5-2012

Statistical Methods for Launch Vehicle Guidance, Navigation, and Control (GN&C) System Design and Analysis

Michael Benjamin Rose
Utah State University

Recommended Citation

STATISTICAL METHODS FOR LAUNCH VEHICLE GUIDANCE, NAVIGATION, AND

CONTROL (GN&C) SYSTEM DESIGN AND ANALYSIS

by

Michael Benjamin Rose

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Approved:

_____   _____
Dr. David K. Geller   Dr. R. Rees Fullmer
Major Professor   Committee Member

_____   _____
Dr. Stephen A. Whitmore   Dr. Charles M. Swenson
Committee Member   Committee Member

_____   _____
Dr. Byard D. Wood   Dr. Mark R. McLellan
Committee Member   Vice President for Research and
                                                    Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2012

Abstract

Statistical Methods for Launch Vehicle Guidance, Navigation, and

Control (GN&C) System Design and Analysis

by

Michael Benjamin Rose, Doctor of Philosophy

Utah State University, 2012

Major Professor: Dr. David K. Geller
Department: Mechanical and Aerospace Engineering

A novel trajectory and attitude control and navigation analysis tool for powered ascent is developed. The tool is capable of rapid trade-space analysis and is designed to ultimately reduce turnaround time for launch vehicle design, mission planning, and redesign work. It is streamlined to quickly determine trajectory and attitude control dispersions, propellant dispersions, orbit insertion dispersions, and navigation errors and their sensitivities to sensor errors, actuator execution uncertainties, and random disturbances.

The tool is developed by applying both Monte Carlo and linear covariance analysis techniques to a closed-loop, launch vehicle guidance, navigation, and control (GN&C) system. The nonlinear dynamics and flight GN&C software models of a closed-loop, six-degree-of-freedom (6-DOF), Monte Carlo simulation are formulated and developed. The nominal reference trajectory (NRT) for the proposed lunar ascent trajectory is defined and generated. The Monte Carlo truth models and GN&C algorithms are linearized about the NRT, the linear covariance equations are formulated, and the linear covariance simulation is developed.

The performance of the launch vehicle GN&C system is evaluated using both Monte Carlo and linear covariance techniques and their trajectory and attitude control dispersion, propellant dispersion, orbit insertion dispersion, and navigation error results are validated and compared. Statistical results from linear covariance analysis are generally within 10% of Monte Carlo results, and in most cases the differences are less than 5%. This is an excellent result given the many complex nonlinearities that are embedded in the ascent GN&C problem. Moreover, the real value

of this tool lies in its speed, where the linear covariance simulation is 1036.62 times faster than the Monte Carlo simulation. Although the application and results presented are for a lunar, single-stage-to-orbit (SSTO), ascent vehicle, the tools, techniques, and mathematical formulations that are discussed are applicable to ascent on Earth or other planets as well as other rocket-powered systems such as sounding rockets and ballistic missiles.

(230 pages)

Public Abstract

Statistical Methods for Launch Vehicle Guidance, Navigation, and

Control (GN&C) System Design and Analysis

A new tool for launch vehicle design and analysis is developed. The tool is capable of rapid analysis of requirements tradeoffs affecting system design and developed to reduce turnaround time for launch vehicle design and mission planning. It is streamlined to quickly determine trajectory and attitude control dispersions which represent how far the actual trajectory is expected to deviate from the nominal flight path, propellant dispersions which represent how much propellant is required to meet mission requirements, and navigation errors which represent how far the navigation filter's estimate of the actual trajectory is expected to deviate from the actual trajectory. Moreover, the tool is able to measure sensitivities to instrument errors, engine performance uncertainties, and random disturbances.

The tool is developed by applying both Monte Carlo and linear covariance analysis techniques to a closed-loop, launch vehicle guidance, navigation, and control (GN&C) system. The nonlinear equations, algorithms, and models for a Monte Carlo simulation are formulated and developed. The nominal reference trajectory (NRT) for the proposed lunar ascent trajectory is defined and generated. The nonlinear equations, algorithms, and models associated with the Monte Carlo simulation are linearized about the NRT. The linear covariance equations are formulated and the linear covariance simulation is developed.

The performance of the launch vehicle GN&C system is evaluated using both Monte Carlo and linear covariance simulations and their results are validated and compared. Statistical results from linear covariance analysis are generally within 10% of Monte Carlo results, and in most cases the differences are less than 5%. This is an excellent result given the many complex nonlinearities that are embedded in the ascent GN&C problem. Moreover, the real value of this tool lies in its speed, where the linear covariance simulation is 1036.62 times faster than the Monte Carlo simulation. Although the application and results presented are for lunar ascent, the tools, techniques, and mathematical formulations that are discussed are applicable to launch vehicles on Earth or other planets as well as other rocket-powered systems such as sounding rockets and ballistic missiles. This research was supported in part by National Aeronautics and Space Administration (NASA)

Johnson Space Center and Draper Laboratory.

Michael Benjamin Rose

Acknowledgments

When I first applied to the program, there were some who were skeptical and others who simply did not believe that an astronomer/physicist could ever succeed in engineering. There was one, however, who saw something in me that the others did not, who was willing to give me a chance, and who never showed any doubt that I would be successful. Thanks, Dr. Geller, for being that person, for teaching me so much, and for never giving up on me, even when I had almost given up on myself.

To my professors, Dr. Rees Fullmer, Dr. Charles Swenson, Dr. Todd Moon, and Dr. Wei Ren, my thanks and gratitude for challenging me and sharing with me so much of your knowledge and experience. Your enthusiasm and love for your various fields of research were absolutely contagious and provided the motivation that I needed to learn all that I could during our time together.

To my dear friends, Dr. Dave Woffinden and Dr. Roger Derek West, thanks for leading the way, for helping me understand things that I seemed to struggle with, for your encouragement and wonderful examples, and for lending a listening ear at the times when I needed it the most. I hope that our friendship will never end. To my friend and former advisor, Dr. Eric Hintz, my thanks and gratitude for helping me lay the foundation that helped me to be successful in this endeavor and for the suggestion that I expand my horizons by pursuing a degree in aerospace engineering.

To Dr. George Davis and Dr. Dave Gaylor at Emergent Space Technologies, thanks for the opportunity to work for you and be a part of your team, for your encouragement and support, and for the computer resources that made this research possible. To all of my friends and colleagues at Draper Laboratory in Houston, Emergent Space Technologies, and Lockheed Martin Space Systems Company in Denver, thanks for making me feel welcome and for providing an environment where I could enjoy learning and working. This research was supported in part by National Aeronautics and Space Administration (NASA) Johnson Space Center (contract NNX09AW27G) and Draper Laboratory.

To the love of my life, Kathy, thanks for being at my side through all of this, for your encouragement and support, and for always believing in me. I could not have done this without you. To my children, Taylor, Hailey, Chase, and Jaxon, thanks for loving me even though I seemed to spend all of my time on this instead of with you. Coming home to you and your mother at the

end of each day made it all worth it. To my parents, brothers, and sisters, thanks for your love, support, and encouragement. To my grandmother, Grandma Rose, thanks for the good genes, for your love and support, and for expressing how proud you are of me.

Lastly, but most important of all, I wish to give all thanks and praise to my loving Heavenly Father, who truly made all of this possible.

Michael Benjamin Rose

Contents

List of Tables

List of Figures

Acronyms

ACS	Attitude Control System
c.m.	Center of Mass
CSD	Complex Step Derivative
DCM	Direction Cosine Matrix
DDR2	Double Data Rate 2
DOF	Degree(s) Of Freedom
ECRV	Exponentially Correlated Random Variable
EKF	Extended Kalman Filter
FC	Fine Count
FPA	Flight Path Angle
GN&C	Guidance, Navigation, and Control
GPS	Global Positioning System
GRAM	Global Reference Atmospheric Model
IGM	Iterative Guidance Mode
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LC	Linear Covariance
LTG	Linear Tangent Guidance
MC	Monte Carlo
MECO	Main Engine Cut Off
MEMS	Micro Electro Mechanical Systems
MOI	Moments Of Inertia
mpg	Miles Per Gallon
NASA	National Aeronautics and Space Administration
NRT	Nominal Reference Trajectory
PD	Proportional Derivative
PEG	Powered Explicit Guidance
PO	Pitch Over
PPM	Parts Per Million
PSD	Positive Semi Definite
RAAN	Right Ascension of the Ascending Node
RCS	Reaction Control System
RSS	Root Sum Square
SAR	Single Axis Rotation
SDRAM	Synchronous Dynamic Random Access Memory
SSTO	Single Stage To Orbit
VR	Vertical Rise

Chapter 1

Introduction

"There is only one way in which a person acquires a new idea: by

the combination or association of two or more ideas he already

has into a new juxtaposition in such a manner as to discover a

relationship among them of which he was not previously aware."

– Francis A. Cartier

A synthesis of Monte Carlo and linear covariance techniques can provide launch vehicle designers and mission planners with a useful state-of-the-art tool for simulating, understanding, and evaluating the performance of the launch vehicle guidance, navigation, and control (GN&C) system during its first critical phase of operation, powered ascent. Although traditional approaches to GN&C system analysis have thus far proved adequate, the need to reduce development costs, shorten turnaround time for planning and redesign, and support rapid trade-space analysis for new GN&C concepts, has sparked renewed interest in the community to explore and develop new approaches or technologies. One approach that has the potential to meet many, if not all, of these needs is the application of linear covariance and Monte Carlo methods in a novel manner so as to exploit their respective strengths while circumventing many of their inherent weaknesses. This dissertation demonstrates how both methods can be used to compliment, support, and validate one another during the development and design process in order to provide greater confidence in the performance results and system design. Moreover, it examines the practical challenges associated with extending and applying linear covariance techniques to powered ascent.

1.1 Problem Definition

The problem that this dissertation considers is that of modeling, analyzing, and seeking to understand the performance of a given launch vehicle GN&C system design as well as conducting the analysis in a rapid fashion so as to foster in-depth trade space studies, shorten turnaround time for planning and redesign, and by so doing reduce some of the development costs. Moreover,

this dissertation also examines the difficult task of validating the implementation of the models and algorithms in a simulation environment. However, no attempt will be made to verify the accuracy of the models and algorithms in relation to the 'real-world' objects that they are meant to represent and describe. To do so would require data from extensive experimentation (e.g., actual launch vehicle, hardware, and software) and is beyond the scope of this research.

The performance of a launch vehicle GN&C system is traditionally evaluated using Monte Carlo methods, i.e, a class of computational algorithms that relies upon repeated random sampling in order to accurately describe a system. Monte Carlo methods are predominantly the method of choice because they provide a thorough analysis of all aspects of GN&C system performance, are capable of capturing the effects of nonlinear phenomena and flaws in model and algorithm design, and allow for replacement of models and algorithms with actual data, hardware, or software as they become available [1, pp. 341].

The problem with Monte Carlo methods, however, is that they are generally slow, time consuming, and can often require significant computing resources. This is due primarily to the fact that the necessary number of samples is dependent upon the desired statistical level of confidence (or acceptable error margin) in the results [2, 3]. In other words, the higher the confidence level, the more samples are required, and ultimately the more time it will take to obtain those samples. Add in the fact that such simulations are generally comprised of high-fidelity nonlinear models and algorithms of the six-degree-of-freedom (6-DOF) vehicle dynamics, sensors, actuators, environment, and GN&C flight software, and it becomes readily apparent why Monte Carlo methods are imprac- tical for in-depth trade-space and sensitivity studies and extremely inefficient for rapid planning and redesign work. Consequently, there has been considerable effort in the literature to develop alternative methods or techniques that are capable of achieving equivalent (or at least comparable) results to those of Monte Carlo methods, but in a fraction of the time [1, 2, 4–6].

One of these alternative techniques is *linear covariance* analysis, which is based upon linear system theory [7] and Kalman filtering for linear systems [8]. Hence, a linear covariance simulation is essentially a linearized version of the nonlinear Monte Carlo simulation and designed specifically to produce the same statistical results as Monte Carlo methods, but with one single run [4]. Also, because it is a linear system it is very fast and thus extraordinarily well-suited for in-depth trade- space and sensitivity studies as well as rapid planning and redesign work.

However, the problem with linear covariance methods is that, like most linearized models, they are generally unable to capture many of the effects of highly nonlinear phenomena. In addition, the general form or implementation of the simulation does not allow for replacement of models and algorithms with actual data, hardware, or software.[1] Moreover, linear covariance techniques are not widely understood by the GN&C community. As a result, their results are generally suspect unless validated with those from Monte Carlo methods.

Despite the skepticism and apprehension that seems to surround them, linear covariance techniques have been successfully applied to many GN&C system problems in aerospace, such as orbital rendezvous and proximity operations [4, 9–11], cislunar trajectories, lunar powered descent and landing [12, 13], interplanetary missions [14], and powered ascent, just to name a few. In the case of powered ascent, however, existing linear covariance simulations are limited to evaluating the performance of the navigation system in an open-loop setting, commonly referred to as *navigation system analysis*. In other words, they are not able to evaluate the performance of the entire GN&C system or examine performance metrics such as trajectory control dispersions, propellant dispersions, orbit insertion dispersions, and true navigation errors, which are key to the powered ascent problem. While an evaluation of the navigation system is important, it is the performance of the entire GN&C system in the presence of sensor errors, actuator uncertainties, and random disturbances that is of greater importance to launch vehicle designers and mission planners.

Several factors have prevented the development of a linear covariance simulation for evaluation of a launch vehicle GN&C system (in a closed-loop setting) during *powered ascent*. First, when a project in industry has need of such a simulation, unless the simulation is already available or in development, it is generally too late or impractical to begin the development process. This is because a linear covariance simulation requires linearized models and algorithms of the vehicle dynamics, actuators, sensors, environment, and GN&C flight software. Moreover, there is always the possibility that linear covariance techniques may not be applicable or valid for a given application. Consequently, there is some initial overhead and risk involved with its development. For this reason, linear covariance simulations tend to be less amenable to development in industry, whereas the university setting has proved to be more conducive. Second, only in the past few years has the fundamental theory behind linear covariance analysis been expanded to include three key principles that are essential to powered ascent, viz., use of inertial sensors for navigation state propagation [4],

---

[1]The reason for this will become evident in a later discussion of the linear covariance formulation and simulation.

closed-loop guidance and control for continuous nonimpulsive maneuvers [4], and the handling of events such as discrete maneuvers, sensor changes, and general GN&C mode changes triggered by some criteria other than time [15]. Lastly, many GN&C algorithms, such as closed-loop guidance, are in general very complex, highly nonlinear, and iterative in nature. As a result, most GN&C flight algorithms pose a substantial challenge when it comes time to linearize them for use in the linear covariance simulation.

1.2 Proposed Solution

In summary, each of the two methods previously described has both strengths and weaknesses that make it more or less suitable for particular tasks, which motivates the purpose of the proposed research. The fundamental thesis of the dissertation is that a synthesis of Monte Carlo and linear covariance techniques will provide launch vehicle designers and mission planners with a useful state-of-the-art tool to rapidly and accurately simulate, evaluate, and characterize the performance of a launch vehicle GN&C system in a closed-loop setting, in the presence of sensor errors, actuator execution uncertainties, and random environment disturbances, and during its first critical phase of operation—powered ascent. As such, the proposed tool will consist of both a linear covariance simulation and a Monte Carlo simulation; the former to support in-depth trade-space and sensitivity studies as well as rapid planning and redesign work, and the latter to provide confidence in and validate the results of the former.

1.2.1 Scope

Although the proposed research will focus on the powered ascent problem for launch vehicle GN&C systems, the tools, techniques, and mathematical formulations will be applicable—in a general sense—to other rocket-powered systems such as sounding rockets and ballistic missiles. There are, however, a few important aspects of the launch vehicle powered ascent problem that this research will not be able to address or capture at this time, either due to time constraints or the sheer magnitude and complexity of the topic. For example, the challenging task of incorporating stochastic models of the atmosphere (e.g., winds, pressure, temperature) in a linear covariance simulation is considered to be a dissertation-sized topic unto itself and therefore beyond the scope of this research. For this reason, the launch scenario considered for this work is a launch from the surface of the Moon, where the operating environment is modeled as the near vacuum of space.

Furthermore, the proposed work does not stem from a specific mission nor is it constrained to a single application. However, in order to properly demonstrate the capabilities of the analysis tool, a specific launch vehicle must be defined and modeled. As such, it is assumed that the launch vehicle is a rigid, single-stage-to-orbit (SSTO) vehicle with time-varying mass and moments of inertia. Thrust will be provided by a single liquid-propellant engine that does not gimbal, is not throttleable, and operates at maximum thrust. Attitude control will be provided by a series of reaction thrusters and a proportional-derivative (PD) controller. The inertial navigation system will employ a strapdown inertial measurement unit (IMU) that provides measurements of the specific force and angular velocity, and an extended Kalman filter (EKF) to process the IMU measurements and maintain an estimate of the navigation states and covariance of the navigation state error. Lastly, the guidance system will operate in both open- and closed-loop modes. Predetermined guidance commands will be used during the initial or open-loop portion of the ascent trajectory and Shuttle's *powered explicit guidance* (PEG) will generate the guidance commands on-the-fly during the closed-loop portion.

1.2.2 Objectives

The objectives used to establish and substantiate the proposed thesis are five-fold: 1) formulate and develop the models, algorithms, and equations for a high-fidelity, 6-DOF, Monte Carlo simulation of the launch vehicle GN&C system, 2) define and develop the nominal reference trajectory for the proposed ascent trajectory, 3) linearize the Monte Carlo simulation models, algorithms, and equations to obtain the linear covariance equations, 4) evaluate the performance of the launch vehicle GN&C system during powered ascent using linear covariance simulation and analysis, and 5) validate the results with Monte Carlo simulation and analysis.

1.3 Dissertation Overview

The remainder of the dissertation is organized as follows. First, a brief summary of background material and related work is presented in Chapter 2. Chapter 3 addresses the first objective and contains the nonlinear models and algorithms developed for the Monte Carlo simulation. The second and third objectives are achieved in Chapters 4 and 5, where the nominal reference trajectory is defined, the linearized models and algorithms are presented, and the linear covariance equations are developed and formulated. In Chapter 6, the launch vehicle, launch scenario, and

key simulation parameters are defined. The results associated with the last two objectives are presented and discussed in Chapter 7. Finally, a summary of the key points of the dissertation, lessons learned, and suggestions for future work is given in Chapter 8.

Chapter 2

Background & Related Work

"Knowledge is of two kinds; we know a subject ourselves,

or we know where we can find information upon it."

– Samuel Johnson

The research in this dissertation is inspired and influenced by the work of a host of engineering professionals from various fields of study. Hence, the primary purpose of this chapter is to provide a brief survey of this related material found in the literature and discuss its relationship with the present work. Moreover, there are a number of basic concepts and general terminology related to launch vehicle powered ascent and GN&C system analysis that are key to following and understanding the present research. As such, the secondary purpose of this chapter is to provide the necessary background material in a concise and comprehensible manner, particularly for those readers who are not well-versed in the subject matter.

2.1 Literature Survey

Examination of the literature reveals four categories or types of analysis related to the present research and where linear covariance techniques stand out as the method of choice. The four types of analysis are sensitivity analysis, navigation error analysis, filter tuning, and consider analysis. Sensitivity analysis is concerned with measuring the sensitivity in the performance of a system to changes in hardware (e.g., sensors and actuators), system environment, and flight computer GN&C algorithms [1, 4, 5]. This type of analysis routinely plays a vital role in major trade space studies and preliminary design work. Navigation error analysis is the study of isolated contributions of individual (or groups of) error sources on the accuracy of the navigation filter's estimate of the true states [1, 5, 11–14, 16]. This type of analysis is fundamental in evaluating the performance of the navigation system and creating navigation error budgets. Filter tuning becomes requisite when the navigation filter is based upon a different, lower-fidelity system model than the truth and refers to the process of 'tuning' the strengths of the process and measurement

noise or adding filter states in order to achieve the best possible estimation performance from a navigation filter [1, 5]. Lastly, consider analysis is used to quantify the individual contributions and effects of problem parameters and states, not modeled in the navigation filter, on the navigation errors [17, 18].

Closer inspection of those papers in the literature that employ the term "linear covariance" reveals two related yet different variations or classes. The first class is linear covariance for navigation system analysis, hereafter referred to simply as *navigation system analysis*. It is the most common form of linear covariance and is usually what most people think of when they hear "linear covariance" mentioned. Navigation system analysis is primarily devoted to the study of the effects of state[1] and measurement[2] uncertainties on the accuracy of the navigation filter's estimate of the true states, which is directly related to the performance of the navigation system. In principle, the true values of the system states are not known, thus requiring an estimate or navigation state vector. Navigation system analysis is not concerned with maintaining an actual estimate of the true states, but rather in the statistical evolution of the covariance of the navigation state error. Moreover, it operates in an open-loop setting and under the assumption that there are no state dispersions, i.e., the true states are equivalent to the nominal states of the system. Navigation system analysis has played an important role in a variety of aerospace applications that are related either directly or indirectly to various aspects of the present work. Examples include attitude determination [17–20], ballistic missile trajectories [21, 22], interplanetary trajectories [14, 16, 23], lunar powered descent and landing [12, 13], orbit determination [24], orbital rendezvous [25] , suboptimal recursive filters [5, 26], and strapdown inertial navigation systems [27].

The second class is linear covariance for GN&C system analysis, hereafter referred to simply as *GN&C system analysis*. It is the least understood form of linear covariance among the GN&C community and is often confused with navigation system analysis. GN&C system analysis, also referred to as closed-loop GN&C analysis in the literature, is the study of the statistical performance of the entire GN&C system in a closed-loop setting and in the presence of sensor errors, actuator execution uncertainties, and random environment disturbances. GN&C system analysis differs from the more common form of linear covariance in several important ways. First, as the name

---

[1]State uncertainties, represented as *process noise*, accumulate over time and are due to errors in modeling the system dynamics.

[2]Measurement uncertainties, represented as *measurement noise*, are due to errors in the measurements and the error in quantities derived from the measurements.

implies, it includes the navigation system as well as the guidance and control systems. Second, it does not operate under the assumption that the state dispersions are zero. Rather, it is able to generate performance metrics such as the covariance of the true state dispersions, the navigation state dispersions, and the true navigation state errors, in addition to the covariance of the filter navigation state errors provided by navigation system analysis. Lastly, it operates in a closed-loop setting, instead of an open-loop setting. This allows it to evaluate not only the performance of each element of the GN&C system, but also the effect that each element may have on another (e.g., navigation on guidance, guidance on control, etc).

Of the limited publications available in the literature on GN&C system analysis, the majority of them have appeared only in the past five years and tend to focus on the general orbital rendezvous problem and its associated applications. For example, in 2006, Geller [4] developed the general theory and demonstrated its potential for use in autonomous onboard mission planning. In 2007, Geller [9] presented a linear covariance analysis of the relative attitude estimation and control problem for orbital rendezvous using angles-only navigation, which was then compared with results obtained through Monte Carlo analysis by Woffinden and Geller [10]. In that same year, Geller [11] also characterized four key parameters to define when onboard autonomous systems are required for orbital rendezvous missions. In 2009, Geller et al. [15] extended linear covariance theory to handle discrete events such as impulsive maneuvers,[3] operational GN&C mode changes, and sensor acquisition changes that are triggered by onboard estimates of position, velocity, or attitude rather than time. Lastly, in 2010, Moesser [29] examined the lunar powered decent problem, which is similar in many regards to the lunar powered ascent problem. Although the papers just mentioned have primarily focused on the orbital rendezvous problem, they contain many useful and important pieces of linear covariance theory that will be used extensively in the present research.

There are also many aspects of the launch vehicle ascent problem that have received considerable attention in the literature, such as flight path constraints [30], guidance [31–39] and control [40–42] techniques, modeling of launch vehicle dynamics [43, 44] and environment [45, 46], trajectory design [47–51], and propulsion system performance [52], all of which may prove beneficial to the proposed research.

---

[3]In 1991, Gossner [28] developed a similar analytical method for handling discrete events in linear covariance analysis. However, this work was restricted to maneuver conditions only.
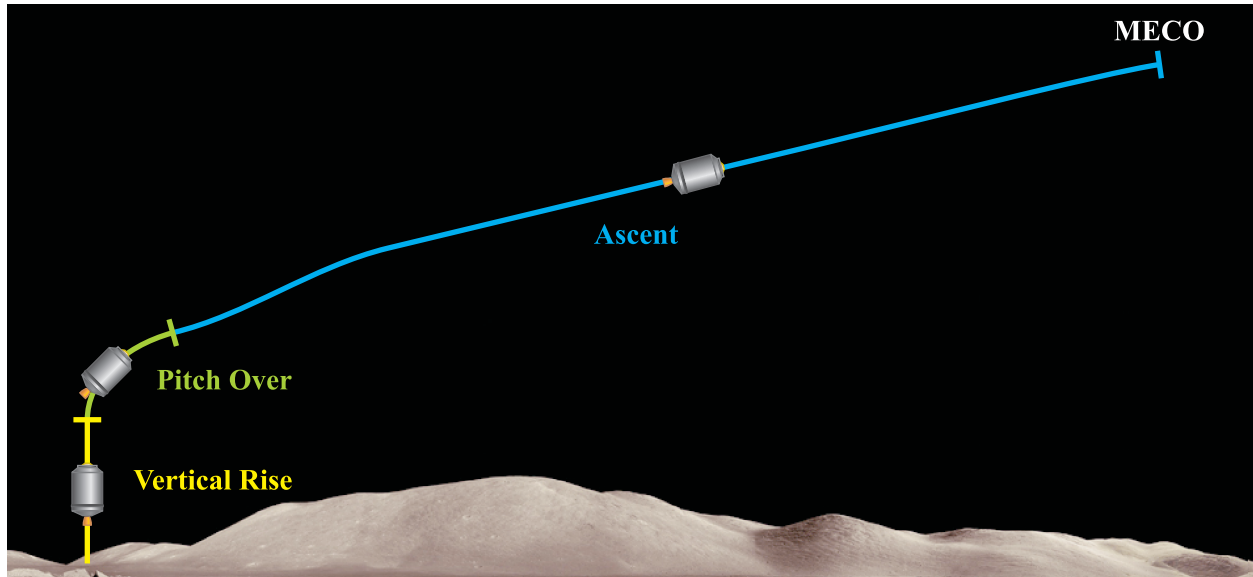
Figure 2.1: Lunar powered ascent profile.

2.2   Ascent to Orbit Flight

The basic ascent-to-orbit problem can be separated into three distinct phases [53]: the powered ascent phase, coasting phase, and target-orbit insertion phase. The powered ascent phase starts at engine ignition or lift off and ends at main engine cut off (MECO). During this segment of the flight, the propulsion system provides continuous, powered thrust, which lasts on the order of a few minutes. At MECO the launch vehicle enters the powerless or coasting phase while traveling along an elliptical transfer orbit. The length of time spent in the coasting phase is dependent upon the orbital parameters of the target orbit. The final or target orbit insertion phase, the shortest in terms of time of the three phases, occurs when a small, impulsive-like maneuver is executed to place the launch vehicle payload or spacecraft in a desired, target orbit.

For this research, the powered ascent phase is separated into three segments (as depicted in Figure 2.1): vertical rise, pitch over, and ascent. The first segment is *vertical rise*, which usually lasts for only a few seconds and whose primary purpose is to safely clear the launch pad or tower and any other potential hazards in the area. The second segment is *pitch over*, which generally spans tens of seconds and whose primary purpose is to reorient the inertial thrust direction from the vertical to the optimal heading for ascent. The third and final segment is *ascent*, which typically lasts a few minutes and ends at MECO.

2.3 GN&C System Analysis

A brief introduction to some important terminology related to GN&C system analysis is given. This is followed by a general overview of the Monte Carlo and linear covariance simulations.

2.3.1 General Terminology

The general terminology is divided into three categories: states, dispersions and navigation errors, and covariances. Each is defined and discussed in greater detail in the following sections.

States

In general, there are two basic types of states: vehicle states and error parameter states. Vehicle states are associated with the dynamics and kinematics of the launch vehicle. Examples include (but are not limited to) position, velocity, attitude, angular velocity, and mass. Error parameter states are used to represent sources of error in the sensors, actuators, and environment models. Examples include (again not limited to) scale factors, misalignments, biases, and disturbances. Moreover, there are four groups or sets of states, each of which is comprised of vehicle states and error parameter states. They are the true states, the nominal states, the navigation states, and the filter design states. The true states, denoted by $\boldsymbol{x} \in \mathbb{R}^{n'}$, represent the true or actual flight path of the launch vehicle, whereas the nominal states, denoted by $\bar{\boldsymbol{x}} \in \mathbb{R}^{n'}$, represent the nominal or desired flight path. The filter design states, denoted by $\mathsf{x} \in \mathbb{R}^{m'}$ where $m' \leq n'$, are used to design the navigation filter and represent the 'truth' for the filter design. Lastly, the navigation states, denoted by $\hat{\boldsymbol{x}} \in \mathbb{R}^{m'}$, are derived from the filter design states and represent the navigation filter's estimate of the true states.

The inertial-to-body attitude quaternion $\boldsymbol{q}_i^b \in \mathbb{R}^4$ is the standard representation of the launch vehicle orientation or attitude. However, due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the state covariance propagation and update equations as well as the state vector update equations. In this approach, the inertial-to-body attitude quaternions $\boldsymbol{q}_i^b, \hat{\boldsymbol{q}}_i^b \in \mathbb{R}^4$ in the true and navigation state vectors are replaced with the Euler rotation vectors $\boldsymbol{\theta}^b, \hat{\boldsymbol{\theta}}^b \in \mathbb{R}^3$, respectively, and the corresponding quaternion kinematic equations are replaced with the linearized Bortz equation [55]. The result is a modified true state vector $\boldsymbol{x}_{\mathrm{m}} \in \mathbb{R}^n$ (where $n = n' - 1$) and a modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}} \in \mathbb{R}^m$ (where $m = m' - 1$).

Dispersions and Navigation Errors

There are two types of dispersions: true state dispersions and navigation state dispersions. The true state dispersions, denoted by $\delta \boldsymbol{x}$, are defined as the difference between the true state vector $\boldsymbol{x}$ and the nominal state vector $\bar{\boldsymbol{x}}$. The navigation state dispersions, denoted by $\delta \hat{\boldsymbol{x}}$, are defined as the difference between the navigation state vector $\hat{\boldsymbol{x}}$ and the nominal state vector $\bar{\boldsymbol{x}}$. Moreover, there are two types of navigation errors: filter navigation state errors and true navigation state errors. The filter navigation state errors, denoted by $\delta \hat{\boldsymbol{e}}$, are defined as the difference between the filter design state vector $\mathbf{x}$ (which represents the true navigation state) and the navigation state vector $\hat{\boldsymbol{x}}$. The true navigation state errors, denoted by $\delta \boldsymbol{e}$, are defined as the difference between the true state dispersions $\delta \boldsymbol{x}$ and the navigation state dispersions $\delta \hat{\boldsymbol{x}}$ or equivalently the difference between the true state vector $\boldsymbol{x}$ and the navigation state vector $\hat{\boldsymbol{x}}$.

Covariances

The covariance of the true state dispersions, denoted by the matrix $\boldsymbol{\mathcal{D}}_{xx} \in \mathbb{R}^{n \times n}$, characterizes how far from the nominal reference trajectory the actual flight path of the launch vehicle may differ due to uncertainties in actuator execution, algorithm selection and performance, navigation error, and various disturbances acting on the launch vehicle. The covariance of the navigation state dispersions, denoted by the matrix $\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}} \in \mathbb{R}^{m \times m}$, characterizes how far from the nominal reference trajectory the estimated state may vary due to uncertainties in actuator execution, algorithm selection and performance, navigation error, and various disturbances acting on the launch vehicle. The covariance of the filter navigation state errors, denoted by the matrix $\hat{\boldsymbol{\mathcal{P}}} \in \mathbb{R}^{m \times m}$, describes the accuracy or precision of the estimated values of the true states attributable to sensor error, modeling error, uncertainties in initial conditions, and the error in quantities derived from the measurements. It is important to emphasize that $\hat{\boldsymbol{\mathcal{P}}}$ is what the navigation filter *thinks* is the covariance of the navigation state error, and is computed by propagating and updating $\hat{\boldsymbol{\mathcal{P}}}$ in the navigation algorithm. The covariance of the true navigation state errors, denoted by the matrix $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{m \times m}$, also describes the accuracy or precision of the estimated values of the true states caused by sensor error, modeling error, uncertainties in initial conditions, and the error in quantities derived from the measurements. However, $\boldsymbol{\mathcal{P}}$ represents the *true* covariance of the navigation error. It is important to note that $\hat{\boldsymbol{\mathcal{P}}}$ and $\boldsymbol{\mathcal{P}}$ are computed in completely different ways and the

only time when they are identical is when the truth models, those that represent the best attempt to model the "real-world" dynamics, and the filter design models, those that are used to formulate and develop the navigation filter, are one and the same.

2.3.2 Monte Carlo Simulation

The high-fidelity Monte Carlo simulation of the launch vehicle GN&C system consists of two main components (as illustrated in Figure 2.2): truth models and GN&C algorithm models. The truth models are comprised of the 6-DOF vehicle dynamics, actuator force and torque models, environment force and torque models, and sensor measurement models. The GN&C algorithm models incorporate the guidance, navigation, and control algorithms needed to maneuver and orient the launch vehicle along the desired or nominal flight path.

From Figure 2.2 it follows that white noise processes $\boldsymbol{w}$, $\boldsymbol{\eta}$, $\boldsymbol{\nu}_k$ and actuator commands $\hat{\boldsymbol{u}}$ drive the truth models, which in turn generate the true state vector $\boldsymbol{x}$ of the system, actuator forces $\boldsymbol{f}$ and torques $\boldsymbol{m}$, and simulated sensor measurements $\tilde{\boldsymbol{y}}$, $\tilde{\boldsymbol{z}}_k$. The navigation algorithm processes the sensor measurement data and produces the navigation state vector $\hat{\boldsymbol{x}}$ and the (filter) navigation state error covariance matrix $\hat{\boldsymbol{\mathcal{P}}}$. The open-loop guidance algorithm employs guidance commands computed and stored prior to launch, whereas the closed-loop algorithm uses the current navigation state, a simplified vehicle dynamics model, and the mission-specific terminal constraints to generate the guidance commands $\hat{\boldsymbol{a}}$. Lastly, the control algorithm utilizes the current navigation state and guidance commands to produce the actuator commands $\hat{\boldsymbol{u}}$, which are then fed to the truth models—thus closing the loop.

The covariances of the true state dispersions $\boldsymbol{\mathcal{D}}_{xx}$, navigation state dispersions $\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}}$, and true navigation state errors $\boldsymbol{\mathcal{P}}$ are determined by collecting the results of N simulation runs,

$$\boldsymbol{\mathcal{D}}_{xx} = \frac{1}{N-1} \sum_{j=1}^{N} \delta\boldsymbol{x}_j \left[\delta\boldsymbol{x}_j\right]^{\mathsf{T}} , \tag{2.1}$$

$$\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}} = \frac{1}{N-1} \sum_{j=1}^{N} \delta\hat{\boldsymbol{x}}_j \left[\delta\hat{\boldsymbol{x}}_j\right]^{\mathsf{T}} , \tag{2.2}$$

$$\boldsymbol{\mathcal{P}} = \frac{1}{N-1} \sum_{j=1}^{N} \delta\boldsymbol{e}_j \left[\delta\boldsymbol{e}_j\right]^{\mathsf{T}} , \tag{2.3}$$
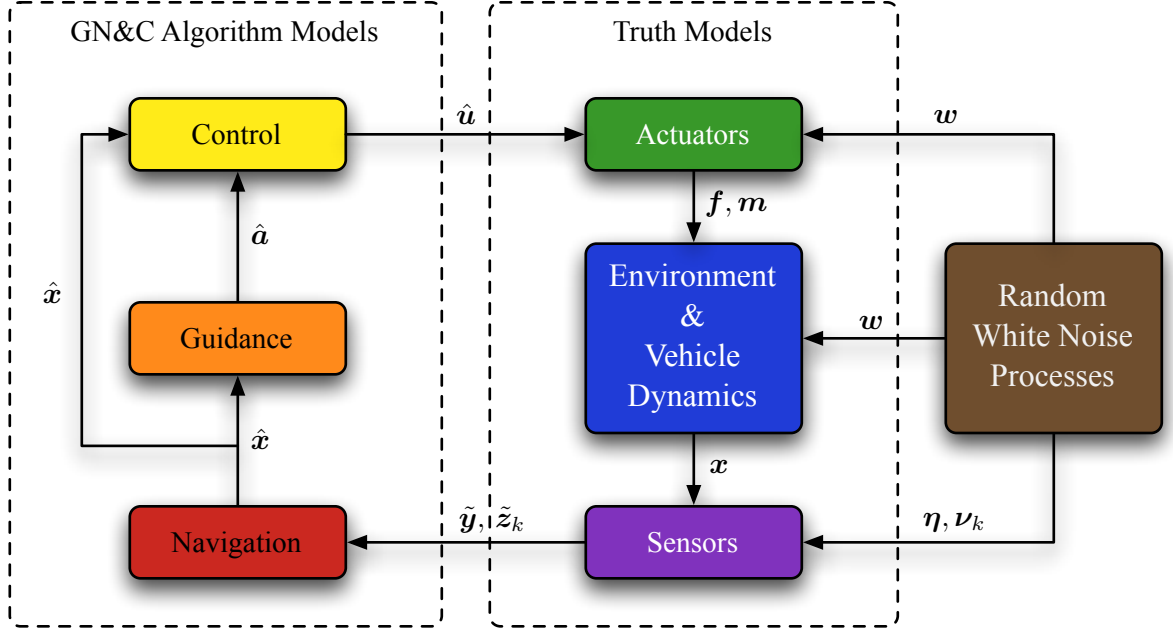
Figure 2.2: Monte Carlo simulation of the launch vehicle GN&C system.

where the output from a single simulation run is the time history of the true state vector $\boldsymbol{x}$, navigation state vector $\hat{\boldsymbol{x}}$, and filter navigation state covariance matrix $\hat{\boldsymbol{\mathcal{P}}}$.

