\boldsymbol{\theta}^B \\ 2 \end{pmatrix} \\ &+\frac{1}{4}\begin{bmatrix} (2I+\delta\boldsymbol{\theta}^B_{\otimes}) & \delta\boldsymbol{\theta}^B \\ (-\delta\boldsymbol{\theta}^B)^T & 2 \end{bmatrix}\begin{pmatrix} \boldsymbol{\omega}^B+\delta\boldsymbol{\omega}^B \\ 0 \end{pmatrix} \end{aligned} \tag{B.8}$$

Expanding the vector-matrix multiplication produces

$$\begin{aligned} \dot{\hat{\boldsymbol{q}}}_{B\rightarrow\hat{B}} =& -\frac{1}{4}\begin{pmatrix} \boldsymbol{\omega}^B\times\delta\boldsymbol{\theta}^B \;+\; 2\boldsymbol{\omega}^B \\ (-\boldsymbol{\omega}^B)^T\;\delta\boldsymbol{\theta}^B \end{pmatrix} \\ &+\frac{1}{4}\begin{pmatrix} 2(\boldsymbol{\omega}^B+\delta\boldsymbol{\omega}^B) \;+\; \delta\boldsymbol{\theta}^B\times(\boldsymbol{\omega}^B+\delta\boldsymbol{\omega}^B) \\ (-\delta\boldsymbol{\theta}^B)^T\;(\boldsymbol{\omega}^B+\delta\boldsymbol{\omega}^B) \end{pmatrix} \end{aligned} \tag{B.9}$$

Neglecting second-order terms and equating the result with the approximate $\dot{\hat{\boldsymbol{q}}}_{B\rightarrow\hat{B}}$ shown in equation B.2 gives

$$\begin{aligned} \begin{pmatrix} \delta\dot{\boldsymbol{\theta}}^B/\,2 \\ 0 \end{pmatrix} \approx& \begin{pmatrix} -(\boldsymbol{\omega}^B\times\delta\boldsymbol{\theta}^B)/4-\boldsymbol{\omega}^B/2 \\ (\boldsymbol{\omega}^B)^T\;\delta\boldsymbol{\theta}^B/4 \end{pmatrix} \\ &+\begin{pmatrix} \boldsymbol{\omega}^B/2+\delta\boldsymbol{\omega}^B/2 \;+\; (\delta\boldsymbol{\theta}^B\times\boldsymbol{\omega}^B)/4 \\ (-\delta\boldsymbol{\theta}^B)^T\;\boldsymbol{\omega}^B/4 \end{pmatrix} \end{aligned} \tag{B.10}$$

Extracting the vector portion of the above expression provides the approximate relationship for the time derivative of the three angle body-frame attitude error.

$$\delta\dot{\boldsymbol{\theta}}^B = \;\delta\boldsymbol{\omega}^B -\; \boldsymbol{\omega}^B\times\delta\boldsymbol{\theta}^B \tag{B.11}$$

Appendix C

Mars Topographic Analysis

This appendix discusses a Mars topographic statistical analysis used for the radar altimeter error model. This model is used within the navigation filter to estimate the measurements of the altimeter. Two of the error terms within the model (k_a, x_t) are elements of the state and therefore have their statistical behavior represented within the state error covariance. As discussed in Section 4.2.2, the estimated radar altimeter measurement is modeled as

$$\hat{h}_m = \hat{h}\left(1 + \hat{k}_a\right) + \hat{x}_t \tag{4.40}$$

The Markov process bias, x_t, is a representation of the variations in surface topography seen by the altimeter. The discrete dynamic description of this Markov process is

$$x_{t\,i+1} = e^{\frac{-\Delta t}{\tau}} x_{t\,i} + w_i \tag{4.42}$$

where w_i is the driving white noise term and

$$\begin{aligned} x_{t\,0} &\sim N(0, \sigma_s^2) \\ w_i &\sim N(0, \tfrac{2\,\sigma_s^2}{\tau}\,\Delta t) \end{aligned}$$

The time constant is defined as

$$\tau = \frac{d}{v_{rel}}$$

and σ_s^2 is the steady-state variance of x_t. The constant d is a reference length of major topographic features and v_{rel} is the surface relative velocity. The time constant is inversely proportional to relative velocity because increased velocity would result in fast variations in topography below the vehicle. This corresponds to a decrease in the time constant, $(v_{rel} \to \infty) \Leftrightarrow (\tau \to 0)$, meaning that the Markov term should approach white noise as velocity goes to infinity. The converse of this makes physical sense as well. A zero surface velocity would have the topography below appearing as a constant random value. This corresponds to an infinite time constant, $(v_{rel} \to 0) \Leftrightarrow (\tau \to \infty)$.

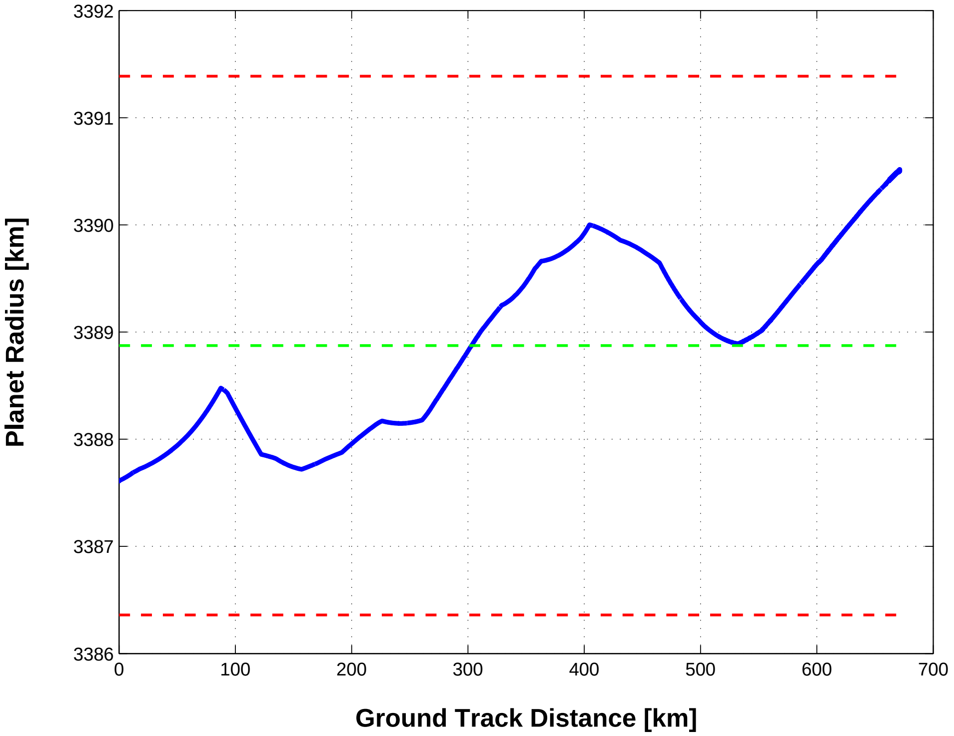


Figure C-1: Mars Topography for Entry Trajectory Ground Track

A sample of the Mars topography beneath the nominal flight trajectory was collected using data from the Mars Orbiter Laser Altimeter (MOLA) aboard the Mars Global Surveyor (MGS) spacecraft. Figure C-1 shows the collected data as planet radius variations versus the traversed ground track distance. The data mean and 3-σ

values are represented as horizontal dashed lines on the plot. The standard deviation (σ_s) of the data indicates a vertical scatter of the topography around the mean radius, m_s. The reference length of major topographic features, d, is a horizontal measure of distance. The resulting statistics from the topography are listed in Table C.1.