2.3.3 Linear Covariance Simulation

The linear covariance simulation is able to produce the same important statistical quantities (i.e., $\boldsymbol{\mathcal{D}}_{xx}$, $\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}}$, $\boldsymbol{\mathcal{P}}$, and $\hat{\boldsymbol{\mathcal{P}}}$) as the Monte Carlo simulation, but it does it in a completely different manner and with only one run of the simulation. It employs an augmented state covariance matrix $\boldsymbol{\mathcal{C}}_{\mathrm{a}}$ where the augmented state vector, denoted by $\mathbf{x} \in \mathbb{R}^{\ell}$ (where $\ell = n + m$), is comprised of both the true state dispersions $\delta\boldsymbol{x}$ and the navigation state dispersions $\delta\hat{\boldsymbol{x}}$. Thus, it follows that the augmented state covariance matrix contains the covariances of the true state dispersions $\boldsymbol{\mathcal{D}}_{xx}$ and the navigation state dispersions $\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}}$. Moreover, it can be shown that the covariance of the true navigation state errors $\boldsymbol{\mathcal{P}}$ is obtained from the augmented state covariance matrix $\boldsymbol{\mathcal{C}}_{\mathrm{a}}$. Figure 2.3 illustrates the high-level structure and flow of the linear covariance simulation of the launch vehicle GN&C system. The time history of the covariances are generated by initializing, propagating, updating, and shaping the augmented state covariance matrix $\boldsymbol{\mathcal{C}}_{\mathrm{a}}$ and initializing, propagating, and updating the filter navigation state covariance matrix $\hat{\boldsymbol{\mathcal{P}}}$.
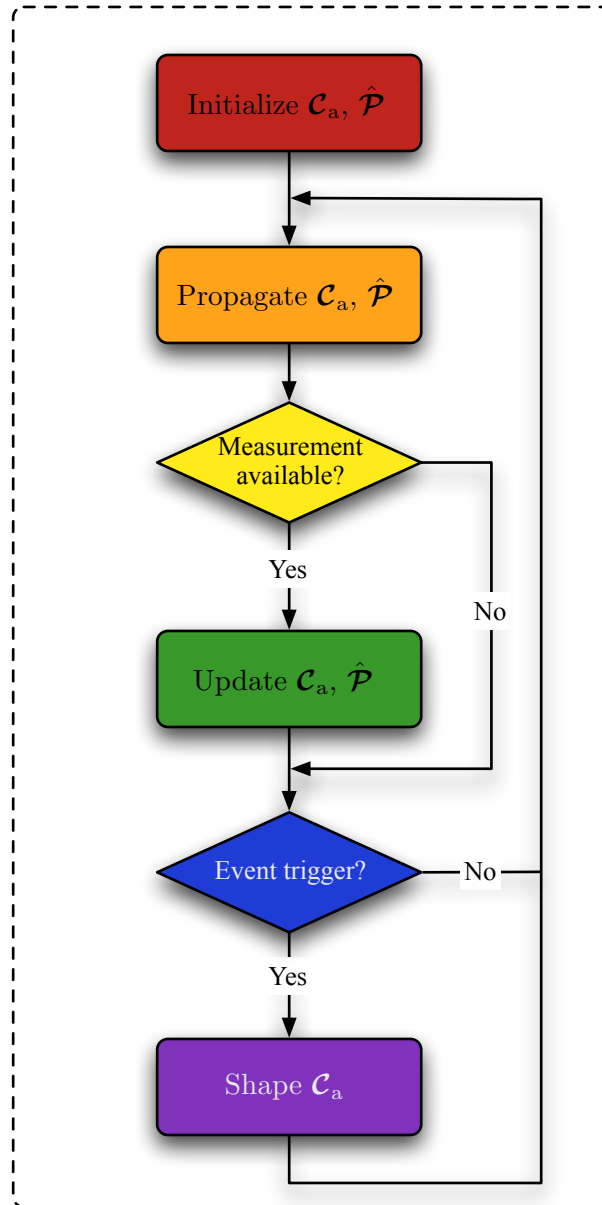
Figure 2.3: Linear covariance simulation of the launch vehicle GN&C system.

Chapter 3

Nonlinear Monte Carlo Models

"As far as the laws of mathematics refer to reality,

they are not certain; and as far as they are certain,

they do not refer to reality."

– Albert Einstein

The purpose of this chapter is to address the first objective of this dissertation, which is to formulate and develop the models, algorithms, and equations for the high-fidelity, 6-DOF, Monte Carlo simulation of the launch vehicle GN&C system. In Chapter 2, a high-level description of the Monte Carlo simulation was given, where it was shown that the simulation can be separated into truth models and GN&C algorithm models. Hence, the chapter is organized as follows. First, the truth models are defined and discussed. The chapter then concludes with a description of the GN&C algorithm models.

3.1 Truth Models

The truth models are comprised of the six-degree-of-freedom vehicle dynamics, environment force and torque models, actuator force and torque models, and sensor measurement models. Each is introduced, defined, and discussed in greater detail in the following sections. The section finishes with definitions of the error parameter states and noise.

3.1.1 Vehicle Dynamics

A launch vehicle experiences six degrees of freedom in Euclidean space. Three translational degrees describe the motion of the center-of-mass (c.m.), and three rotational degrees describe the orientation of the vehicle. If the c.m. of the launch vehicle is used as the reference point, the translational and rotational motions can be described separately, where Newton's second law governs the translational dynamics and Euler's law governs the rotational or attitude dynamics, both of which are referenced to an inertial reference frame.

Under the assumption that the launch vehicle can be represented as a rigid body,[1] the 6-DOF equations of motion that completely describe the launch vehicle in three-dimensional Euclidean space can be decoupled and separated into the translational and rotational dynamics. For this reason, the launch vehicle is hereafter modeled as a rigid body (i.e., fixed c.m.) with the added complexity that the mass and associated mass moments-of-inertia (MOI) are allowed to change with time. Although it would be more realistic to consider a time varying center of mass for a launch vehicle, it can be shown that the translational and rotational dynamics are interdependent, and therefore cannot be decoupled. Consequently, the inclusion of a time varying c.m. is reserved for future work.

Translational Motion

The three-DOF translational dynamics of the launch vehicle's c.m. with respect to the inertial frame are given by

$$\dot{\boldsymbol{r}}^i = \boldsymbol{v}^i \, , \tag{3.1}$$

$$\dot{\boldsymbol{v}}^i = \boldsymbol{a}_{\mathrm{g}}^i(\boldsymbol{r}^i) + [\boldsymbol{T}(\boldsymbol{q}_i^b)]^\intercal \, \boldsymbol{a}_{\mathrm{ng}}^b \, , \tag{3.2}$$

where $\boldsymbol{r}^i$ is the position vector with respect to the inertial frame expressed in inertial coordinates, hereafter referred to as simply the *inertial position*, $\boldsymbol{v}^i$ is the velocity vector with respect to the inertial frame expressed in inertial coordinates, hereafter referred to as simply the *inertial velocity*, $\boldsymbol{a}_{\mathrm{g}}^i$ is the gravitational acceleration vector represented in inertial coordinates, $\boldsymbol{a}_{\mathrm{ng}}^b$ is the nongravitational acceleration vector represented in body coordinates, and $\boldsymbol{T}(\boldsymbol{q}_i^b)$ is the inertial-to-body transformation matrix (using quaternion parameterization). Note that items in parentheses denote that the variable to the immediate left is a function of the variables in parentheses. For example, $\boldsymbol{a}_{\mathrm{g}}^i(\boldsymbol{r}^i)$ denotes that the gravitational acceleration vector $\boldsymbol{a}_{\mathrm{g}}^i$ is a function of the inertial position vector $\boldsymbol{r}^i$.

---

[1]A rigid body is generally defined as a collection of elements or particles where the relative distances between every pair of particles are time invariant. See Rao [56, pp. 104] and Zipfel [57, pp. 18].

Rotational Motion

The three-DOF rotational dynamics of the launch vehicle about its c.m. with respect to the inertial frame are described by

$$\dot{\boldsymbol{q}}_i^b = -\frac{1}{2}\boldsymbol{q}_\omega(\boldsymbol{\omega}_{ib}^b) \otimes \boldsymbol{q}_i^b \ , \tag{3.3}$$

$$\dot{\boldsymbol{\omega}}_{ib}^b = [\boldsymbol{J}_b^b]^{-1}\big[\boldsymbol{m}_{\text{acs}}^b - \boldsymbol{\omega}_{ib}^b \times \boldsymbol{J}_b^b\,\boldsymbol{\omega}_{ib}^b\big] + \boldsymbol{\alpha}_{\text{ng}}^b \ , \tag{3.4}$$

where $\boldsymbol{q}_i^b$ is the attitude quaternion of the body frame with respect to the inertial frame expressed in body coordinates, hereafter referred to as simply the *inertial-to-body attitude*, $\boldsymbol{\omega}_{ib}^b$ is the angular velocity vector of the body frame with respect to the inertial frame expressed in body coordinates, hereafter referred to as simply the *body angular velocity*, and $\boldsymbol{q}_\omega$ is the pure quaternion of the angular velocity vector $\boldsymbol{\omega}_{ib}^b$ as given by

$$\boldsymbol{q}_\omega = \begin{bmatrix} \boldsymbol{\omega}_{ib}^b \\ 0 \end{bmatrix} . \tag{3.5}$$

Moreover, $\boldsymbol{J}_b^b$ is the mass MOI tensor with respect to the body frame and expressed in body coordinates, $\boldsymbol{m}_{\text{acs}}^b$ is the net applied moment or torque from the attitude control system (ACS) expressed in body coordinates, and $\boldsymbol{\alpha}_{\text{ng}}^b$ is the nongravitational angular acceleration vector expressed in body coordinates.

Note that the minus sign in the quaternion kinematics equation (3.3) is not a typographical error, but rather the result of the quaternion convention chosen for this research. The quaternion $\boldsymbol{q}_a^b$, representing the rotation from frame a to frame b, is defined as

$$\boldsymbol{q}_a^b = \begin{bmatrix} \mathbf{v} \\ s \end{bmatrix} = \begin{bmatrix} -\mathbf{u}^a \, \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} , \tag{3.6}$$

where $\mathbf{v}$ is the vector component of the quaternion, s is the scalar component of the quaternion, and a minus sign appears on the vector component. Quaternion multiplication, denoted by the $\otimes$ operator, is defined such that quaternions are multiplied in the "natural order" or the same order as direction cosine matrices (DCMs). Note that this is the same convention used on the Space Shuttle.

Mass

The mass of a launch vehicle is often referred to in the literature as the *wet mass* and defined as the mass of the launch vehicle, its contents, and propellant (or fuel). Thus, the differential equation that describes how the wet mass changes with time is given by

$$\dot{m}_b = -\beta \ , \tag{3.7}$$

where β is the (constant) mass flow rate of the propulsion system.

3.1.2 Environment Models

The environment models consist of both the gravitational and nongravitational acceleration as well as the nongravigational angular acceleration. For this research, it is assumed that there are no gravitational angular accelerations acting on the launch vehicle.

Gravitational Acceleration

The vector $\boldsymbol{a}_{\mathrm{g}}^{i}$ in the velocity dynamics equation (3.2) represents the vector sum of all gravitational (or volume) accelerations acting on the launch vehicle, expressed in inertial coordinates. In this study, only the acceleration due to a single planetary body, as described by Newton's law of gravitation, is considered. The inertial acceleration due to gravity can be written as

$$\boldsymbol{a}_{\mathrm{g}}^{i}(\boldsymbol{r}^{i}) = -\frac{\mu}{\|\boldsymbol{r}^{i}\|^{2}}\mathbf{u}_{r}^{i} \ , \tag{3.8}$$

where μ is the gravitational constant of the planetary body and $\mathbf{u}_{r}^{i}$ is the unit vector in the direction of the inertial position. Note that this equation is valid under the assumption that the planetary body and launch vehicle can be treated as rigid bodies (or point masses), each located at their respective c.m.. As a result, the gravitational acceleration equation (3.8) is commonly referred to as a *point mass* gravity model.

Nongravitational Acceleration

The vector $\boldsymbol{a}_{\mathrm{ng}}^{b}$ in the velocity dynamics equation (3.2) represents the vector sum of all nongravitational (or surface) accelerations acting on the launch vehicle, expressed in body coordi-

nates. In this study, only the acceleration due to thrust and the acceleration due to an unmodeled sinusoidal disturbance (e.g., fuel-slosh) are considered, which is given by

$$a_{\text{ng}}^b(m_b, f_{\text{thr}}^b, a_{\text{od}}^b) = \frac{f_{\text{thr}}^b(p_{\text{thr}}^b, w_{\text{thr}}^b)}{m_b} + a_{\text{od}}^b \; , \tag{3.9}$$

where f_{thr}^b is the applied thrust force vector expressed in body coordinates, m_b is the wet mass, and a_{od}^b represents the unmodeled sinusoidal disturbance acceleration expressed in body coordinates. The unmodeled disturbance acceleration is defined as

$$a_{\text{od}}^b = c_1^b \sin(\omega_{\text{od}} t) + c_2^b \cos(\omega_{\text{od}} t) \; , \tag{3.10}$$

where ω_{od} is the natural frequency of the oscillations, and the sinusoidal disturbance error source parameters, $p_{\text{od}}^b = (c_1^b, c_2^b)$, represent the translational acceleration amplitudes of the oscillations in the body frame and are modeled as first-order Gauss-Markov processes, as defined in § 3.1.5.

Nongravitational Angular Acceleration

The vector α_{ng}^b in the angular velocity dynamics equation [see Eq. (3.4)] represents the vector sum of all unmodeled nongravitational (or volume) angular accelerations acting on the launch vehicle, expressed in body coordinates. Only the angular acceleration due to an unmodeled sinusoidal disturbance (e.g., fuel-slosh) is considered, which is given by

$$\alpha_{\text{ng}}^b = \alpha_{\text{od}}^b \equiv d_1^b \sin(\omega_{\text{od}} t) + d_2^b \cos(\omega_{\text{od}} t) \; , \tag{3.11}$$

where ω_{od} is the natural frequency of the oscillations, and the sinusoidal disturbance error source parameters, $p_{\text{od}}^b = (d_1^b, d_2^b)$, represent the rotational acceleration amplitudes of the oscillations in the body frame and are modeled as first-order Gauss-Markov processes, as defined in § 3.1.5.

3.1.3 Actuator Models

Two systems of actuators are considered for this research: propulsion and attitude control. Each is introduced, defined, and discussed in greater detail in the following sections.

Propulsion System

The primary purpose of the propulsion system is to lift the launch vehicle from the surface of and place it into orbit about a planetary body. If the main engine(s) is able to gimbal, then a secondary purpose may be to provide limited attitude control during powered ascent. Moreover, the propulsion system provides a surface force, commonly referred to as the *thrust force*, on the launch vehicle. This thrust force is the direct result of mass being expelled at high speeds from the engine, typically in the form of hot exhaust gases.

In general, the engine and its performance is completely characterized by two parameters:

1. *specific impulse*, denoted by I_{sp} (with units of s), and defined as the total impulse (or change in momentum) per unit weight of propellant [58], and

2. *vacuum thrust*, denoted by T_{vac} (with units of N or lbf), and defined as the average maximum thrust of the engine in vacuum.

From these parameters, two additional parameters of interest can be obtained. The first is the *effective exhaust velocity*, denoted by c (with units of m/s), defined as the average nominal speed of the exhaust gases exiting the engine in vacuum, and written as [58]

$$c = I_{sp}\, g_0 \; , \tag{3.12}$$

where g_0 is the standard acceleration of gravity at sea level on earth. The second is the *mass flow rate*, denoted by β (with units of kg/s) , defined as the rate at which mass (i.e., propellant or fuel) is converted to exhaust gases and ejected from the engine, and written as [58]

$$\beta = \frac{T}{I_{sp}\, g_0} \; , \tag{3.13}$$

where T is the thrust. It follows from this equation that the only way to increase the thrust is to either increase the velocity at which mass is expelled from the engine (i.e., the effective exhaust velocity or specific impulse) or increase the amount of mass expelled from the engine (i.e., the mass flow rate). The specific impulse or I_{sp} is a measure of the efficiency of the engine, much like *miles per gallon* (mpg) is for automobile engines, and considered to be a time-invariant quantity. The

mass flow rate, however, is an adjustable quantity and used to vary (or throttle) the thrust level of the engine in flight.

The thrust force is a vector quantity described by Newton's third law,

$$\boldsymbol{f}_{\text{thr}}^b \equiv -\boldsymbol{f}_{\text{ex}}^b \,, \tag{3.14}$$

where $\boldsymbol{f}_{\text{thr}}^b$ is the thrust force imparted on the vehicle (expressed in body coordinates), and $\boldsymbol{f}_{\text{ex}}^b$ is the force imparted on the exhaust gases (expressed in body coordinates). This is illustrated in Figure 3.1, where the exhaust velocity vector $\boldsymbol{v}_{\text{ex}}^b$, expressed in body coordinates, is defined as

$$\boldsymbol{v}_{\text{ex}}^b = -c\,\mathbf{b}_1^b = -I_{\text{sp}}\,g_0\,\mathbf{b}_1^b \,, \tag{3.15}$$

where $\mathbf{b}_1^b$ is the unit basis vector of the body frame b. The force $\boldsymbol{f}_{\text{ex}}^b$ imparted on the exhaust gases, as shown in Figure 3.1, can be written as

$$\boldsymbol{f}_{\text{ex}}^b = \beta\,\boldsymbol{v}_{\text{ex}}^b \,. \tag{3.16}$$

Using Eqs. (3.16), (3.15), and (3.13) in Eq. (3.14), the thrust force vector, expressed in body coordinates, can be written in the following equivalent forms,

$$\boldsymbol{f}_{\text{thr}}^b = -\beta\,\boldsymbol{v}_{\text{ex}}^b = \beta\,I_{\text{sp}}\,g_0\,\mathbf{b}_1^b = T\,\mathbf{b}_1^b \,. \tag{3.17}$$

Finally, the thrust force vector, expressed in inertial coordinates, is given by

$$\boldsymbol{f}_{\text{thr}}^i = [\boldsymbol{T}(\boldsymbol{q}_i^b)]^\intercal\,\boldsymbol{f}_{\text{thr}}^b \,. \tag{3.18}$$

In this study, the propulsion system consists of a single liquid-propellant engine with the following simplifications:

- engine does not gimbal or pivot but is mounted in a fixed orientation with respect to the body frame;

- engine is not throttleable (i.e., the mass flow rate is held constant);

- engine operates at maximum thrust (i.e., $T = T_{\text{vac}}$).

From Eqs. (3.18) and (3.17) it follows that errors in the desired inertial thrust force vector $\boldsymbol{f}_{\text{thr}}^i$ are due to errors in the inertial attitude of the launch vehicle $\boldsymbol{T}(\boldsymbol{q}_i^b)$, and variability in the engine mounting, mass flow rate β, and exhaust velocity vector $\boldsymbol{v}_{\text{ex}}^p$. The maneuver execution errors due to errors in the inertial attitude of the launch vehicle are modeled explicitly in the attitude or rotational dynamics and not discussed here [see § 3.1.1]. From the simplifications listed above, it is assumed that the engine mounting and mass flow rate are known to sufficient accuracy that the errors in the desired thrust force vector due to variability in these quantities can be neglected. Consequently, only the thrust errors due to variability in the exhaust velocity or specific impulse are considered here.

A thruster that expels mass consistently at speeds in excess of or less than the effective exhaust velocity (each time it is fired) is referred to as running 'hot' or 'cold', respectively. This results in some variability in the actual magnitude of the exhaust velocity vector. Furthermore, the exit velocity of the exhaust gases is not uniform over the entire cross-section of the nozzle, resulting in some variability in the actual direction of the exhaust velocity vector. Consequently, these variations cause errors, hereafter referred to as *thruster execution errors*, in the applied or actual thrust force vector. The source of the errors can be represented by scale factors $\boldsymbol{s}_{\text{thr}}^b$, misalignments $\boldsymbol{\epsilon}_{\text{thr}}^b$, random biases $\boldsymbol{b}_{\text{thr}}^b$, and noise $\boldsymbol{w}_{\text{thr}}^b$ [59]. The thruster error parameter states, $\boldsymbol{p}_{\text{thr}}^b = (\boldsymbol{s}_{\text{thr}}^b, \boldsymbol{\epsilon}_{\text{thr}}^b, \boldsymbol{b}_{\text{thr}}^b)$, are modeled as first-order Gauss-Markov processes, as defined in § 3.1.5, and $\boldsymbol{w}_{\text{thr}}^b$ is process noise, as defined in § 3.1.6. After incorporating all of these error sources into the actuator model, the applied thrust force vector, expressed in body coordinates, is given by

$$\boldsymbol{f}_{\text{thr}}^b(\boldsymbol{p}_{\text{thr}}^b, \boldsymbol{w}_{\text{thr}}^b) = \left(\boldsymbol{I}_{3\times3} - [\boldsymbol{\epsilon}_{\text{thr}}^b \times]\right)\left\{\left(\boldsymbol{I}_{3\times3} + [\boldsymbol{s}_{\text{thr}}^b \searrow]\right) T_{\text{vac}}\, \mathbf{b}_1^b + \boldsymbol{b}_{\text{thr}}^b + \boldsymbol{w}_{\text{thr}}^b\right\}. \tag{3.19}$$

Attitude Control System

The primary purpose of the attitude control system (ACS) is to provide active attitude control of the launch vehicle about its three body axes. If the ACS is comprised entirely of small thrusters or 'RCS jets',[2] then a secondary purpose may be to provide limited propulsion or thrust,

---

[2]Often times in the literature, when the ACS is comprised of only small thrusters or jets, the ACS is commonly referred to as the *reaction control system* (RCS); hence the appellation 'RCS jets'.
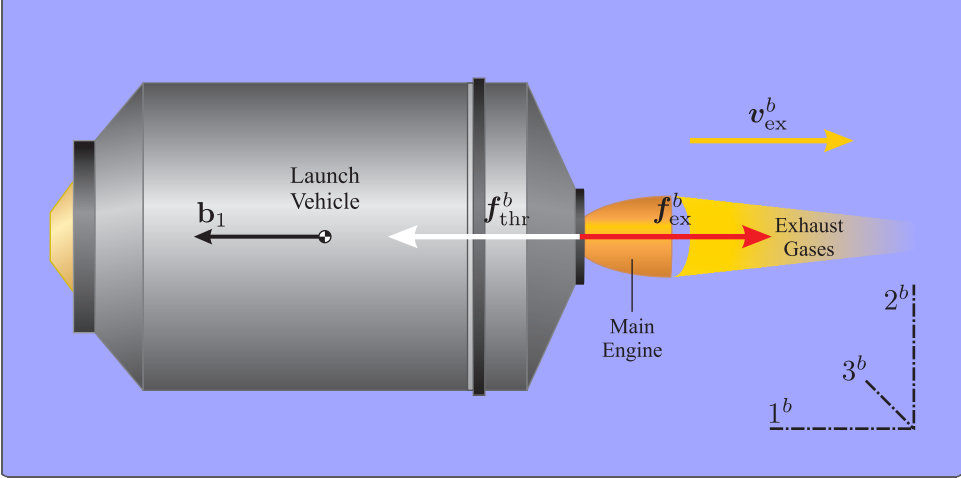
Figure 3.1: Propulsion system.

in the event of main engine failure, during powered ascent. Such was the case with the Apollo lunar ascent module. In general, two configurations of actuators are used for attitude control of the launch vehicle. The first was alluded to in the discussion of the propulsion system, i.e., a gimbaled main engine(s) for control about the body pitch and yaw axes, and smaller thrusters for control about the body roll axis. The second arises when the main engine(s) is fixed (i.e., does not gimbal), and RCS jets are used exclusively for control about all three body axes. Since the main engine considered in this study does not gimbal, only the second configuration is discussed here.

Each ACS thruster (or RCS jet), like the main engine, can be characterized by its specific impulse, vacuum thrust, effective exhaust velocity, and mass flow rate, and imparts a thrust force on the launch vehicle. In general, however, the ACS thrusters are oriented and fired in such a manner that neither their individual or combined thrust force is directed through the vehicle center of mass, resulting in an applied torque or moment about the center of mass, hereafter referred to as the *net ACS moment*.

The ACS induced moment, like the thrust force, is a vector quantity and is defined as the vector sum of moments from N ACS thrusters,

$$\boldsymbol{m}_{\text{acs}}^b = \sum_{j=1}^{N} \boldsymbol{m}_{\text{acs},j}^b \ , \tag{3.20}$$

where $\boldsymbol{m}_{\text{acs},j}^b$ is the moment vector from the jth ACS thruster (expressed in body coordinates). Since any torque or moment can be expressed as the vector product of the applied force and the

moment arm or linear distance from the point of application of the force to the point of reference (i.e., center of mass), the net ACS moment vector $\boldsymbol{m}_{\text{acs}}^b$ can also be written as

$$\boldsymbol{m}_{\text{acs}}^b = \sum_{j}^{N} \left(\boldsymbol{r}_{\text{acs},j}^b \times \boldsymbol{f}_{\text{acs},j}^b \right) , \tag{3.21}$$

where $\boldsymbol{r}_{\text{acs},j}^b$ is the moment arm of the jth ACS thruster (expressed in body coordinates), and $\boldsymbol{f}_{\text{acs},j}^b$ is the applied thrust force vector of the jth ACS thruster (expressed in body coordinates).

In this study, the ACS consists of eight thrusters strategically mounted to the exterior of the launch vehicle, as illustrated in Figure 3.2. Rather than model each of the thrusters individually, a simpler and more unified approach is taken where only the net ACS moment about the launch vehicle center of mass is modeled. The ACS thrusters are susceptible to the same variabilities as the main thruster. As a result, these variabilities cause errors, hereafter referred to as *ACS execution errors*, in the applied or actual net ACS moment vector. These errors can be represented by scale factors $\boldsymbol{s}_{\text{acs}}^b$, orthogonality misalignments $\boldsymbol{\epsilon}_{\text{acs}}^b$, random biases $\boldsymbol{b}_{\text{acs}}^b$, and noise $\boldsymbol{w}_{\text{acs}}^b$. The ACS error parameter states, $\boldsymbol{p}_{\text{acs}}^b = (\boldsymbol{s}_{\text{acs}}^b, \boldsymbol{\epsilon}_{\text{acs}}^b, \boldsymbol{b}_{\text{acs}}^b)$, are modeled as first-order Gauss-Markov processes, as defined in § 3.1.5, and $\boldsymbol{w}_{\text{acs}}^b$ is process noise, as defined in § 3.1.6. After incorporating all of these error sources into the error model, the applied net ACS moment vector, expressed in body coordinates, is given by

$$\boldsymbol{m}_{\text{acs}}^b(\hat{\boldsymbol{m}}_{\text{com}}^b, \boldsymbol{p}_{\text{acs}}^b, \boldsymbol{w}_{\text{acs}}^b) = \left(\boldsymbol{I}_{3\times3} - [\boldsymbol{\epsilon}_{\text{acs}}^b \times] \right) \left\{ \left(\boldsymbol{I}_{3\times3} + [\boldsymbol{s}_{\text{acs}}^b \diagdown] \right) \hat{\boldsymbol{m}}_{\text{com}}^b + \boldsymbol{b}_{\text{acs}}^b + \boldsymbol{w}_{\text{acs}}^b \right\} , \tag{3.22}$$

where $\hat{\boldsymbol{m}}_{\text{com}}^b$ is the commanded ACS moment issued by the attitude controller [see § 3.2.3].

3.1.4 Inertial Measurement Unit

The purpose of the *inertial measurement unit* (IMU) is to measure the motion of the launch vehicle in three-dimensional Euclidean space relative to an inertial frame of reference. For this reason, the IMU is generally the primary sensor on launch vehicles and a variety of other spacecraft. Moreover, it is generally comprised of two inertial instrument packages: accelerometers and gyros. These inertial instruments can be mounted on a set of rotating gimbals, called a *navigation platform*, such that no matter how the vehicle rotates the instruments maintain their inertial (or local-level) alignment. Hence, the navigation platform is commonly called a *space*
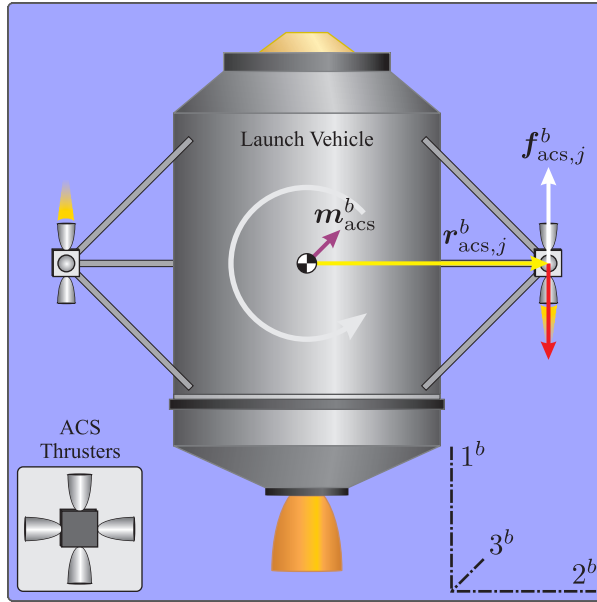
Figure 3.2: Attitude control system.

stabilized or *inertial platform*. Alternatively, the inertial instruments can be rigidly mounted to the vehicle structure, in what is referred to as a *strapdown system*, so that they rotate with the vehicle. Consequently, the provided measurements are obtained in the rotating IMU case frame and must be transformed to the desired frame of reference.

A strapdown IMU has several advantages over its counterpart, the *inertial platform* system, in that it is generally lighter, cheaper, simpler, more robust, more reliable, and more easily configured for odd-shaped spaces [60]. The disadvantages, however, are usually lower accuracy, complex sensor alignment and calibration procedures, and gyros required to handle a large dynamic range of rotation rates. In general, the IMU is installed with a position offset from the launch vehicle center of mass, as depicted in Figure 3.3. This position is not known perfectly, and will vary with time as fuel is expended. Moreover, if the IMU position offset is not properly accounted for, attitude motion will inadvertently be measured as acceleration.

The IMU is located within the vehicle body frame, however, it measures the motion of the vehicle with respect to its own case or platform frame, hereafter referred to as the *IMU case frame*. This frame is defined by the orthogonal triad of basis vectors $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$, and can be thought of as "fiducial lines which are physically inscribed" [61] on the IMU. The *IMU coordinate system*, denoted with a superscript c, is the preferred coordinate system of the IMU case frame. Hence, the coordinate axes 1^c, 2^c, and 3^c are aligned with the basis vectors $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$, respectively. The
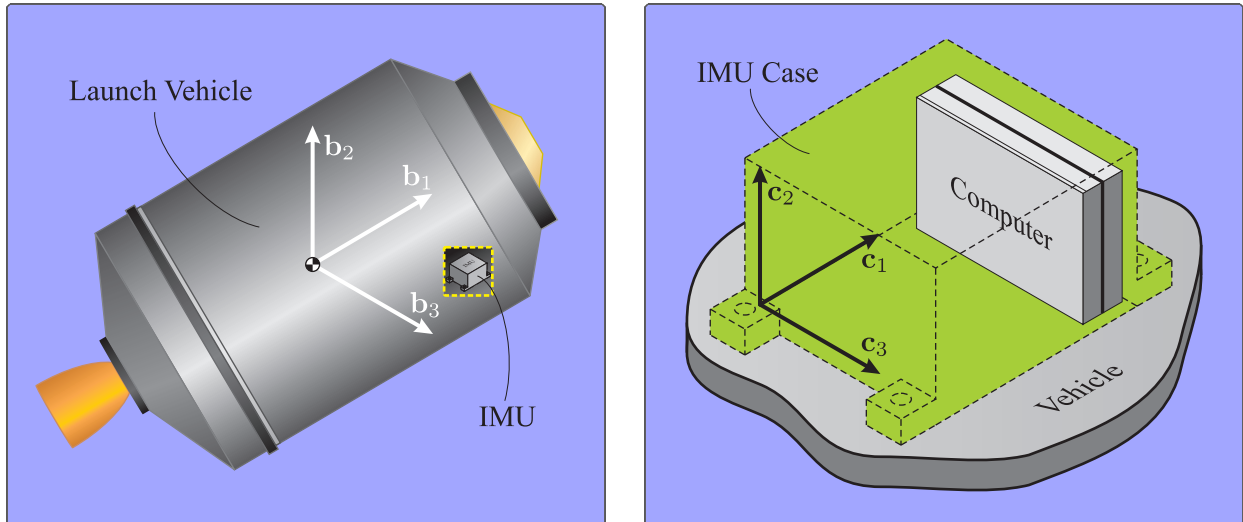
Figure 3.3: (*left*) IMU position is generally offset from the vehicle center of mass. (*right*) IMU case frame.

relationship between the IMU case and vehicle body frames is usually determined prior to launch through some form of instrument alignment procedures, and represented with the IMU-to-body transformation matrix $\boldsymbol{T}_c^b$.

For this research, only the strapdown implementation of the IMU is considered, as shown in Figure 3.4. It is also assumed that the IMU and associated case frame are collocated at the vehicle center of mass and the IMU-to-body transformation matrix $\boldsymbol{T}_c^b$ is known perfectly. Note that in general the IMU is not collocated with the center of mass, and the orientation of its frame with respect to the body frame is not perfectly known.

Accelerometer

An *accelerometer* is a mechanical device that uses the "inertial" properties of matter or light to measure the *specific force*, i.e., the nongravitational force per unit mass. The specific force (also called the g-force or mass-specific force) has units of acceleration or m/s^2. So it is not actually a force at all, but a type of acceleration. To be more specific, the specific force is a "proper" acceleration, which is the acceleration relative to free-fall or an inertial path. Neither the acceleration of gravity or "force of gravity" contribute to proper accelerations, thus the accelerometer does not measure the acceleration due to gravity. There are a variety of accelerometers used in aerospace, such as the pendulous accelerometer (in both open- and closed-loop varieties), the vibrating beam
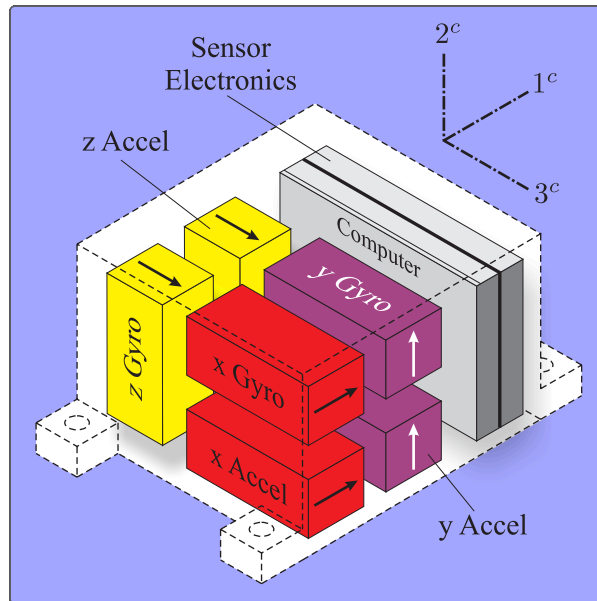
Figure 3.4: Strapdown IMU configuration.

accelerometer, the pendulous-integrating gyro accelerometer, and the micro-electro-mechanical systems (MEMS) accelerometer.

In general, three single-DOF accelerometers are mounted to the IMU case in such a manner that their input axes instrument an orthogonal set and together are able to measure the (nongravitational) acceleration in three dimensions, as depicted in Figure 3.5. Since perfect alignment of the accelerometer input axes with the IMU coordinate axes can never be achieved, despite even the most accurate alignment procedures, there exists some uncertainty in the coordinate transformation between the two right-handed, orthogonal systems of coordinates. If it is assumed that the two coordinate systems are nearly coincident, i.e., they differ by only "small" angles, then the angle from the coordinate axes 1^c to 1^a, 2^c to 2^a, and 3^c to 3^a are denoted by ϵ_1, ϵ_2, and ϵ_3 (respectively), as depicted in Figure 3.5. Thus, an orthogonal "small-angle" transformation between two almost coincident coordinate systems is given by [61, pp. 21]

$$\boldsymbol{T}_c^a = \boldsymbol{I}_{3\times3} - [\boldsymbol{\epsilon}\times] \ , \tag{3.23}$$
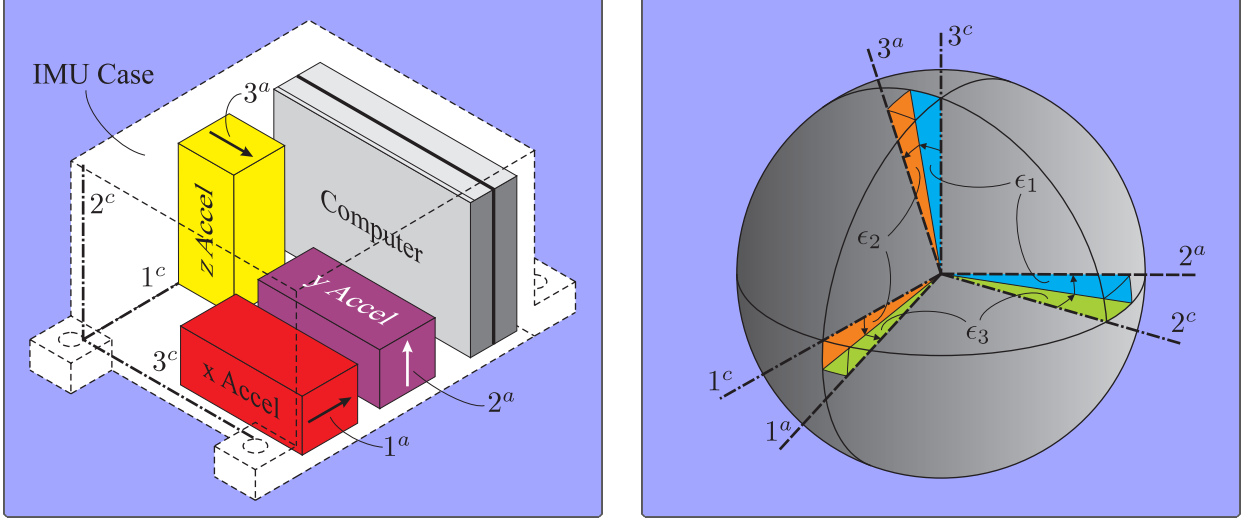
Figure 3.5: (*left*) Accelerometer input axes form orthogonal coordinate set. (*right*) Inertial instrument small-angle orthogonality geometry on the unit sphere.

where $[\boldsymbol{\epsilon}\times]$ is the skew symmetric form of the three small angles, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_3)$, defined as

$$[\boldsymbol{\epsilon}\times] \equiv \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix} . \tag{3.24}$$

Ideally, an accelerometer is sensitive to the specific force in only one precisely defined direction, called the *input axis*, which is generally perpendicular to its mounting surface. However, misalignments of the (maximum) *sensitivity axis* with respect to the input axis, due to manufacturing flaws and inaccuracies, cause the instrument to respond to components of the specific force in the plane parallel to the mounting surface, as shown in Figure 3.6. This sensitivity in the other directions is commonly referred to as the *cross-axis* or *transverse sensitivity*. Thus, the input axes are physically instrumented by the sensitive axes of the accelerometers, which constitute a nonorthogonal set of axes. Hence, special treatment is necessary in order to relate these axes to the orthogonal set of input axes. If it is assumed that the accelerometer's sensitive axes are nonorthogonal, but differ by only "small" angles with respect to the input axes, then the angles between the coordinate axes 1^a and 1^s, 2^a and 2^s, and 3^a and 3^s are denoted by γ_1, γ_2, and γ_3, respectively. These three angles are each broken up into two separate, independent rotations about the orthogonal input axes, as illustrated in Figure 3.6. Thus, the coordinate transformation relating
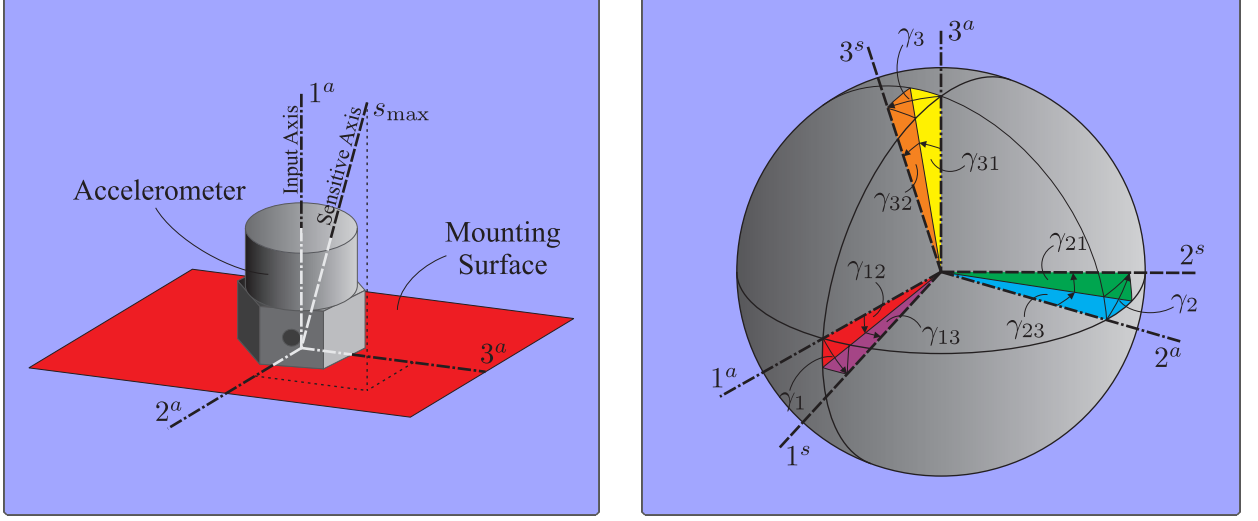
Figure 3.6: (*left*) Accelerometer input axes and sensitive axes geometry. (*right*) Inertial instrument small-angle nonorthogonality geometry on the unit sphere.

the nonorthogonoal sensitive axes to the input axes is given by [61, pp. 39]

$$\boldsymbol{T}_a^s = \boldsymbol{I}_{3\times3} - \boldsymbol{\Gamma}(\boldsymbol{\gamma}) \,, \tag{3.25}$$

where $\boldsymbol{\Gamma}(\boldsymbol{\gamma})$ is the nonorthogonal form of the six small angles, $\boldsymbol{\gamma} = (\gamma_{12}, \gamma_{13}, \gamma_{21}, \gamma_{23}, \gamma_{31}, \gamma_{32})$, defined as

$$\boldsymbol{\Gamma}(\boldsymbol{\gamma}) \equiv \begin{bmatrix} 0 & -\gamma_{13} & \gamma_{12} \\ \gamma_{23} & 0 & -\gamma_{21} \\ -\gamma_{32} & \gamma_{31} & 0 \end{bmatrix} \,. \tag{3.26}$$

An instrument's internal or input units are practically never the same as its output units. For example, the internal units of an accelerometer might be *volts*, whereas the output units are generally those of acceleration or m/s^2. Thus, some internal unit conversion factor is necessary. Moreover, the sensitivity of the instrument over a wide range of inputs is generally not a linear one-to-one relationship, i.e., the output of the device is not exactly proportional to the input by some constant, nonunitary factor. Thus, a *scale factor* is defined as the ratio between changes in the input and output signals [62–64]. Ideally, the mapping from internal to output units is a linear relationship, hence the scale factor can be computed as the slope of the best straight line fitted to data obtained by varying the input over a wide range of values and recording the output. If it is assumed that the accelerometer's scale factors s_1, s_2, and s_3 are linear, then the mapping is given

by [61, pp. 90]

$$S = I_{3\times3} + [s\searrow] \, , \tag{3.27}$$

where $[s\searrow]$ is the diagonal matrix of the scale factors, $s = (s_1, s_2, s_3)$, defined as

$$[s\searrow] = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} . \tag{3.28}$$

From the preceding discussion, it follows that errors in the measured nongravitational acceleration with respect to the body frame are typically due to scale factor uncertainties, orthogonal and nonorthogonal misalignments, random and systematic biases, and noise. After incorporating all of these error sources into a stochastic error model, the measured nongravitational acceleration vector, expressed in body coordinates, is given by

$$\tilde{a}_{\mathrm{ng}}^s = \left(I_{3\times3} + [s_{\tilde{a}}\searrow]\right)\left\{T_a^s \, T_c^a \, T_b^c \, a_{\mathrm{ng}}^b + b_{\tilde{a}}^s + \eta_{\tilde{a}}^s\right\} \, ,$$
$$= \left(I_{3\times3} + [s_{\tilde{a}}\searrow]\right)\left\{\left(I_{3\times3} - \Gamma(\gamma_{\tilde{a}})\right)\left(I_{3\times3} - [\epsilon_{\tilde{a}}\times]\right) a_{\mathrm{ng}}^b + b_{\tilde{a}}^s + \eta_{\tilde{a}}^s\right\} \, . \tag{3.29}$$

The accelerometer error parameter states, $p_{\tilde{a}} = (s_{\tilde{a}}, \gamma_{\tilde{a}}, \epsilon_{\tilde{a}}, b_{\tilde{a}}^s)$, are modeled as first-order Gauss-Markov processes, as defined in § 3.1.5, and $\eta_{\tilde{a}}^s$ is measurement noise, as defined in § 3.1.6.

Gyroscope

A gyroscope or gyro is a mechanical device that exploits the "inertial" properties of matter or light to measure the angular rate or velocity of the IMU case frame as it rotates with respect to the inertial reference frame. There are a variety of gyros used in aerospace, such as the mechanical single-DOF gyro (in both open- and closed-loop varieties), the mechanical two-DOF gyro, the vibrating gyro, the optical gyro (i.e., ring laser, interferometric fiber optic), and the MEMS gyro. A set of three single-DOF gyros is considered in this study. Moreover, all of the uncertainties associated with inertial instruments discussed previously are applicable to the gyro. Thus, it follows that errors in the measured angular velocity of the body frame with respect to the inertial frame are typically due to scale factor uncertainties, orthogonal and nonorthogonal misalignments, random and systematic biases, and noise. After incorporating all of these error sources into a stochastic

error model, the measured angular velocity vector, expressed in body coordinates, is given by

$$\tilde{\boldsymbol{\omega}}_{ib}^s = \left(\boldsymbol{I}_{3\times3} + [\boldsymbol{s}_{\tilde{\omega}}\diagdown]\right)\left\{\boldsymbol{T}_a^s\,\boldsymbol{T}_c^a\,\boldsymbol{T}_b^c\,\boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_{\tilde{\omega}}^s + \boldsymbol{\eta}_{\tilde{\omega}}^s\right\} ,$$

$$= \left(\boldsymbol{I}_{3\times3} + [\boldsymbol{s}_{\tilde{\omega}}\diagdown]\right)\left\{\left(\boldsymbol{I}_{3\times3} - \boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{\omega}})\right)\left(\boldsymbol{I}_{3\times3} - [\boldsymbol{\epsilon}_{\tilde{\omega}}\times]\right)\boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_{\tilde{\omega}}^s + \boldsymbol{\eta}_{\tilde{\omega}}^s\right\} . \tag{3.30}$$

The gyro error parameter states, $\boldsymbol{p}_{\tilde{\omega}} = (\boldsymbol{s}_{\tilde{\omega}}, \boldsymbol{\gamma}_{\tilde{\omega}}, \boldsymbol{\epsilon}_{\tilde{\omega}}, \boldsymbol{b}_{\tilde{\omega}}^s)$, are modeled as first-order Gauss-Markov processes, as defined in § 3.1.5, and $\boldsymbol{\eta}_{\tilde{\omega}}^s$ is measurement noise, as defined in § 3.1.6.

3.1.5 Error Parameters

All expected error parameters and unmodeled sources of error incorporated in the nonlinear models are modeled as continuous first-order Gauss-Markov processes, also known as *exponentially correlated random variables* (ECRVs), with dynamics given by

$$\dot{p} = -\frac{1}{\tau_p}p + w_p , \qquad p(t_0) \sim \mathcal{N}(0, \sigma_p^2) , \tag{3.31}$$

where τ_p is the associated time constant or correlation time and w_p is Gaussian white noise with

$$E\{w_p\} = 0 , \tag{3.32}$$

$$E\{w_p(t)\,w_p(t')\} = \frac{2\,\sigma_p^2}{\tau_p}\,\delta(t - t') , \tag{3.33}$$

where σ_p^2 is the steady-state variance of p, $E\{\,\}$ is the expectation operator, and $\delta(t-t')$ is the Dirac delta function. Modeling the error parameters as ECRVs provides a great deal of flexibility, this is because, depending on the time constant τ_p, the error parameter can range from being constant like a bias (where $\tau_p \to \infty$) to something more like white noise (where $\tau_p \to 0$). Hence, a large valued time constant results in a slowly changing random variable, whereas a small valued time constant results in a rapidly changing random variable.

3.1.6 Noise

There are two types of noise used in this research, viz. process noise and measurement noise. Process noise, denoted by the vector $\boldsymbol{w}(t)$, is modeled as uncorrelated Gaussian white noise

with mean and covariance, respectively,

$$E\{\boldsymbol{w}(t)\} = \boldsymbol{0} \ , \tag{3.34}$$

$$E\Big\{\boldsymbol{w}(t)\,\big[\boldsymbol{w}(t')\big]^{\mathsf{T}}\Big\} = \boldsymbol{S}_w(t)\,\delta(t - t') \ , \tag{3.35}$$

where $\boldsymbol{S}_w(t)$ is a diagonal matrix that represents the *strength* of the process noise and essentially defines the quality or accuracy of the truth models. Measurement noise, denoted with the vector $\boldsymbol{\eta}(t)$, is also modeled as uncorrelated Gaussian white noise with mean and covariance, respectively,

$$E\{\boldsymbol{\eta}(t)\} = \boldsymbol{0} \ , \tag{3.36}$$

$$E\Big\{\boldsymbol{\eta}(t)\,\big[\boldsymbol{\eta}(t')\big]^{\mathsf{T}}\Big\} = \boldsymbol{S}_\eta(t)\,\delta(t - t') \ , \tag{3.37}$$

where $\boldsymbol{S}_\eta(t)$ is also a diagonal matrix that represents the *strength* of the measurement noise and also defines the quality or accuracy of the measurement models.

3.2 GN&C Algorithm Models

The GN&C algorithm models are comprised of the inertial navigation, ascent guidance and steering, and attitude control. Each is introduced, defined, and discussed in greater detail in the following sections.

3.2.1 Inertial Navigation

Navigation is the determination of the position and velocity of a vehicle or body relative to some frame of reference. For example, terrestrial navigation is relative to the Earth and celestial navigation is relative to the stars, hence inertial navigation is relative to an inertial reference frame. Often times, the attitude determination problem is solved separately from the translational determination problem. For this research, however, a fully integrated navigation structure is considered, where *inertial navigation* refers to the determination of the position, velocity, and attitude of the launch vehicle relative to the inertial reference frame.

Inertial navigation on launch vehicles is generally accomplished with an unaided *inertial navigation system* (INS), which consists of an IMU and a data processing algorithm or navigation filter. The INS is unaided in the sense that no external measurements (e.g., optic or radiometric)

are used during powered ascent to aid in the estimation process. Moreover, the INS operates on the principle of *dead-reckoning*, i.e., the estimation of the current vehicle states using prior estimates or knowledge of the states, measured or computed rates, and the elapsed time. In other words, the IMU-measured angular velocity is numerically integrated to obtain the attitude and corresponding inertial-to-body transformation matrix, which is then used to transform the IMU-measured specific force to inertial coordinates. The specific force, now expressed in inertial coordinates, is then added to the inertial gravitational acceleration, which typically comes from an onboard gravity model, to obtain the total inertial acceleration. Finally, the total inertial acceleration is numerically integrated once to obtain the inertial velocity and a second time to obtain the inertial position.

The most extensively used data processing algorithm for nonlinear state estimation and inertial navigation on launch vehicles is the extended Kalman filter (EKF), an extension of the optimal linear estimator developed and proposed by Kalman [8] in 1960. In general, the EKF incorporates the standard two-stage process (as illustrated in Figure 3.7): propagate and update. The propagation stage makes use of the nonlinear system dynamics model and linearized error dynamics, and the update stages makes use of the nonlinear measurement models and linearized measurement sensitivities. Since the IMU (which provides the continuous inertial measurements $\tilde{\boldsymbol{y}}$) is the only sensor considered for this research, it follows that additional sensors (which generally provide discrete measurements $\tilde{\boldsymbol{z}}_k$) and the update stage of the filter, depicted in Figure 3.7, are not considered here but left for future work. For a brief history of the EKF, the reader is referred to the work of Jazwinski [65]. For a detailed examination of the practical aspects of filter design, the reader is referred to the work of Gelb [5], whereas discussions on the more theoretical aspects of Kalman filtering can be found in the works of Jazwinski [65], Maybeck [1], Tapley, Shutz, and Born [24], and Crassidis and Junkins [66].

Filter Design Models

The *true system models* or simply the "truth" models, defined previously in § 3.1, are generally the best representation of the "real world" for the research effort. In the development of the navigation filter, the *filter design models* are the *truth* models of the system as far as the the filter is concerned, but are usually simplified versions of the true system models. In other words, they are scaled to capture the needed complexity required for state estimation but reduced
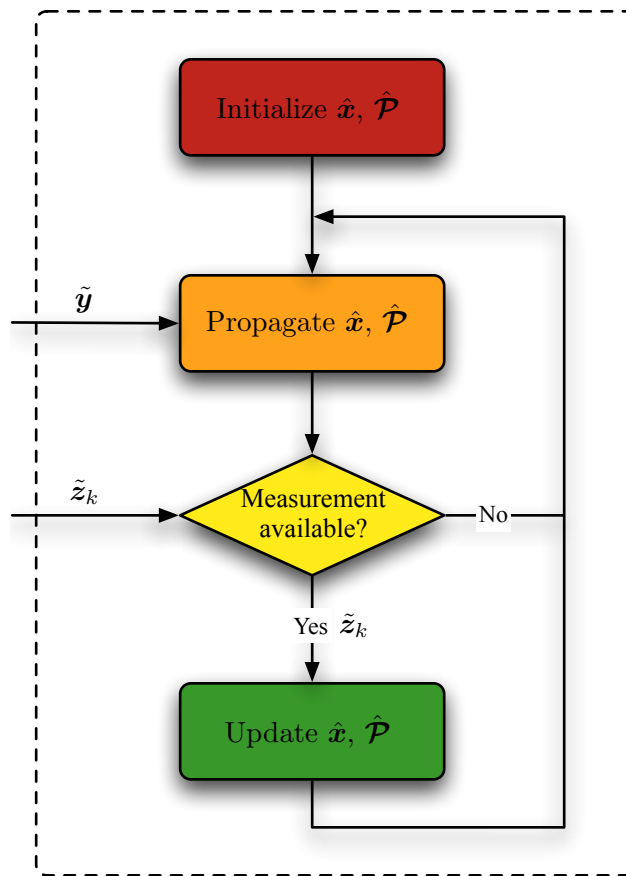
Figure 3.7: Flow diagram of the extended Kalman filter for inertial navigation.

to meet computational limitations or requirements. For example, the gravitational acceleration in the "truth" models may include an n-body gravity model and harmonic terms to respectively account for the gravitational effects of other bodies (e.g., Sun, moon, or other planets) and the nonspherical shape of the primary body, while the filter design models may include only a two-body gravity model that assumes that the primary body is a point-mass. The inertial navigation filter formulated and developed for this research includes models of the continuous inertial measurements from the strapdown IMU, the translational and rotational dynamics of the launch vehicle, and the dynamics of the modeled error parameters.

The continuous inertial measurements processed by the filter are solely those from the IMU, viz. the specific force $\tilde{a}^s \in \mathbb{R}^3$ and angular rate (or angular velocity) $\tilde{\omega}^s \in \mathbb{R}^3$ as measured respectively in the sensor frames by three accelerometers and three gyroscopes. It is assumed that both sets of measurements are corrupted by scale factor (s) errors, orthogonal (ϵ) and nonorthogonal (γ) misalignment uncertainties, random and systematic biases (b), and noise (η). Each of these

measurement errors are described in greater detail in the IMU model as given in § 3.1.4. Hence, the corrupted accelerometer measurement $\tilde{\boldsymbol{a}}^s$, expressed in sensor coordinates, is given by

$$\tilde{\boldsymbol{a}}^s(\boldsymbol{a}_{\text{ng}}^b, \boldsymbol{p}_{\tilde{a}}, \boldsymbol{\eta}_{\tilde{a}}^s) = \left(\boldsymbol{I}_{3\times3} + [\boldsymbol{s}_{\tilde{a}}\searrow]\right)\left\{\left(\boldsymbol{I}_{3\times3} - \boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{a}})\right)\left(\boldsymbol{I}_{3\times3} - [\boldsymbol{\epsilon}_{\tilde{a}}\times]\right)\boldsymbol{a}_{\text{ng}}^b + \boldsymbol{b}_{\tilde{a}}^s + \boldsymbol{\eta}_{\tilde{a}}^s\right\}, \qquad (3.38)$$

where $\boldsymbol{a}_{\text{ng}}^b \in \mathbb{R}^3$ is the *true* specific force (or nongravitational acceleration) vector expressed in body coordinates, $[\boldsymbol{s}_{\tilde{a}}\searrow]$ is the diagonal matrix of the accelerometer scale factor error vector $\boldsymbol{s}_{\tilde{a}} \in \mathbb{R}^3$ as defined in Eq. (3.28) on page 31, $\boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{a}})$ is the nonorthogonal form of the accelerometer nonorthogonal misalignment error vector $\boldsymbol{\gamma}_{\tilde{a}} \in \mathbb{R}^6$ as defined in Eq. (3.26) on page 30, and $[\boldsymbol{\epsilon}_{\tilde{a}}\times]$ is the skew symmetric form of the accelerometer orthogonal misalignment error vector $\boldsymbol{\epsilon}_{\tilde{a}} \in \mathbb{R}^3$ as defined in Eq. (3.24) on page 29. Moreover, the accelerometer error parameters $\boldsymbol{p}_{\tilde{a}} = (\boldsymbol{s}_{\tilde{a}}, \boldsymbol{\gamma}_{\tilde{a}}, \boldsymbol{\epsilon}_{\tilde{a}}, \boldsymbol{b}_{\tilde{a}}^s)$ are modeled as first-order Gauss-Markov processes as defined in § 3.1.5 on page 32, and the accelerometer measurement noise $\boldsymbol{\eta}_{\tilde{a}}^s \in \mathbb{R}^3$ is modeled as Gaussian white noise with

$$E\{\boldsymbol{\eta}_{\tilde{a}}^s(t)\} = \boldsymbol{0}, \qquad (3.39)$$

$$E\left\{\boldsymbol{\eta}_{\tilde{a}}^s(t)\left[\boldsymbol{\eta}_{\tilde{a}}^s(t')\right]^{\mathsf{T}}\right\} = \sigma_{\eta_{\tilde{a}}}^2 \boldsymbol{I}_{3\times3}\,\delta(t - t'). \qquad (3.40)$$

Similarly, the corrupted gyro measurement $\tilde{\boldsymbol{\omega}}^s$, expressed in sensor coordinates, is given by

$$\tilde{\boldsymbol{\omega}}^s(\boldsymbol{\omega}_{ib}^b, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s) = \left(\boldsymbol{I}_{3\times3} + [\boldsymbol{s}_{\tilde{\omega}}\searrow]\right)\left\{\left(\boldsymbol{I}_{3\times3} - \boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{\omega}})\right)\left(\boldsymbol{I}_{3\times3} - [\boldsymbol{\epsilon}_{\tilde{\omega}}\times]\right)\boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_{\tilde{\omega}}^s + \boldsymbol{\eta}_{\tilde{\omega}}^s\right\}, \qquad (3.41)$$

where $\boldsymbol{\omega}_{ib}^b \in \mathbb{R}^3$ is the *true* angular velocity of the launch vehicle with respect to the inertial frame expressed in body coordinates, $[\boldsymbol{s}_{\tilde{\omega}}\searrow]$ is the diagonal matrix of the gyro scale factor error vector $\boldsymbol{s}_{\tilde{\omega}} \in \mathbb{R}^3$ as defined in Eq. (3.28), $\boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{\omega}})$ is the nonorthogonal form of the gyro nonorthogonal misalignment error vector $\boldsymbol{\gamma}_{\tilde{\omega}} \in \mathbb{R}^6$ as defined in Eq. (3.26), and $[\boldsymbol{\epsilon}_{\tilde{\omega}}\times]$ is the skew symmetric form of the gyro orthogonal misalignment error vector $\boldsymbol{\epsilon}_{\tilde{\omega}} \in \mathbb{R}^3$ as defined in Eq. (3.24). In addition, the gyro error parameters $\boldsymbol{p}_{\tilde{\omega}} = (\boldsymbol{s}_{\tilde{\omega}}, \boldsymbol{\gamma}_{\tilde{\omega}}, \boldsymbol{\epsilon}_{\tilde{\omega}}, \boldsymbol{b}_{\tilde{\omega}}^s)$ are modeled as first-order Gauss-Markov processes as defined in § 3.1.5 on page 32, and the gyro measurement noise $\boldsymbol{\eta}_{\tilde{\omega}}^s \in \mathbb{R}^3$ is modeled as Gaussian

white noise with

$$E\{\boldsymbol{\eta}_{\tilde{\omega}}^s(t)\} = \mathbf{0} \ , \tag{3.42}$$

$$E\Big\{\boldsymbol{\eta}_{\tilde{\omega}}^s(t) \left[\boldsymbol{\eta}_{\tilde{\omega}}^s(t')\right]^\mathsf{T}\Big\} = \sigma_{\eta_{\tilde{\omega}}}^2 \boldsymbol{I}_{3\times 3}\, \delta(t-t') \ . \tag{3.43}$$

Since continuous accelerometer measurements are available, the differential equations of motion that describe the translational dynamics of the launch vehicle's c.m. with respect to the inertial frame, as governed by Newton's second law, are written as

$$\dot{\boldsymbol{r}}^i = \boldsymbol{v}^i \ , \tag{3.44}$$

$$\dot{\boldsymbol{v}}^i = \boldsymbol{a}_{\mathrm{g}}^i(\boldsymbol{r}^i) + \left[\boldsymbol{T}(\boldsymbol{q}_i^b)\right]^\mathsf{T} \boldsymbol{a}_{\mathrm{ng}}^b(\tilde{\boldsymbol{a}}^s, \boldsymbol{p}_{\tilde{a}}, \boldsymbol{\eta}_{\tilde{a}}^s) \ , \tag{3.45}$$

where $\boldsymbol{a}_{\mathrm{g}}^i \in \mathbb{R}^3$ is the gravitational acceleration vector, expressed in inertial coordinates, and defined by the two-body gravity (or point-mass) model

$$\boldsymbol{a}_{\mathrm{g}}^i(\boldsymbol{r}^i) = -\frac{\mu}{\left\|\boldsymbol{r}^i\right\|^3}\boldsymbol{r}^i \ , \tag{3.46}$$

$\boldsymbol{T}(\boldsymbol{q}_i^b) \in \mathbb{R}^{3\times 3}$ is the inertial-to-body transformation matrix, and $\boldsymbol{a}_{\mathrm{ng}}^b \in \mathbb{R}^3$ is the nongravitational acceleration vector, expressed in body coordinates, recovered from the accelerometer measurement $\tilde{\boldsymbol{a}}^s$ by eliminating the effects of the accelerometer errors $\boldsymbol{p}_{\tilde{a}}$ and noise $\boldsymbol{\eta}_{\tilde{a}}^s$ through error compensation, as given by

$$\boldsymbol{a}_{\mathrm{ng}}^b(\tilde{\boldsymbol{a}}^s, \boldsymbol{p}_{\tilde{a}}, \boldsymbol{\eta}_{\tilde{a}}^s) = \left(\boldsymbol{I}_{3\times 3} + [\boldsymbol{\epsilon}_{\tilde{a}}\times]\right)\left(\boldsymbol{I}_{3\times 3} + \boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{a}})\right)\Big\{\left(\boldsymbol{I}_{3\times 3} - [\boldsymbol{s}_{\tilde{a}}\diagdown]\right)\tilde{\boldsymbol{a}}^s - \boldsymbol{b}_{\tilde{a}}^s - \boldsymbol{\eta}_{\tilde{a}}^s\Big\} \ . \tag{3.47}$$

This expression is the result of solving Eq. (3.38) for $\boldsymbol{a}_{\mathrm{ng}}^b$ and using the first-order inverse approximation [67]

$$\left(\boldsymbol{I}_{n\times n} + \boldsymbol{A}\right)^{-1} \cong \left(\boldsymbol{I}_{n\times n} - \boldsymbol{A}\right) \ , \tag{3.48}$$

which can be derived from the *Sherman-Morrison lemma* [66, Eq. (A.20)]

$$\left(\boldsymbol{I}_{n\times n} + \boldsymbol{A}\right)^{-1} = \boldsymbol{I}_{n\times n} - \boldsymbol{A}\left(\boldsymbol{I}_{n\times n} + \boldsymbol{A}\right)^{-1} \ , \tag{3.49}$$

by recursively substituting the above expression into itself and keeping only the first-order terms. Note that Eq. (3.48) holds if the elements of the matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ are 'small', which for this particular application is met by definition of the IMU sensor error parameters $(\boldsymbol{s}, \boldsymbol{\gamma}, \boldsymbol{\epsilon})$ [see § 3.1.4 on page 25].

Since continuous gyro measurements are also available, the only differential equation of motion needed to describe the rotational dynamics of the launch vehicle about its c.m. with respect to the inertial frame is

$$\dot{\boldsymbol{q}}_i^b = -\frac{1}{2}\boldsymbol{q}_\omega(\boldsymbol{\omega}_{ib}^b) \otimes \boldsymbol{q}_i^b \ , \tag{3.50}$$

where $\boldsymbol{q}_\omega \in \mathbb{R}^4$ is the pure quaternion defined as

$$\boldsymbol{q}_\omega(\boldsymbol{\omega}_{ib}^b) \equiv \begin{bmatrix} \boldsymbol{\omega}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s) \\ 0 \end{bmatrix} \ , \tag{3.51}$$

and $\boldsymbol{\omega}_{ib}^b \in \mathbb{R}^3$ is the angular velocity of the launch vehicle, expressed in body coordinates, recovered from the gyro measurement $\tilde{\boldsymbol{\omega}}^s$ by eliminating the effects of the the gyro errors $\boldsymbol{p}_{\tilde{\omega}}$ and noise $\boldsymbol{\eta}_{\tilde{\omega}}^s$ through error compensation, as given by

$$\boldsymbol{\omega}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s) = \left(\boldsymbol{I}_{3\times3} + [\boldsymbol{\epsilon}_{\tilde{\omega}}\times]\right)\left(\boldsymbol{I}_{3\times3} + \boldsymbol{\Gamma}(\boldsymbol{\gamma}_{\tilde{\omega}})\right)\left\{\left(\boldsymbol{I}_{3\times3} - [\boldsymbol{s}_{\tilde{\omega}}\diagdown]\right)\tilde{\boldsymbol{\omega}}^s - \boldsymbol{b}_{\tilde{\omega}}^s - \boldsymbol{\eta}_{\tilde{\omega}}^s\right\} \ . \tag{3.52}$$

This expression is the result of solving Eq. (3.41) for $\boldsymbol{\omega}_{ib}^b$ and using the first-order inverse approximation given in Eq. (3.48). Note that the gyro measurement provides the angular velocity of the launch vehicle's center-of-mass with respect to the inertial frame, thus eliminating the need to explicitly integrate Euler's equation, include an angular velocity state, or maintain models of the vehicle's mass moments-of-inertia, attitude control system, or any other disturbance torques acting on the launch vehicle. This is referred to in the literature as *model replacement* [4, 54, 55, 68], the advantage being that it greatly reduces the complexity of the filter. The disadvantage, however, being that because an angular velocity state is not explicitly maintained, its corresponding covariance is not maintained either and hence unavailable.

The filter is capable of modeling error parameters to help mitigate the effects of various errors, such as accelerometer and gyro errors, through error compensation, with examples given in Eqs. (3.47) and (3.52). As mentioned earlier, these error parameters are modeled as first-order

Gauss-Markov processes as defined in § 3.1.5 on page 32. For this research, the filter is setup to model the accelerometer and gyro error parameters $\boldsymbol{p}_{\tilde{a}}$ and $\boldsymbol{p}_{\tilde{\omega}}$ respectively, which were introduced and defined previously, and the ACS error parameters $\boldsymbol{p}_{\mathrm{acs}}$, which will be introduced and defined later in § 3.2.3 on page 53.

The differential equations of motion given in Eqs. (3.44), (3.45), (3.50), and (3.31) can be written in the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), t\big) - \mathbf{B}\,\boldsymbol{\eta}(t) + \mathbf{G}\,\mathbf{w}(t) \ , \tag{3.53}$$

where $\mathbf{f} : \mathbb{R}^{m'} \to \mathbb{R}^{m'}$ is a nonlinear vector function of the true navigation state vector $\mathbf{x}(t) \in \mathbb{R}^{m'}$ given by

$$\mathbf{x} = \big(\boldsymbol{r}^i,\, \boldsymbol{v}^i,\, \boldsymbol{q}_i^b,\, \boldsymbol{p}\big) \ , \tag{3.54}$$

$\boldsymbol{\eta}(t) \in \mathbb{R}^l = \big(\boldsymbol{\eta}_{\tilde{a}}^s, \boldsymbol{\eta}_{\tilde{\omega}}^s\big)$ is the continuous inertial measurement noise vector, and $\mathbf{w}(t) \in \mathbb{R}^p$ is a vector of Gaussian white noise terms from Eq. (3.31) associated with the error parameter states $\boldsymbol{p}(t) \in \mathbb{R}^p = (\boldsymbol{p}_{\tilde{a}}, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{p}_{\mathrm{acs}})$. Moreover, $\mathbf{B} \in \mathbb{R}^{m' \times l}$ and $\mathbf{G} \in \mathbb{R}^{m' \times p}$ are matrices of ones and zeros that respectively map $\boldsymbol{\eta}$ and $\mathbf{w}$ to the proper states in $\mathbf{x}$. It is important to note that the function $\mathbf{f}$ in Eq. (3.53) is not written explicitly as a function of the continuous inertial measurements $\tilde{\boldsymbol{y}}(t) \in \mathbb{R}^l = (\tilde{\boldsymbol{a}}^s, \tilde{\boldsymbol{\omega}}^s)$ because, as far as the filter is concerned, the measurements are deterministic inputs.

Navigation State Propagation

Now that the filter design models have been defined, the next step in developing the navigation filter is to formulate the navigation state propagation equations. Since the desired navigation filter is to be an EKF, it is assumed that

$$E\big\{\mathbf{x}(t)\big\} = \hat{\boldsymbol{x}}(t) \ , \quad \forall \, t \ . \tag{3.55}$$

In other words, the conditional mean of $\mathbf{x}(t)$ is $\hat{\boldsymbol{x}}(t)$, i.e., the *estimate* of the true navigation state or simply the *navigation state*. Therefore, the navigation state propagation equations are obtained by taking the expectation value of the filter design models, such that the differential equations of

motion become

$$\dot{\hat{\boldsymbol{r}}}^i = \hat{\boldsymbol{v}}^i \ , \tag{3.56}$$

$$\dot{\hat{\boldsymbol{v}}}^i = \hat{\boldsymbol{a}}_{\mathrm{g}}^i(\hat{\boldsymbol{r}}^i) + \big[\hat{\boldsymbol{T}}(\hat{\boldsymbol{q}}_i^b)\big]^{\mathsf{T}} \hat{\boldsymbol{a}}_{\mathrm{ng}}^b(\tilde{\boldsymbol{a}}^s, \hat{\boldsymbol{p}}_{\tilde{a}}) \ , \tag{3.57}$$

$$\dot{\hat{\boldsymbol{q}}}_i^b = -\frac{1}{2}\hat{\boldsymbol{q}}_\omega(\hat{\boldsymbol{\omega}}_{ib}^b) \otimes \hat{\boldsymbol{q}}_i^b \ , \tag{3.58}$$

$$\dot{\hat{p}}_j = -\frac{1}{\tau_{\hat{p}_j}}\hat{p}_j \ , \quad j = 1, 2, \ldots, p \tag{3.59}$$

where

$$\hat{\boldsymbol{a}}_{\mathrm{g}}^i(\hat{\boldsymbol{r}}^i) = -\frac{\mu}{\left\lVert \hat{\boldsymbol{r}}^i \right\rVert^3}\hat{\boldsymbol{r}}^i \ , \tag{3.60}$$

$$\hat{\boldsymbol{a}}_{\mathrm{ng}}^b(\tilde{\boldsymbol{a}}^s, \hat{\boldsymbol{p}}_{\tilde{a}}) = \big(\boldsymbol{I}_{3\times3} + [\hat{\boldsymbol{\epsilon}}_{\tilde{a}}\times]\big)\big(\boldsymbol{I}_{3\times3} + \boldsymbol{\Gamma}(\hat{\boldsymbol{\gamma}}_{\tilde{a}})\big)\Big\{\big(\boldsymbol{I}_{3\times3} - [\hat{\boldsymbol{s}}_{\tilde{a}}\searrow]\big)\tilde{\boldsymbol{a}}^s - \hat{\boldsymbol{b}}_{\tilde{a}}^s\Big\} \ , \tag{3.61}$$

$$\hat{\boldsymbol{q}}_\omega(\hat{\boldsymbol{\omega}}_{ib}^b) = \begin{bmatrix} \hat{\boldsymbol{\omega}}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \hat{\boldsymbol{p}}_{\tilde{\omega}}) \\ 0 \end{bmatrix} \ , \tag{3.62}$$

and

$$\hat{\boldsymbol{\omega}}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \hat{\boldsymbol{p}}_{\tilde{\omega}}) = \big(\boldsymbol{I}_{3\times3} + [\hat{\boldsymbol{\epsilon}}_{\tilde{\omega}}\times]\big)\big(\boldsymbol{I}_{3\times3} + \boldsymbol{\Gamma}(\hat{\boldsymbol{\gamma}}_{\tilde{\omega}})\big)\Big\{\big(\boldsymbol{I}_{3\times3} - [\hat{\boldsymbol{s}}_{\tilde{\omega}}\searrow]\big)\tilde{\boldsymbol{\omega}}^s - \hat{\boldsymbol{b}}_{\tilde{\omega}}^s\Big\} \ , \tag{3.63}$$

Moreover, it follows that the expectation value of Eq. (3.53) is

$$\dot{\hat{\boldsymbol{x}}}(t) = \hat{\boldsymbol{f}}\big(\hat{\boldsymbol{x}}(t), t\big) \ , \tag{3.64}$$

where $\hat{\boldsymbol{f}} : \mathbb{R}^{m'} \to \mathbb{R}^{m'}$ is a nonlinear vector function of the navigation states $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^{m'}$ given by

$$\hat{\boldsymbol{x}} = \big(\hat{\boldsymbol{r}}^i, \ \hat{\boldsymbol{v}}^i, \ \hat{\boldsymbol{q}}_i^b, \ \hat{\boldsymbol{p}}\big) \ , \tag{3.65}$$

where $\hat{\boldsymbol{p}}(t) \in \mathbb{R}^p = (\hat{\boldsymbol{p}}_{\tilde{a}}, \ \hat{\boldsymbol{p}}_{\tilde{\omega}}, \ \hat{\boldsymbol{p}}_{\mathrm{acs}})$.