Description	**Value**	**Units**
topography mean, m_s	3388.373	$[km]$
topography standard deviation, σ_s	0.8379	$[km]$
reference length, r_{mars}	90	$[km]$

Table C.1: Mars Topography Statistics

Appendix D

State Dynamics Equation

The state dynamics equation is defined as

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{A}\,\delta\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{\epsilon} \tag{4.70}$$

where $\boldsymbol{A}$ is the state dynamics matrix, $\delta\boldsymbol{x}$ is the state error vector, and $\boldsymbol{B}$ is the coefficient matrix for the white noise vector $\boldsymbol{\epsilon}$. These terms are defined as

$$\delta\boldsymbol{x} = \begin{pmatrix} \delta\boldsymbol{r}^{I} & \delta\boldsymbol{v}^{I} & \delta\boldsymbol{\theta}^{B} & \delta\boldsymbol{b}_g & \delta\boldsymbol{s}_g & \delta\boldsymbol{\gamma}_g & \delta\boldsymbol{b}_a & \delta\boldsymbol{s}_a & \delta\boldsymbol{\gamma}_a & \delta x_t & \delta k_a & \delta k_v \end{pmatrix}^T$$

$$\boldsymbol{\epsilon} = \begin{pmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_g \\ \boldsymbol{\epsilon}_{accel} \\ w \end{pmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \left.\dfrac{\partial\, \boldsymbol{g}}{\partial\, \boldsymbol{r}^I}\right|_{\hat{\boldsymbol{r}}^I} & 0 & -\underset{\hat{B}\to I}{\hat{\boldsymbol{T}}}(\boldsymbol{a}_m^B)_\otimes & 0 & 0 & 0 & -\underset{\hat{B}\to I}{\hat{\boldsymbol{T}}} & -\underset{\hat{B}\to I}{\hat{\boldsymbol{T}}}\boldsymbol{D}_a & -\underset{\hat{B}\to I}{\hat{\boldsymbol{T}}}\boldsymbol{F}_a & 0 & 0 & 0 \\ 0 & 0 & -(\boldsymbol{\omega}_m^B)_\otimes & -\boldsymbol{I} & -\boldsymbol{D}_g & -\boldsymbol{F}_g & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \underset{\hat{B}\to I}{\hat{\boldsymbol{T}}} & 0 & \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\right] = \boldsymbol{Q}_{\boldsymbol{\epsilon}}$$

where

$$\boldsymbol{Q}_{\boldsymbol{\epsilon}} = \left[\begin{array}{llll} \boldsymbol{Q}_a & 0 & 0 & 0 \\ 0 & \boldsymbol{Q}_g & 0 & 0 \\ 0 & 0 & \boldsymbol{Q}_{accel} & 0 \\ 0 & 0 & 0 & Q_{alt} \end{array}\right]$$

See equation 4.80, page 76.

Appendix E

Initial Filter Covariance at Atmosphere Interface

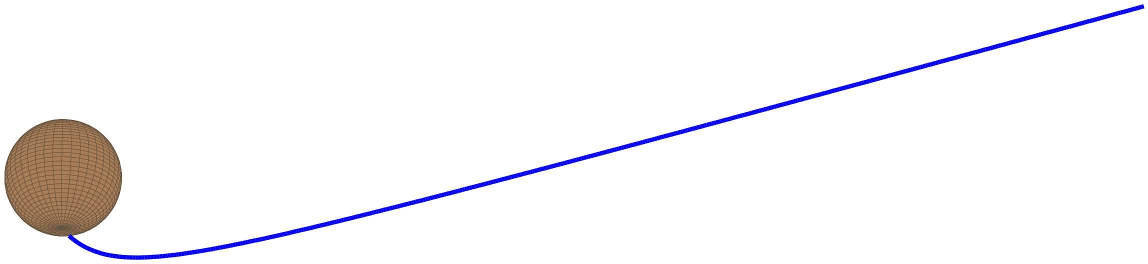


Figure E-1: 12 Hour Cruise Trajectory Prior to Atmosphere Interface

As the entry vehicle approaches Mars, final navigation measurements are used to initialize the onboard navigation filter. This update is performed using the Deep Space Network (DSN) earth-based antenna for position and velocity and a cruise stage mounted star camera for attitude. This occurs approximately twelve hours prior to atmospheric entry interface (EI). From this point, the entry vehicle separates from the cruise stage and begins its coast to EI. The position and velocity correlation matrix for this initial EI-12 hour point is provided below.

$$\begin{bmatrix} 1 & 0.9262992 & 0.6755059 & -0.9967651 & -0.8826522 & -0.745258 \\ 0.9262992 & 1 & 0.4450082 & -0.931506 & -0.8566474 & -0.5574377 \\ 0.6755059 & 0.4450082 & 1 & -0.6388666 & -0.7219254 & -0.9516894 \\ -0.9967651 & -0.931506 & -0.6388666 & 1 & 0.8488419 & 0.6976657 \\ -0.8826522 & -0.8566474 & -0.7219254 & 0.8488419 & 1 & 0.8665107 \\ -0.745258 & -0.5574377 & -0.9516894 & 0.6976657 & 0.8665107 & 1 \end{bmatrix}$$

This matrix is in the LVLH coordinate system organized in the order of altitude, cross track, and downrange. The corresponding 1-σ position and velocity errors in meters and meters-per-second are

$$\tilde{x} = \begin{pmatrix} 1.643865771e+04 & 4.942183642e+03 & 3.991479429e+03 \end{pmatrix}$$
$$\tilde{v} = \begin{pmatrix} 2.768808352e-01 & 1.386667238e-01 & 9.012280708e-02 \end{pmatrix}$$

The star camera attitude measurement system provides attitude information to the navigation system with a 20 arcsecond (1-σ) accuracy per axis.

The entry vehicle navigation filter uses the initial state information and this associated uncertainty to propagate the state information through time. No additional external measurements or navigation aids are available during this final cruise phase. The IMU gyros are used during this "dead-reckoning" orbital cruise period to assist in the state estimation. Two IMU systems are under consideration. The Honeywell MIMU and Litton LN200. Using the performance specifications of the gyros within these two IMUs, the estimated states and error covariance are propagated forward in time to EI. The resulting position/velocity error covariance matrix associated with the ARES entry interface location after this 12 hour cruise is

$$\begin{bmatrix} 7.3472E{+}06 & 1.1651E{+}07 & -1.5816E{+}07 & -9.1142E{+}03 & -3.0358E{+}02 & -8.6052E{+}03 \\ 1.1651E{+}07 & 2.6611E{+}07 & -4.1133E{+}07 & -2.0888E{+}04 & 1.3634E{+}03 & -2.3854E{+}04 \\ -1.5816E{+}07 & -4.1133E{+}07 & 6.9142E{+}07 & 3.3660E{+}04 & -4.3329E{+}03 & 4.0160E{+}04 \\ -9.1142E{+}03 & -2.0888E{+}04 & 3.3660E{+}04 & 1.6924E{+}01 & -1.6104E{+}00 & 1.9336E{+}01 \\ -3.0358E{+}02 & 1.3634E{+}03 & -4.3329E{+}03 & -1.6104E{+}00 & 9.6137E{-}01 & -2.5465E{+}00 \\ -8.6052E{+}03 & -2.3854E{+}04 & 4.0160E{+}04 & 1.9336E{+}01 & -2.5465E{+}00 & 2.3540E{+}01 \end{bmatrix}$$

This matrix is presented in the inertial X-Y-Z coordinate system in meters and meters-per-second. The entry vehicle is spinning during the 12-hour cruise. The resulting roll/pitch/yaw attitude errors are presented in Table E.1. These are the attitude errors that result from a 12 hour cruise to EI using the specified gyros.

	MIMU	**LN200**
roll [rad]	0.03767	0.75352
pitch [rad]	0.00037	0.00435
yaw [rad]	0.00037	0.00435

Table E.1: Entry Interface Attitude Errors

For a secondary analysis, the LN200 is updated with a star camera measurement one hour prior to EI. The pre-EI analysis is executed again, but from one hour before EI and with the attitude state uncertainty reset to 20 arcseconds for each attitude axis. The resulting EI attitude uncertainty for this scenario is seen in Table E.2.

	LN200
roll [rad]	0.06281
pitch [rad]	0.00126
yaw [rad]	0.00126

Table E.2: LN200 Entry Interface Errors for EI-1Hr Attitude Update

Appendix F

Additional Monte Carlo Results

Figures F-1 through F-3 present the LN200/MER limited run Monte Carlo results for the scenario where a star camera attitude update is made available one hour prior to atmospheric interface. Figures F-4 through F-6 present the same results for the LN200/MSP instrument scenario.