Navigation State Error Covariance Propagation

Now that the filter navigation state propagation equations have been formulated, the final step in developing the navigation filter is to formulate the filter navigation state error covariance propagation equations. First, the filter navigation state error $\delta\hat{\boldsymbol{e}} \in \mathbb{R}^{m'}$ is defined as the difference between the true navigation states **x** (which represents the "truth" for the filter design) and the

navigation states $\hat{\boldsymbol{x}}$ (which is the filter's estimate of the true states),

$$\delta\hat{\boldsymbol{e}}(t) \equiv \mathbf{x}(t) - \hat{\boldsymbol{x}}(t) \ , \tag{3.66}$$

with

$$E\big\{\delta\hat{\boldsymbol{e}}(t)\big\} = \mathbf{0} \ , \ \ \forall \ t \tag{3.67}$$

$$E\Big\{\delta\hat{\boldsymbol{e}}(t)\big[\delta\hat{\boldsymbol{e}}(t)\big]^{\mathsf{T}}\Big\} = \hat{\boldsymbol{\mathcal{P}}}(t) \ , \tag{3.68}$$

where $\hat{\boldsymbol{\mathcal{P}}}(t) \in \mathbb{R}^{m\times m}$ is the filter navigation state error covariance matrix, which represents the filter's measure of the estimation accuracy [5]. Next, Eq. (3.66) is substituted in Eq. (3.53) and $\mathbf{f}$ is expanded in a first-order Taylor series about the current estimate (conditional mean) of the true navigation state $\hat{\boldsymbol{x}}(t)$,

$$\mathbf{f}\big(\mathbf{x}(t),t\big) \approx \hat{\boldsymbol{f}}\big(\hat{\boldsymbol{x}}(t),t\big) + \frac{\partial\mathbf{f}\big(\mathbf{x}(t),t\big)}{\mathbf{x}(t)}\bigg|_{\mathbf{x}(t)=\hat{\boldsymbol{x}}(t)} \big(\mathbf{x}(t) - \hat{\boldsymbol{x}}(t)\big) \ ,$$

such that the linear differential equation is obtained

$$\delta\dot{\hat{\boldsymbol{e}}}(t) = \mathbf{F}_{\mathsf{x}}\big(\hat{\boldsymbol{x}}(t),t\big)\,\delta\hat{\boldsymbol{e}}(t) - \mathbf{B}\,\boldsymbol{\eta}(t) + \mathbf{G}\,\mathbf{w}(t) \ , \tag{3.69}$$

where $\mathbf{F}_{\mathsf{x}} \in \mathbb{R}^{m\times m}$ is the Jacobian defined as

$$\mathbf{F}_{\mathsf{x}}\big(\hat{\boldsymbol{x}}(t),t\big) \equiv \frac{\partial\mathbf{f}\big(\mathbf{x}(t),t\big)}{\mathbf{x}_{\mathrm{m}}(t)}\bigg|_{\mathbf{x}(t)=\hat{\boldsymbol{x}}(t)} \ , \tag{3.70}$$

and $\mathbf{B} \in \mathbb{R}^{m\times l}$ and $\mathbf{G} \in \mathbb{R}^{m\times p}$ are now matrices of ones and zeros that respectively map $\boldsymbol{\eta}$ and $\mathbf{w}$ to the proper states in $\mathbf{x}_{\mathrm{m}}$.

Recall that the inertial-to-body attitude quaternion $\boldsymbol{q}_i^b \in \mathbb{R}^4$ is the standard representation of the launch vehicle attitude. However, due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the filter navigation state error covariance propagation equations. In this approach, the inertial-to-body attitude quaternions $\boldsymbol{q}_i^b$, $\hat{\boldsymbol{q}}_i^b \in \mathbb{R}^4$ respectively in the true navigation state vector $\mathbf{x} \in \mathbb{R}^{m'}$ defined in Eq. (3.54) and the navigation state vector $\hat{\boldsymbol{x}} \in \mathbb{R}^{m'}$ defined in Eq. (3.65) are replaced with the

Euler rotation vectors $\boldsymbol{\theta}_i^b$, $\hat{\boldsymbol{\theta}}_i^b \in \mathbb{R}^3$, which represent the attitude of the body frame b with respect to the inertial frame i expressed in body coordinates, resulting in a *modified true navigation state vector* $\mathbf{x}_{\mathrm{m}}(t) \in \mathbb{R}^m$ (where $m = m' - 1$), given by

$$\mathbf{x}_{\mathrm{m}} = \left(\boldsymbol{r}^i,\, \boldsymbol{v}^i,\, \boldsymbol{\theta}_i^b,\, \boldsymbol{p}\right) , \tag{3.71}$$

and a *modified navigation state vector* $\hat{\boldsymbol{x}}_{\mathrm{m}} \in \mathbb{R}^m$ given by

$$\hat{\boldsymbol{x}}_{\mathrm{m}} = \left(\hat{\boldsymbol{r}}^i,\, \hat{\boldsymbol{v}}^i,\, \hat{\boldsymbol{\theta}}_i^b,\, \hat{\boldsymbol{p}}\right) . \tag{3.72}$$

In addition, the quaternion kinematics in Eqs. (3.50) and (3.58) are replaced with the linearized Bortz equation [55]. As such, the linearized filter navigation state error dynamics equation, given in Eq. (3.69), is based upon the modified true navigation state vector $\mathbf{x}_{\mathrm{m}}$ and modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}}$, i.e., $\delta\hat{\boldsymbol{e}}(t) \in \mathbb{R}^m$, and the Jacobian $\mathbf{F}_{\mathsf{x}}$ has been defined and mapping matrices $\mathbf{B}$ and $\mathbf{G}$ have been redefined accordingly.

Given the definition of the modified true navigation state vector in Eq. (3.71), the Jacobian $\mathbf{F}_{\mathsf{x}}$, defined in Eq. (3.70), can be expanded and written as

$$\mathbf{F}_{\mathsf{x}}\big(\hat{\boldsymbol{x}}(t), t\big) \equiv \left.\frac{\partial \mathbf{f}\big(\mathbf{x}(t), t\big)}{\mathbf{x}_{\mathrm{m}}(t)}\right|_{\mathbf{x}(t) = \hat{\boldsymbol{x}}(t)} = \begin{bmatrix} \left.\dfrac{\partial \dot{\boldsymbol{r}}^i}{\partial \mathbf{x}_{\mathrm{m}}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}} \\[2ex] \left.\dfrac{\partial \dot{\boldsymbol{v}}^i}{\partial \mathbf{x}_{\mathrm{m}}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}} \\[2ex] \left.\dfrac{\partial \dot{\boldsymbol{\theta}}_i^b}{\partial \mathbf{x}_{\mathrm{m}}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}} \\[2ex] \left.\dfrac{\partial \dot{p}_j}{\partial \mathbf{x}_{\mathrm{m}}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}} \\[1ex] \vdots \end{bmatrix} . \tag{3.73}$$

The elements of the Jacobian $\mathbf{F}_{\mathsf{x}}$ associated with the inertial position kinematics, defined in Eq. (3.44), are given by

$$\left.\frac{\partial \dot{\boldsymbol{r}}^i}{\partial \mathbf{x}_{\mathrm{m}}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}} = \left.\frac{\partial \boldsymbol{v}^i}{\partial \mathbf{x}_{\mathrm{m}}}\right|_{\mathbf{x}=\hat{\boldsymbol{x}}} = \frac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} . \tag{3.74}$$

Given the definition of the modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}}$ in Eq. (3.72), it follows that the partial derivative of the inertial velocity $\hat{\boldsymbol{v}}^i$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ can be expanded and written in terms of each of the states contained in the modified navigation state

vector as follows

$$\frac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} = \left[\begin{array}{cccc} \dfrac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{r}}^i} & \dfrac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{v}}^i} & \dfrac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{\theta}}_i^b} & \dfrac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{p}}} \end{array} \right] . \tag{3.75}$$

Each of these partial derivatives can then be evaluated, thereby resulting in

$$\frac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} = \left[\begin{array}{cccc} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times p} \end{array} \right] . \tag{3.76}$$

Note that the partial derivative of the inertial velocity $\hat{\boldsymbol{v}}^i$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ is an example of a simple or basic partial derivative, i.e., evaluating the partial derivative yields a matrix of ones and zeros. This is due to the fact that the inertial velocity $\hat{\boldsymbol{v}}^i$ is itself a navigation state and a member of the modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}}$. Rather than expand and write out these basic partial derivatives explicitly, they are hereafter expressed only in their general form, i.e., $\frac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}$. Subsequently, it can be shown that the elements of the Jacobian $\mathsf{F_x}$ associated with the inertial velocity dynamics, defined in Eq. (3.45), are given by

$$\frac{\partial \dot{\boldsymbol{v}}^i}{\partial \mathsf{x}_{\mathrm{m}}} \bigg|_{\mathsf{x}=\hat{\boldsymbol{x}}} = \frac{\partial \boldsymbol{a}_{\mathrm{g}}^i(\boldsymbol{r}^i)}{\partial \mathsf{x}_{\mathrm{m}}} \bigg|_{\mathsf{x}=\hat{\boldsymbol{x}}} + \left[\hat{\boldsymbol{T}}(\hat{\boldsymbol{q}}_i^b)\right]^{\mathsf{T}} \left(-[\hat{\boldsymbol{a}}_{\mathrm{ng}}^b \times] \frac{\partial \hat{\boldsymbol{\theta}}_i^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} + \frac{\partial \boldsymbol{a}_{\mathrm{ng}}^b(\tilde{\boldsymbol{a}}^s, \boldsymbol{p}_{\tilde{a}}, \boldsymbol{\eta}_{\tilde{a}}^s)}{\partial \mathsf{x}_{\mathrm{m}}} \bigg|_{\mathsf{x}=\hat{\boldsymbol{x}}} \right) , \tag{3.77}$$

where $\hat{\boldsymbol{a}}_{\mathrm{ng}}^b$ is the compensated accelerometer measurement defined in Eq. (3.61) and

$$\frac{\partial \boldsymbol{a}_{\mathrm{g}}^i(\boldsymbol{r}^i)}{\partial \mathsf{x}_{\mathrm{m}}} \bigg|_{\mathsf{x}=\hat{\boldsymbol{x}}} = -\frac{\mu}{\left\|\hat{\boldsymbol{r}}^i\right\|^3} \left(\boldsymbol{I}_{3\times3} - 3\, \hat{\mathbf{u}}_r^i \left[\hat{\mathbf{u}}_r^i\right]^{\mathsf{T}} \right) \frac{\partial \hat{\boldsymbol{r}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} , \tag{3.78}$$

where $\hat{\mathbf{u}}_r^i$ is the unit vector in the direction of the (estimated) inertial position vector $\hat{\boldsymbol{r}}^i$. Moreover, the partial derivative of $\boldsymbol{a}_{\mathrm{ng}}^b$ with respect to x_{m} is given by

$$\begin{aligned} \frac{\partial \boldsymbol{a}_{\mathrm{ng}}^b(\tilde{\boldsymbol{a}}^s, \boldsymbol{p}_{\tilde{a}}, \boldsymbol{\eta}_{\tilde{a}}^s)}{\partial \mathsf{x}_{\mathrm{m}}} \bigg|_{\mathsf{x}=\hat{\boldsymbol{x}}} &= \left(\boldsymbol{I}_{3\times3} + [\hat{\boldsymbol{\epsilon}}_{\tilde{a}} \times] \right) \left(\boldsymbol{I}_{3\times3} + \boldsymbol{\Gamma}(\hat{\boldsymbol{\gamma}}_{\tilde{a}}) \right) \left(-[\tilde{\boldsymbol{a}}^s \diagdown] \frac{\partial \hat{\boldsymbol{s}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \frac{\partial \hat{\boldsymbol{b}}_{\tilde{a}}^s}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} \right) \\ &\quad - \left(\boldsymbol{I}_{3\times3} + [\hat{\boldsymbol{\epsilon}}_{\tilde{a}} \times] \right) [\blacktriangle \#] \frac{\partial \hat{\boldsymbol{\gamma}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - [\blacklozenge \times] \frac{\partial \hat{\boldsymbol{\epsilon}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} , \end{aligned} \tag{3.79}$$

where $[\blacktriangle \#]$ is the nonorthogonal form, defined in Eq. (4.39) on page 64, of the vector $\blacktriangle \in \mathbb{R}^3$ defined as

$$\blacktriangle \equiv \left(\boldsymbol{I}_{3\times3} - [\hat{\boldsymbol{s}}_{\tilde{a}} \diagdown] \right) \tilde{\boldsymbol{a}}^s - \hat{\boldsymbol{b}}_{\tilde{a}}^s , \tag{3.80}$$

and $[\blacklozenge\times]$ is the skew symmetric form, defined in Eq. (3.24) on page 29, of the vector $\blacklozenge \in \mathbb{R}^3$ defined as

$$\blacklozenge \equiv \left(\boldsymbol{I}_{3\times 3} + \boldsymbol{\Gamma}(\hat{\boldsymbol{\gamma}}_{\tilde{a}})\right)\left\{\left(\boldsymbol{I}_{3\times 3} - [\hat{\boldsymbol{s}}_{\tilde{a}}\diagdown]\right)\tilde{\boldsymbol{a}}^s - \hat{\boldsymbol{b}}_{\tilde{a}}^s\right\} . \tag{3.81}$$

Furthermore, it can be shown that the elements of the Jacobian $\boldsymbol{\mathsf{F}}_\times$ associated with the attitude quaternion kinematics, defined in Eq. (3.50), are given by

$$\left.\frac{\partial \dot{\boldsymbol{\theta}}_i^b}{\partial \boldsymbol{\mathsf{x}}_m}\right|_{\boldsymbol{\mathsf{x}}=\hat{\boldsymbol{x}}} = \left.\frac{\partial \boldsymbol{\omega}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s)}{\partial \boldsymbol{\mathsf{x}}_m}\right|_{\boldsymbol{\mathsf{x}}=\hat{\boldsymbol{x}}} - [\hat{\boldsymbol{\omega}}_{ib}^b\times]\frac{\partial \hat{\boldsymbol{\theta}}_i^b}{\partial \hat{\boldsymbol{x}}_m} , \tag{3.82}$$

where the above expression is commonly referred to as the linearized Bortz equation, $\hat{\boldsymbol{\omega}}_{ib}^b$ is the compensated gyro measurement defined in Eq. (3.63), and the partial derivative of $\boldsymbol{\omega}_{ib}^b$ with respect to $\boldsymbol{\mathsf{x}}_m$ is given by

$$\left.\frac{\partial \boldsymbol{\omega}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s)}{\partial \boldsymbol{\mathsf{x}}_m}\right|_{\boldsymbol{\mathsf{x}}=\hat{\boldsymbol{x}}} = \left(\boldsymbol{I}_{3\times 3} + [\hat{\boldsymbol{\epsilon}}_{\tilde{\omega}}\times]\right)\left(\boldsymbol{I}_{3\times 3} + \boldsymbol{\Gamma}(\hat{\boldsymbol{\gamma}}_{\tilde{\omega}})\right)\left(-[\tilde{\boldsymbol{\omega}}^s\diagdown]\frac{\partial \hat{\boldsymbol{s}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_m} - \frac{\partial \hat{\boldsymbol{b}}_{\tilde{\omega}}^s}{\partial \hat{\boldsymbol{x}}_m}\right)$$
$$- \left(\boldsymbol{I}_{3\times 3} + [\hat{\boldsymbol{\epsilon}}_{\tilde{\omega}}\times]\right)[\blacktriangledown\#]\frac{\partial \hat{\boldsymbol{\gamma}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_m} - [\bigstar\times]\frac{\partial \hat{\boldsymbol{\epsilon}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_m} , \quad (3.83)$$

where $[\blacktriangledown\#]$ is the nonorthogonal form, defined in Eq. (4.39) on page 64, of the vector $\blacktriangledown \in \mathbb{R}^3$ defined as

$$\blacktriangledown \equiv \left(\boldsymbol{I}_{3\times 3} - [\hat{\boldsymbol{s}}_{\tilde{\omega}}\diagdown]\right)\tilde{\boldsymbol{\omega}}^s - \hat{\boldsymbol{b}}_{\tilde{\omega}}^s , \tag{3.84}$$

and $[\bigstar\times]$ is the skew symmetric form, defined in Eq. (3.24) on page 29, of the vector $\bigstar \in \mathbb{R}^3$ defined as

$$\bigstar \equiv \left(\boldsymbol{I}_{3\times 3} + \boldsymbol{\Gamma}(\hat{\boldsymbol{\gamma}}_{\tilde{\omega}})\right)\left\{\left(\boldsymbol{I}_{3\times 3} - [\hat{\boldsymbol{s}}_{\tilde{\omega}}\diagdown]\right)\tilde{\boldsymbol{\omega}}^s - \hat{\boldsymbol{b}}_{\tilde{\omega}}^s\right\} . \tag{3.85}$$

Lastly, the elements of the Jacobian $\boldsymbol{\mathsf{F}}_\times$ associated with the jth error parameter dynamics, defined in Eq. (3.31), are given by

$$\left.\frac{\partial \dot{p}_j}{\partial \boldsymbol{\mathsf{x}}_m}\right|_{\boldsymbol{\mathsf{x}}=\hat{\boldsymbol{x}}} = -\frac{1}{\tau_{\hat{p}_j}}\frac{\partial \hat{p}_j}{\partial \hat{\boldsymbol{x}}_m} . \tag{3.86}$$

From linear system theory [7], it follows that the solution to the linear differential equation specified in Eq. (3.69) is given by

$$\delta \hat{\boldsymbol{e}}(t) = \hat{\boldsymbol{\Phi}}(t, t_0)\, \delta \hat{\boldsymbol{e}}(t_0) - \int_{t_0}^t \hat{\boldsymbol{\Phi}}(t, \tau)\, \boldsymbol{\mathsf{B}}\, \boldsymbol{\eta}(\tau)\, d\tau + \int_{t_0}^t \hat{\boldsymbol{\Phi}}(t, \tau)\, \boldsymbol{\mathsf{G}}\, \boldsymbol{\mathsf{w}}(\tau)\, d\tau , \tag{3.87}$$

where the filter navigation state error transition matrix $\hat{\boldsymbol{\Phi}}(t, t_0) \in \mathbb{R}^{m \times m}$ is the unique solution to the matrix differential equation

$$\dot{\hat{\boldsymbol{\Phi}}}(t, t_0) = \mathsf{F}_\mathsf{x}\big(\hat{\boldsymbol{x}}(t), t\big)\, \hat{\boldsymbol{\Phi}}(t, t_0) \;, \tag{3.88}$$

with the initial condition $\hat{\boldsymbol{\Phi}}(t_0, t_0) = \boldsymbol{I}_{m \times m}$. Then from the definition given in Eq. (3.68), it can be shown that the discrete form of the Ricatti equation or filter navigation state error covariance propagation equation is given by

$$\hat{\boldsymbol{\mathcal{P}}}(t) = \hat{\boldsymbol{\Phi}}(t, t_0)\, \hat{\boldsymbol{\mathcal{P}}}(t_0)\, \big[\hat{\boldsymbol{\Phi}}(t, t_0)\big]^\mathsf{T} + \hat{\boldsymbol{Q}}_\eta(t) + \hat{\boldsymbol{Q}}_\mathsf{w}(t) \;, \tag{3.89}$$

where $\hat{\boldsymbol{Q}}_\eta(t)$ and $\hat{\boldsymbol{Q}}_\mathsf{w}(t)$ are the integrals defined as

$$\hat{\boldsymbol{Q}}_\eta(t) = \int_{t_0}^{t} \hat{\boldsymbol{\Phi}}(t, \tau)\, \mathsf{B}\, \hat{\boldsymbol{S}}_\eta(\tau)\, \big[\mathsf{B}\big]^\mathsf{T} \big[\hat{\boldsymbol{\Phi}}(t, \tau)\big]^\mathsf{T} d\tau \;, \tag{3.90}$$

$$\hat{\boldsymbol{Q}}_\mathsf{w}(t) = \int_{t_0}^{t} \hat{\boldsymbol{\Phi}}(t, \tau)\, \mathsf{G}\, \hat{\boldsymbol{S}}_\mathsf{w}(\tau)\, \big[\mathsf{G}\big]^\mathsf{T} \big[\hat{\boldsymbol{\Phi}}(t, \tau)\big]^\mathsf{T} d\tau \;, \tag{3.91}$$

and $\hat{\boldsymbol{S}}_\eta$ and $\hat{\boldsymbol{S}}_\mathsf{w}$ respectively represent the *strength* of the continuous inertial measurement noise $\boldsymbol{\eta}(t)$ and the Gaussian white noise $\mathsf{w}(t)$ associated with the error parameter states $\hat{\boldsymbol{p}}(t)$. Rather than numerically integrating $\hat{\boldsymbol{\Phi}}(t, t_0)$, $\hat{\boldsymbol{Q}}_\eta(t)$, and $\hat{\boldsymbol{Q}}_\mathsf{w}(t)$, it is assumed that the simulation time step $dt = t - t_0$ is 'small' enough such that the following approximations hold

$$\hat{\boldsymbol{\Phi}}(t, t_0) \cong \boldsymbol{I}_{m \times m} + \mathsf{F}_\mathsf{x}\big(\hat{\boldsymbol{x}}(t), t\big)\, dt + \frac{\big[\mathsf{F}_\mathsf{x}\big(\hat{\boldsymbol{x}}(t), t\big)\, dt\big]^2}{2!} + \cdots + \frac{\big[\mathsf{F}_\mathsf{x}\big(\hat{\boldsymbol{x}}(t), t\big)\, dt\big]^n}{n!} \;, \tag{3.92}$$

$$\hat{\boldsymbol{Q}}_\eta(t) \cong \hat{\boldsymbol{\Phi}}(t, t)\, \mathsf{B}\, \hat{\boldsymbol{S}}_\eta(t)\, \big[\mathsf{B}\big]^\mathsf{T} \big[\hat{\boldsymbol{\Phi}}(t, t)\big]^\mathsf{T} dt = \mathsf{B}\, \hat{\boldsymbol{S}}_\eta(t)\, \big[\mathsf{B}\big]^\mathsf{T} dt \;, \tag{3.93}$$

$$\hat{\boldsymbol{Q}}_\mathsf{w}(t) \cong \hat{\boldsymbol{\Phi}}(t, t)\, \mathsf{G}\, \hat{\boldsymbol{S}}_\mathsf{w}(t)\, \big[\mathsf{G}\big]^\mathsf{T} \big[\hat{\boldsymbol{\Phi}}(t, t)\big]^\mathsf{T} dt = \mathsf{G}\, \hat{\boldsymbol{S}}_\mathsf{w}(t)\, \big[\mathsf{G}\big]^\mathsf{T} dt \;. \tag{3.94}$$

Hence, for this research, these approximations are used in conjunction with the discrete Ricatti equation, defined in Eq. (3.89), to propagate the filter navigation state error covariance $\hat{\boldsymbol{\mathcal{P}}}$.

3.2.2 Ascent Guidance and Steering

In general, guidance for powered ascent consists of both open-loop and closed-loop algorithms in order to achieve mission requirements. The open-loop guidance is employed during the

early phases of powered ascent[3] (e.g., vertical rise, pitch over, gravity turn) and consists of prede-termined body attitude and attitude rate commands to steer the vehicle along a designated ascent profile designed to satisfy in-flight constraints (e.g., dynamic pressure, angle of attack, sideslip angle). The closed-loop guidance operates during the last phase of powered ascent[4] and utilizes the current navigation state solution along with a model of the system dynamics to compute the fuel-optimal thrust pointing commands needed to satisfy terminal or engine cut-off constraints (e.g., altitude, velocity, flight path angle, orbital plane).

Open-Loop Guidance

For the vertical rise segment, the launch vehicle attitude is held constant with respect to the rotating planet. Thus, the commanded attitude rate or angular velocity vector $\hat{\boldsymbol{\omega}}_{ib,\text{com}}^{i}$ is simply equal to the angular velocity of the planet $\boldsymbol{\omega}_{ip}^{i}$ with respect to the inertial frame,

$$\hat{\boldsymbol{\omega}}_{ib,\text{com}}^{i} = \boldsymbol{\omega}_{ip}^{i} \ . \tag{3.95}$$

The commanded inertial-to-body attitude quaternion $\hat{\boldsymbol{q}}_{i,\text{com}}^{\hat{b}}$ can be expressed in terms of transformation matrices as

$$\hat{\boldsymbol{T}}(\hat{\boldsymbol{q}}_{i,\text{com}}^{\hat{b}}) = \bar{\boldsymbol{T}}_{g}^{b}(t_0)\, \boldsymbol{T}_{p}^{g}\, \boldsymbol{T}_{i}^{p} \ , \tag{3.96}$$

where $\bar{\boldsymbol{T}}_{g}^{b}(t_0)$ is the nominal planetographic-to-body transformation matrix at time t_0, $\boldsymbol{T}_{p}^{g}$ is the planet-fixed-to-planetographic transformation matrix, and $\boldsymbol{T}_{i}^{p}$ is the inertial-to-planet-fixed transformation matrix. During vertical rise, the transformation matrices $\bar{\boldsymbol{T}}_{g}^{b}$ and $\boldsymbol{T}_{p}^{g}$ are time invariant, where as $\boldsymbol{T}_{i}^{p}$ is time varying due to the rotation of the planet. See Appendix A for definitions of these frames and coordinate transformation matrices.

For the pitch-over maneuver segment, a quaternion-based eigenaxis slew command propagator was developed and used to eliminate the large unrealistic torque requirements observed with incremental step-input commands, and to improve the angular response of the vehicle by gradually reorienting the vehicle over a time period that meets the capabilities of the actuators. The command propagator operates under the premise that any attitude transformation can be achieved by a single axis rotation (SAR), i.e., a single rotation about a special vector called an eigenvector

---

[3]In the literature, this is often referred to as the *first stage*, however, not necessarily in reference to the stages or segments of a launch vehicle.
[4]This is commonly referred to in the literature as the *second stage*.

or eigenaxis $\hat{\mathbf{e}}$. Given the maximum angular acceleration $\alpha_{\max}$ and velocity $\omega_{\max}$ capabilities of the actuators, the command propagator generates the necessary angular position $\hat{\theta}$ and velocity $\hat{\omega}$ values that produce what is commonly known as a *trapezoidal velocity profile*. In other words, the vehicle accelerates until it reaches the maximum allowable angular velocity. Once the vehicle reaches this maximum rate, it continues to rotate at this angular speed until it is time to begin deceleration. At this time the vehicle decelerates until the desired attitude and angular velocity are achieved.

The commanded angular velocity vector $\hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^{i}$ can be written in terms of the commanded angular velocity magnitude $\hat{\omega}$ and the eigenaxis $\hat{\mathbf{e}}^{i}$ expressed in inertial coordinates,

$$\hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^{i} = \hat{\omega}\,\hat{\mathbf{e}}^{i} \;. \tag{3.97}$$

The commanded inertial-to-body attitude quaternion $\hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}}$ can be written in terms of the commanded angular position $\hat{\theta}$, the eigenaxis $\hat{\mathbf{e}}^{b}$ expressed in body coordinates, and the current nominal inertial-to-body quaternion $\bar{\boldsymbol{q}}_{i}^{b}$,

$$\hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}} = \hat{\boldsymbol{q}}_{b}^{\hat{b}} \otimes \bar{\boldsymbol{q}}_{i}^{b} \;, \tag{3.98}$$

where

$$\hat{\boldsymbol{q}}_{b}^{\hat{b}} = \begin{bmatrix} -\hat{\mathbf{e}}^{b}\sin(\hat{\theta}/2) \\ \cos(\hat{\theta}/2) \end{bmatrix} \;. \tag{3.99}$$

Note that the negative sign in the previous expression is not a typographical error, but rather the result of the quaternion convention chosen for this research.

Closed-Loop Guidance

The literature is replete with closed-loop guidance algorithms that utilize various methods or approaches to compute the fuel optimal pointing commands [31–33, 35, 36, 69] necessary to steer the launch vehicle into the desired orbit. Very few of these algorithms, however, are ever tested or flown on an actual launch vehicle. The closed-loop vacuum guidance algorithm currently flown on the Space Shuttle [70], proposed for future lunar ascent missions [48], and hence selected for this research is known as *powered explicit guidance* (PEG).

PEG, originally called Linear Tangent Guidance (LTG), was developed by Roland F. Jaggers, circa 1970, while working for Boeing in support of the new Space Shuttle program. The innovative vector-based LTG theory was a product of Jagger's intimate knowledge of the Saturn Iterative Guidance Mode (IGM) and extensive experience developing optimal trajectories at NASA Marshall Space Flight Center. After LTG was selected for Space Shuttle ascent to orbit, around late 1972, it was refined and generalized by a team comprised of NASA Johnson Space Center engineers and supporting contractor personnel to handle all Shuttle exoatmospheric maneuvers and requirements, including nominal ascent, ascent aborts, and deorbit. In June of 1974 LTG was renamed PEG. PEG was later chosen for Orion orbit insertion, deorbit, and rendezvous burn guidance, and saw use in trade studies for Altair lunar landing guidance [71]. Only a brief discussion of PEG is given here. A general overview of the Space Shuttle implementation of PEG can be found in McHenry et al. [70], whereas the original theoretical formulation of PEG is given in Jaggers [72].

PEG is the explicit solution to the two point boundary-value problem of exoatmospheric powered flight guidance and trajectory optimization for launch vehicles [72]. It is "explicit" in the sense that the formulas for the steering commands are derived as direct solutions to the equations of motion, and expressed in terms of and valid for any values of the current (initial) and desired (final) boundary conditions [73]. The current navigation state solution for the inertial position and velocity vectors provide the initial boundary conditions and anywhere from one to six terminal or cut-off constraints provide the final boundary conditions.[5] For ascent, five cut-off constraints are necessary [72]; they are (planetocentric) altitude, inertial velocity magnitude, (inertial) flight path angle, inclination, and right ascension of the descending node[6] [70], all of which are mission dependent and determined *a priori*.

PEG is formulated under four important assumptions: (1) the aerodynamic forces acting on the vehicle are negligible, (2) the engine exhaust velocity is constant, (3) either the thrust (mass flow rate) or acceleration is constant,[7] and (4) the target conditions are independent of time and functions only of the estimated (navigation) inertial position and velocity states, $\hat{\boldsymbol{r}}^i$ and $\hat{\boldsymbol{v}}^i$, respectively [72]. Moreover, PEG uses a vector form of the *linear tangent steering law* to generate real time values of the unit inertial thrust vector, $\mathbf{u}_f^i$, and the rate of change of the said

---

[5]PEG is flexible in the sense that it is capable of handling many different types of maneuvers along with a wide variety of cut-off constraints. For example, in order to meet six terminal conditions, engine throttling is required.[70]

[6]In this research, the ascending node, which differs from the descending node by a factor of π or 180 deg, will be used instead.

[7]Constant acceleration requires engine throttling.

In addition to computing the steering parameters $\mathbf{u}^i_\lambda$, $\dot{\boldsymbol{\lambda}}$, and t_λ, PEG also calculates the remaining burn time t_{go}, i.e., the time-to-go until the terminal targets are achieved and MECO should occur. As the remaining burn time approaches zero, however, small perturbations or variations in the flight path result in large changes in the guidance commands. In other words, PEG becomes rapidly unstable as t_{go} goes to zero [72]. To avoid this, when the time-to-go reaches a certain predefined value, PEG terminates and guidance modes to 'fine count', where the last commanded unit inertial thrust vector $\mathbf{u}^i_{\mathrm{f}}$ is held constant, t_{go} is decremented until it reaches zero and the engine shutdown command is issued.

The guidance/control steering interface, hereafter referred to simply as *steering*, operates in conjunction with PEG and refers to the process of converting the steering parameters $\boldsymbol{\lambda}_\nu$, $\dot{\boldsymbol{\lambda}}$, and t_λ to body attitude and attitude rate commands, which are used by the attitude control system to generate the actuator commands. This process can be described as follows: Given the steering parameters $\mathbf{u}^i_\lambda$, $\dot{\boldsymbol{\lambda}}$, and t_λ, the commanded unit inertial thrust vector $\mathbf{u}^i_{\mathrm{f}}$ is computed using Eq. (3.103). Next, the commanded inertial-to-desired-body attitude quaternion $\hat{\boldsymbol{q}}^{\hat{b}}_{i,\mathrm{com}}$ is computed by first forming the desired-body-to-inertial transformation matrix $\boldsymbol{T}^i_{\hat{b}}$ then converting it to a quaternion.[10]

The desired-body-to-inertial transformation matrix $\boldsymbol{T}^i_{\hat{b}}$ can be formed by defining three orthogonal unit vectors (as illustrated in Figure 3.8b) that represent the body x, y, z axes, each expressed in inertial coordinates, as given by

$$\boldsymbol{T}^i_{\hat{b}} = \begin{bmatrix} \mathbf{u}^i_x & \mathbf{u}^i_y & \mathbf{u}^i_z \end{bmatrix} . \tag{3.104}$$

The unit inertial thrust vector $\mathbf{u}^i_{\mathrm{f}}$ is defined to point in the same direction as the body x axis,

$$\mathbf{u}^i_x = \mathbf{u}^i_{\mathrm{f}} . \tag{3.105}$$

By convention, the body y axis is defined to point in the direction normal to the plane formed by the unit inertial thrust vector $\mathbf{u}^i_{\mathrm{f}}$ and the (estimated) inertial position vector $\hat{\boldsymbol{r}}^i$,

$$\mathbf{u}^i_y = \frac{\mathbf{u}^i_{\mathrm{f}} \times \hat{\boldsymbol{r}}^i}{\left\| \mathbf{u}^i_{\mathrm{f}} \times \hat{\boldsymbol{r}}^i \right\|} . \tag{3.106}$$

---

[10]The process or method of converting a DCM to a quaternion is not discussed here, but can be found in texts by Sidi [75, pp. 318-327], Markley [76], or Riegsecker [77].
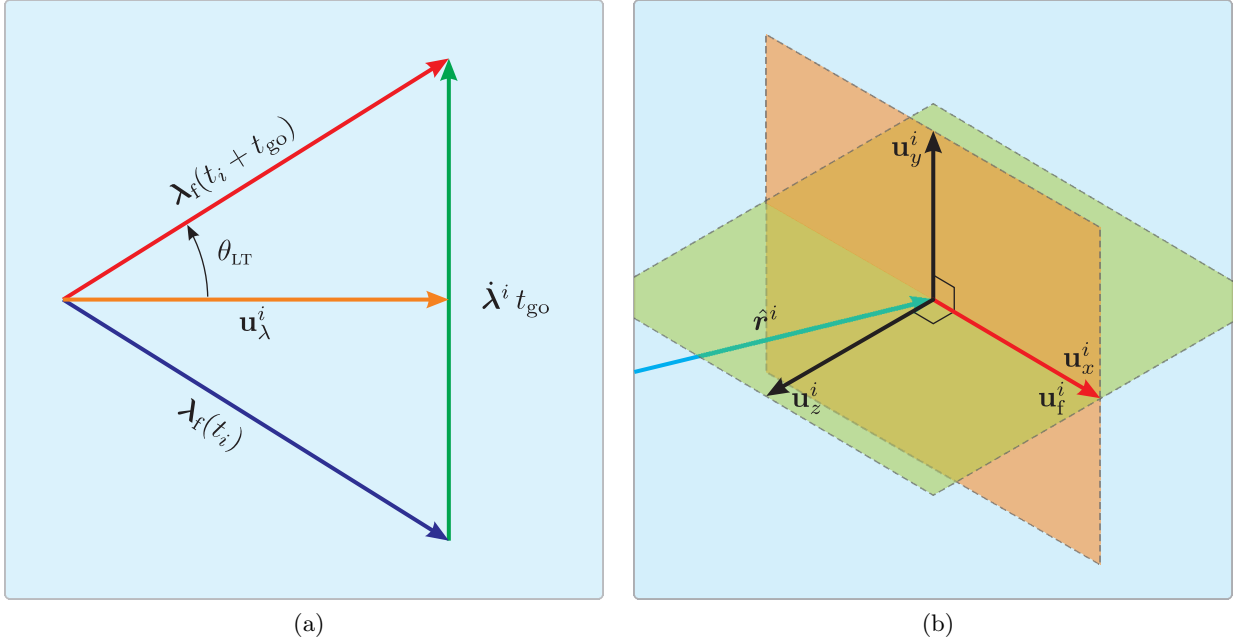
Figure 3.8: (a) Vector diagram of the linear tangent steering law. (b) The orthogonal unit vectors of the body r, p, y axes, expressed in inertial coordinates, that are used to form the inertial-to-body-desired DCM.