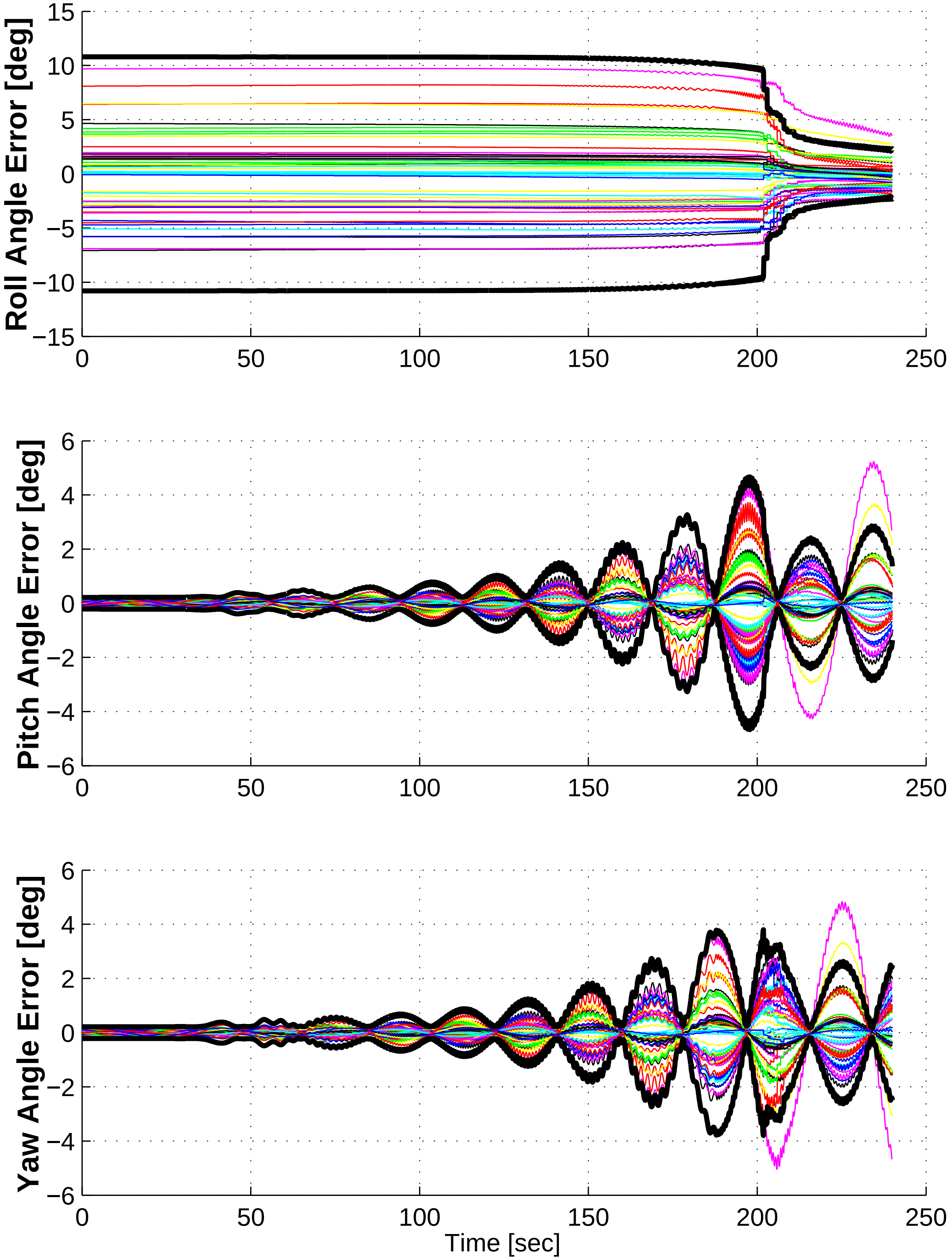


Figure F-1: Augmented LN200\MER Scenario: Attitude Error Monte Carlo Analysis

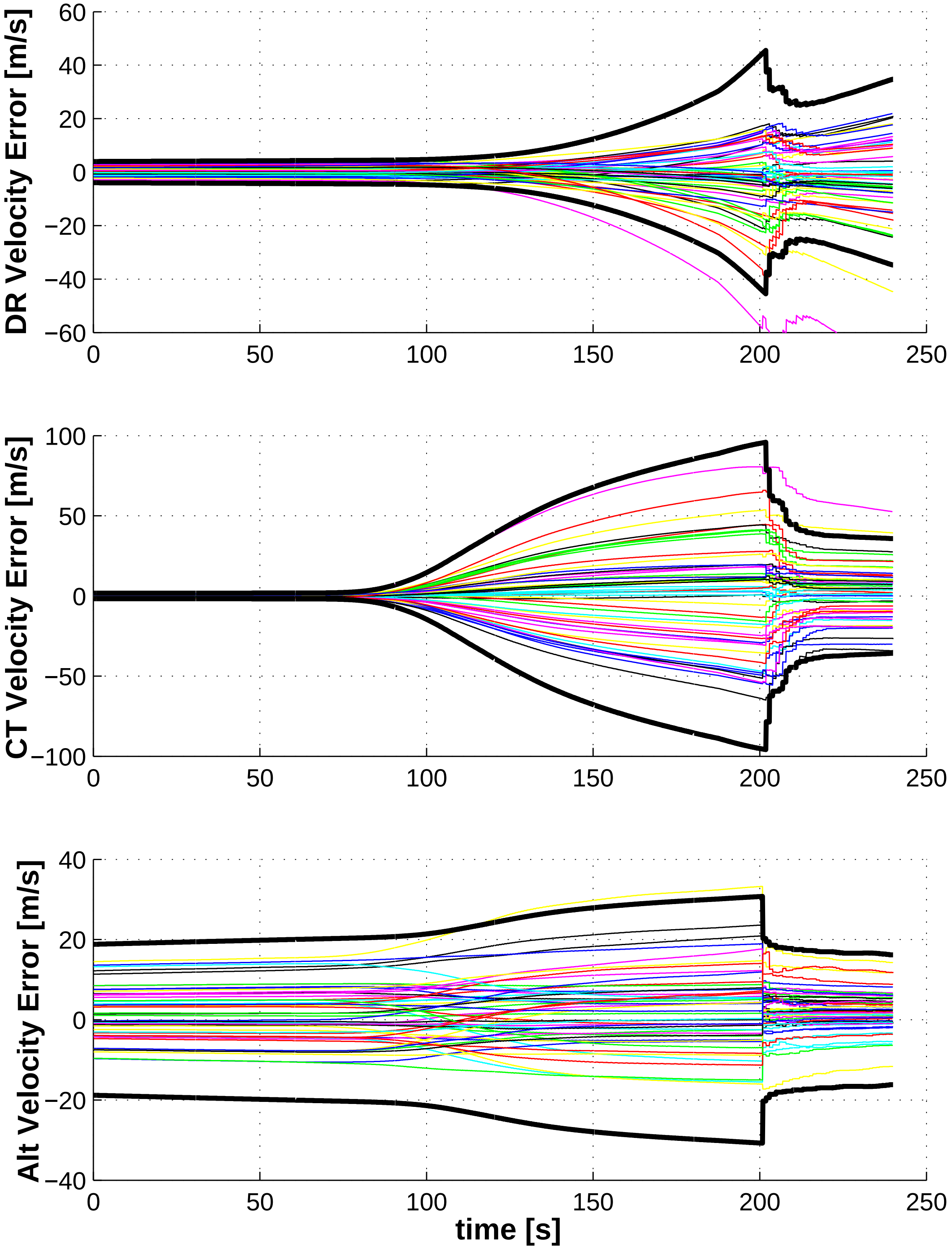


Figure F-2: Augmented LN200\MER Scenario: Velocity Error Monte Carlo Analysis

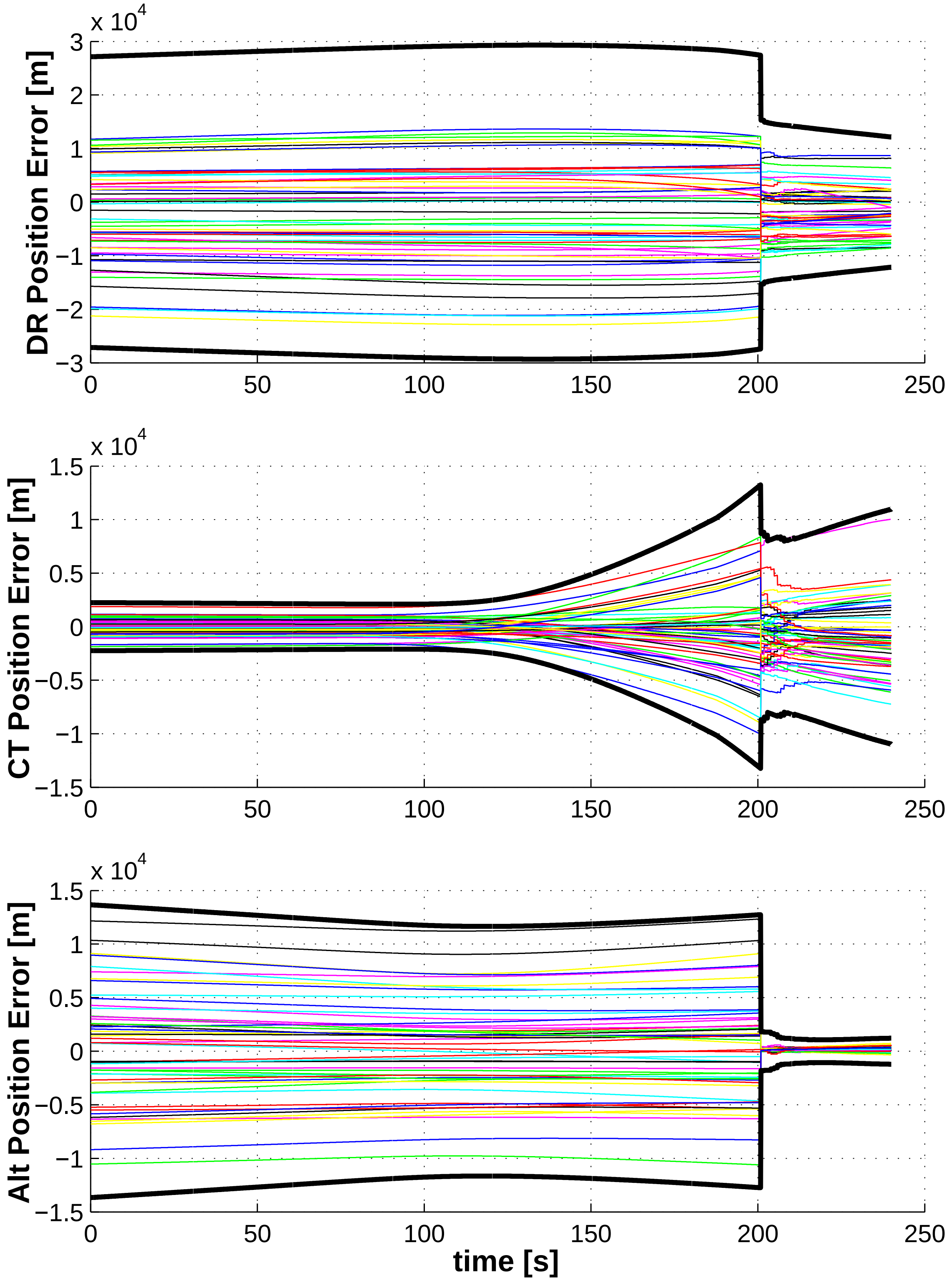


Figure F-3: Augmented LN200\MER Scenario: Position Error Monte Carlo Analysis

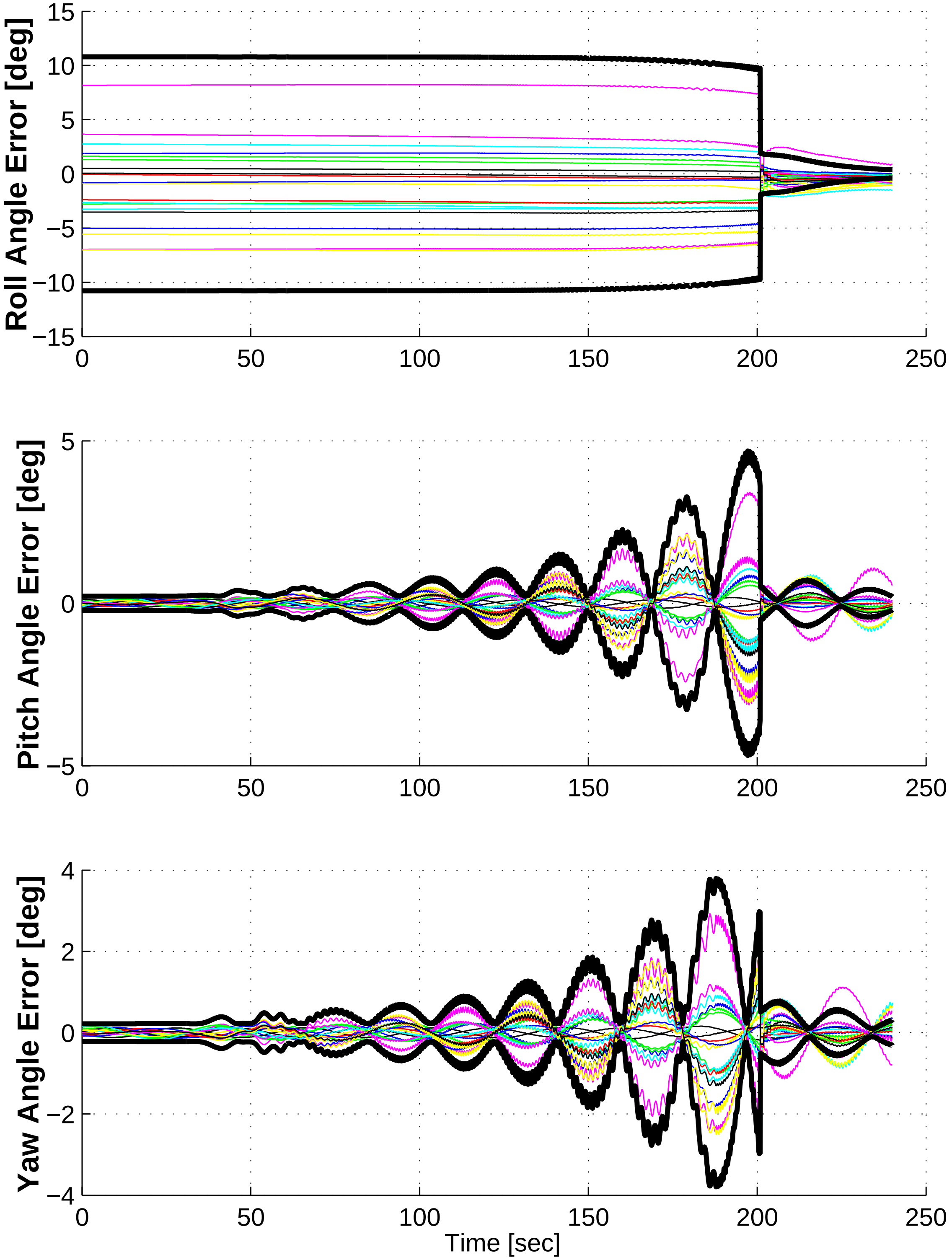


Figure F-4: Augmented LN200\MSP Scenario: Attitude Error Monte Carlo Analysis

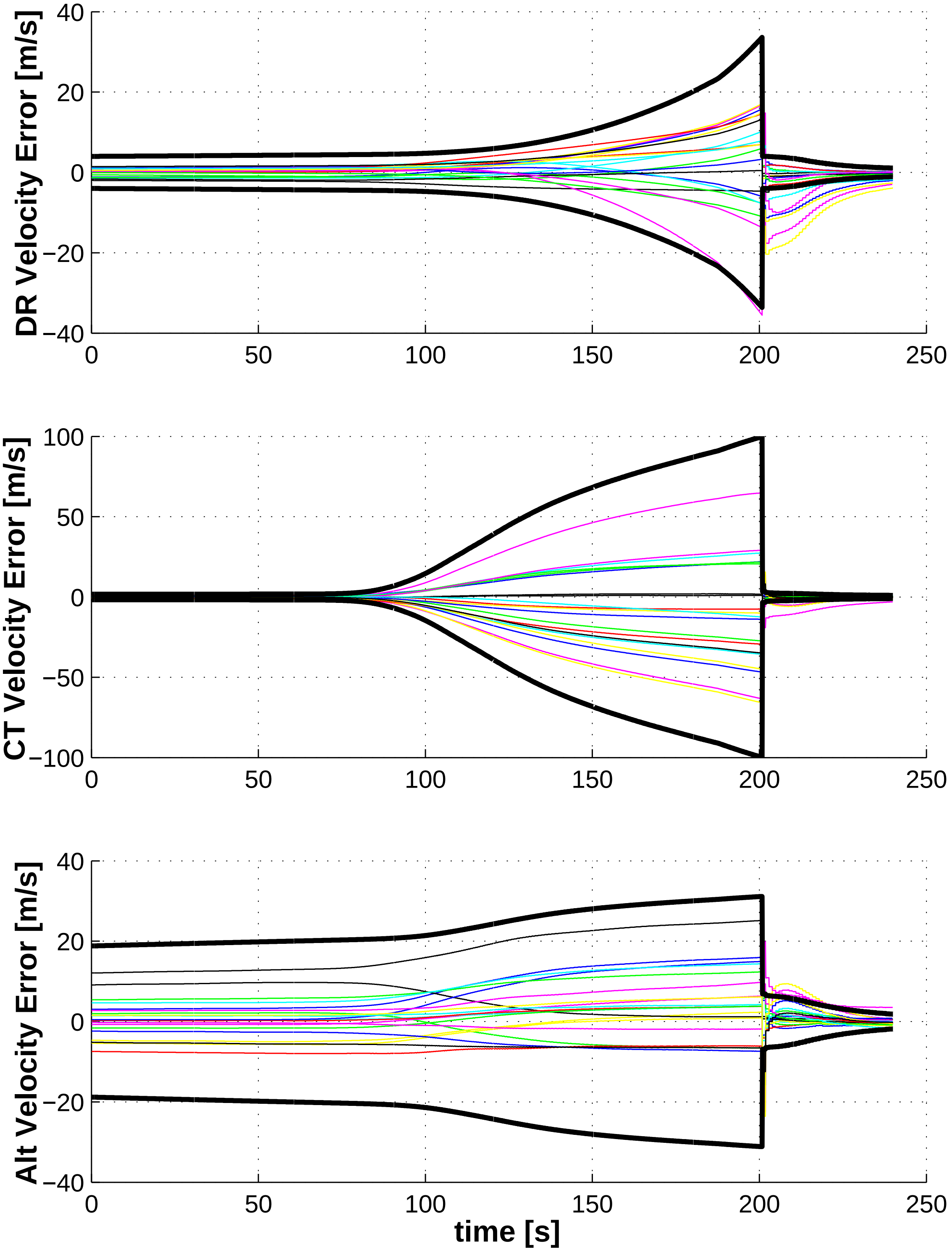


Figure F-5: Augmented LN200\MSP Scenario: Velocity Error Monte Carlo Analysis

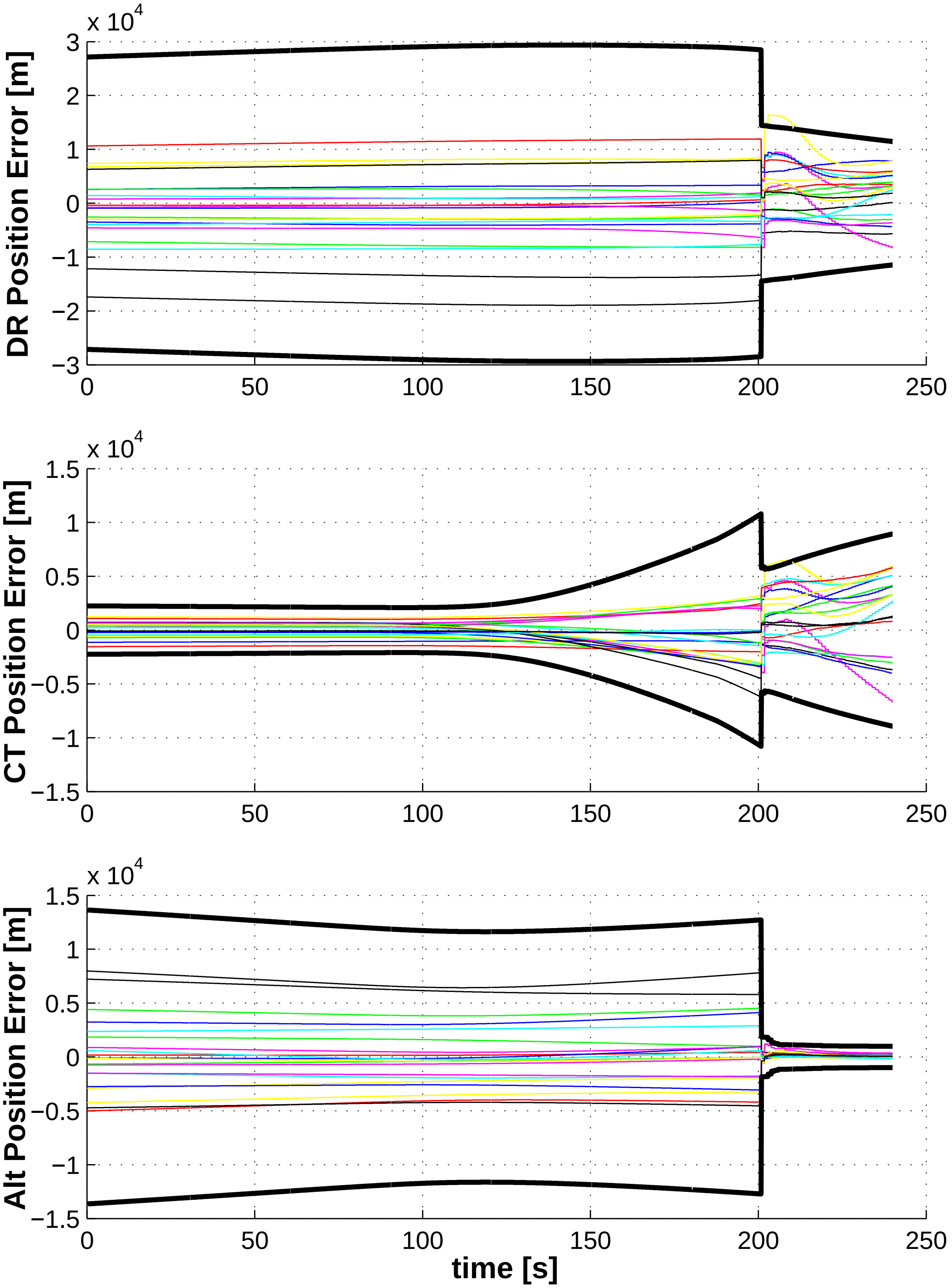


Figure F-6: Augmented LN200\MSP Scenario: Position Error Monte Carlo Analysis

Appendix G

Measurement Sensitivity Matrix Derivation

Altimeter Partial Derivative

The derivation of the altimeter partial derivative is as follows. The altimeter model is formulated as

$$h_m = h\left(1 + k_a\right) + x_t + \nu_a$$

where h_m is the measured distance, h is the true distance, k_a is the scale factor, x_t is the Markov topography process, and ν_a is the sensor noise. The partial with respect to several of the states is zero.

$$\frac{\partial h_m}{\partial \boldsymbol{v}^I} = \frac{\partial h_m}{\partial \boldsymbol{b}_g} = \frac{\partial h_m}{\partial \boldsymbol{s}_g} = \frac{\partial h_m}{\partial \boldsymbol{\gamma}_g} = \frac{\partial h_m}{\partial \boldsymbol{b}_a} = \frac{\partial h_m}{\partial \boldsymbol{s}_a} = \frac{\partial h_m}{\partial \boldsymbol{\gamma}_a} = \frac{\partial h_m}{\partial k_v} = 0$$

The partial with respect to x_t is

$$\frac{\partial h_m}{\partial x_t} = 1$$

The partial with respect to k_a is

$$\frac{\partial h_m}{\partial k_a} = h$$

The partial with respect to position is

$$\frac{\partial h_m}{\partial \boldsymbol{r}^I} = (1+k_a)\frac{\partial h}{\partial \boldsymbol{r}^I}$$

The partial of h with respect to position is found by differentiating the quadratic equation of geometry

$$\left|\boldsymbol{r}^I\right|^2 + 2\,\boldsymbol{r}^I \cdot (\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B)\,h + h^2 = R_s^2$$

with respect to position.