The body z axis completes the right-handed triad,

$$\mathbf{u}_z^i = \mathbf{u}_x^i \times \mathbf{u}_y^i \ . \tag{3.107}$$

Finally, the commanded angular velocity vector, expressed in inertial coordinates, is computed using the following expression [70]

$$\hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^i = \mathbf{u}_\lambda^i \times \dot{\boldsymbol{\lambda}}^i \ . \tag{3.108}$$

3.2.3 Attitude Control

Attitude control, as it pertains to launch vehicles, is the process of computing the actuator commands necessary to perform rotational maneuvers in the presence of random disturbances and attitude uncertainties, while satisfying constraints associated with the mission and vehicle. Although there exists a wide variety of control methods or techniques available in the literature, ranging from the very complex to the ultra simple, classical control techniques for space vehicles, as outlined in the texts of Wie [78] and Sidi [75], are sufficient for the needs of this study. This is primarily due to the fact that the focus of this research is on guidance and navigation system

analysis and design and not on control system design and analysis. Consequently, only the closed-loop proportional-derivative (PD) feedback controller is considered for this research.

For this research, it is assumed that the main engine of the launch vehicle is not throttleable nor is it gimbaled. Consequently, the translational motion is controlled by adjusting the direction of the inertial thrust vector over time, which is accomplished through control of the rotational motion or attitude of the launch vehicle. A quaternion-based control law is chosen for the attitude controller because the attitude of the launch vehicle and associated kinematics are already expressed in terms of the quaternion, and the quaternion control law is better suited for large attitude maneuvers and more efficient than the simpler Euler angles control law in minimizing the length of the total angular path followed by the vehicle during a particular maneuver [75, §7.2.5].

Quaternion Attitude Control Law

The attitude controller uses the current estimated attitude and angular velocity from navigation and the desired attitude and angular velocity commands from guidance to compute the attitude and angular velocity error vectors. The attitude error quaternion that represents the rotation from the current attitude to the desired attitude is defined as the product of the commanded attitude quaternion $\hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}}$ and the current estimate of the body-to-inertial attitude quaternion $\hat{\boldsymbol{q}}_b^i$,

$$\delta\hat{\boldsymbol{q}}_b^{\hat{b}} = \hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}} \otimes [\hat{\boldsymbol{q}}_i^b]^{-1} \ . \tag{3.109}$$

The attitude error quaternion can also be expressed in terms of the unit vector along the single axis of rotation $\mathbf{u}^b$ and the rotation angle θ about the axis of rotation,

$$\delta\hat{\boldsymbol{q}}_b^{\hat{b}} = \begin{bmatrix} -\mathbf{u}^b \sin(\theta/2) \\ \cos(\theta/2) \end{bmatrix} \ . \tag{3.110}$$

From this latter expression, the attitude error vector $\hat{\boldsymbol{\theta}}_{\mathrm{err}}^b$, also referred to as the Euler rotation vector [77], is given by

$$\hat{\boldsymbol{\theta}}_{\mathrm{err}}^b = \theta\,\mathbf{u}^b \ . \tag{3.111}$$

The angular velocity error vector is defined as the difference between the desired angular velocity vector $\hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^i$ and the compensated gyro measurement $\hat{\boldsymbol{\omega}}_{ib}^b$,

$$\hat{\boldsymbol{\omega}}_{\text{err}}^b = \hat{\boldsymbol{T}}(\hat{\boldsymbol{q}}_i^b)\,\hat{\boldsymbol{\omega}}_{ib,\text{com}}^i - \hat{\boldsymbol{\omega}}_{ib}^b \ . \tag{3.112}$$

Note that this expression assumes that a gyro measurement is available such that the angular velocity state need not be estimated. As a result, the compensated gyro measurement is used in place of the estimated angular velocity state.

The required moment or torque to achieve the guidance commands, expressed in body coordinates, is then defined as

$$\hat{\boldsymbol{m}}_{\text{err}}^b = \hat{\boldsymbol{K}}_\theta^b\,\hat{\boldsymbol{\theta}}_{\text{err}}^b + \hat{\boldsymbol{K}}_\omega^b\,\hat{\boldsymbol{\omega}}_{\text{err}}^b \ , \tag{3.113}$$

where $\hat{\boldsymbol{K}}_\theta^b$ and $\hat{\boldsymbol{K}}_\omega^b$ are the proportional and derivative control gain matrices, respectively. It can be shown that for a given natural frequency ω_{acs}, desired damping ratio ζ_{acs}, and mass MOI tensor $\boldsymbol{J}_b^b$, the proportional and derivative control gains are given by [79, Section 5.2.3]

$$\hat{\boldsymbol{K}}_\theta^b = \omega_{\text{acs}}^2\,\boldsymbol{J}_b^b \ , \tag{3.114}$$

$$\hat{\boldsymbol{K}}_\omega^b = 2\,\zeta_{\text{acs}}\,\omega_{\text{acs}}\,\boldsymbol{J}_b^b \ . \tag{3.115}$$

For this research, it is assumed that the time-varying mass MOI tensor $\boldsymbol{J}_b^b$ is computed prior to launch and stored onboard. Consequently, the time-varying control gains $\hat{\boldsymbol{K}}_\theta^b$ and $\hat{\boldsymbol{K}}_\omega^b$ are independent of the estimated states of the system.

ACS Error Compensation

The actual moment applied by the ACS is generally corrupted by actuator errors modeled here as scale factor uncertainties, orthogonality misalignments, random biases, and noise. If these scale factor, misalignment, and bias errors are observable and estimated in the navigation algorithm as error parameter states, then the effects of these errors can be mitigated to some extent through error compensation. This is accomplished by modifying the commanded moment issued to the ACS $\hat{\boldsymbol{m}}_{\text{err}}^b$, defined in Equation (3.113), in the following manner,

$$\hat{\boldsymbol{m}}_{\text{com}}^b(\hat{\boldsymbol{m}}_{\text{err}}^b, \hat{\boldsymbol{p}}_{\text{acs}}^b) = \left(\boldsymbol{I}_{3\times3} - [\hat{\boldsymbol{s}}_{\text{acs}}^b \diagdown]\right)\Big\{\left(\boldsymbol{I}_{3\times3} + [\hat{\boldsymbol{\epsilon}}_{\text{acs}}^b\times]\right)\hat{\boldsymbol{m}}_{\text{err}}^b(\hat{\boldsymbol{\theta}}_{\text{err}}^b, \hat{\boldsymbol{\omega}}_{\text{err}}^b) - \hat{\boldsymbol{b}}_{\text{acs}}^b\Big\} \ , \tag{3.116}$$

where $\hat{\boldsymbol{s}}_{\text{acs}}^b$, $\hat{\boldsymbol{\epsilon}}_{\text{acs}}^b$, and $\hat{\boldsymbol{b}}_{\text{acs}}^b$ are the estimated ACS scale factor, orthogonality, and bias vectors,

respectively. These estimated ACS error parameters states $\hat{\boldsymbol{p}}^b_{\mathrm{acs}} = (\hat{\boldsymbol{s}}^b_{\mathrm{acs}}, \hat{\boldsymbol{\epsilon}}^b_{\mathrm{acs}}, \hat{\boldsymbol{b}}^b_{\mathrm{acs}})$ are modeled as first-order Gauss-Markov processes, as defined by Eq. (3.59).

Chapter 4

Linearized Models and Algorithms

"The moment a person forms a theory, his imagination

sees in every object only the traits which favor that theory."

– Thomas Jefferson

In the preceding chapter, the nonlinear truth models and GN&C algorithm models associated with the Monte Carlo simulation were defined. The purpose of this chapter is to address the next step in the development process, which is to define the nominal reference trajectory (NRT) and linearize the Monte Carlo nonlinear models about the NRT. Hence the chapter is organized as follows: First, the NRT is defined, followed by a brief discussion of the general method for linearizing a nonlinear system of equations about the NRT. With this foundation in hand, the nonlinear truth models and GN&C algorithm models are expressed in their general forms, linearized about the NRT, and the associated partial derivatives are presented.

Before proceeding, it is important to note that the presentation of the linearized models and algorithms herein closely follows that of the previous chapter. However, out of necessity, there are a number of equations referenced that have not yet appeared in the material. Although referencing in this manner is somewhat unorthodox, this approach facilitates easy finding of those expressions that appear in the equation at hand but who are not defined until later in the chapter.

4.1   Nominal Reference Trajectory

The NRT is the path that the launch vehicle would follow in the absence of sensor measurement errors, actuator execution variabilities, and random environment disturbances. It is generated with a single, error-less, noise-less run of the Monte Carlo simulation, and consists of the nominal states $\bar{\boldsymbol{x}}$, nominal actuator commands $\bar{\boldsymbol{u}}$, nominal guidance commands $\bar{\boldsymbol{\xi}}$, and nominal sensor measurements $\bar{\boldsymbol{y}}$ and $\bar{\boldsymbol{z}}_k$, where the bar ¯ above the variable denotes the nominal value.

4.2 Linearization of Nonlinear Systems of Equations

A nonlinear system of equations can be linearized using a Taylor-series expansion about some operating point, such as the NRT introduced above. The linearization process is best illustrated with the following example. The true state dynamics defined in Eqs. (3.1)-(3.7) can be written in the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big) \ , \tag{4.1}$$

where $\boldsymbol{f} : \mathbb{R}^{n'} \times \mathbb{R}^a \times \mathbb{R}^s \to \mathbb{R}^{n'}$ is a nonlinear vector function of the true states $\boldsymbol{x}(t) \in \mathbb{R}^{n'}$, actuator commands $\hat{\boldsymbol{u}}(t) \in \mathbb{R}^a$, and state process noise $\boldsymbol{w}(t) \in \mathbb{R}^s$. Let

$$\delta\boldsymbol{x}(t) \equiv \boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t) \quad \text{and} \quad \delta\hat{\boldsymbol{u}}(t) \equiv \hat{\boldsymbol{u}}(t) - \bar{\boldsymbol{u}}(t) \ , \tag{4.2}$$

where $\delta\boldsymbol{x}(t)$ and $\delta\hat{\boldsymbol{u}}(t)$ represent deviations from the nominal or dispersions; more specifically, $\delta\boldsymbol{x}(t)$ represents the true state dispersions. It is important to note that an expression for the state process noise has been omitted because the nominal value of the state process noise is defined to be zero (i.e., $\bar{\boldsymbol{w}}(t) = \boldsymbol{0}$) such that the corresponding deviation from the nominal $\delta\boldsymbol{w}(t)$ is simply the state process noise $\boldsymbol{w}(t)$,

$$\delta\boldsymbol{w}(t) \equiv \boldsymbol{w}(t) \ . \tag{4.3}$$

Thus, the nominal state dynamics equation is given by

$$\dot{\bar{\boldsymbol{x}}}(t) = \boldsymbol{f}\big(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t), t\big) \ . \tag{4.4}$$

Given the expressions in Eq. (4.2), the nonlinear true state dynamics equation can now be written as

$$\dot{\bar{\boldsymbol{x}}}(t) + \delta\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\bar{\boldsymbol{x}}(t) + \delta\boldsymbol{x}(t), \bar{\boldsymbol{u}}(t) + \delta\hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big) \ . \tag{4.5}$$

Next, $\boldsymbol{f}$ is expanded in a Taylor series about the NRT $\bar{\boldsymbol{x}}(t)$ and $\bar{\boldsymbol{u}}(t)$. Then, under the assumption that the magnitude of the deviations from the nominal $\delta\boldsymbol{x}(t)$ and $\delta\hat{\boldsymbol{u}}(t)$ are small relative to the magnitude of the nominal values, only the first-order terms of the series are retained, resulting in

the following truncated Taylor series expansion of $\boldsymbol{f}$,

$$\boldsymbol{f}\big(\bar{\boldsymbol{x}}(t) + \delta\boldsymbol{x}(t), \bar{\boldsymbol{u}}(t) + \delta\hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big) \cong \boldsymbol{f}\big(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t), t\big) + \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\boldsymbol{x}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} (\boldsymbol{x} - \bar{\boldsymbol{x}})$$

$$+ \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\hat{\boldsymbol{u}}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} (\hat{\boldsymbol{u}} - \bar{\boldsymbol{u}})$$

$$+ \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\boldsymbol{w}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} (\boldsymbol{w} - \bar{\boldsymbol{w}}) \ ,$$

where $\frac{\partial\boldsymbol{f}(\boldsymbol{x}, \hat{\boldsymbol{u}}, \boldsymbol{w}, t)}{\partial\boldsymbol{x}}\big|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}}$ denotes the Jacobian of $\boldsymbol{f}$ with respect to the true states $\boldsymbol{x}$ evaluated at the NRT $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$. As a result, Eq. (4.5) can now be written as

$$\dot{\bar{\boldsymbol{x}}}(t) + \delta\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t), t\big) + \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\boldsymbol{x}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} \delta\boldsymbol{x}(t)$$

$$+ \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\hat{\boldsymbol{u}}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} \delta\hat{\boldsymbol{u}}(t) + \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\boldsymbol{w}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} \boldsymbol{w}(t) \ .$$

Using the expression for the nominal state dynamics defined in Eq. (4.4) to cancel terms on both sides of the previous expression, it follows that the linear time-varying true-state dynamics equation is given by

$$\delta\dot{\boldsymbol{x}}(t) = \boldsymbol{F}_x(t)\,\delta\boldsymbol{x}(t) + \boldsymbol{F}_{\hat{u}}(t)\,\delta\hat{\boldsymbol{u}}(t) + \boldsymbol{F}_w(t)\,\boldsymbol{w}(t) \ , \tag{4.6}$$

where the Jacobians $\boldsymbol{F}_x(t) \in \mathbb{R}^{n\times n}$, $\boldsymbol{F}_{\hat{u}}(t) \in \mathbb{R}^{n\times u}$, and $\boldsymbol{F}_w(t) \in \mathbb{R}^{n\times s}$ are respectively defined as

$$\boldsymbol{F}_x(t) = \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\boldsymbol{x}_{\mathrm{m}}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} \ , \quad \boldsymbol{F}_{\hat{u}}(t) = \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\hat{\boldsymbol{u}}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} \ , \tag{4.7}$$

and

$$\boldsymbol{F}_w(t) = \frac{\partial\boldsymbol{f}\big(\boldsymbol{x}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{w}(t), t\big)}{\partial\boldsymbol{w}(t)}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} \ . \tag{4.8}$$

Recall that the inertial-to-body attitude quaternion $\boldsymbol{q}_i^b \in \mathbb{R}^4$ is the standard representation of the launch vehicle attitude. However, due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the state covariance propagation and update equations as well as the state vector update equations. In this approach, the inertial-to-body attitude quaternion $\boldsymbol{q}_i^b \in \mathbb{R}^4$ in the true state vector $\boldsymbol{x} \in \mathbb{R}^{n'}$, given

by

$$x = \left(r^i, \ v^i, \ q_i^b, \ \omega_{ib}^b, \ m_b, \ p \right) , \tag{4.9}$$

is replaced with the Euler rotation vector $\boldsymbol{\theta}_i^b \in \mathbb{R}^3$, which represents the attitude of the body frame b with respect to the inertial frame i expressed in body coordinates, resulting in a modified true state vector $\boldsymbol{x}_\mathrm{m} \in \mathbb{R}^n$ (where $n = n' - 1$), given by

$$\boldsymbol{x}_\mathrm{m} = \left(\boldsymbol{r}^i, \ \boldsymbol{v}^i, \ \boldsymbol{\theta}_i^b, \ \boldsymbol{\omega}_{ib}^b, \ m_b, \ \boldsymbol{p} \right) . \tag{4.10}$$

Moreover, the quaternion kinematics equation is replaced with the linearized Bortz equation [55]. As such, the linearized true state dynamics equation, given in Eq. (4.6), is based upon the modified true state vector $\boldsymbol{x}_\mathrm{m}$, i.e., $\delta \boldsymbol{x}(t) \in \mathbb{R}^n$, and the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ have been defined accordingly.

4.3 Truth Models

The truth models defined in Chapter 3 are comprised of the launch vehicle dynamics, environment acceleration models, actuator force and torque models, and inertial measurement models. Since the first three are directly related to the true state dynamics, the following three sections define the elements of the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ associated with the launch vehicle dynamics, environment models, and actuator models. In the fourth section, the inertial measurement models are linearized about the NRT.

4.3.1 Launch Vehicle Dynamics

The linearized launch vehicle dynamics include the inertial position $\boldsymbol{r}^i$, inertial velocity $\boldsymbol{v}^i$, body attitude $\boldsymbol{\theta}_i^b$, body angular velocity $\boldsymbol{\omega}_{ib}^b$, and wet mass m_b.

Inertial Position

The elements of the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ associated with the inertial position kinematics, defined in Eq. (3.1) on page 17, are respectively given by

$$\left. \frac{\partial \dot{\boldsymbol{r}}^i}{\partial \boldsymbol{x}_\mathrm{m}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \frac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{x}_\mathrm{m}} , \quad \left. \frac{\partial \dot{\boldsymbol{r}}^i}{\partial \hat{\boldsymbol{u}}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{3 \times a} , \quad \text{and} \quad \left. \frac{\partial \dot{\boldsymbol{r}}^i}{\partial \boldsymbol{w}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{3 \times s} . \tag{4.11}$$

Given the definition of the modified true state vector $\boldsymbol{x}_{\mathrm{m}}$ in Eq. (4.10), it follows that the partial derivative of the inertial velocity $\boldsymbol{v}^i$ with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ can be expanded and written in terms of each of the states contained in the modified true state vector as follows

$$\frac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{x}_{\mathrm{m}}} = \left[\begin{array}{cccccc} \dfrac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{r}^i} & \dfrac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{v}^i} & \dfrac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{\theta}_i^b} & \dfrac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{\omega}_{ib}^b} & \dfrac{\partial \boldsymbol{v}^i}{\partial m_b} & \dfrac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{p}} \end{array} \right] . \tag{4.12}$$

Each of these partial derivatives can then be evaluated, thereby resulting in

$$\frac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{x}_{\mathrm{m}}} = \left[\begin{array}{cccccc} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times p} \end{array} \right] . \tag{4.13}$$

Note that the partial derivative of the inertial velocity $\boldsymbol{v}^i$ with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ is an example of a simple or basic partial derivative, i.e., evaluating the partial derivative yields a matrix of ones and zeros. This is because the inertial velocity $\boldsymbol{v}^i$ itself is a true state and a member of the modified true state vector $\boldsymbol{x}_{\mathrm{m}}$. Rather than expand and write out these partial derivatives explicitly, they are hereafter expressed only in their general form, i.e., $\frac{\partial \boldsymbol{v}^i}{\partial \boldsymbol{x}_{\mathrm{m}}}$. Moreover, to help distinguish them from the other more complex partial derivatives, the evaluation operator (denoted by the vertical bar | on the right hand side of the partial derivative) is subsequently implied and therefore left off.

Inertial Velocity

It can be shown that the elements of the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ associated with the inertial velocity dynamics, defined in Eq. (3.2) on page 17, are respectively given by

$$\left. \frac{\partial \dot{\boldsymbol{v}}^i}{\partial \boldsymbol{x}_{\mathrm{m}}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \left. \frac{\partial \boldsymbol{a}_{\mathrm{g}}^i}{\partial \boldsymbol{x}_{\mathrm{m}}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} + \left[\bar{\boldsymbol{T}}(\bar{\boldsymbol{q}}_i^b) \right]^{\mathsf{T}} \left(-\left[\bar{\boldsymbol{a}}_{\mathrm{ng}}^b \times \right] \frac{\partial \boldsymbol{\theta}_i^b}{\partial \boldsymbol{x}_{\mathrm{m}}} + \left. \frac{\partial \boldsymbol{a}_{\mathrm{ng}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \right) , \tag{4.14}$$

$$\left. \frac{\partial \dot{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{u}}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{3\times a} , \quad \text{and} \quad \left. \frac{\partial \dot{\boldsymbol{v}}^i}{\partial \boldsymbol{w}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \left[\bar{\boldsymbol{T}}(\bar{\boldsymbol{q}}_i^b) \right]^{\mathsf{T}} \left. \frac{\partial \boldsymbol{a}_{\mathrm{ng}}^b}{\partial \boldsymbol{w}} \right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} , \tag{4.15}$$

where it can be shown that the nominal nongravitational acceleration $\bar{\boldsymbol{a}}_{\mathrm{ng}}^b$, expressed in body coordinates, is given by

$$\bar{\boldsymbol{a}}_{\mathrm{ng}}^b = \frac{\bar{T}_{\mathrm{vac}}}{\bar{m}_b} \mathbf{b}_1^b , \tag{4.16}$$

and the partial derivatives of the gravitational and nongravitational acceleration $\boldsymbol{a}_{\mathrm{g}}^i$, $\boldsymbol{a}_{\mathrm{ng}}^b$ with

respect to the modified true states $\boldsymbol{x}_\mathrm{m}$ and state process noise $\boldsymbol{w}$ are specified respectively in Eqs. (4.21), (4.22), and (4.23).

Body Attitude

It can be shown that the elements of the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ associated with the attitude quaternion kinematics, defined in Eq. (3.3) on page 18, are respectively given by

$$\left.\frac{\partial \dot{\boldsymbol{\theta}}_i^b}{\partial \boldsymbol{x}_\mathrm{m}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \frac{\partial \boldsymbol{\omega}_{ib}^b}{\partial \boldsymbol{x}_\mathrm{m}} - \left[\bar{\boldsymbol{\omega}}_{ib}^b \times\right] \frac{\partial \boldsymbol{\theta}_i^b}{\partial \boldsymbol{x}_\mathrm{m}} \;, \quad \left.\frac{\partial \dot{\boldsymbol{\theta}}_i^b}{\partial \hat{\boldsymbol{u}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{3\times a} \;, \quad \text{and} \quad \left.\frac{\partial \dot{\boldsymbol{\theta}}_i^b}{\partial \boldsymbol{w}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{3\times s} \;, \qquad (4.17)$$

where the first expression above is the *linearized Bortz equation.*

Body Angular Velocity

It can be shown that the elements of the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ associated with the body angular velocity dynamics, defined in Eq. (3.4) on page 18, are respectively given by

$$\left.\frac{\partial \dot{\boldsymbol{\omega}}_{ib}^b}{\partial \boldsymbol{x}_\mathrm{m}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \left[\bar{\boldsymbol{J}}_b^b\right]^{-1} \left\{ \left.\frac{\partial \boldsymbol{m}_\mathrm{acs}^b}{\partial \boldsymbol{x}_\mathrm{m}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} + \left(\left[\bar{\boldsymbol{J}}_b^b\,\bar{\boldsymbol{\omega}}_{ib}^b \times\right] - \left[\bar{\boldsymbol{\omega}}_{ib}^b \times\right]\bar{\boldsymbol{J}}_b^b \right)\frac{\partial \boldsymbol{\omega}_{ib}^b}{\partial \boldsymbol{x}_\mathrm{m}} \right\} + \left.\frac{\partial \boldsymbol{\alpha}_\mathrm{d}^b}{\partial \boldsymbol{x}_\mathrm{m}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \;, \qquad (4.18)$$

$$\left.\frac{\partial \dot{\boldsymbol{\omega}}_{ib}^b}{\partial \hat{\boldsymbol{u}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \left[\bar{\boldsymbol{J}}_b^b\right]^{-1} \left.\frac{\partial \boldsymbol{m}_\mathrm{acs}^b}{\partial \hat{\boldsymbol{u}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \;, \quad \text{and} \quad \left.\frac{\partial \dot{\boldsymbol{\omega}}_{ib}^b}{\partial \boldsymbol{w}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \left[\bar{\boldsymbol{J}}_b^b\right]^{-1} \left.\frac{\partial \boldsymbol{m}_\mathrm{acs}^b}{\partial \boldsymbol{w}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \;, \qquad (4.19)$$

where the partial derivatives of the applied ACS torque $\boldsymbol{m}_\mathrm{acs}^b$ with respect to the modified true states $\boldsymbol{x}_\mathrm{m}$, actuator commands $\hat{\boldsymbol{u}}$, and state process noise $\boldsymbol{w}$ are respectively defined in Eqs. (4.29) and (4.30), and the partial derivative of the sinusoidal disturbance angular acceleration $\boldsymbol{\alpha}_\mathrm{d}^b$ with respect to the modified true states $\boldsymbol{x}_\mathrm{m}$ is given in Eq. (4.26).

Wet Mass

The elements of the Jacobians $\boldsymbol{F}_x$, $\boldsymbol{F}_{\hat{u}}$, and $\boldsymbol{F}_w$ associated with the wet mass dynamics, defined in Eq. (3.7) on page 19, are respectively given by

$$\frac{\partial \dot{m}_b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{1\times n} \;, \quad \frac{\partial \dot{m}_b}{\partial \hat{\boldsymbol{u}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{1\times a} \;, \quad \text{and} \quad \frac{\partial \dot{m}_b}{\partial \boldsymbol{w}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \boldsymbol{0}_{1\times s} \;. \tag{4.20}$$

4.3.2 Environment Models

The environment models include the gravitational acceleration $\boldsymbol{a}_{\mathrm{g}}^i$, the nongravitational acceleration $\boldsymbol{a}_{\mathrm{ng}}^b$, and the rotational sinusoidal disturbance $\boldsymbol{\alpha}_{\mathrm{d}}^b$.

Gravitational Acceleration

The partial derivative of the gravitational acceleration $\boldsymbol{a}_{\mathrm{g}}^i$, defined in Eq. (3.8) on page 19, with respect to the true states $\boldsymbol{x}$ is given by

$$\frac{\partial \boldsymbol{a}_{\mathrm{g}}^i}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = -\frac{\mu}{\left\|\bar{\boldsymbol{r}}^i\right\|^3}\left(\boldsymbol{I}_{3\times 3} - 3\,\bar{\mathbf{u}}_r^i\left[\bar{\mathbf{u}}_r^i\right]^\mathsf{T}\right)\frac{\partial \boldsymbol{r}^i}{\partial \boldsymbol{x}_{\mathrm{m}}} \;, \tag{4.21}$$

where $\bar{\mathbf{u}}_r^i$ is the unit vector in the direction of the nominal inertial position $\bar{\boldsymbol{r}}^i$.

Nongravitational Acceleration

The partial derivatives of the nongravitational acceleration $\boldsymbol{a}_{\mathrm{ng}}^b$, defined in Eq. (3.9) on page 20, with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ and state process noise $\boldsymbol{w}$ are respectively given by

$$\frac{\partial \boldsymbol{a}_{\mathrm{ng}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \frac{1}{\bar{m}_b}\frac{\partial \boldsymbol{f}_{\mathrm{thr}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} - \frac{\bar{\boldsymbol{f}}_{\mathrm{thr}}^b}{\bar{m}_b^2}\frac{\partial m_b}{\partial \boldsymbol{x}_{\mathrm{m}}} + \frac{\partial \boldsymbol{a}_{\mathrm{d}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \tag{4.22}$$

and

$$\frac{\partial \boldsymbol{a}_{\mathrm{ng}}^b}{\partial \boldsymbol{w}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \frac{1}{\bar{m}_b}\frac{\partial \boldsymbol{f}_{\mathrm{thr}}^b}{\partial \boldsymbol{w}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \;, \tag{4.23}$$

where it can be shown the nominal thrust force $\bar{\boldsymbol{f}}_{\mathrm{thr}}^b$, expressed in body coordinates, is given by

$$\bar{\boldsymbol{f}}_{\mathrm{thr}}^b = \bar{T}_{\mathrm{vac}}\mathbf{b}_1^b \;, \tag{4.24}$$

the partial derivatives of the thrust force $\boldsymbol{f}_{\mathrm{thr}}^b$ with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ and state process noise $\boldsymbol{w}$ are respectively given in Eqs. (4.27) and (4.28), and the partial derivative of the sinusoidal disturbance acceleration $\boldsymbol{a}_{\mathrm{od}}^b$ with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ is specified in

Eq. (4.25).

Sinusoidal Disturbance

The partial derivative of the sinusoidal disturbance acceleration $\boldsymbol{a}_{\mathrm{d}}^b$, defined in Eq. (3.10) on page 20, with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ is given by

$$\frac{\partial \boldsymbol{a}_{\mathrm{d}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} = \sin(\omega_{\mathrm{d}} t)\frac{\partial \boldsymbol{c}_1}{\partial \boldsymbol{x}_{\mathrm{m}}} + \cos(\omega_{\mathrm{d}} t)\frac{\partial \boldsymbol{c}_2}{\partial \boldsymbol{x}_{\mathrm{m}}} \ , \tag{4.25}$$

and the partial derivative of the sinusoidal disturbance angular acceleration $\boldsymbol{\alpha}_{\mathrm{d}}^b$, defined in Eq. (3.11) on page 20, with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ is given by

$$\frac{\partial \boldsymbol{\alpha}_{\mathrm{d}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} = \sin(\omega_{\mathrm{d}} t)\frac{\partial \boldsymbol{d}_1}{\partial \boldsymbol{x}_{\mathrm{m}}} + \cos(\omega_{\mathrm{d}} t)\frac{\partial \boldsymbol{d}_2}{\partial \boldsymbol{x}_{\mathrm{m}}} \ . \tag{4.26}$$

4.3.3 Actuator Models

The actuator models include the thrust force $\boldsymbol{f}_{\mathrm{thr}}^b$ and ACS torque $\boldsymbol{m}_{\mathrm{acs}}^b$.

Thrust Force

The partial derivatives of the applied thrust force $\boldsymbol{f}_{\mathrm{thr}}^b$, defined in Eq. (3.19) on page 23, with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$ and state process noise $\boldsymbol{w}$ are respectively given by

$$\frac{\partial \boldsymbol{f}_{\mathrm{thr}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} = \left[\bar{T}_{\mathrm{vac}}\mathbf{b}_1^b \diagdown\right]\frac{\partial \boldsymbol{s}_{\mathrm{thr}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \left[\bar{T}_{\mathrm{vac}}\mathbf{b}_1^b \times\right]\frac{\partial \boldsymbol{\epsilon}_{\mathrm{thr}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \frac{\partial \boldsymbol{b}_{\mathrm{thr}}}{\partial \boldsymbol{x}_{\mathrm{m}}} \tag{4.27}$$

and

$$\frac{\partial \boldsymbol{f}_{\mathrm{thr}}^b}{\partial \boldsymbol{w}}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} = \frac{\partial \boldsymbol{w}_{\mathrm{thr}}}{\partial \boldsymbol{w}} \ . \tag{4.28}$$

ACS Torque

The partial derivatives of the applied ACS torque $\boldsymbol{m}_{\mathrm{acs}}^b$, defined in Eq. (3.22) on page 25, with respect to the modified true states $\boldsymbol{x}_{\mathrm{m}}$, actuator commands $\hat{\boldsymbol{u}}$, and state process noise $\boldsymbol{w}$ are respectively given by

$$\frac{\partial \boldsymbol{m}_{\mathrm{acs}}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, \bar{\boldsymbol{w}}} = \left[\bar{\boldsymbol{m}}_{\mathrm{com}}^b \diagdown\right]\frac{\partial \boldsymbol{s}_{\mathrm{acs}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \left[\bar{\boldsymbol{m}}_{\mathrm{com}}^b \times\right]\frac{\partial \boldsymbol{\epsilon}_{\mathrm{acs}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \frac{\partial \boldsymbol{b}_{\mathrm{acs}}}{\partial \boldsymbol{x}_{\mathrm{m}}} \tag{4.29}$$

and

$$\frac{\partial \boldsymbol{m}_{\text{acs}}^{b}}{\partial \hat{\boldsymbol{u}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \frac{\partial \hat{\boldsymbol{m}}_{\text{com}}^{b}}{\partial \hat{\boldsymbol{u}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} \quad, \quad \text{and} \quad \frac{\partial \boldsymbol{m}_{\text{acs}}^{b}}{\partial \boldsymbol{w}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},\bar{\boldsymbol{w}}} = \frac{\partial \boldsymbol{w}_{\text{acs}}}{\partial \boldsymbol{w}} \quad, \tag{4.30}$$

where $\bar{\boldsymbol{m}}_{\text{com}}^{b}$ is the nominal commanded ACS torque.

4.3.4 Inertial Measurements

The continuous inertial measurements $\tilde{\boldsymbol{y}} = (\tilde{\boldsymbol{a}}_{\text{ng}}^{b}, \tilde{\boldsymbol{\omega}}_{ib}^{b})$ defined in Eqs. (3.29) and (3.30) can be written in the form

$$\tilde{\boldsymbol{y}}(t) = \boldsymbol{c}\big(\boldsymbol{x}(t), \boldsymbol{w}(t), \boldsymbol{\eta}(t), t\big) \ , \tag{4.31}$$

where $\boldsymbol{c} : \mathbb{R}^{n'} \times \mathbb{R}^{s} \times \mathbb{R}^{l} \to \mathbb{R}^{c}$ is a nonlinear vector function of the true states $\boldsymbol{x}(t) \in \mathbb{R}^{n'}$, state process noise $\boldsymbol{w}(t) \in \mathbb{R}^{s}$, and continuous measurement noise $\boldsymbol{\eta}(t) \in \mathbb{R}^{l}$. Next, let

$$\delta\tilde{\boldsymbol{y}}(t) \equiv \tilde{\boldsymbol{y}}(t) - \bar{\boldsymbol{y}}(t) \ , \tag{4.32}$$

where $\delta\tilde{\boldsymbol{y}}(t)$ represents the deviation from the nominal. Moreover, expressions for the state process noise and continuous measurement noise have been omitted because the nominal values are defined to be zero (i.e., $\bar{\boldsymbol{w}}(t) = \boldsymbol{0}$ and $\bar{\boldsymbol{\eta}}(t) = \boldsymbol{0}$) such that the corresponding deviations from the nominal $\delta\boldsymbol{w}(t)$ and $\delta\boldsymbol{\eta}(t)$ are respectively the state process noise $\boldsymbol{w}(t)$ and continuous measurement noise $\boldsymbol{\eta}(t)$. As such, it follows that the nominal continuous inertial measurements $\bar{\boldsymbol{y}}(t)$ is given by

$$\bar{\boldsymbol{y}}(t) = \boldsymbol{c}\big(\bar{\boldsymbol{x}}(t), t\big) \ . \tag{4.33}$$

Then, following the linearization process outlined in § 4.2, the linear time-varying inertial measurements equation is given by

$$\delta\tilde{\boldsymbol{y}}(t) = \boldsymbol{C}_{x}(t)\,\delta\boldsymbol{x}(t) + \boldsymbol{C}_{w}(t)\,\boldsymbol{w}(t) + \boldsymbol{C}_{\eta}(t)\,\boldsymbol{\eta}(t) \ , \tag{4.34}$$

where the Jacobians $\boldsymbol{C}_{x}(t) \in \mathbb{R}^{c \times n}$, $\boldsymbol{C}_{w}(t) \in \mathbb{R}^{c \times s}$, and $\boldsymbol{C}_{\eta}(t) \in \mathbb{R}^{c \times l}$ are respectively defined as

$$\boldsymbol{C}_{x}(t) = \frac{\partial \boldsymbol{c}\big(\boldsymbol{x}(t), \boldsymbol{w}(t), \boldsymbol{\eta}(t), t\big)}{\partial \boldsymbol{x}_{\text{m}}(t)}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{w}},\bar{\boldsymbol{\eta}}} \quad, \quad \boldsymbol{C}_{w}(t) = \frac{\partial \boldsymbol{c}\big(\boldsymbol{x}(t), \boldsymbol{w}(t), \boldsymbol{\eta}(t), t\big)}{\partial \boldsymbol{w}(t)}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{w}},\bar{\boldsymbol{\eta}}} \quad, \tag{4.35}$$

and

$$C_\eta(t) = \left. \frac{\partial c\big(x(t), w(t), \eta(t), t\big)}{\partial \eta(t)} \right|_{\bar{x},\bar{w},\bar{\eta}} . \tag{4.36}$$

Recall that due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the state covariance propagation and update equations as well as the state vector update equations. As such, the linearized inertial measurements equation, given in Eq. (4.34), is based upon the modified true state vector x_{m}, i.e., $\delta x(t) \in \mathbb{R}^n$, and the Jacobians C_x, C_w, and C_η have been defined accordingly.

Accelerometer Measurement

It can be shown that the elements of the Jacobians C_x, C_w, and C_η associated with the accelerometer measurement $\tilde{a}_{\mathrm{ng}}^b$, defined in Eq. (3.29) on page 31, are respectively given by

$$\left. \frac{\partial \tilde{a}_{\mathrm{ng}}^b}{\partial x_{\mathrm{m}}} \right|_{\bar{x},\bar{w},\bar{\eta}} = \left. \frac{\partial a_{\mathrm{ng}}^b}{\partial x_{\mathrm{m}}} \right|_{\bar{x},\bar{w},\bar{\eta}} + \left[\bar{a}_{\mathrm{ng}}^b \diagdown\right] \frac{\partial s_{\tilde{a}}}{\partial x_{\mathrm{m}}} + \left[\bar{a}_{\mathrm{ng}}^b \#\right] \frac{\partial \gamma_{\tilde{a}}}{\partial x_{\mathrm{m}}} + \left[\bar{a}_{\mathrm{ng}}^b \times\right] \frac{\partial \epsilon_{\tilde{a}}}{\partial x_{\mathrm{m}}} + \frac{\partial b_{\tilde{a}}}{\partial x_{\mathrm{m}}} \tag{4.37}$$

and

$$\left. \frac{\partial \tilde{a}_{\mathrm{ng}}^b}{\partial w} \right|_{\bar{x},\bar{w},\bar{\eta}} = \left. \frac{\partial a_{\mathrm{ng}}^b}{\partial w} \right|_{\bar{x},\bar{w},\bar{\eta}}, \quad \text{and} \quad \left. \frac{\partial \tilde{a}_{\mathrm{ng}}^b}{\partial \eta} \right|_{\bar{x},\bar{w},\bar{\eta}} = \frac{\partial \eta_{\tilde{a}}}{\partial \eta}, \tag{4.38}$$

where the nominal nongravitational acceleration $\bar{a}_{\mathrm{ng}}^b$ is defined in Eq. (4.16), $\left[a\#\right]$ is the nonorthogonal form of the three-dimensional vector $a = (a_1, a_2, a_3)$ defined as

$$\left[a\#\right] \equiv \begin{bmatrix} a_3 & -a_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_3 & a_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_2 & -a_1 \end{bmatrix}, \tag{4.39}$$

and the partial derivatives of the nongravitational acceleration a_{ng}^b with respect to the modified true states x_{m} and state process noise w are specified respectively in Eqs. (4.22) and (4.23).

Gyro Measurement

It can be shown that the elements of the Jacobians C_x, C_w, and C_η associated with the gyro measurement $\tilde{\omega}_{ib}^b$, defined in Eq. (3.30) on page 32, are respectively given by

$$\left.\frac{\partial \tilde{\boldsymbol{\omega}}_{ib}^b}{\partial \boldsymbol{x}_{\mathrm{m}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{w}},\bar{\boldsymbol{\eta}}} = \frac{\partial \boldsymbol{\omega}_{ib}^b}{\partial \boldsymbol{x}_{\mathrm{m}}} + \left[\bar{\boldsymbol{\omega}}_{ib}^b \diagdown\right]\frac{\partial \boldsymbol{s}_{\tilde{\omega}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \left[\bar{\boldsymbol{\omega}}_{ib}^b \#\right]\frac{\partial \boldsymbol{\gamma}_{\tilde{\omega}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \left[\bar{\boldsymbol{\omega}}_{ib}^b \times\right]\frac{\partial \boldsymbol{\epsilon}_{\tilde{\omega}}}{\partial \boldsymbol{x}_{\mathrm{m}}} + \frac{\partial \boldsymbol{b}_{\tilde{\omega}}}{\partial \boldsymbol{x}_{\mathrm{m}}} \tag{4.40}$$

and

$$\left.\frac{\partial \tilde{\boldsymbol{\omega}}_{ib}^b}{\partial \boldsymbol{w}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{w}},\bar{\boldsymbol{\eta}}} = \mathbf{0}_{3\times s} , \quad \text{and} \quad \left.\frac{\partial \tilde{\boldsymbol{\omega}}_{ib}^b}{\partial \boldsymbol{\eta}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{w}},\bar{\boldsymbol{\eta}}} = \frac{\partial \boldsymbol{\eta}_{\tilde{\omega}}}{\partial \boldsymbol{\eta}} , \tag{4.41}$$

where $\bar{\boldsymbol{\omega}}_{ib}^b$ is the nominal body angular velocity.

4.4   GN&C Algorithm Models

4.4.1   Navigation State Dynamics

The navigation state dynamics defined in Eqs. (3.56)-(3.59) can be written in the form

$$\dot{\hat{\boldsymbol{x}}}(t) = \hat{\boldsymbol{f}}\big(\hat{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), t\big) , \tag{4.42}$$

where $\hat{\boldsymbol{f}} : \mathbb{R}^{m'} \times \mathbb{R}^c \to \mathbb{R}^{m'}$ is a nonlinear function of the navigation states $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^{m'}$ and continuous inertial measurements $\tilde{\boldsymbol{y}}(t) \in \mathbb{R}^c$. Next, let

$$\delta\hat{\boldsymbol{x}}(t) \equiv \hat{\boldsymbol{x}}(t) - \bar{\mathbf{x}}(t) , \tag{4.43}$$

where $\bar{\mathbf{x}}(t)$ denotes the nominal value of the true navigation states and $\delta\hat{\boldsymbol{x}}(t)$ represents the deviation from the nominal or navigation state dispersions. Given this expression and Eq. (4.32), it follows that the nominal navigation state dynamics can be written as

$$\dot{\bar{\mathbf{x}}}(t) = \hat{\boldsymbol{f}}\big(\bar{\mathbf{x}}(t), \bar{\boldsymbol{y}}(t), t\big) . \tag{4.44}$$

Then, following the linearization process outlined in § 4.2, linear time-varying navigation-state dynamics equation is given by

$$\delta\dot{\hat{\boldsymbol{x}}}(t) = \hat{\boldsymbol{F}}_{\hat{x}}(t)\, \delta\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{F}}_{\tilde{y}}(t)\, \delta\tilde{\boldsymbol{y}}(t) , \tag{4.45}$$

where the Jacobians $\hat{\boldsymbol{F}}_{\hat{x}}(t) \in \mathbb{R}^{m \times m}$ and $\hat{\boldsymbol{F}}_{\tilde{y}}(t) \in \mathbb{R}^{m \times c}$ are respectively defined as

$$\hat{\boldsymbol{F}}_{\hat{x}}(t) = \left. \frac{\partial \hat{\boldsymbol{f}}\left(\hat{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), t\right)}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}(t)} \right|_{\bar{\mathbf{x}}, \bar{\boldsymbol{y}}} \quad \text{and} \quad \hat{\boldsymbol{F}}_{\tilde{y}}(t) = \left. \frac{\partial \hat{\boldsymbol{f}}\left(\hat{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), t\right)}{\partial \tilde{\boldsymbol{y}}(t)} \right|_{\bar{\mathbf{x}}, \bar{\boldsymbol{y}}} . \tag{4.46}$$

Recall that due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the state covariance propagation and update equations as well as the state vector update equations. In this approach, the inertial-to-body attitude quaternion $\hat{\boldsymbol{q}}_i^b \in \mathbb{R}^4$ in the navigation state vector $\hat{\boldsymbol{x}} \in \mathbb{R}^{m'}$, given by

$$\hat{\boldsymbol{x}} = \left(\hat{\boldsymbol{r}}^i,\ \hat{\boldsymbol{v}}^i,\ \hat{\boldsymbol{q}}_i^b, \hat{\boldsymbol{p}} \right) , \tag{4.47}$$

is replaced with the Euler rotation vector $\hat{\boldsymbol{\theta}}^b \in \mathbb{R}^3$, which represents the estimated attitude of the body frame b with respect to the inertial frame i expressed in body coordinates, resulting in a modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}} \in \mathbb{R}^m$ (where $m = m' - 1$), given by

$$\hat{\boldsymbol{x}}_{\mathrm{m}} = \left(\hat{\boldsymbol{r}}^i,\ \hat{\boldsymbol{v}}^i,\ \hat{\boldsymbol{\theta}}_i^b,\ \hat{\boldsymbol{p}} \right) . \tag{4.48}$$

Moreover, the quaternion kinematics equation is replaced with the linearized Bortz equation [55]. As such, the linearized navigation state dynamics equation, given in Eq. (4.45), is based upon the modified navigation state vector $\hat{\boldsymbol{x}}$, i.e., $\delta \hat{\boldsymbol{x}}(t) \in \mathbb{R}^m$, and the Jacobians $\hat{\boldsymbol{F}}_{\hat{x}}$ and $\hat{\boldsymbol{F}}_{\tilde{y}}$ have been defined accordingly.

Inertial Position

The elements of the Jacobians $\hat{\boldsymbol{F}}_{\hat{x}}$ and $\hat{\boldsymbol{F}}_{\tilde{y}}$ associated with the inertial position kinematics, defined in Eq. (3.56) on page 40, are respectively given by

$$\left. \frac{\partial \dot{\hat{\boldsymbol{r}}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} \right|_{\bar{\mathbf{x}}, \bar{\boldsymbol{y}}} = \frac{\partial \hat{\boldsymbol{v}}^i}{\partial \hat{\boldsymbol{x}}} \quad \text{and} \quad \left. \frac{\partial \dot{\hat{\boldsymbol{r}}}^i}{\partial \tilde{\boldsymbol{y}}} \right|_{\bar{\mathbf{x}}, \bar{\boldsymbol{y}}} = \boldsymbol{0}_{3 \times c} . \tag{4.49}$$

Inertial Velocity

It can be shown that the elements of the Jacobians $\hat{\boldsymbol{F}}_{\hat{x}}$ and $\hat{\boldsymbol{F}}_{\tilde{y}}$ associated with the inertial velocity dynamics, defined in Eq. (3.57) on page 40, are respectively given by

$$\left.\frac{\partial \dot{\hat{\boldsymbol{v}}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = \left.\frac{\partial \hat{\boldsymbol{a}}_{\mathrm{g}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} + \left[\bar{\boldsymbol{T}}(\bar{\boldsymbol{q}}_i^b)\right]^{\mathsf{T}} \left(-\left[\bar{\boldsymbol{a}}_{\mathrm{ng}}^b \times\right] \frac{\partial \hat{\boldsymbol{\theta}}_i^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} + \left.\frac{\partial \hat{\boldsymbol{a}}_{\mathrm{ng}}^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}}\right) \tag{4.50}$$

and

$$\left.\frac{\partial \dot{\hat{\boldsymbol{v}}}^i}{\partial \tilde{\boldsymbol{y}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = \left[\bar{\boldsymbol{T}}(\bar{\boldsymbol{q}}_i^b)\right]^{\mathsf{T}} \left.\frac{\partial \hat{\boldsymbol{a}}_{\mathrm{ng}}^b}{\partial \tilde{\boldsymbol{y}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} \quad, \tag{4.51}$$

where the nominal accelerometer measurement $\bar{\boldsymbol{a}}_{\mathrm{ng}}^b$ is equal to the nominal nongravitational acceleration defined in Eq. (4.16) on page 59, the partial derivative of the gravitational acceleration $\hat{\boldsymbol{a}}_{\mathrm{g}}^i$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ is given in Eq. (4.54), and the partial derivatives of the compensated accelerometer measurement $\hat{\boldsymbol{a}}_{\mathrm{ng}}^b$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ and the continuous inertial measurements $\tilde{\boldsymbol{y}}$ are respectively specified in Eqs. (4.55) and (4.56).

Body Attitude

It can be shown that the elements of the Jacobians $\hat{\boldsymbol{F}}_{\hat{x}}$ and $\hat{\boldsymbol{F}}_{\tilde{y}}$ associated with the attitude quaternion kinematics, defined in Eq. (3.58) on page 40, are respectively given by

$$\left.\frac{\partial \dot{\hat{\boldsymbol{\theta}}}_i^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}} = \left.\frac{\partial \hat{\boldsymbol{\omega}}_{ib}^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}} - \left[\bar{\boldsymbol{\omega}}_{ib}^b \times\right] \frac{\partial \hat{\boldsymbol{\theta}}_i^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} \tag{4.52}$$

and

$$\left.\frac{\partial \dot{\hat{\boldsymbol{\theta}}}_i^b}{\partial \tilde{\boldsymbol{y}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}} = \left.\frac{\partial \hat{\boldsymbol{\omega}}_{ib}^b}{\partial \tilde{\boldsymbol{y}}}\right|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}}} \quad, \tag{4.53}$$

where the first expression above is the *linearized Bortz equation*, the nominal gyro measurement $\bar{\boldsymbol{\omega}}_{ib}^b$ is equal to the nominal angular velocity and the partial derivatives of the compensated gyro measurement $\hat{\boldsymbol{\omega}}_{ib}^b$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ and the continuous inertial measurements $\tilde{\boldsymbol{y}}$ are respectively specified in Eqs. (4.57) and (4.58).

Gravitational Acceleration

The partial derivative of the gravitational acceleration $\hat{\boldsymbol{a}}_{\mathrm{g}}^i$ with respect to the navigation states $\hat{\boldsymbol{x}}$ is given by

$$\left.\frac{\partial \hat{\boldsymbol{a}}_{\mathrm{g}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = -\frac{\mu}{\left\|\bar{\boldsymbol{r}}^i\right\|^3}\left(\boldsymbol{I}_{3\times 3} - 3\,\bar{\mathbf{u}}_{\hat{r}}^i\left[\bar{\mathbf{u}}_{\hat{r}}^i\right]^\mathsf{T}\right)\frac{\partial \hat{\boldsymbol{r}}^i}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\ , \tag{4.54}$$

where $\bar{\mathbf{u}}_{\hat{r}}^i$ is the unit vector in the direction of the nominal inertial position $\bar{\boldsymbol{r}}^i$.

Compensated Accelerometer Measurement

The partial derivatives of the compensated accelerometer measurement $\hat{\boldsymbol{a}}_{\mathrm{ng}}^b$, defined in Eq. (3.61) on page 40, with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ and continuous inertial measurements $\tilde{\boldsymbol{y}}$ are respectively given by

$$\left.\frac{\partial \hat{\boldsymbol{a}}_{\mathrm{ng}}^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = -\left[\bar{\boldsymbol{a}}_{\mathrm{ng}}^b\diagdown\right]\frac{\partial \hat{\boldsymbol{s}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \left[\bar{\boldsymbol{a}}_{\mathrm{ng}}^b\#\right]\frac{\partial \hat{\boldsymbol{\gamma}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \left[\bar{\boldsymbol{a}}_{\mathrm{ng}}^b\times\right]\frac{\partial \hat{\boldsymbol{\epsilon}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \frac{\partial \hat{\boldsymbol{b}}_{\tilde{a}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} \tag{4.55}$$

and

$$\left.\frac{\partial \hat{\boldsymbol{a}}_{\mathrm{ng}}^b}{\partial \tilde{\boldsymbol{y}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = \frac{\partial \tilde{\boldsymbol{a}}_{\mathrm{ng}}^b}{\partial \tilde{\boldsymbol{y}}}\ , \tag{4.56}$$

where it is straightforward to show that the nominal accelerometer measurement $\bar{\boldsymbol{a}}_{\mathrm{ng}}^b$ is equal to the nominal nongravitational acceleration defined in Eq. (4.16).

Compensated Gyro Measurement

The partial derivatives of the compensated accelerometer measurement $\hat{\boldsymbol{\omega}}_{ib}^b$, defined in Eq. (3.63) on page 40, with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\mathrm{m}}$ and continuous inertial measurements $\tilde{\boldsymbol{y}}$ are respectively given by

$$\left.\frac{\partial \hat{\boldsymbol{\omega}}_{ib}^b}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = -\left[\bar{\boldsymbol{\omega}}_{ib}^b\diagdown\right]\frac{\partial \hat{\boldsymbol{s}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \left[\bar{\boldsymbol{\omega}}_{ib}^b\#\right]\frac{\partial \hat{\boldsymbol{\gamma}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \left[\bar{\boldsymbol{\omega}}_{ib}^b\times\right]\frac{\partial \hat{\boldsymbol{\epsilon}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} - \frac{\partial \hat{\boldsymbol{b}}_{\tilde{\omega}}}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}} \tag{4.57}$$

and

$$\left.\frac{\partial \hat{\boldsymbol{\omega}}_{ib}^b}{\partial \tilde{\boldsymbol{y}}}\right|_{\bar{\mathbf{x}},\bar{\boldsymbol{y}}} = \frac{\partial \tilde{\boldsymbol{\omega}}_{ib}^b}{\partial \tilde{\boldsymbol{y}}}\ , \tag{4.58}$$

where it is readily shown that the nominal gyro measurement $\bar{\boldsymbol{\omega}}_{ib}^b$ is equal to the nominal angular velocity.

4.4.2 Filter Navigation State Error

The inertial navigation filter developed in § 3.2.1 for the Monte Carlo simulation is modified here for use in the linear covariance simulation. First, a linearized EKF is simply a standard Kalman filter, such that

$$E\{\mathbf{x}(t)\} = \bar{\mathbf{x}}(t) , \quad \forall\, t . \tag{4.59}$$

In other words, the conditional mean of $\mathbf{x}(t)$ is the *nominal value* of the true navigation state $\bar{\mathbf{x}}(t)$. Next, the linearized filter navigation state error dynamics equation, defined in Eq. (3.69) on page 41, is rewritten as

$$\delta\dot{\hat{e}}(t) = \mathbf{F}_\mathsf{x}(t)\,\delta\hat{e}(t) - \mathbf{B}\,\boldsymbol{\eta}(t) + \mathbf{G}\,\mathbf{w}(t) , \tag{4.60}$$

where $\mathbf{F}_\mathsf{x}(t) \in \mathbb{R}^{m\times m}$ is the Jacobian defined as

$$\mathbf{F}_\mathsf{x}(t) \equiv \left.\frac{\partial\mathbf{f}\big(\mathbf{x}(t),t\big)}{\mathbf{x}_\mathrm{m}(t)}\right|_{\mathbf{x}(t)=\bar{\boldsymbol{x}}(t)} , \tag{4.61}$$

and $\mathbf{B} \in \mathbb{R}^{m\times l}$ and $\mathbf{G} \in \mathbb{R}^{m\times p}$ are matrices of ones and zeros that respectively map $\boldsymbol{\eta}$ and $\mathbf{w}$ to the proper states in $\mathbf{x}_\mathrm{m}$. The elements of the Jacobian $\mathbf{F}_\mathsf{x}$ associated with the inertial position kinematics, defined in Eq. (3.44) on page 37, are given by

$$\left.\frac{\partial\dot{\boldsymbol{r}}^i}{\partial\mathbf{x}_\mathrm{m}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial\boldsymbol{r}^i}{\partial\mathbf{x}_\mathrm{m}} . \tag{4.62}$$

Next, it can be shown that the elements of the Jacobian $\mathbf{F}_\mathsf{x}$ associated with the inertial velocity dynamics, defined in Eq. (3.45) on page 37, are given by

$$\left.\frac{\partial\dot{\boldsymbol{v}}^i}{\partial\mathbf{x}_\mathrm{m}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = \left.\frac{\partial\boldsymbol{a}_\mathrm{g}^i(\boldsymbol{r}^i)}{\partial\mathbf{x}_\mathrm{m}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} + [\bar{\boldsymbol{T}}(\bar{\boldsymbol{q}}_i^b)]^\mathsf{T}\left(-[\bar{\boldsymbol{a}}_\mathrm{ng}^b\times]\frac{\partial\boldsymbol{\theta}_i^b}{\partial\mathbf{x}_\mathrm{m}} + \left.\frac{\partial\boldsymbol{a}_\mathrm{ng}^b(\tilde{\boldsymbol{a}}^s,\boldsymbol{p}_{\tilde{a}},\boldsymbol{\eta}_{\tilde{a}}^s)}{\partial\mathbf{x}_\mathrm{m}}\right|_{\mathbf{x}=\bar{\mathbf{x}}}\right) , \tag{4.63}$$

where the partial derivative of the gravitational acceleration vector $\boldsymbol{a}_\mathrm{g}^i$, defined in Eq. (3.46) on page 37, with respect to $\mathbf{x}_\mathrm{m}$ is given by

$$\left.\frac{\partial\boldsymbol{a}_\mathrm{g}^i(\boldsymbol{r}^i)}{\partial\mathbf{x}_\mathrm{m}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = -\frac{\mu}{\left\|\bar{\boldsymbol{r}}^i\right\|^3}\left(\boldsymbol{I}_{3\times3} - 3\,\bar{\mathbf{u}}_r^i\,[\bar{\mathbf{u}}_r^i]^\mathsf{T}\right)\frac{\partial\boldsymbol{r}^i}{\partial\mathbf{x}_\mathrm{m}} , \tag{4.64}$$

where $\bar{\mathbf{u}}_r^i$ is the unit vector in the direction of the nominal inertial position vector $\bar{\boldsymbol{r}}^i$. Moreover, $\bar{\boldsymbol{a}}_{\text{ng}}^b$ is the nominal value of the compensated nongravitational acceleration vector defined in Eq. (3.47) on page 37, and the partial derivative of the compensated nongravitational acceleration vector $\boldsymbol{a}_{\text{ng}}^b$ with respect to $\mathbf{x}_{\text{m}}$ is given by

$$\left.\frac{\partial \boldsymbol{a}_{\text{ng}}^b(\tilde{\boldsymbol{a}}^s, \boldsymbol{p}_{\tilde{a}}, \boldsymbol{\eta}_{\tilde{a}}^s)}{\partial \mathbf{x}_{\text{m}}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = -[\bar{\tilde{\boldsymbol{a}}}^s \diagdown]\frac{\partial \boldsymbol{s}_{\tilde{a}}}{\partial \mathbf{x}_{\text{m}}} - [\bar{\tilde{\boldsymbol{a}}}^s \#]\frac{\partial \boldsymbol{\gamma}_{\tilde{a}}}{\partial \mathbf{x}_{\text{m}}} - [\bar{\tilde{\boldsymbol{a}}}^s \times]\frac{\partial \boldsymbol{\epsilon}_{\tilde{a}}}{\partial \mathbf{x}_{\text{m}}} - \frac{\partial \boldsymbol{b}_{\tilde{a}}^s}{\partial \mathbf{x}_{\text{m}}} \ , \tag{4.65}$$

where $\bar{\tilde{\boldsymbol{a}}}^s$ is the nominal accelerometer measurement. Next, it can be shown that the elements of the Jacobian $\mathbf{F}_{\mathsf{x}}$ associated with the attitude quaternion kinematics, defined in Eq. (3.50) on page 38, are given by

$$\left.\frac{\partial \dot{\boldsymbol{\theta}}_i^b}{\partial \mathbf{x}_{\text{m}}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = \left.\frac{\partial \boldsymbol{\omega}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s)}{\partial \mathbf{x}_{\text{m}}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} - [\bar{\boldsymbol{\omega}}_{ib}^b \times]\frac{\partial \boldsymbol{\theta}_i^b}{\partial \mathbf{x}_{\text{m}}} \ , \tag{4.66}$$

where the above expression is commonly referred to as the *linearized Bortz equation*, $\bar{\boldsymbol{\omega}}_{ib}^b$ is the nominal value of the compensated gyro measurement defined in Eq. (3.52) on page 38, and the partial derivative of the compensated gyro measurement $\boldsymbol{\omega}_{ib}^b$ with respect to $\mathbf{x}_{\text{m}}$ is given by

$$\left.\frac{\partial \boldsymbol{\omega}_{ib}^b(\tilde{\boldsymbol{\omega}}^s, \boldsymbol{p}_{\tilde{\omega}}, \boldsymbol{\eta}_{\tilde{\omega}}^s)}{\partial \mathbf{x}_{\text{m}}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = -[\bar{\tilde{\boldsymbol{\omega}}}^s \diagdown]\frac{\partial \boldsymbol{s}_{\tilde{\omega}}}{\partial \mathbf{x}_{\text{m}}} - [\bar{\tilde{\boldsymbol{\omega}}}^s \#]\frac{\partial \boldsymbol{\gamma}_{\tilde{\omega}}}{\partial \mathbf{x}_{\text{m}}} - [\bar{\tilde{\boldsymbol{\omega}}}^s \times]\frac{\partial \boldsymbol{\epsilon}_{\tilde{\omega}}}{\partial \mathbf{x}_{\text{m}}} - \frac{\partial \boldsymbol{b}_{\tilde{\omega}}^s}{\partial \mathbf{x}_{\text{m}}} \ , \tag{4.67}$$

where $\bar{\tilde{\boldsymbol{\omega}}}^s$ is the nominal gyro measurement. Lastly, the elements of the Jacobian $\mathbf{F}_{\mathsf{x}}$ associated with the jth error parameter dynamics, defined in Eq. (3.31) on page 32, are given by

$$\left.\frac{\partial \dot{p}_j}{\partial \mathbf{x}_{\text{m}}}\right|_{\mathbf{x}=\bar{\mathbf{x}}} = -\frac{1}{\tau_{p_j}}\frac{\partial p_j}{\partial \mathbf{x}_{\text{m}}} \ . \tag{4.68}$$

4.4.3 Guidance Commands

Let the guidance commands $\hat{\boldsymbol{\xi}} = (\hat{\boldsymbol{q}}_{i,\text{com}}^b, \hat{\boldsymbol{\omega}}_{ib,\text{com}}^i)$ defined in § 3.2.2 be written as

$$\hat{\boldsymbol{\xi}}(t) = \hat{\boldsymbol{h}}\big(\hat{\boldsymbol{x}}(t), t\big) \ , \tag{4.69}$$

where $\hat{\boldsymbol{h}} : \mathbb{R}^{m'} \to \mathbb{R}^{g'}$ is a nonlinear function of the navigation states $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^{m'}$. Next, let

$$\delta\hat{\boldsymbol{\xi}}(t) \equiv \hat{\boldsymbol{\xi}}(t) - \bar{\boldsymbol{\xi}}(t) \ , \tag{4.70}$$

where $\bar{\boldsymbol{\xi}}(t)$ denotes the nominal guidance commands and $\delta\hat{\boldsymbol{\xi}}(t)$ represents the deviation from the nominal. Given this expression and a similar expression for the navigation states in Eq. (4.43), it follows that the nominal guidance commands $\bar{\boldsymbol{\xi}}(t)$ can be written as

$$\bar{\boldsymbol{\xi}}(t) = \hat{\boldsymbol{h}}\big(\bar{\mathbf{x}}(t), t\big) \ . \tag{4.71}$$

Then, following the linearization process outlined in § 4.2, the linear time-varying guidance commands equation is given by

$$\delta\hat{\boldsymbol{\xi}}(t) = \hat{\boldsymbol{H}}_{\hat{x}}(t)\, \delta\hat{\boldsymbol{x}}(t) \ , \tag{4.72}$$

where the Jacobian $\hat{\boldsymbol{H}}_{\hat{x}}(t) \in \mathbb{R}^{g\times m}$ is defined as

$$\hat{\boldsymbol{H}}_{\hat{x}}(t) = \left.\frac{\partial\hat{\boldsymbol{h}}\big(\hat{\boldsymbol{x}}(t), t\big)}{\partial\hat{\boldsymbol{x}}_{\mathrm{m}}(t)}\right|_{\bar{\mathbf{x}}} \ . \tag{4.73}$$

Note that these last two equations represent the linearized guidance algorithm. Also, recall that due to state covariance matrix singularity issues associated with the quaternion [54], the inertial-to-body attitude quaternions $\boldsymbol{q}_i^b$, $\hat{\boldsymbol{q}}_i^b \in \mathbb{R}^4$ in the true and navigation state vectors were replaced with the Euler rotation vectors $\boldsymbol{\theta}_i^b$, $\hat{\boldsymbol{\theta}}_i^b \in \mathbb{R}^3$, resulting in modified true and modified navigation state vectors. Since one of the guidance commands is an attitude quaternion, it follows that this modified state vector approach must also be applied here. Thus, the commanded attitude quaternion $\hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}} \in \mathbb{R}^4$ in the guidance commands vector $\hat{\boldsymbol{\xi}} \in \mathbb{R}^{g'}$, given by

$$\hat{\boldsymbol{\xi}} = \big(\hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}}, \hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^i\big) \ , \tag{4.74}$$

is replaced with the Euler rotation vector $\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}} \in \mathbb{R}^3$, which represents the desired attitude of the body frame b with respect to the inertial frame i expressed in desired body coordinates, resulting in a modified guidance commands vector $\hat{\boldsymbol{\xi}}_{\mathrm{m}} \in \mathbb{R}^g$ (where $g = g' - 1$), given by

$$\hat{\boldsymbol{\xi}}_{\mathrm{m}} = \big(\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}, \hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^i\big) \ . \tag{4.75}$$

As such, the linearized guidance commands equation, given in Eq. (4.72), is based upon the modified guidance commands vector $\hat{\boldsymbol{\xi}}_{\mathrm{m}}$, i.e., $\delta\hat{\boldsymbol{\xi}}(t) \in \mathbb{R}^g$, and the dimensions of the Jacobian $\hat{\boldsymbol{H}}_{\hat{x}}$ have been

defined accordingly.

Recall from § 3.2.2 that during vertical rise and pitch over, guidance operates in an open-loop fashion where the guidance commands issued by guidance were computed prior to launch and stored in the flight computer. Since these precomputed guidance commands are based upon the desired or NRT and not on the current navigation states, it follows that they are the nominal guidance commands during this period and therefore must satisfy Eq. (4.71). Consequently, $\delta\hat{\boldsymbol{\xi}}(t) = \mathbf{0}$ and it follows that the Jacobian $\hat{\boldsymbol{H}}_{\hat{x}}$ must also be zero (or the null matrix). In other words, the partial derivatives of the guidance commands with respect to the navigation states $\hat{\boldsymbol{x}}$ during vertical rise and pitch over are zero. From the end of pitch over to when guidance modes to fine count, guidance operates in a closed-loop fashion where the guidance commands are computed on-the-fly with PEG and the current navigation solution. Thus, it follows that the guidance commands during this period satisfy Eq. (4.69) and that Eqs. (4.72) and (4.73) represent the linearized closed-loop guidance algorithm or PEG.

Until this point, it has been possible to formulate closed-form analytic expressions for all of the requisite partial derivatives. However, since PEG is a very complex, highly nonlinear algorithm and contains an iterative solver, it follows naturally that closed-form analytic expressions for the partial derivatives associated with PEG are not obtainable and therefore numerical methods must be employed. The problem with numerical methods is that they are never exact and are at best approximations of the values or functions that they represent. For example, if standard finite-divided difference methods, such as the forward difference or central difference formulas, are employed to evaluate the partial derivatives, then one is faced with the well-known "step-size dilemma," i.e., what step size to use. If the step size is too small, then the obtained value of the derivative may be subject to subtractive cancellation or round-off errors, which are due to the finite precision of the digital computer. Conversely, if the step size is too large, then the obtained value of the derivative may be subject to truncation errors, which are due to the finite number of terms in the Taylor series that were included in the derivative approximation. In order to find the so-called "optimal" step size, i.e., the step size that minimizes both round-off and truncation errors for a given function and computer, one must generally devote a considerable amount of time and effort.

It is worth noting that higher-order finite-divided difference methods, such as Richardson Extrapolation and Romberg Integration, have been developed to obtain more accurate values of the

derivative [80]. They basically employ more higher-order terms in the Taylor series in the derivative approximation so as to reduce the effect of truncation errors. They are, however, still subject to round-off errors and the step-size dilemma. Moreover, they require a minimum of four[1] separate evaluations of the function or algorithm, thus there is also a computational penalty associated with these higher-order methods.

Complex-Step Derivative

One numerical method that is not subject to the step-size dilemma and has been shown to achieve near analytic accuracy (to within machine precision) when computing the first derivative of real-valued functions or algorithms is the *complex-step derivative approximation* [81], which states that if a function $f(x)$ is analytic, i.e., differentiable in the complex plane, then the first derivative of the function at the operating point x_0 is given by

$$f'(x_0) \approx \frac{\text{Im}\{f(x_0 + ih)\}}{h} \, ,$$
(4.76)

where $\text{Im}\{f\}$ represents the imaginary part of the function f evaluated at the complex value $x_0 + ih$, and h denotes the step size, i.e., the length of the interval along x over which the approximation is made. Notice that Eq. (4.76) does not contain a difference operation, thus it is not subject to subtractive cancellation or round-off errors. Therefore, extremely small step sizes (e.g., $h = 10^{-40}$) can be used with no loss of accuracy or precision. As a result, the truncation error in the approximation, which is of $\mathcal{O}(h^2)$, can be either eliminated altogether or minimized and rendered negligible [81, 82]. Furthermore, Eq. (4.76) involves only one[2] function evaluation, thus the computational penalty associated with this method is simply due to the time required to perform complex arithmetic. General implementation procedures and common issues (and their solutions) associated with several popular programming languages (e.g., MATLAB, C/C++, FORTRAN) are outlined and discussed in Martins et al. [81]. There are, however, a few additional issues that were discovered in the course of this research that are not mentioned in Ref. [81] or the literature.

In order to apply the complex-step derivative (CSD) method to PEG or any other computer

---

[1]This is for a scalar single-variable function using Richardson Extrapolation. The number of function evaluations increases for vector and multivariable functions as well as for Romberg Integration with more than one iteration. [80]

[2]This is for a scalar single-variable function. The number of function evaluations increases for vector and multivariable functions.

algorithm for that matter, the algorithm and all of its subfunctions must be defined to handle complex values. The reason for this is that the derivative information is captured and maintained in the imaginary part of the variables. Therefore, should one of the subfunctions inadvertently discard or improperly modify the imaginary part of a complex variable, then the derivative information would be lost, resulting in an incorrect derivative. Hence, one issue that was discovered is that the intrinsic MATLAB functions `norm` and `atan2`, both of which are employed by the guidance algorithm, are not defined to handle complex values. Since the functions are intrinsic to MATLAB and cannot be edited, it became necessary to develop and create customized functions that are able to handle both real and complex values, then replace all calls to the intrinsic functions in the guidance algorithm with calls to the replacement functions. The details of how the customized functions were developed and coded are not discussed here but can be found in Appendix B.1.

Another issue that was encountered is how to properly apply the CSD method to vector functions or algorithms when one or more of the input or output variables is an attitude quaternion. In this research, PEG is the vector algorithm of interest, the navigation states $\hat{x}$ are the inputs, the guidance commands $\hat{\boldsymbol{\xi}}$ are the outputs, and both the input and output variables include attitude quaternions. There are actually two separate problems associated with this issue. The first problem deals with how the perturbation step is added to the input quaternion. This is because the perturbation step isn't added to the quaternion in the same manner as a vector and the method requires that the perturbation be added to the imaginary part of the nominal input variable. The second problem, however, deals with how the derivative information is extracted from the output quaternion. This is because the derivative information is contained in the imaginary part of the four-dimensional quaternion whereas the partial derivative is of the three-dimensional attitude vector. The details of how to properly apply the CSD method to quaternions is not discussed here but can be found in Appendix B.2.

Fine-Count Partial-Derivative Dilemma

During fine count, the last commanded inertial thrust direction computed by PEG is held constant, which corresponds to a constant attitude command and a zero attitude rate command. Since the guidance commands during fine count are not a function of the current navigation states, one might conclude that the associated partial derivatives must be zero, just as those during vertical

rise and pitch over were zero. This is a correct conclusion for the partial derivatives associated with the zero attitude rate command, because from a Monte Carlo perspective the attitude rate command is invariant from sample run to sample run. This is an incorrect deduction, however, for the partial derivatives associated with the constant attitude command, because the last attitude command computed by PEG will undoubtedly vary from sample run to sample run due to variations in the sample trajectories and the corresponding navigation states. In other words, the variation in the commanded attitude at some time t_c during fine count, observed from collecting N Monte Carlo samples, is due to variations in the navigation states at the last time that PEG was called or time t_{e-1} (where t_{e-1} denotes the time step just before the event time t_e and $t_{e-1} < t_c$). Thus, it follows that *the partial derivatives associated with the constant attitude command with respect to the current modified navigation states during fine count should not be zero.* Consequently, one might conclude that since the attitude command is held constant, the associated partial derivatives must also be constant and therefore equal to those computed at time t_{e-1}. However, when the partial derivatives are applied in this fashion, it is equivalent to stating that the variation in the commanded attitude at time t_c is somehow due to variations in the current navigation states at time t_c, which contradicts the statement above that the guidance commands during fine count are not a function of the current navigation states. Thus, it follows that *the partial derivatives associated with the constant attitude command with respect to the current modified navigation states during fine count should not be constant.* As a result, one is faced with the dilemma of how to properly compute the partial derivatives associated with the constant attitude command during fine count, where PEG is no longer called.