$$2(\boldsymbol{r}^I)^T + 2(\boldsymbol{r}^I)^T\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\,\frac{\partial h}{\partial \boldsymbol{r}^I} + 2\,h\frac{\partial}{\partial \boldsymbol{r}^I}\Big((\boldsymbol{r}^I)^T\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\Big) + 2\,h\frac{\partial h}{\partial \boldsymbol{r}^I} = 0$$

$$(\boldsymbol{r}^I)^T + (\boldsymbol{r}^I)^T\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\,\frac{\partial h}{\partial \boldsymbol{r}^I} + \,h(\boldsymbol{i}_a^B)^T\,\boldsymbol{T}_{I\to B} + \,h\frac{\partial h}{\partial \boldsymbol{r}^I} = 0$$

Solving this equation for the partial derivative produces

$$\frac{\partial h}{\partial \boldsymbol{r}^I} = \frac{-\left[\,(\boldsymbol{r}^I)^T + \,h\,(\boldsymbol{i}_a^B)^T\boldsymbol{T}_{I\to B}\right]}{(\boldsymbol{r}^I)^T\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B + \,h}$$

The partial with respect to attitude is

$$\frac{\partial h_m}{\partial \boldsymbol{\theta}} = (1+k_a)\frac{\partial h}{\partial \boldsymbol{\theta}}$$

The partial of h with respect to attitude is also found by differentiating the quadratic equation of geometry

$$h\,\frac{\partial}{\partial \boldsymbol{\theta}}\Big(2(\boldsymbol{r}^I)^T\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\Big) + \Big(2(\boldsymbol{r}^I)^T\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\Big)\frac{\partial h}{\partial \boldsymbol{\theta}} + 2\,h\frac{\partial h}{\partial \boldsymbol{\theta}} = 0$$

$$h\,\frac{\partial}{\partial \boldsymbol{\theta}}\Big(2(\boldsymbol{r}^I)^T\,\hat{\boldsymbol{T}}_{\hat{B}\to I}(\boldsymbol{I}-\delta\boldsymbol{\theta}_\otimes)\boldsymbol{i}_a^B\Big) + \Big(2(\boldsymbol{r}^I)^T\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\Big)\frac{\partial h}{\partial \boldsymbol{\theta}} + 2\,h\frac{\partial h}{\partial \boldsymbol{\theta}} = 0$$

$$h\,\frac{\partial}{\partial \boldsymbol{\theta}}\Big(2(\boldsymbol{r}^I)^T\,\hat{\boldsymbol{T}}_{\hat{B}\to I}(-\delta\boldsymbol{\theta}_\otimes)\boldsymbol{i}_a^B\Big) + \Big(2(\boldsymbol{r}^I)^T\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\Big)\frac{\partial h}{\partial \boldsymbol{\theta}} + 2\,h\frac{\partial h}{\partial \boldsymbol{\theta}} = 0$$

$$h\,\frac{\partial}{\partial \boldsymbol{\theta}}\Big(2(\boldsymbol{r}^{I})^{T}\,\hat{\boldsymbol{T}}_{\hat{B}\to I}(\boldsymbol{i}_a^B)_{\otimes}\,\delta\boldsymbol{\theta}\Big) + \Big(2(\boldsymbol{r}^{I})^{T}\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\Big)\frac{\partial h}{\partial \boldsymbol{\theta}} + 2\,h\frac{\partial h}{\partial \boldsymbol{\theta}} = 0$$

Solving this equation for the partial derivative produces

$$\frac{\partial h}{\partial \boldsymbol{\theta}} = \frac{-\Big[h\,(\boldsymbol{r}^{I})^{T}\hat{\boldsymbol{T}}_{\hat{B}\to I}\,\big(\boldsymbol{i}_a^B\big)_{\otimes}\Big]}{(\boldsymbol{r}^{I})^{T}\,\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B + \,h}$$

Velocimeter Partial Derivative

The derivation of the velocimeter partial derivative is as follows. The model for this instrument is formulated as

$$\boldsymbol{v}_m^B = \boldsymbol{T}_{I\to B}\,\big(\boldsymbol{v}^I - \boldsymbol{v}_s\big)\,(1 + k_v) + \boldsymbol{\nu}_v$$

where $\boldsymbol{v}_m^B$ is the surface relative measured velocity vector in body coordinates, $\boldsymbol{v}^I$ is the true inertial vehicle velocity, $\boldsymbol{v}_s$ is the planet surface velocity, k_v is the scale factor, and $\boldsymbol{\nu}_v$ is the sensor noise. The surface velocity is defined as

$$\boldsymbol{v}_s = \;\boldsymbol{\omega}\times\boldsymbol{R}_s$$

where $\boldsymbol{\omega}$ is the rotational rate of the planet and $\boldsymbol{R}_s$ is a vector to the point where the radar beam intercepts the planet surface. It is defined as (see Figure 3-4)

$$\boldsymbol{R}_s = \boldsymbol{r}^I + h\,\left(\boldsymbol{T}_{B\to I}\boldsymbol{i}_a^B\right)$$

The partial with respect to several of the states is zero.

$$\frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{b}_g} = \frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{s}_g} = \frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{\gamma}_g} = \frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{b}_a} = \frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{s}_a} = \frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{\gamma}_a} = \frac{\partial \boldsymbol{v}_m^B}{\partial x_t} = \frac{\partial \boldsymbol{v}_m^B}{\partial k_a} = 0$$

The partial with respect to velocity is

$$\begin{aligned}\frac{\partial \boldsymbol{v}_m^B}{\partial \boldsymbol{v}^I} &= \frac{\partial}{\partial \boldsymbol{v}^I}\left(\boldsymbol{T}_{I\to B}\boldsymbol{v}^I(1 + k_v)\right)\\ &= (1 + k_v)\boldsymbol{T}_{I\to B}\end{aligned}$$

The partial with respect to k_v is

$$\begin{aligned}\frac{\partial \boldsymbol{v}_m^B}{\partial k_v} &= \frac{\partial}{\partial k_v}\big(\boldsymbol{T}_{I\rightarrow B}(\boldsymbol{v}^I - \boldsymbol{v}_s)(1+k_v)\big)\\ &= \boldsymbol{T}_{I\rightarrow B}(\boldsymbol{v}^I - \boldsymbol{v}_s)\end{aligned}$$

The partial with respect to position is

$$\begin{aligned}\frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{r}^I} &= \frac{\partial}{\partial \boldsymbol{r}^I}\big(-\boldsymbol{T}_{I\rightarrow B}\boldsymbol{v}_s(1+k_v)\big)\\ &= \frac{\partial}{\partial \boldsymbol{r}^I}\big(-\boldsymbol{T}_{I\rightarrow B}\left[\boldsymbol{\omega}_{\otimes}\boldsymbol{r}^I + \boldsymbol{\omega}_{\otimes} h\, \boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\ \right](1+k_v)\big)\\ &= -(1+k_v)\boldsymbol{T}_{I\rightarrow B}\boldsymbol{\omega}_{\otimes}\left[\boldsymbol{I} + \boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\frac{\partial \hat{h}}{\partial \boldsymbol{r}^I}\right]\end{aligned}$$