The first approach that was attempted was to write the partial derivative of the constant attitude command at time t_e (the first time step into fine count) in terms of the partial derivative computed at time t_{e-1} (the time step just before guidance modes to fine count and the last time that PEG was called) as follows

$$\left. \frac{\partial \hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}(t_e)}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}(t_e)} \right|_{\bar{\mathbf{x}}} = \frac{\partial \hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}(t_{e-1})}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}(t_{e-1})} \left. \frac{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}(t_{e-1})}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}(t_e)} \right|_{\bar{\mathbf{x}}},$$

where it is noted that $\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}(t_{e-1}) = \hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}(t_e)$ because the attitude command is held constant during fine count. Furthermore, it is straightforward to show that the last term on the right-hand

side is simply the state transition matrix $\hat{\mathbf{\Phi}}(t_e, t_{e-1})$, thus the previous expression can be written as

$$\left.\frac{\partial \hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}(t_{e-1})}{\partial \hat{\boldsymbol{x}}_\text{m}(t_e)}\right|_{\bar{\mathbf{x}}} = \left.\frac{\partial \hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}(t_{e-1})}{\partial \hat{\boldsymbol{x}}_\text{m}(t_{e-1})} \hat{\mathbf{\Phi}}(t_e, t_{e-1})\right|_{\bar{\mathbf{x}}} . \tag{4.77}$$

This expression can then be readily adapted for any future time $t > t_{e-1}$. Unfortunately, this state transition matrix approach was not able to properly capture the effects that the constant attitude command has on the true and navigation states during fine count, as seen with Monte Carlo analysis. Hence, it was subsequently abandoned.

A second approach was undertaken that seems to avoid this whole fine-count partial-derivative dilemma altogether, i.e., it does not require the partial derivatives of the attitude command during fine count at all. More importantly, it is able to correctly capture the effects that the constant attitude command has on the true and navigation states during fine count. In this approach, a pseudo-state $\hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}$ (i.e, the Euler rotation vector representation of the attitude command quaternion $\hat{\boldsymbol{q}}_{i,\text{com}}^{\hat{b}}$) is created and appended to the modified true state vector $\boldsymbol{x}_\text{m}$ as follows

$$\boldsymbol{x}_\text{m} = \left(\boldsymbol{r}^i, \, \boldsymbol{v}^i, \, \boldsymbol{\theta}_i^b, \, \boldsymbol{\omega}_{ib}^b, \, m_b, \, \boldsymbol{p}, \, \hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}\right) . \tag{4.78}$$

Then, each time that PEG is called, the updated attitude command is applied as an instantaneous state covariance correction, which uses the partial derivatives of the attitude command. Finally, when guidance modes to fine count and PEG is no longer called, no state covariance corrections are made, thus eliminating the need for the partial derivatives of the attitude command during fine count. The details of the state covariance correction are not discussed here, but can be found in § 5.5.3.

It is important to note a few things at this point. First, this new pseudo-state is only for linear covariance purposes, i.e., it is only used in the formulation of the linear covariance equations and in the linear covariance simulation. Actually, it is common practice in linear covariance to append extra states in order to capture the covariance of certain aspects of the problem that are not normally captured with the traditional states of the system. Second, a pseudo-state for the attitude rate command $\hat{\boldsymbol{\omega}}_{ib,\text{com}}^i$ could also have been created and appended in the same manner. However, since the partial derivatives associated with the attitude rate command were not a factor in the

fine-count partial-derivative dilemma,[3] this option was not explored. Third, the attitude command

pseudo-state was initially appended to the modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}}$, primarily because

the attitude command is a function of the navigation states $\hat{\boldsymbol{x}}$ and it made more sense. However,

this doesn't work because current linear covariance theory, particularly the theory associated with

event triggers, is unable to handle the case where the navigation state vector (or modified navigation

state vector) contains states that are not in the true state vector or are not linear (or nonlinear)

functions of the true states. For this reason, the pseudo-state was appended to the modified true

state vector instead.

4.4.4 Actuator Commands

Since the modified attitude command $\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}$ is now a true state, it follows that the actuator

commands defined in Eq. (3.116) can be written in the form

$$\hat{\boldsymbol{u}}(t) = \hat{\boldsymbol{g}}\big(\hat{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), \boldsymbol{x}(t), t\big) \;, \tag{4.79}$$

where $\hat{\boldsymbol{g}} : \mathbb{R}^{m'} \times \mathbb{R}^{c} \times \mathbb{R}^{n'} \to \mathbb{R}^{a}$ is a nonlinear function of the navigation states $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^{m'}$,

continuous inertial measurements $\tilde{\boldsymbol{y}}(t) \in \mathbb{R}^{c}$, and true states $\boldsymbol{x}(t) \in \mathbb{R}^{n'}$. Given the expressions in

Eqs. (4.2), (4.32), and (4.43), it follows that the nominal actuator commands $\bar{\boldsymbol{u}}(t)$ can be written

as

$$\bar{\boldsymbol{u}}(t) = \hat{\boldsymbol{g}}\big(\bar{\mathbf{x}}(t), \bar{\boldsymbol{y}}(t), \bar{\boldsymbol{x}}(t), t\big) \;. \tag{4.80}$$

Then, following the linearization process outlined in §4.2, the linear time-varying actuator com-

mands equation is given by

$$\delta\hat{\boldsymbol{u}}(t) = \hat{\boldsymbol{G}}_{\hat{x}}(t)\,\delta\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\delta\tilde{\boldsymbol{y}}(t) + \hat{\boldsymbol{G}}_{x}(t)\,\delta\boldsymbol{x}(t) \;, \tag{4.81}$$

where the Jacobians $\hat{\boldsymbol{G}}_{\hat{x}}(t) \in \mathbb{R}^{a \times m}$, $\hat{\boldsymbol{G}}_{\tilde{y}}(t) \in \mathbb{R}^{a \times c}$ and $\hat{\boldsymbol{G}}_{x}(t) \in \mathbb{R}^{a \times n}$, are respectively defined as

$$\hat{\boldsymbol{G}}_{\hat{x}}(t) = \left.\frac{\partial\hat{\boldsymbol{g}}\big(\hat{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), \boldsymbol{x}(t), t\big)}{\partial\hat{\boldsymbol{x}}_{\mathrm{m}}(t)}\right|_{\bar{\mathbf{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}} \;, \quad \hat{\boldsymbol{G}}_{\tilde{y}}(t) = \left.\frac{\partial\hat{\boldsymbol{g}}\big(\hat{\boldsymbol{x}}(t), \tilde{\boldsymbol{y}}(t), \boldsymbol{x}(t), t\big)}{\partial\tilde{\boldsymbol{y}}(t)}\right|_{\bar{\mathbf{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{x}}} \;, \tag{4.82}$$

---

[3]During fine count, the attitude rate command $\hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^{i}$ is like an open loop command, i.e., $\hat{\boldsymbol{\omega}}_{ib,\mathrm{com}}^{i}$ is **always** zero.

and

$$\hat{G}_x(t) = \left.\frac{\partial \hat{g}\big(\hat{x}(t), \tilde{y}(t), x(t), t\big)}{\partial x_{\mathrm{m}}(t)}\right|_{\bar{x},\bar{y},\bar{x}} .$$
(4.83)

Recall that due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the state covariance propagation and update equations as well as the state vector update equations. As such, the linearized actuator commands equation, given in Eq. (4.81), is based upon the modified navigation state vector $\hat{x}_{\mathrm{m}}$ and modified true state vector x_{m}, i.e., $\delta\hat{x}(t) \in \mathbb{R}^m$ and $\delta x(t) \in \mathbb{R}^n$, and the Jacobians $\hat{G}_{\hat{x}}$, $\hat{G}_{\tilde{y}}$, and $\hat{G}_x$ have been defined accordingly.

Compensated Torque Command

The elements of the Jacobians $\hat{G}_{\hat{x}}$, $\hat{G}_{\tilde{y}}$, and $\hat{G}_x$ associated with the compensated ACS torque command $\hat{m}^b_{\mathrm{com}}$, defined in Eq. (3.116), are respectively given by

$$\left.\frac{\partial \hat{m}^b_{\mathrm{com}}}{\partial \hat{x}_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}} = \left.\frac{\partial \hat{m}^b_{\mathrm{err}}}{\partial \hat{x}_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}} - \big[\bar{m}^b_{\mathrm{err}}\diagdown\big]\frac{\partial \hat{s}_{\mathrm{acs}}}{\partial \hat{x}_{\mathrm{m}}} - \big[\bar{m}^b_{\mathrm{err}}\times\big]\frac{\partial \hat{\epsilon}_{\mathrm{acs}}}{\partial \hat{x}_{\mathrm{m}}} - \frac{\partial \hat{b}_{\mathrm{acs}}}{\partial \hat{x}_{\mathrm{m}}} ,$$
(4.84)

$$\left.\frac{\partial \hat{m}^b_{\mathrm{com}}}{\partial \tilde{y}}\right|_{\bar{x},\bar{y},\bar{x}} = \left.\frac{\partial \hat{m}^b_{\mathrm{err}}}{\partial \tilde{y}}\right|_{\bar{x},\bar{y},\bar{x}} \quad \text{and} \quad \left.\frac{\partial \hat{m}^b_{\mathrm{com}}}{\partial x_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}} = \left.\frac{\partial \hat{m}^b_{\mathrm{err}}}{\partial x_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}} ,$$
(4.85)

where $\bar{m}^b_{\mathrm{err}}$ is the nominal value of the required ACS torque and the partial derivatives of the required ACS torque $\hat{m}^b_{\mathrm{err}}$ with respect to the modified navigation states $\hat{x}_{\mathrm{m}}$, continuous inertial measurements $\tilde{y}$, and modified true states x_{m} are specified in Eqs. (4.86) and (4.87).

Attitude Control Law

It can be shown that the partial derivatives of the required ACS torque $\hat{m}^b_{\mathrm{err}}$, defined in Eq. (3.113), with respect to the modified navigation states $\hat{x}_{\mathrm{m}}$, continuous inertial measurements $\tilde{y}$, and modified true states x_{m} are respectively given by

$$\left.\frac{\partial \hat{m}^b_{\mathrm{err}}}{\partial \hat{x}_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}} = \Big(\bar{K}^b_\omega\big[\bar{T}(\bar{q}^b_i)\,\bar{\omega}^i_{ib,\mathrm{com}}\times\big] - \bar{K}^b_\theta\Big)\frac{\partial \hat{\theta}^b_i}{\partial \hat{x}_{\mathrm{m}}} + \bar{K}^b_\omega\,\bar{T}(\bar{q}^b_i)\left.\frac{\partial \hat{\omega}^i_{ib,\mathrm{com}}}{\partial \hat{x}_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}}$$

$$- \bar{K}^b_\omega\left.\frac{\partial \hat{\omega}^b_{ib}}{\partial \hat{x}_{\mathrm{m}}}\right|_{\bar{x},\bar{y},\bar{x}} ,$$
(4.86)

$$\frac{\partial \hat{\boldsymbol{m}}_{\text{err}}^{b}}{\partial \tilde{\boldsymbol{y}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{x}}} = -\bar{\boldsymbol{K}}_{\omega}^{b}\,\frac{\partial \hat{\boldsymbol{\omega}}_{ib}^{b}}{\partial \tilde{\boldsymbol{y}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{x}}} \quad \text{and} \quad \frac{\partial \hat{\boldsymbol{m}}_{\text{err}}^{b}}{\partial \boldsymbol{x}_{\text{m}}}\bigg|_{\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{x}}} = \bar{\boldsymbol{K}}_{\theta}^{b}\frac{\partial \hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}}{\partial \boldsymbol{x}_{\text{m}}} \;, \tag{4.87}$$

where $\bar{\boldsymbol{K}}_{\theta}^{b}$ and $\bar{\boldsymbol{K}}_{\omega}^{b}$ are respectively the nominal proportional and derivative attitude control gains, $\bar{\boldsymbol{\omega}}_{ib,\text{com}}^{i}$ is the nominal attitude rate command from guidance, and the partial derivatives of the compensated gyro measurement $\hat{\boldsymbol{\omega}}_{ib}^{b}$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\text{m}}$ and inertial measurements $\tilde{\boldsymbol{y}}$ are respectively specified in Eqs. (4.57) and (4.58). The partial derivative of the attitude rate command $\hat{\boldsymbol{\omega}}_{ib,\text{com}}^{i}$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\text{m}}$ is computed with the complex-step derivative approximation described in §4.4.3. The partial derivative of the attitude command $\hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}$ with respect to the modified true states $\boldsymbol{x}_{\text{m}}$,which appears in Eq. (4.87), is due to the attitude command being added to the true states, as described in §4.4.3. The partial derivative of the attitude command $\hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}$ with respect to the modified navigation states $\hat{\boldsymbol{x}}_{\text{m}}$ does not appear here, but is used in the state covariance correction [see §5.5.3].

In summary, the NRT was defined and the nonlinear truth models and GN&C algorithm models associated with the Monte Carlo simulation were linearized about the NRT. However, since PEG is a very complex, highly nonlinear algorithm, closed-form analytic expressions of the partial derivatives could not be obtained, hence the CSD method was employed to get the most accurate numerical partial derivatives possible. The linearized models derived herein will be used in the next chapter to formulate the linear covariance equations. Lastly, in order to solve the fine-count partial-derivative dilemma, the guidance attitude command vector pseudo-state was created and appended to the modified true state vector, so that each time that PEG is called the updated attitude command is applied as an instantaneous state covariance correction. Then when guidance modes to fine count and PEG is no longer called, no more state covariance corrections are made, thereby capturing the effects of the constant attitude command on both the true states and navigation states while eliminating the need for the partial derivatives of the attitude command during fine count.

Chapter 5

Linear Covariance Formulation

"Simplicity is the ultimate sophistication."

– Leonardo da Vinci

In the preceding chapter, the NRT was defined and the nonlinear truth models and GN&C algorithm models were linearized about the nominal reference trajectory. The purpose of this chapter is to present the final step in the development process, which is to formulate the linear covariance equations that constitute the linear covariance simulation, thereby accomplishing the third objective of this dissertation. The chapter is organized as follows: First, the linearized propagation equations for the true state dispersions and navigation state dispersions are obtained, which are then used to define the augmented state vector and associated propagation equation. Next, the true navigation state errors are derived and expressed in terms of the augmented state vector. This is followed by a reformulation of the event trigger theory needed to handle discrete events triggered by something other than time. Lastly, the linear covariance equations are formulated and presented. This includes initialization, propagation, correction, and shaping of the augmented state covariance matrix and initialization and propagation of the filter navigation state error covariance matrix, as well as covariance post-processing.

5.1   Linearized Propagation Equations

Now that linearized expressions have been obtained for the true state dynamics, inertial measurements, navigation state dynamics, and actuator commands, it follows that the linearized actuator commands, given in Eq. (4.81), can be substituted into the linearized true state dynamics, given in Eq. (4.6), such that

$$\delta\dot{\boldsymbol{x}}(t) = \left[\boldsymbol{F}_x(t) + \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_x(t)\right]\delta\boldsymbol{x}(t) + \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\hat{x}}(t)\,\delta\hat{\boldsymbol{x}}(t) + \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\delta\tilde{\boldsymbol{y}}(t) + \boldsymbol{F}_w(t)\,\boldsymbol{w}(t)\ .$$

Next, the linearized inertial measurements, given in Eq. (4.34), can be substituted into the above expression, such that the linearized propagation equation for the true state dispersions is given by

$$\delta\dot{\boldsymbol{x}}(t) = \left\{\boldsymbol{F}_x(t) + \boldsymbol{F}_{\hat{u}}(t)\left[\hat{\boldsymbol{G}}_x(t) + \hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\boldsymbol{C}_x(t)\right]\right\}\delta\boldsymbol{x}(t) + \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\hat{x}}(t)\,\delta\hat{\boldsymbol{x}}(t)$$
$$+ \left[\boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\boldsymbol{C}_w(t) + \boldsymbol{F}_w(t)\right]\boldsymbol{w}(t) + \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\boldsymbol{C}_\eta(t)\,\boldsymbol{\eta}(t)\;. \tag{5.1}$$

Similarly, the linearized inertial measurements, given in Eq. (4.34), can be substituted into the linearized navigation state dynamics, given in Eq. (4.45), such that the linearized propagation equation for the navigation state dispersions is given by

$$\delta\dot{\hat{\boldsymbol{x}}}(t) = \hat{\boldsymbol{F}}_{\tilde{y}}(t)\,\boldsymbol{C}_x(t)\,\delta\boldsymbol{x}(t) + \hat{\boldsymbol{F}}_{\hat{x}}(t)\,\delta\hat{\boldsymbol{x}}(t) + \hat{\boldsymbol{F}}_{\tilde{y}}(t)\,\boldsymbol{C}_w(t)\,\boldsymbol{w}(t) + \hat{\boldsymbol{F}}_{\tilde{y}}(t)\,\boldsymbol{C}_\eta(t)\,\boldsymbol{\eta}(t)\;. \tag{5.2}$$

5.2 Augmented Linear System

Let the true state dispersions $\delta\boldsymbol{x} \in \mathbb{R}^n$ and navigation state dispersions $\delta\hat{\boldsymbol{x}} \in \mathbb{R}^m$ be formed into an augmented state vector $\mathbf{x}_\mathrm{a} \in \mathbb{R}^\ell$ (where $\ell = n + m$),

$$\mathbf{x}_\mathrm{a} = \left(\delta\boldsymbol{x},\,\delta\hat{\boldsymbol{x}}\right)\;, \tag{5.3}$$

such that Eqs. (5.1) and (5.2) can be condensed to the following linear augmented-state propagation equation,

$$\dot{\mathbf{x}}_\mathrm{a}(t) = \mathbf{F}(t)\,\mathbf{x}_\mathrm{a}(t) + \mathbf{W}(t)\,\boldsymbol{w}(t) + \mathbf{G}(t)\,\boldsymbol{\eta}(t)\;, \tag{5.4}$$

where the matrices $\mathbf{F}(t) \in \mathbb{R}^{\ell\times\ell}$, $\mathbf{W}(t) \in \mathbb{R}^{\ell\times s}$, and $\mathbf{G}(t) \in \mathbb{R}^{\ell\times c}$ are defined as

$$\mathbf{F}(t) = \begin{bmatrix} \boldsymbol{F}_x(t) + \boldsymbol{F}_{\hat{u}}(t)\left[\hat{\boldsymbol{G}}_x(t) + \hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\boldsymbol{C}_x(t)\right] & \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\hat{x}}(t) \\ \hat{\boldsymbol{F}}_{\tilde{y}}(t)\,\boldsymbol{C}_x(t) & \hat{\boldsymbol{F}}_{\hat{x}}(t) \end{bmatrix}, \tag{5.5}$$

$$\mathbf{W}(t) = \begin{bmatrix} \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\boldsymbol{C}_w(t) + \boldsymbol{F}_w(t) \\ \hat{\boldsymbol{F}}_{\tilde{y}}(t)\,\boldsymbol{C}_w(t) \end{bmatrix}, \quad \text{and} \quad \mathbf{G}(t) = \begin{bmatrix} \boldsymbol{F}_{\hat{u}}(t)\,\hat{\boldsymbol{G}}_{\tilde{y}}(t)\,\boldsymbol{C}_\eta(t) \\ \hat{\boldsymbol{F}}_{\tilde{y}}(t)\,\boldsymbol{C}_\eta(t) \end{bmatrix}. \tag{5.6}$$

5.3 True Navigation State Errors

The true navigation state errors $\delta\boldsymbol{e}(t) \in \mathbb{R}^{m'}$ are defined as the difference between the true navigation states $\mathbf{x}(t) \in \mathbb{R}^{m'}$ and the navigation states $\hat{\boldsymbol{x}}(t) \in \mathbb{R}^{m'}$

$$\delta\boldsymbol{e}(t) \equiv \mathbf{x}(t) - \hat{\boldsymbol{x}}(t) \ . \tag{5.7}$$

where the true navigation states can be written in the general form

$$\mathbf{x}(t) = \boldsymbol{m}\big(\boldsymbol{x}(t)\big) \ , \tag{5.8}$$

where $\boldsymbol{m} : \mathbb{R}^{n'} \to \mathbb{R}^{m'}$ is a nonlinear function that maps the true states $\boldsymbol{x} \in \mathbb{R}^{n'}$ to the true navigation states $\mathbf{x}$. It is important to emphasize that this mapping function represents the general case, i.e., it is valid for both full- and reduced-state filters as well as for both inertial- and relative-state filters. Next, given that the true states $\boldsymbol{x}$ can be written as

$$\delta\boldsymbol{x}(t) \equiv \boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t) \ , \tag{5.9}$$

it follows that $\boldsymbol{m}$ can be linearized about the nominal reference trajectory, following the process outlined in § 4.2, such that Eq. (5.8) becomes

$$\mathbf{x}(t) = \boldsymbol{m}\big(\bar{\boldsymbol{x}}(t)\big) + \boldsymbol{M}_x(t)\,\delta\boldsymbol{x}(t) \ , \tag{5.10}$$

where the Jacobian $\boldsymbol{M}_x(t) \in \mathbb{R}^{m \times n}$ is defined as

$$\boldsymbol{M}_x(t) = \left. \frac{\partial \boldsymbol{m}\big(\boldsymbol{x}(t)\big)}{\partial \boldsymbol{x}(t)} \right|_{\boldsymbol{x}(t) = \bar{\boldsymbol{x}}(t)} \ . \tag{5.11}$$

Recall from the previous chapter that due to state covariance matrix singularity issues associated with the quaternion, the inertial-to-body attitude quaternions $\boldsymbol{q}_i^b$, $\hat{\boldsymbol{q}}_i^b$ in the true and navigation state vectors were respectively replaced with the Euler rotation vectors $\boldsymbol{\theta}_i^b$, $\hat{\boldsymbol{\theta}}_i^b$, resulting in modified true and modified navigation state vectors. As such, the linearized mapping function, given in Eq. (5.10), is derived using the modified true state vector, i.e., $\delta\boldsymbol{x} \in \mathbb{R}^n$, and the dimensions of the mapping function Jacobian $\boldsymbol{M}_x$ have been defined accordingly. Next, the navigation states $\hat{\boldsymbol{x}}$ can

be written as

$$\hat{\boldsymbol{x}}(t) = \bar{\mathbf{x}}(t) + \delta\hat{\boldsymbol{x}}(t) = \boldsymbol{m}\big(\bar{\boldsymbol{x}}(t)\big) + \delta\hat{\boldsymbol{x}}(t) \; , \tag{5.12}$$

which is also derived using the modified navigation state vector, for the reasons just described. Substituting this expression and Eq. (5.10) into the definition given in Eq. (5.7) and canceling terms yields the following expression for the true navigation state errors in terms of the true state dispersions $\delta\boldsymbol{x}$ and navigation state dispersions $\delta\hat{\boldsymbol{x}}$,

$$\delta\boldsymbol{e}(t) = \boldsymbol{M}_x(t)\,\delta\boldsymbol{x}(t) - \delta\hat{\boldsymbol{x}}(t) \; . \tag{5.13}$$

which can also be expressed in terms of the augmented state $\mathbf{x}_{\mathrm{a}}$

$$\delta\boldsymbol{e}(t) = \left[\begin{array}{cc} \boldsymbol{M}_x(t) & -\boldsymbol{I}_{m\times m} \end{array} \right] \mathbf{x}_{\mathrm{a}}(t) \; . \tag{5.14}$$

5.4 Event Triggers

Recall that the time-to-go parameter t_{go}, calculated by PEG and discussed in § 3.2.2, is dependent upon the current estimated flight path (i.e., navigation states) and the terminal targets, hence variations in either will result in variations in the burn time. Since the terminal targets are constant, it follows that, from a Monte Carlo perspective, the time when guidance modes from PEG to fine count occurs relative to lift-off (t_0) will vary from sample run to sample run due to variations in the flight path. The problem is that a linear covariance simulation has only one flight path (i.e., the nominal reference trajectory) from which to generate and capture the covariance of this time-varying event, not N (where $N > 1$) trajectories like Monte Carlo methods. Hence, the purpose of this section is to address this problem of how to properly handle discrete events that are triggered by some criteria other than time, known as *event triggers*, in a linear covariance simulation.

An *event trigger* is a condition or constraint that can be expressed mathematically as a function of the estimated or navigation states $\hat{\boldsymbol{x}}$, [15]

$$\Psi\big[\hat{\boldsymbol{x}}(t_{\mathrm{e}})\big] = 0 \; . \tag{5.15}$$

When this condition is satisfied, a discrete event at time t_{e} is triggered. Examples of discrete

events include (but are not limited to) impulsive maneuvers, sensor changes, and general GN&C mode changes. Examples of trigger conditions include (but are not limited to) estimates of the altitude, downrange position, velocity magnitude, and the remaining burn time. The theory and techniques developed for implementing event triggers in a linear covariance simulation are presented and outlined in Geller et al. [15]. It is important to point out, however, that their formulation is only applicable to the special case where the navigation algorithm incorporates a full-state filter, i.e., the navigation state vector $\hat{\boldsymbol{x}} \in \mathbb{R}^{n'}$ contains the same states as the true or environment state vector $\boldsymbol{x} \in \mathbb{R}^{n'}$. In other words, it does not allow for a reduced-state filter (e.g., $\hat{\boldsymbol{x}} \in \mathbb{R}^{m'}$, $\boldsymbol{x} \in \mathbb{R}^{n'}$, where $m' \leq n'$) or a relative-state filter in lieu of the more traditional inertial-state filter. In order for the event trigger theory and techniques to be suitable for a wide variety of problems in aerospace and this research, a modified formulation was developed for the more general case where the filter design states (and true navigation states) $\mathbf{x} \in \mathbb{R}^{m'}$, from which the navigation states $\hat{\boldsymbol{x}} \in \mathbb{R}^{m'}$ are derived, are defined to be a nonlinear function of the true states $\boldsymbol{x} \in \mathbb{R}^{n'}$, as given by Eq. (5.8).

From a Monte Carlo perspective, an event occurs for a given sample when the navigation states $\hat{\boldsymbol{x}}(t_{\mathrm{e}})$ at the time t_{e} satisfies the condition in Eq. (5.15). At the nominal time $\bar{t}_{\mathrm{e}}$ of the event, the true states $\boldsymbol{x}(\bar{t}_{\mathrm{e}})$ and navigation states $\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})$ for a given Monte Carlo sample can be respectively written as

$$\boldsymbol{x}(\bar{t}_{\mathrm{e}}) = \bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \delta\boldsymbol{x}(\bar{t}_{\mathrm{e}}) \tag{5.16}$$

and

$$\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) = \bar{\mathbf{x}}(\bar{t}_{\mathrm{e}}) + \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) = \boldsymbol{m}\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big) + \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \ . \tag{5.17}$$

The time derivative of the first expression is simply

$$\dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) = \dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}}) + \delta\dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \ , \tag{5.18}$$

whereas the time derivative of second expression will require a few more steps. First, the time derivative of the nonlinear term on the right-hand side of Eq. (5.17) can be expanded using the chain rule as follows

$$\frac{d\boldsymbol{m}\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big)}{dt} = \frac{\partial\boldsymbol{m}\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big)}{\partial\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})} \frac{d\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})}{dt} = \frac{\partial\boldsymbol{m}\big(\boldsymbol{x}(t_{\mathrm{e}})\big)}{\partial\boldsymbol{x}(t_{\mathrm{e}})}\bigg|_{\bar{\boldsymbol{x}}} \dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}}) = \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}}) \ .$$

From this expression and Eq. (5.17), it follows that the time derivative of the navigation state can now be written as

$$\dot{\hat{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}}) = \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}}) + \delta\dot{\hat{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\;. \tag{5.19}$$

Moreover, at the sample time t_{e} of the event, the true states and navigation states for a given Monte Carlo sample can be respectively approximated to first order as

$$\boldsymbol{x}(t_{\mathrm{e}}) \cong \boldsymbol{x}(\bar{t}_{\mathrm{e}}) + \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\,[t_{\mathrm{e}} - \bar{t}_{\mathrm{e}}] = \boldsymbol{x}(\bar{t}_{\mathrm{e}}) + \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\;, \tag{5.20}$$

and

$$\hat{\boldsymbol{x}}(t_{\mathrm{e}}) \cong \hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \dot{\hat{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,[t_{\mathrm{e}} - \bar{t}_{\mathrm{e}}] = \hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \dot{\hat{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\;, \tag{5.21}$$

where $\delta t_{\mathrm{e}} = t_{\mathrm{e}} - \bar{t}_{\mathrm{e}}$ represents the event time dispersion. Next, Eqs. (5.16) and (5.18) can be substituted into the first expression to yield

$$\boldsymbol{x}(t_{\mathrm{e}}) \cong \bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \delta\boldsymbol{x}(\bar{t}_{\mathrm{e}}) + \dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}} + \overbrace{\delta\dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}}^{\text{2nd order}}\;.$$

Thus to first order, the true states $\boldsymbol{x}(t_{\mathrm{e}})$ for a given Monte Carlo sample at the sample time t_{e} of the event can be written as

$$\boldsymbol{x}(t_{\mathrm{e}}) = \bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \delta\boldsymbol{x}(\bar{t}_{\mathrm{e}}) + \dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\;. \tag{5.22}$$

After applying the same methodology to Eq. (5.21), it follows that to first order the navigation states $\hat{\boldsymbol{x}}(t_{\mathrm{e}})$ can be written as

$$\hat{\boldsymbol{x}}(t_{\mathrm{e}}) = \boldsymbol{m}\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big) + \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\;. \tag{5.23}$$

Substituting this last expression into the event trigger condition in Eq. (5.15) produces

$$\Psi\Big[\boldsymbol{m}\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big) + \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\Big] = 0\;. \tag{5.24}$$

The condition function on the left hand side of this expression can then be linearized about the nominal using a truncated first-order Taylor series

$$\Psi\left[m\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big)+\delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})+\boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\right]\cong\Psi\left[m\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big)\right]+\frac{\partial\Psi[\hat{\boldsymbol{x}}(t_{\mathrm{e}})]}{\partial\hat{\boldsymbol{x}}(t_{\mathrm{e}})}\bigg|_{\bar{\boldsymbol{x}}}\left[\delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})+\boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}\right].$$

Using this result in Eq. (5.24) and noting that by definition

$$\Psi\left[m\big(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\big)\right]=0\ , \tag{5.25}$$

it follows that to first order

$$\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\,\delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})+\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\,\boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\,\delta t_{\mathrm{e}}=0\ ,$$

where the Jacobian $\boldsymbol{\Psi}_{\hat{x}}\in\mathbb{R}^{1\times m}$ is defined as

$$\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})=\frac{\partial\Psi\left[\hat{\boldsymbol{x}}(t_{\mathrm{e}})\right]}{\partial\hat{\boldsymbol{x}}(t)}\bigg|_{\hat{\boldsymbol{x}}=\bar{\boldsymbol{x}}}. \tag{5.26}$$

Solving this expression for the event time dispersion δt_{e} yields the first expression of interest

$$\delta t_{\mathrm{e}}=-\left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\,\boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_{\mathrm{e}})\right]^{-1}\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\,\delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\ . \tag{5.27}$$

At first glance, it would appear that this equation is valid if and only if the quantity in the square brackets is invertible. A simple dimensional analysis using matrix multiplication, however, is sufficient to show that the quantity actually reduces to a scalar, which is always invertible under the condition that it is nonzero. Eq. (5.27) is important because it represents to first order the difference between the time that the event occurs in the nominal reference trajectory and the time that the event occurs for a given Monte Carlo sample.

The event time dispersion δt_{e} given Eq. (5.27) can also be written in terms of of the augmented states $\mathbf{x}_{\mathrm{a}}$ at the nominal time $\bar{t}_{\mathrm{e}}$ of the event as follows

$$
\begin{aligned}
\delta t_{\mathrm{e}} &= \mathbf{0}_{1 \times n}\, \delta \boldsymbol{x}(\bar{t}_{\mathrm{e}}) + \left\{-\left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\right\} \delta \hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \\
&= \left[\begin{array}{cc} \mathbf{0}_{1 \times n} & -\left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}}) \end{array}\right] \left[\begin{array}{c} \delta \boldsymbol{x}(\bar{t}_{\mathrm{e}}) \\ \delta \hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \end{array}\right] \\
&= \boldsymbol{A}(\bar{t}_{\mathrm{e}})\, \mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})\ .
\end{aligned}
$$

The variance of the event time $\sigma_{t_{\mathrm{e}}}^2$ is determined by squaring the previous expression and taking the expected value over all Monte Carlo samples at the nominal event time $\bar{t}_{\mathrm{e}}$ as follows

$$
\sigma_{t_{\mathrm{e}}}^2 = E\{\delta t_{\mathrm{e}}^2\} = E\left\{\boldsymbol{A}(\bar{t}_{\mathrm{e}})\, \mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})\, [\boldsymbol{A}(\bar{t}_{\mathrm{e}})\, \mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\right\} = E\left\{\boldsymbol{A}(\bar{t}_{\mathrm{e}})\, \mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})\, [\mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\, [\boldsymbol{A}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\right\}\ .
$$

Since $\boldsymbol{A}(\bar{t}_{\mathrm{e}})$ is a known nonstochastic quantity, it can be moved out of the expectation operator

$$
\sigma_{t_{\mathrm{e}}}^2 = \boldsymbol{A}(\bar{t}_{\mathrm{e}})\, E\left\{\mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})\, [\mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\right\}\, [\boldsymbol{A}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\ .
$$

Then, recognizing that $\boldsymbol{\mathcal{C}}_{\mathrm{a}}(\bar{t}_{\mathrm{e}}) = E\left\{\mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})\, [\mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\right\}$, it follows that the second expression of interest is given by

$$
\sigma_{t_{\mathrm{e}}}^2 = \boldsymbol{A}(\bar{t}_{\mathrm{e}})\, \boldsymbol{\mathcal{C}}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})\, [\boldsymbol{A}(\bar{t}_{\mathrm{e}})]^{\mathsf{T}}\ , \tag{5.28}
$$

where $\boldsymbol{\mathcal{C}}_{\mathrm{a}}(\bar{t}_{\mathrm{e}})$ is the covariance of the augmented states at the nominal event time $\bar{t}_{\mathrm{e}}$. Eq. (5.28) is important because it represents to first order the variance of the event time as a function of the augmented state covariance at the nominal event time along the nominal reference trajectory.

From a Monte Carlo perspective, the next step is to collect all of the true and navigation states at their respective event trigger times and evaluate the covariance of these states. Analytically, this is achieved by first substituting the event time dispersion in Eq. (5.27) into Eqs. (5.22)

and (5.23) to obtain the states at their respective trigger times,

$$\boldsymbol{x}(t_{\mathrm{e}}) = \bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) + \delta\boldsymbol{x}(\bar{t}_{\mathrm{e}}) - \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \ , \tag{5.29}$$

$$\hat{\boldsymbol{x}}(t_{\mathrm{e}}) = \boldsymbol{m}\!\left[\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right] + \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) - \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \ . \tag{5.30}$$

Since the mean values of the true dispersions $\delta\boldsymbol{x}(\bar{t}_{\mathrm{e}})$ and navigation dispersions $\delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})$ are both zero at the nominal event time $\bar{t}_{\mathrm{e}}$, the mean values of the states at the sample event time t_{e} are equal to the nominal states at the nominal event time,

$$E\{\boldsymbol{x}(t_{\mathrm{e}})\} = E\{\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\} \ , \tag{5.31}$$

$$E\{\hat{\boldsymbol{x}}(t_{\mathrm{e}})\} = E\{\bar{\mathbf{x}}(\bar{t}_{\mathrm{e}})\} = E\{\boldsymbol{m}\!\left(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right)\} \ . \tag{5.32}$$

This is important because it allows the covariance of the states at the trigger times to be computed analytically. Next, the augmented states $\mathbf{x}_{\mathrm{a}}(t_{\mathrm{e}})$ at the sample event time t_{e} can be written as follows

$$\mathbf{x}_{\mathrm{a}}(t_{\mathrm{e}}) = \begin{bmatrix} \delta\boldsymbol{x}(t_{\mathrm{e}}) \\ \delta\hat{\boldsymbol{x}}(t_{\mathrm{e}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(t_{\mathrm{e}}) - \bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \\ \hat{\boldsymbol{x}}(t_{\mathrm{e}}) - \boldsymbol{m}\!\left(\bar{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right) \end{bmatrix}$$

$$= \begin{bmatrix} \delta\boldsymbol{x}(\bar{t}_{\mathrm{e}}) - \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \\ \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) - \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \delta\hat{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \end{bmatrix} \ ,$$

which can then be expressed in the traditional state-space form

$$\mathbf{x}_{\mathrm{a}}(t_{\mathrm{e}}) = \boldsymbol{\Phi}_{\mathrm{s}}(\bar{t}_{\mathrm{e}})\, \mathbf{x}_{\mathrm{a}}(\bar{t}_{\mathrm{e}}) \ , \tag{5.33}$$

where $\boldsymbol{\Phi}_{\mathrm{s}}(\bar{t}_{\mathrm{e}})$ is the *covariance shaping matrix*,

$$\boldsymbol{\Phi}_{\mathrm{s}}(\bar{t}_{\mathrm{e}}) = \begin{bmatrix} \boldsymbol{I}_{n\times n} & -\dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}}) \\ \boldsymbol{0}_{m\times n} & \boldsymbol{I}_{m\times m} - \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}}) \left[\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}})\, \boldsymbol{M}_x(\bar{t}_{\mathrm{e}})\, \dot{\boldsymbol{x}}(\bar{t}_{\mathrm{e}})\right]^{-1} \boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{\mathrm{e}}) \end{bmatrix} \ . \tag{5.34}$$

Since the mean value of the augmented states $\mathbf{x}_{\mathrm{a}}(t_{\mathrm{e}})$ by definition is zero, it follows that the third and final expression of interest is the covariance of the augmented states immediately after the

nominal event time $\bar{t}_\mathrm{e}$, which is given by

$$\mathcal{C}_\mathrm{a}^{+s}(\bar{t}_\mathrm{e}) = \boldsymbol{\Phi}_\mathrm{s}(\bar{t}_\mathrm{e})\, \mathcal{C}_\mathrm{a}^{-s}(\bar{t}_\mathrm{e})\, \left[\boldsymbol{\Phi}_\mathrm{s}(\bar{t}_\mathrm{e})\right]^\mathsf{T} , \tag{5.35}$$

where the superscripts $^{-s}$ and $^{+s}$ on the augmented state covariance matrix denote *before* and *after* *shaping*, respectively. Eq. (5.35) is important because application of the covariance shaping matrix allows the linear covariance simulation to capture the effects of the time-varying discrete event. From a Monte Carlo perspective, it is equivalent to resetting the simulation clock to a common time $\bar{t}_\mathrm{e}$ for each sample as the event is triggered such that all subsequent results are then measured with respect to the common time $\bar{t}_\mathrm{e}$.