The partial with respect to attitude is

$$\begin{aligned}\frac{\partial \hat{\boldsymbol{v}}_m^B}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}}\big(-\boldsymbol{T}_{I\rightarrow B}\boldsymbol{v}_s(1+k_v)\big)\\ &= -(1+k_v)\left[\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{T}_{I\rightarrow B}(\boldsymbol{\theta})\boldsymbol{v}_s) + \frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{T}_{I\rightarrow B}\boldsymbol{v}_s(\boldsymbol{\theta}))\right]\\ &= -(1+k_v)\left[\frac{\partial}{\partial \boldsymbol{\theta}}((\boldsymbol{I}+\delta\boldsymbol{\theta}_{\otimes})\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s) + \boldsymbol{T}_{I\rightarrow B}\frac{\partial}{\partial \boldsymbol{\theta}}(\,h\,\boldsymbol{\omega}_{\otimes}\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B)\right]\\ &= -(1+k_v)\left[\frac{\partial}{\partial \boldsymbol{\theta}}(\delta\boldsymbol{\theta}_{\otimes}\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s) + \boldsymbol{T}_{I\rightarrow B}\frac{\partial}{\partial \boldsymbol{\theta}}(\,h\,\boldsymbol{\omega}_{\otimes}\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B)\right]\\ &= (1+k_v)\left(\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s\right)_{\otimes} - (1+k_v)\,\boldsymbol{T}_{I\rightarrow B}\left[\boldsymbol{\omega}_{\otimes}\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\frac{\partial h}{\partial \boldsymbol{\theta}} +\, h\,\boldsymbol{\omega}_{\otimes}\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B)\right]\\ &= (1+k_v)\left(\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s\right)_{\otimes} - (1+k_v)\,\boldsymbol{T}_{I\rightarrow B}\left[\boldsymbol{\omega}_{\otimes}\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\frac{\partial h}{\partial \boldsymbol{\theta}} +\, h\,\boldsymbol{\omega}_{\otimes}\hat{\boldsymbol{T}}_{\hat{B}\rightarrow I}\frac{\partial}{\partial \boldsymbol{\theta}}((\boldsymbol{I}-\delta\boldsymbol{\theta}_{\otimes})\boldsymbol{i}_a^B)\right]\\ &= (1+k_v)\left(\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s\right)_{\otimes} - (1+k_v)\,\boldsymbol{T}_{I\rightarrow B}\left[\boldsymbol{\omega}_{\otimes}\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\frac{\partial h}{\partial \boldsymbol{\theta}} +\, h\,\boldsymbol{\omega}_{\otimes}\hat{\boldsymbol{T}}_{\hat{B}\rightarrow I}\frac{\partial}{\partial \boldsymbol{\theta}}(-\delta\boldsymbol{\theta}_{\otimes}\boldsymbol{i}_a^B)\right]\\ &= (1+k_v)\left(\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s\right)_{\otimes} - (1+k_v)\,\boldsymbol{T}_{I\rightarrow B}\boldsymbol{\omega}_{\otimes}\left[\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\frac{\partial h}{\partial \boldsymbol{\theta}} +\, h\,\hat{\boldsymbol{T}}_{\hat{B}\rightarrow I}(\boldsymbol{i}_a^B)_{\otimes}\right]\\ &= (1+k_v)\left[\left(\hat{\boldsymbol{T}}_{I\rightarrow \hat{B}}\boldsymbol{v}_s\right)_{\otimes} -\ \boldsymbol{T}_{I\rightarrow B}\boldsymbol{\omega}_{\otimes}\left(\boldsymbol{T}_{B\rightarrow I}\boldsymbol{i}_a^B\,\frac{\partial h}{\partial \boldsymbol{\theta}} + h\,\hat{\boldsymbol{T}}_{\hat{B}\rightarrow I}(\boldsymbol{i}_a^B)_{\otimes}\right)\right]\end{aligned}$$

Appendix H

LN200 Results Comparison

Figures H-1 through H-3 compare the navigation uncertainty for the nominal ($8.75km$ instrument activation) scenario using the LN200 IMU combined with the two instrument packages under consideration: the MER altimeter, and the MSP altimeter/velocimeter.

Figures H-4 through H-6 show only the terminal principal state errors as a function of initial measurement altitude for this same entry and descent scenario.

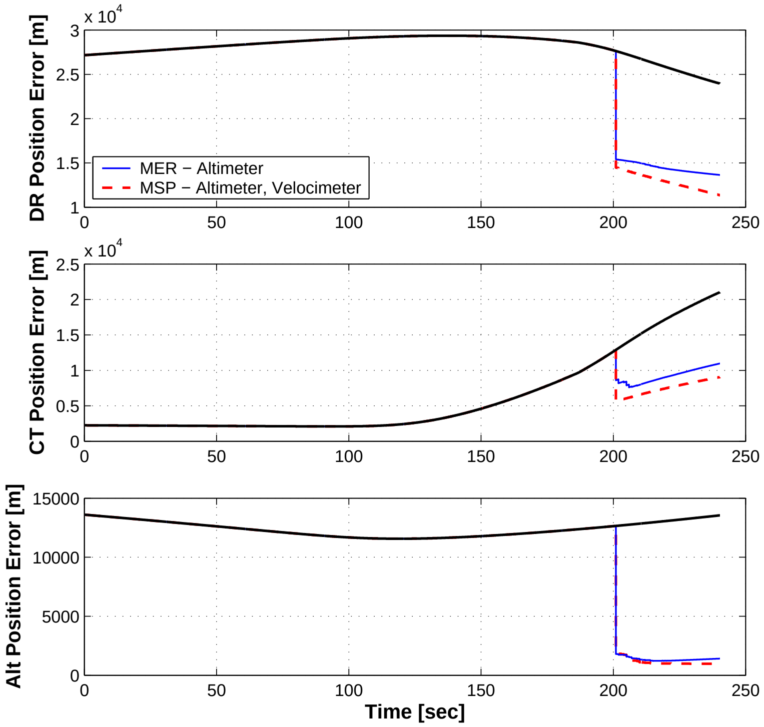


Figure H-1: Nominal LN200 Scenario: Position Error Comparison

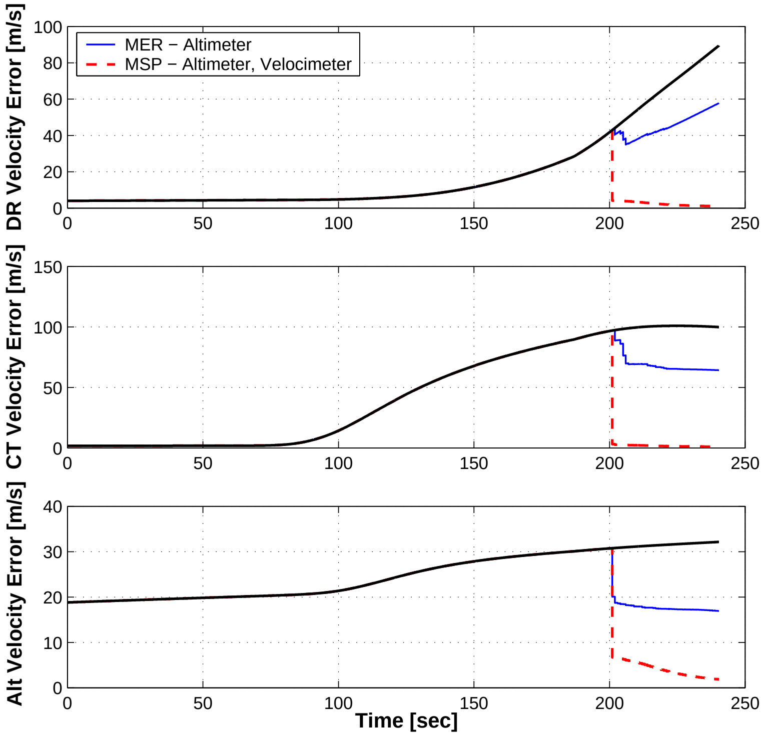


Figure H-2: Nominal LN200 Scenario: Velocity Error Comparison

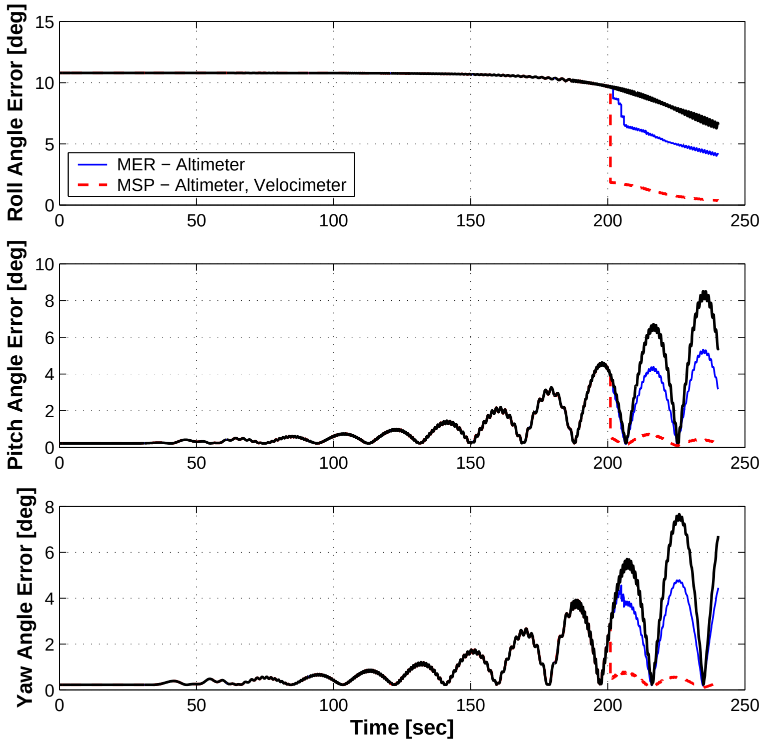


Figure H-3: Nominal LN200 Scenario: Attitude Error Comparison

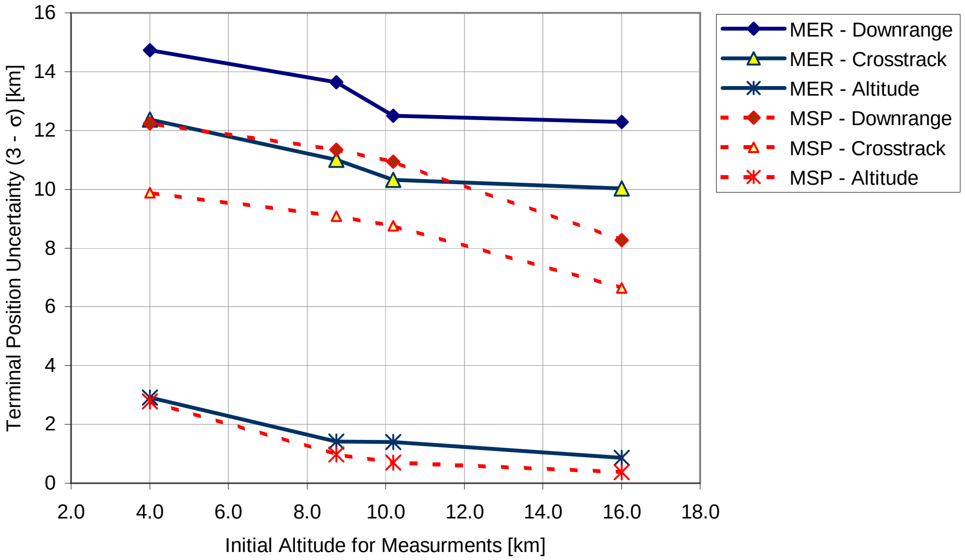


Figure H-4: LN200 Scenario: Terminal Position Error Comparison

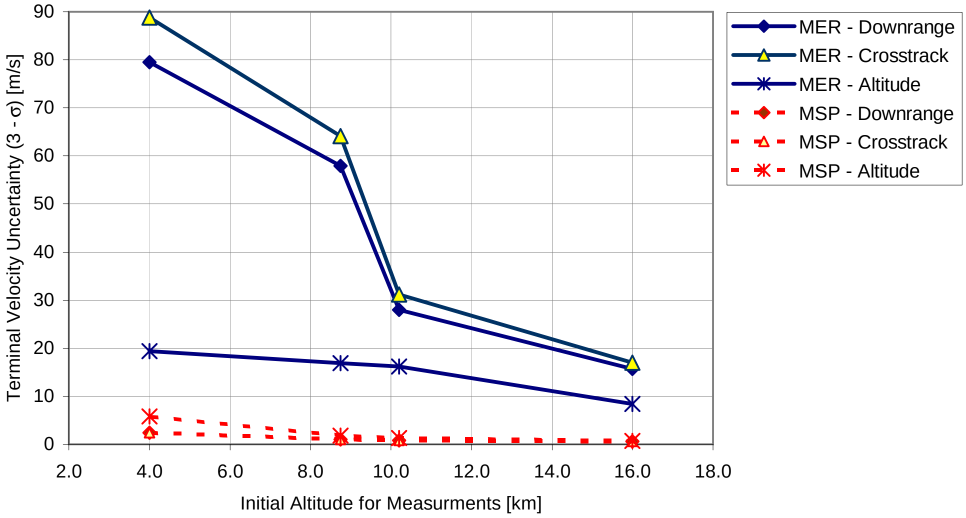


Figure H-5: LN200 Scenario: Terminal Velocity Error Comparison

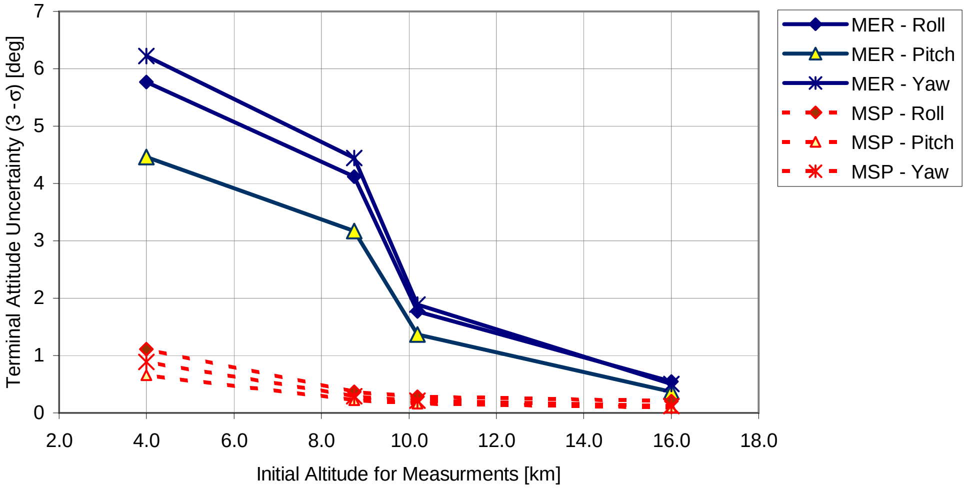


Figure H-6: LN200 Scenario: Terminal Attitude Error Comparison

Bibliography

[1] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston, Virginia, 1999.

[2] Robert Bishop and Timothy Crain. Mars Entry Navigation: Atmospheric Interface Through Parachute Deploy. *AIAA*, (2002-4501), August 2002.

[3] R.A. Blanchard, R.D. Braun, P.H. Kallemeyn, S.W. Thurman, and D.A. Spencer. Mars Pathfinder Entry, Descent, and Landing Reconstruction. In *Journal of Spacecraft and Rockets*, Volume 36, pages 357–366. AIAA, May 1999.

[4] Glen Bobskill, Paresh Parikh, Ramadas Prabhu, and Erik Tyler. Aerodynamic Database Development for Mars Smart Lander Vehicle Configuration. *AIAA*, (2002-4411), 2002.

[5] Robert Brown and Patrick Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, Inc., New York, third edition, 1997.

[6] Russell Carpenter. Strapdown IMU Error Model. *NASA Johnson Space Center*, (EG-97-103), September 1997.

[7] Steven Chapra and Raymond Canale. *Numerical Methods for Engineers.* McGraw-Hill, Boston, third edition, 1998.

[8] Edward J. Fallon. System Design Overview of the Mars Pathfinder Parachute Decelerator Subsystem. *AIAA*, (97-1511), 1997.

[9] Arthur Gelb, editor. *Applied Optimal Estimation.* The MIT Press, Cambridge, Massachusetts, 1974.

[10] C. G. Justus and D. L. Johnson. *Mars Global Rererence Atmospheric Model 2001 Version.* NASA Marshall Space Flight Center, Alabama, 2001.

[11] C. G. Justus and D. L. Johnson. Mars Global Rererence Atmospheric Model 2001 Version: Users Guide. *NASA Marshall Space Flight Center*, (TM-2001-210961), 2001.

[12] William Lear. Kalman Filtering Techniques. *NASA Johnson Space Center*, (JSC-20688), September 1985.

[13] Joel Levin. *ARES - An Aerial Regional-scale Environmental Survey of Mars - Volume 1: Concept Study Report.* NASA Langley Research Center, 2003.

[14] Mary Kae Lockwood, Ken Sutton, Ram Prabhu, Richard Powell, Claude Graves, Chirold Epp, and Gilbert Carman. Entry Configurations and Performance Comparisons For The Mars Smart Lander. *AIAA*, (2002-4407), 2002.

[15] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, Volume 141 of *Mathematics in Science and Engineering.* Academic Press, New York, San Francisco, London, 1979.

[16] Eric M. Queen and Ben Raiszadeh. Mars Smart Lander Parachute Simulation Model. *AIAA*, (2002-4617), 2002.

[17] Ralph Roncoli, Bill Strauss, and Dolan Highsmith. *Mars Exploration Rover Project Planetary Constants and Models - Version 2.* NASA, November 2002. Interoffice Memorandum.

[18] Wallace E. VanderVelde. Class notes, Stochastic Estimation and Control. Massachusetts Institute Of Technology, May 2003.

[19] James R. Wertz. *Spacecraft Attitude Determination and Control.* Kluwer Academic Publishers, Dordrecht, Boston, London, 1978.