5.5 Covariance Equations

By definition, the augmented state covariance matrix $\mathcal{C}_\mathrm{a} \in \mathbb{R}^{\ell \times \ell}$ is given by

$$\mathcal{C}_\mathrm{a}(t) = E\left\{\mathbf{x}_\mathrm{a}(t) \left[\mathbf{x}_\mathrm{a}(t)\right]^\mathsf{T}\right\} . \tag{5.36}$$

Substituting Eq. (5.3) into the above expression, it is straightforward to show that the augmented state covariance matrix can be written in the following block-partition form,

$$\mathcal{C}_\mathrm{a}(t) = \begin{bmatrix} \boldsymbol{\mathcal{D}}_{xx}(t) & \boldsymbol{\mathcal{D}}_{x\hat{x}}(t) \\ \left[\boldsymbol{\mathcal{D}}_{x\hat{x}}(t)\right]^\mathsf{T} & \boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}}(t) \end{bmatrix} , \tag{5.37}$$

where the covariance of the true state dispersions $\boldsymbol{\mathcal{D}}_{xx}(t) \in \mathbb{R}^{n \times n}$ is given by

$$\boldsymbol{\mathcal{D}}_{xx}(t) = E\left\{\delta\boldsymbol{x}(t) \left[\delta\boldsymbol{x}(t)\right]^\mathsf{T}\right\} , \tag{5.38}$$

the covariance of the navigation state dispersions $\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}} \in \mathbb{R}^{m \times m}$ is given by

$$\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}}(t) = E\left\{\delta\hat{\boldsymbol{x}}(t) \left[\delta\hat{\boldsymbol{x}}(t)\right]^\mathsf{T}\right\} , \tag{5.39}$$

and the cross correlation $\boldsymbol{\mathcal{D}}_{x\hat{x}} \in \mathbb{R}^{n \times m}$ is given by

$$\boldsymbol{\mathcal{D}}_{x\hat{x}}(t) = E\left\{\delta\boldsymbol{x}(t) \left[\delta\hat{\boldsymbol{x}}(t)\right]^\mathsf{T}\right\} . \tag{5.40}$$

By definition, the true navigation state error covariance matrix $\mathcal{P} \in \mathbb{R}^{m \times m}$ is given by

$$\mathcal{P}(t) = E\left\{\delta e(t) \left[\delta e(t)\right]^{\mathsf{T}}\right\} . \tag{5.41}$$

From Eqs. (5.37)–(5.39) it follows that the covariance matrices of the true and navigation state dispersions can also be respectively written as

$$\mathcal{D}_{xx}(t) = \left[\begin{array}{cc} I_{n \times n} & 0_{n \times m} \end{array}\right] \mathcal{C}_{\mathrm{a}}(t) \left[\begin{array}{cc} I_{n \times n} & 0_{n \times m} \end{array}\right]^{\mathsf{T}} \tag{5.42}$$

and

$$\mathcal{D}_{\hat{x}\hat{x}}(t) = \left[\begin{array}{cc} 0_{m \times n} & I_{m \times m} \end{array}\right] \mathcal{C}_{\mathrm{a}}(t) \left[\begin{array}{cc} 0_{m \times n} & I_{m \times m} \end{array}\right]^{\mathsf{T}} . \tag{5.43}$$

Moreover, from Eqs. (5.41) and (5.14) it follows that the covariance matrix of the true navigation state errors can written as

$$\mathcal{P}(t) = \left[\begin{array}{cc} M_x(t) & -I_{m \times m} \end{array}\right] \mathcal{C}_{\mathrm{a}}(t) \left[\begin{array}{cc} M_x(t) & -I_{m \times m} \end{array}\right]^{\mathsf{T}} . \tag{5.44}$$

Hence, given the augmented state covariance matrix $\mathcal{C}_{\mathrm{a}}(t)$, the covariance matrices of the true state dispersions, navigation state dispersions, and true navigation state errors can be respectively computed with these last three expressions.

5.5.1 Covariance Initialization

This section describes the methodology used to initialize the augmented state covariance matrix $\mathcal{C}_{\mathrm{a}}$ and the filter navigation state error covariance matrix $\hat{\mathcal{P}}$, where it is assumed that the modified true state vector $x_{\mathrm{m}} \in \mathbb{R}^n$ and modified navigation state vector $\hat{x}_{\mathrm{m}} \in \mathbb{R}^m$ are respectively given by

$$x_{\mathrm{m}} = \left(r^i, \ v^i, \ \theta_i^b, \ \omega_{ib}^b, \ m_b, \ p, \ \hat{\theta}_{i,\mathrm{com}}^{\hat{b}}\right) \tag{5.45}$$

and

$$\hat{x}_{\mathrm{m}} = \left(\hat{r}^i, \ \hat{v}^i, \ \hat{\theta}_i^b, \ \hat{p}\right) . \tag{5.46}$$

It is important to point out that, other than the pseudo-state $\hat{\theta}_{i,\mathrm{com}}^b$, the modified navigation state vector $\hat{x}_{\mathrm{m}}$ contains only a subset of the true states for two reasons. First, it is assumed that an

IMU is present, which provides nearly continuous measurements of the specific force and angular velocity. As such, the body angular velocity vector $\hat{\boldsymbol{\omega}}_{ib}^b$ is not estimated and therefore not included in the modified navigation state vector. Second, the wet mass is assumed to be a known quantity (i.e., as a function of time), thus it is not estimated nor included in the modified navigation state vector.

True State Dispersions

The covariance matrix of the true state dispersions $\boldsymbol{\mathcal{D}}_{xx} \in \mathbb{R}^{n \times n}$ at time t_0 can be written as

$$
\boldsymbol{\mathcal{D}}_{xx}(t_0) =
\begin{bmatrix}
\boldsymbol{\mathcal{D}}_{rr}^i(t_0) & \boldsymbol{\mathcal{D}}_{rv}^i(t_0) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times q} & \mathbf{0}_{3\times3} \\
\left[\boldsymbol{\mathcal{D}}_{rv}^i(t_0)\right]^{\mathsf{T}} & \boldsymbol{\mathcal{D}}_{vv}^i(t_0) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times q} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{\mathcal{D}}_{\theta\theta}^b(t_0) & \boldsymbol{\mathcal{D}}_{\theta\omega}^b(t_0) & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times q} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \left[\boldsymbol{\mathcal{D}}_{\theta\omega}^b(t_0)\right]^{\mathsf{T}} & \boldsymbol{\mathcal{D}}_{\omega\omega}^b(t_0) & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times q} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \sigma_{m_b}^2 & \mathbf{0}_{1\times q} & \mathbf{0}_{1\times3} \\
\mathbf{0}_{q\times3} & \mathbf{0}_{q\times3} & \mathbf{0}_{q\times3} & \mathbf{0}_{q\times3} & \mathbf{0}_{q\times1} & \boldsymbol{\mathcal{D}}_{pp}(t_0) & \mathbf{0}_{q\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times q} & \mathbf{0}_{3\times3}
\end{bmatrix} ,
\qquad (5.47)
$$

where the elements along the main diagonal are defined as

$$
E\left\{\delta \boldsymbol{r}^i(t_0)\left[\delta \boldsymbol{r}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{rr}^i(t_0) , \qquad E\left\{\delta \boldsymbol{v}^i(t_0)\left[\delta \boldsymbol{v}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{vv}^i(t_0) , \qquad (5.48)
$$

$$
E\left\{\delta \boldsymbol{\theta}_i^b(t_0)\left[\delta \boldsymbol{\theta}_i^b(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{\theta\theta}^b(t_0) , \qquad E\left\{\delta \boldsymbol{\omega}_{ib}^b(t_0)\left[\delta \boldsymbol{\omega}_{ib}^b(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{\omega\omega}^b(t_0) , \qquad (5.49)
$$

$$
E\left\{\delta m_b(t_0)\,\delta m_b(t_0)\right\} = \sigma_{m_b}^2(t_0) , \qquad E\left\{\delta \boldsymbol{p}(t_0)\left[\delta \boldsymbol{p}(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{pp}(t_0) , \qquad (5.50)
$$

and

$$
E\left\{\delta \hat{\boldsymbol{\theta}}_{i,\text{com}}^b(t_0)\left[\delta \hat{\boldsymbol{\theta}}_{i,\text{com}}^b(t_0)\right]^{\mathsf{T}}\right\} = \mathbf{0}_{3\times3} . \qquad (5.51)
$$

Moreover, it is assumed that, with a few notable exceptions, all of the true states are initially uncorrelated, as denoted in Eq. (5.47) with a light-gray color. Due to the nature of the problem, it will be shown that the inertial position and velocity states are initially correlated, as well as the

body attitude and angular velocity states. Hence, the nonzero off-diagonal elements are defined as

$$E\left\{\delta\boldsymbol{r}^i(t_0)\left[\delta\boldsymbol{v}^i(t_0)\right]^\mathsf{T}\right\} = \boldsymbol{\mathcal{D}}_{rv}^i(t_0) \tag{5.52}$$

and

$$E\left\{\delta\boldsymbol{\theta}_i^b(t_0)\left[\delta\boldsymbol{\omega}_{ib}^b(t_0)\right]^\mathsf{T}\right\} = \boldsymbol{\mathcal{D}}_{\theta\omega}^b(t_0) \ . \tag{5.53}$$

For a launch vehicle sitting on the launch pad, it is more intuitive to define the covariance matrix of the inertial position true dispersions at time t_0 in terms of the planetographic coordinate system (i.e., north-east-down) [see § A.2.4] and then transform it to the inertial coordinate system. As such, the initial covariance matrix of the position true state dispersions, assuming that there is no initial correlation between the three position components, is a diagonal matrix $\boldsymbol{\mathcal{D}}_{rr}^g \in \mathbb{R}^{3\times3}$ with the *north*, *east*, and *down* variances $\sigma_{r_\mathrm{N}}^2$, $\sigma_{r_\mathrm{E}}^2$, and $\sigma_{r_\mathrm{D}}^2$, respectively, along the main diagonal,

$$\boldsymbol{\mathcal{D}}_{rr}^g(t_0) = \begin{bmatrix} \sigma_{r_\mathrm{N}}^2 & 0 & 0 \\ 0 & \sigma_{r_\mathrm{E}}^2 & 0 \\ 0 & 0 & \sigma_{r_\mathrm{D}}^2 \end{bmatrix} \ . \tag{5.54}$$

Then, it can be shown that the initial covariance matrix of the inertial position true state dispersions is given by

$$\boldsymbol{\mathcal{D}}_{rr}^i(t_0) = \boldsymbol{T}_g^i(t_0)\,\boldsymbol{\mathcal{D}}_{rr}^g(t_0)\left[\boldsymbol{T}_g^i(t_0)\right]^\mathsf{T} \ , \tag{5.55}$$

where $\boldsymbol{T}_g^i$ is the planetographic-to-inertial transformation matrix at time t_0. Note that $\boldsymbol{\mathcal{D}}_{rr}^i(t_0)$ is a nondiagonal matrix. This arises from the fact that the planetographic and inertial coordinate frames are generally not coincident, i.e., mathematically $\boldsymbol{T}_g^i \neq \boldsymbol{I}_{3\times3}$.

Since the inertial position of the launch vehicle at time t_0 is fixed with respect to the planet, it follows that the initial inertial velocity vector $\boldsymbol{v}^i(t_0)$ is given by

$$\boldsymbol{v}^i(t_0) = \boldsymbol{\omega}_{ip}^i \times \boldsymbol{r}^i(t_0) \ , \tag{5.56}$$

where $\boldsymbol{\omega}_{ip}^i$ is the angular velocity vector of the planet, expressed in inertial coordinates. With this linear expression and Eqs. (5.48) and (5.52), it is then straightforward to show that the initial

covariance matrix of the inertial velocity true state dispersions can be written as

$$\mathcal{D}_{vv}^i(t_0) = \left[\boldsymbol{\omega}_{ip}^i\times\right] \mathcal{D}_{rr}^i(t_0) \left[\boldsymbol{\omega}_{ip}^i\times\right]^\mathsf{T} , \qquad (5.57)$$

and the initial cross correlation matrix of the inertial position and velocity true state dispersions can be written as

$$\mathcal{D}_{rv}^i(t_0) = \mathcal{D}_{rr}^i(t_0) \left[\boldsymbol{\omega}_{ip}^i\times\right]^\mathsf{T} , \qquad (5.58)$$

where $\left[\boldsymbol{\omega}_{ip}^i\times\right]$ is the cross-product form of the planet's inertial angular velocity vector.

The covariance matrix of the body attitude true state dispersions at time t_0, assuming that there is no initial correlation between the three attitude components, is a diagonal matrix $\mathcal{D}_{\theta\theta}^b \in \mathbb{R}^{3\times3}$ with the body x, y, and z variances $\sigma_{\theta_x}^2$, $\sigma_{\theta_y}^2$, and $\sigma_{\theta_z}^2$, respectively, along the main diagonal. Since the attitude of the launch vehicle is also initially fixed with respect to the planet, it follows that the body angular velocity vector $\boldsymbol{\omega}_{ib}^b$ at time t_0 is given by

$$\boldsymbol{\omega}_{ib}^b(t_0) = \boldsymbol{T}_i^b(t_0)\,\boldsymbol{\omega}_{ip}^i , \qquad (5.59)$$

where $\boldsymbol{T}_i^b = \boldsymbol{T}(\boldsymbol{q}_i^b)$ is the initial inertial-to-body transformation matrix and a nonlinear function of the inertial-to-body attitude quaternion at time t_0. This expression is then linearized about the nominal to yield the following linear expression for the initial body angular velocity true dispersions,

$$\delta\boldsymbol{\omega}_{ib}^b(t_0) = \left[\bar{\boldsymbol{T}}_i^b(t_0)\,\boldsymbol{\omega}_{ip}^i\times\right] \delta\boldsymbol{\theta}_i^b(t_0) , \qquad (5.60)$$

where $\bar{\boldsymbol{T}}_i^b(t_0)$ is the *nominal* initial inertial-to-body transformation matrix and $\delta\boldsymbol{\theta}_i^b(t_0)$ is the body attitude true state dispersions at time t_0. With this expression and Eqs. (5.49) and (5.53), it can be shown that the initial covariance matrix of the body angular velocity true state dispersions can be written as

$$\mathcal{D}_{\omega\omega}^b(t_0) = \left[\bar{\boldsymbol{T}}_i^b(t_0)\,\boldsymbol{\omega}_{ip}^i\times\right] \mathcal{D}_{\theta\theta}^b(t_0) \left[\bar{\boldsymbol{T}}_i^b(t_0)\,\boldsymbol{\omega}_{ip}^i\times\right]^\mathsf{T} , \qquad (5.61)$$

and the initial cross correlation matrix of the body attitude and angular velocity true state dispersions can be written as

$$\mathcal{D}_{\theta\omega}^b(t_0) = \mathcal{D}_{\theta\theta}^b(t_0) \left[\bar{\boldsymbol{T}}_i^b(t_0)\,\boldsymbol{\omega}_{ip}^i\times\right]^\mathsf{T} . \qquad (5.62)$$

The initial variance of the wet mass true state dispersions is $\sigma_{m_b}^2$ and the initial covariance matrix of the error parameter true state dispersions is a diagonal matrix $\boldsymbol{\mathcal{D}}_{pp} \in \mathbb{R}^{q \times q}$ with the error parameter variances $\sigma_{p_k}^2$ $(k = 1, 2, \ldots, q)$ along the main diagonal. Lastly, the initial covariance matrix of the pseudo-state dispersions is simply the 3×3 null matrix, i.e., $\mathbf{0}_{3 \times 3}$.

Navigation State Errors

The initial covariance matrix of the true navigation state errors $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{m \times m}$ can be written as

$$\boldsymbol{\mathcal{P}}(t_0) = \begin{bmatrix} \boldsymbol{\mathcal{P}}_{\mathsf{rr}}^i(t_0) & \boldsymbol{\mathcal{P}}_{\mathsf{rv}}^i(t_0) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times o} \\ \left[\boldsymbol{\mathcal{P}}_{\mathsf{rv}}^i(t_0)\right]^{\mathsf{T}} & \boldsymbol{\mathcal{P}}_{\mathsf{vv}}^i(t_0) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times o} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \boldsymbol{\mathcal{P}}_{\theta\theta}^b(t_0) & \mathbf{0}_{3 \times o} \\ \mathbf{0}_{o \times 3} & \mathbf{0}_{o \times 3} & \mathbf{0}_{o \times 3} & \boldsymbol{\mathcal{P}}_{\mathsf{pp}}(t_0) \end{bmatrix}, \tag{5.63}$$

where the elements along the main diagonal are defined as

$$E\left\{\delta \boldsymbol{e}_{\mathsf{r}}^i(t_0) \left[\delta \boldsymbol{e}_{\mathsf{r}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{P}}_{\mathsf{rr}}^i(t_0) , \qquad E\left\{\delta \boldsymbol{e}_{\mathsf{v}}^i(t_0) \left[\delta \boldsymbol{e}_{\mathsf{v}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{P}}_{\mathsf{vv}}^i(t_0) , \tag{5.64}$$

$$E\left\{\delta \boldsymbol{e}_{\theta}^b(t_0) \left[\delta \boldsymbol{e}_{\theta}^b(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{P}}_{\theta\theta}^b(t_0) , \qquad E\left\{\delta \boldsymbol{e}_{\mathsf{p}}(t_0) \left[\delta \boldsymbol{e}_{\mathsf{p}}(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{P}}_{\mathsf{pp}}(t_0) . \tag{5.65}$$

Moreover, it is assumed that, with the notable exception of the inertial position and velocity states, all of the true navigation states are initially uncorrelated. Hence, the only nonzero off-diagonal element is defined as

$$E\left\{\delta \boldsymbol{e}_{\mathsf{r}}^i(t_0) \left[\delta \boldsymbol{e}_{\mathsf{v}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{P}}_{\mathsf{rv}}^i(t_0) . \tag{5.66}$$

Like the inertial position true state dispersions, the covariance matrix of the inertial position true navigation state errors at time t_0 is defined in terms of the planetographic coordinate system, then transformed to the inertial coordinate system. As such, the initial covariance matrix of the position true navigation state errors, assuming that there is no initial correlation between the three position components, is a diagonal matrix $\boldsymbol{\mathcal{P}}_{\mathsf{rr}}^g \in \mathbb{R}^{3 \times 3}$ with the *north*, *east*, and *down* variances $\sigma_{\mathsf{r_N}}^2$, $\sigma_{\mathsf{r_E}}^2$, and $\sigma_{\mathsf{r_D}}^2$, respectively, along the main diagonal. It then follows that the initial covariance matrix of the inertial position true navigation state errors is given by

$$\boldsymbol{\mathcal{P}}_{\mathsf{rr}}^i(t_0) = \boldsymbol{T}_g^i(t_0) \, \boldsymbol{\mathcal{P}}_{\mathsf{rr}}^g \left[\boldsymbol{T}_g^i(t_0)\right]^{\mathsf{T}} . \tag{5.67}$$

The covariance matrix of the inertial velocity true navigation state errors at time t_0 is found in the same manner as that of the inertial velocity true state dispersions [see Eq. (5.57)] and therefore is given by

$$\boldsymbol{\mathcal{P}}_{\mathsf{vv}}^i(t_0) = \left[\boldsymbol{\omega}_{ip}^i\times\right] \boldsymbol{\mathcal{P}}_{\mathsf{rr}}^i(t_0) \left[\boldsymbol{\omega}_{ip}^i\times\right]^\mathsf{T} . \tag{5.68}$$

Moreover, it can be shown that the cross correlation matrix of the inertial position and velocity true navigation state errors is given by

$$\boldsymbol{\mathcal{P}}_{\mathsf{rv}}^i(t_0) = \boldsymbol{\mathcal{P}}_{\mathsf{rr}}^i(t_0) \left[\boldsymbol{\omega}_{ip}^i\times\right]^\mathsf{T} . \tag{5.69}$$

The covariance matrix of the body attitude true navigation state errors at time t_0, assuming that there is no initial correlation between the three attitude components, is a diagonal matrix $\boldsymbol{\mathcal{P}}_{\theta\theta}^b \in \mathbb{R}^{3\times 3}$ with the body x, y, and z variances $\sigma_{\theta_x}^2$, $\sigma_{\theta_y}^2$, and $\sigma_{\theta_z}^2$, respectively, along the main diagonal. In like manner, the initial covariance matrix of the navigation error parameter state errors is a diagonal matrix $\boldsymbol{\mathcal{P}}_{\mathsf{pp}} \in \mathbb{R}^{o\times o}$ with the error parameter variances $\sigma_{\mathsf{p}_k}^2$ $(k = 1, 2, \ldots, o)$ along the main diagonal. Lastly, the covariance matrix of the filter navigation state errors $\hat{\boldsymbol{\mathcal{P}}} \in \mathbb{R}^{m\times m}$ at time t_0 is assumed to be equal to that of the true navigation state error, i.e.,

$$\hat{\boldsymbol{\mathcal{P}}}(t_0) = \boldsymbol{\mathcal{P}}(t_0) ,$$

where $\boldsymbol{\mathcal{P}}(t_0)$ is defined in Eq. (5.63).

Navigation State Dispersions

The initial covariance matrix of the navigation state dispersions $\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}} \in \mathbb{R}^{m\times m}$ can be written as

$$\boldsymbol{\mathcal{D}}_{\hat{x}\hat{x}}(t_0) = \begin{bmatrix} \boldsymbol{\mathcal{D}}_{\hat{r}\hat{r}}^i(t_0) & \boldsymbol{\mathcal{D}}_{\hat{r}\hat{v}}^i(t_0) & 0_{3\times 3} & 0_{3\times o} \\ \left[\boldsymbol{\mathcal{D}}_{\hat{r}\hat{v}}^i(t_0)\right]^\mathsf{T} & \boldsymbol{\mathcal{D}}_{\hat{v}\hat{v}}^i(t_0) & 0_{3\times 3} & 0_{3\times o} \\ 0_{3\times 3} & 0_{3\times 3} & \boldsymbol{\mathcal{D}}_{\hat{\theta}\hat{\theta}}^b(t_0) & 0_{3\times o} \\ 0_{o\times 3} & 0_{o\times 3} & 0_{o\times 3} & \mathbf{0}_{o\times o} \end{bmatrix} , \tag{5.70}$$

where the elements along the main diagonal are defined as

$$E\left\{\delta\hat{\boldsymbol{r}}^i(t_0)\left[\delta\hat{\boldsymbol{r}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{\hat{r}\hat{r}}^i(t_0) \ , \qquad E\left\{\delta\hat{\boldsymbol{v}}^i(t_0)\left[\delta\hat{\boldsymbol{v}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{\hat{v}\hat{v}}^i(t_0) \ , \qquad (5.71)$$

$$E\left\{\delta\hat{\boldsymbol{\theta}}_i^b(t_0)\left[\delta\hat{\boldsymbol{\theta}}_i^b(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{\hat{\theta}\hat{\theta}}^b(t_0) \ , \qquad E\left\{\delta\hat{\boldsymbol{p}}(t_0)\left[\delta\hat{\boldsymbol{p}}(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{0}_{o\times o} \ . \qquad (5.72)$$

Moreover, it is assumed that, with the notable exception of the inertial position and velocity states, all of the navigation states are initially uncorrelated. Hence, the only nonzero off-diagonal element is defined as

$$E\left\{\delta\hat{\boldsymbol{r}}^i(t_0)\left[\delta\hat{\boldsymbol{v}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}_{\hat{r}\hat{v}}^b(t_0) \ . \qquad (5.73)$$

From the definition of the true navigation state error, it follows that the inertial position navigation state dispersion $\delta\hat{\boldsymbol{r}}^i$ can be written as the sum of the inertial position true state dispersion $\delta\boldsymbol{r}^i$ and the inertial position true navigation state error $\delta\boldsymbol{e}_{\mathsf{r}}^i$,

$$\delta\hat{\boldsymbol{r}}^i = \delta\boldsymbol{r}^i + \delta\boldsymbol{e}_{\mathsf{r}}^i \ . \qquad (5.74)$$

Using this expression and assuming that the inertial position true state dispersions and true navigation state errors are initially uncorrelated (i.e., $E\left\{\delta\boldsymbol{r}^i(t_0)\left[\delta\boldsymbol{e}_{\mathsf{r}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{0}_{3\times 3}$), it is straightforward to show that the covariance matrix of the inertial position navigation state dispersions at time t_0 is given by

$$\boldsymbol{\mathcal{D}}_{\hat{r}\hat{r}}^i(t_0) = \boldsymbol{\mathcal{D}}_{rr}^i(t_0) + \boldsymbol{\mathcal{P}}_{\mathsf{rr}}^i(t_0) \ . \qquad (5.75)$$

In like manner, it can be shown that the initial covariance matrices of the inertial velocity and body attitude navigation state dispersions are respectively given by

$$\boldsymbol{\mathcal{D}}_{\hat{v}\hat{v}}^i(t_0) = \boldsymbol{\mathcal{D}}_{vv}^i(t_0) + \boldsymbol{\mathcal{P}}_{\mathsf{vv}}^i(t_0) \ , \qquad (5.76)$$

$$\boldsymbol{\mathcal{D}}_{\hat{\theta}\hat{\theta}}^b(t_0) = \boldsymbol{\mathcal{D}}_{\theta\theta}^b(t_0) + \boldsymbol{\mathcal{P}}_{\theta\theta}^b(t_0) \ , \qquad (5.77)$$

and the initial cross correlation matrix of the inertial position and velocity navigation state dispersions can be written as

$$\boldsymbol{\mathcal{D}}^i_{\hat{r}\hat{v}}(t_0) = \left[\boldsymbol{\mathcal{D}}^i_{rr}(t_0) + \boldsymbol{\mathcal{P}}^i_{\mathsf{rr}}(t_0)\right]\left[\boldsymbol{\omega}^i_{ip}\times\right]^{\mathsf{T}},$$
$$= \boldsymbol{\mathcal{D}}^i_{rv}(t_0) + \boldsymbol{\mathcal{P}}^i_{\mathsf{rv}}(t_0) \ . \tag{5.78}$$

Augmented State Cross Correlations

The initial augmented state cross correlations matrix $\boldsymbol{\mathcal{D}}_{x\hat{x}} \in \mathbb{R}^{n\times m}$ can be written as

$$\boldsymbol{\mathcal{D}}_{x\hat{x}}(t_0) = \begin{bmatrix} \boldsymbol{\mathcal{D}}^i_{r\hat{r}}(t_0) & \boldsymbol{\mathcal{D}}^i_{r\hat{v}}(t_0) & 0_{3\times3} & 0_{3\times o} \\ \boldsymbol{\mathcal{D}}^i_{v\hat{r}}(t_0) & \boldsymbol{\mathcal{D}}^i_{v\hat{v}}(t_0) & 0_{3\times3} & 0_{3\times o} \\ 0_{3\times3} & 0_{3\times3} & \boldsymbol{\mathcal{D}}^b_{\theta\hat{\theta}}(t_0) & 0_{3\times o} \\ 0_{3\times3} & 0_{3\times3} & \boldsymbol{\mathcal{D}}^b_{\omega\hat{\theta}}(t_0) & 0_{3\times o} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times o} \\ 0_{q\times3} & 0_{q\times3} & 0_{q\times3} & 0_{q\times o} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times o} \end{bmatrix} , \tag{5.79}$$

where the nonzero elements are defined as

$$E\left\{\delta\boldsymbol{r}^i(t_0)\left[\delta\hat{\boldsymbol{r}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}^i_{r\hat{r}}(t_0) , \qquad E\left\{\delta\boldsymbol{v}^i(t_0)\left[\delta\hat{\boldsymbol{v}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}^i_{v\hat{v}}(t_0) , \tag{5.80}$$

$$E\left\{\delta\boldsymbol{r}^i(t_0)\left[\delta\hat{\boldsymbol{v}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}^i_{r\hat{v}}(t_0) , \qquad E\left\{\delta\boldsymbol{v}^i(t_0)\left[\delta\hat{\boldsymbol{r}}^i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}^i_{v\hat{r}}(t_0) , \tag{5.81}$$

$$E\left\{\delta\boldsymbol{\theta}^b_i(t_0)\left[\delta\hat{\boldsymbol{\theta}}^b_i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}^b_{\theta\hat{\theta}}(t_0) , \qquad E\left\{\delta\boldsymbol{\omega}^b_{ib}(t_0)\left[\delta\hat{\boldsymbol{\theta}}^b_i(t_0)\right]^{\mathsf{T}}\right\} = \boldsymbol{\mathcal{D}}^b_{\omega\hat{\theta}}(t_0) . \tag{5.82}$$

It is straightforward to show that the cross correlation matrix of the inertial position true and navigation state dispersions at time t_0 is simply equal to the initial covariance matrix of the inertial position true state dispersions, i.e.,

$$\boldsymbol{\mathcal{D}}^i_{r\hat{r}}(t_0) = \boldsymbol{\mathcal{D}}^i_{rr}(t_0) \ . \tag{5.83}$$

Likewise, the initial cross correlation matrix of the inertial velocity true and navigation state dispersions is given by

$$\boldsymbol{\mathcal{D}}^i_{v\hat{v}}(t_0) = \boldsymbol{\mathcal{D}}^i_{vv}(t_0) \ . \tag{5.84}$$

Next, it can be shown that the initial cross correlation matrix of the inertial position true state dispersions and the inertial velocity navigation state dispersions is given by

$$\boldsymbol{\mathcal{D}}_{r\hat{v}}^{i}(t_0) = \boldsymbol{\mathcal{D}}_{rr}^{i}(t_0) \left[\boldsymbol{\omega}_{ip}^{i}\times\right]^{\mathsf{T}} , \tag{5.85}$$

and the initial cross correlation matrix of the inertial velocity true state dispersions and the inertial position navigation state dispersions is given by

$$\boldsymbol{\mathcal{D}}_{v\hat{r}}^{i}(t_0) = \left[\boldsymbol{\omega}_{ip}^{i}\times\right] \boldsymbol{\mathcal{D}}_{rr}^{i}(t_0) . \tag{5.86}$$

Lastly, it follows that the cross correlation matrix of the body attitude true and navigation state dispersions at time t_0 is given by

$$\boldsymbol{\mathcal{D}}_{\theta\hat{\theta}}^{b}(t_0) = \boldsymbol{\mathcal{D}}_{\theta\theta}^{b}(t_0) , \tag{5.87}$$

and the cross correlation matrix of the body angular velocity true state dispersions and the body attitude navigation state dispersions at time t_0 is given by

$$\boldsymbol{\mathcal{D}}_{\omega\hat{\theta}}^{b}(t_0) = \left[\bar{\boldsymbol{T}}_{i}^{b}(t_0)\,\boldsymbol{\omega}_{ip}^{i}\times\right] \boldsymbol{\mathcal{D}}_{\theta\theta}^{b}(t_0) . \tag{5.88}$$

Due to the nature of the problem, the true and navigation state dispersions are highly correlated, which is illustrated by the fact that there are nonzero elements in the augmented state cross correlation matrix defined in Eq (5.79). Moreover, the inertial velocity true and navigation state dispersions are highly correlated with the inertial position true and navigation state dispersions, and the body angular velocity true state dispersions are highly correlated with the body attitude true state dispersions, which is evidenced by the nonzero off-diagonal elements in the initial true state dispersion, navigation state dispersion, and true navigation state error covariance matrices defined in Eqs. (5.47), (5.70), and (5.63), respectively.

When states are highly correlated, i.e., the correlation coefficients are close to ± 1, it has been observed that the covariance matrix will generally contain small negative eigenvalues, so that numerically it is no longer positive-semi-definite (PSD). This can cause numerical instabilities as the covariance matrix is propagated [83] in the linear covariance simulation and will undoubtedly cause problems in Monte Carlo simulation initialization routine because the states of the system

are initialized with the associated eigenvalues and eigenvectors of the initial covariance matrix. If the states are highly correlated only at time t_0, then one way to ensure that the initial covariance matrix is PSD is to add some additional variance to the diagonal elements of the correlated states. For example, if the inertial velocity state is highly correlated with the inertial position state, then one would simply add some additional variance to the diagonal elements of the inertial velocity state covariance until the negative eigenvalues disappear. There is, however, no general rule of thumb for the amount of additional variance to add. It follows naturally though that it should be large enough so that the covariance matrix is PSD, yet small enough that the modified values of the state variance still represent the original values.

5.5.2 Covariance Propagation

From linear system theory, it follows that the solution to Eq. (5.4) is given by [7]

$$\mathbf{x}_a(t) = \mathbf{\Phi}_a(t, t_0)\, \mathbf{x}_a(t_0) + \int_{t_0}^{t} \mathbf{\Phi}_a(t, \tau)\, \mathbf{G}(\tau)\, \boldsymbol{\eta}(\tau)\, d\tau + \int_{t_0}^{t} \mathbf{\Phi}_a(t, \tau)\, \mathbf{W}(\tau)\, \boldsymbol{w}(\tau)\, d\tau \ , \qquad (5.89)$$

where the augmented state transition matrix $\mathbf{\Phi}_a(t, t_0) \in \mathbb{R}^{\ell \times \ell}$ is the unique solution to the matrix differential equation

$$\dot{\mathbf{\Phi}}_a(t, t_0) = \mathbf{F}(t)\, \mathbf{\Phi}_a(t, t_0) \ , \qquad (5.90)$$

with the initial condition $\mathbf{\Phi}_a(t_0, t_0) = \boldsymbol{I}_{\ell \times \ell}$. Using Eq. (5.89) in the definition of the augmented state dispersion covariance matrix given in Eq. (5.36) and utilizing the covariance of the process noise $\boldsymbol{S}_w$ and measurement noise $\boldsymbol{S}_\eta$ respectively defined in Eqs. (3.35) and (3.37) to replace terms, it follows that the *augmented state covariance propagation equation* is given by

$$\boldsymbol{\mathcal{C}}_a(t) = \mathbf{\Phi}_a(t, t_0)\, \boldsymbol{\mathcal{C}}_a(t_0)\, \big[\mathbf{\Phi}_a(t, t_0)\big]^{\mathsf{T}} + \boldsymbol{Q}_w(t) + \boldsymbol{Q}_\eta(t) \ , \qquad (5.91)$$

where

$$\boldsymbol{Q}_w(t) = \int_{t_0}^{t} \mathbf{\Phi}_a(t, \tau)\, \mathbf{W}(\tau)\, \boldsymbol{S}_w(\tau)\, \big[\mathbf{W}(\tau)\big]^{\mathsf{T}} \big[\mathbf{\Phi}_a(t, \tau)\big]^{\mathsf{T}} d\tau \ . \qquad (5.92)$$

and

$$\boldsymbol{Q}_\eta(t) = \int_{t_0}^{t} \mathbf{\Phi}_a(t, \tau)\, \mathbf{G}(\tau)\, \boldsymbol{S}_\eta(\tau)\, \big[\mathbf{G}(\tau)\big]^{\mathsf{T}} \big[\mathbf{\Phi}_a(t, \tau)\big]^{\mathsf{T}} d\tau \ . \qquad (5.93)$$

In lieu of having to numerically integrate Eq. (5.90) and solve the integrals in Eqs. (5.92) and (5.93),

it is assumed that the simulation time step $dt = t - t_0$ is 'small' enough such that the following are valid approximations,

$$\boldsymbol{\Phi}_{\mathrm{a}}(t, t_0) \cong \boldsymbol{I}_{\ell \times \ell} + \mathbf{F}(t)\, dt + \frac{\left[\mathbf{F}(t)\, dt\right]^2}{2!} + \frac{\left[\mathbf{F}(t)\, dt\right]^3}{3!} + \cdots + \frac{\left[\mathbf{F}(t)\, dt\right]^n}{n!} \ , \tag{5.94}$$

$$\boldsymbol{Q}_w(t) \cong \boldsymbol{\Phi}_{\mathrm{a}}(t, t)\, \mathbf{W}(t)\, \boldsymbol{S}_w(t) \left[\mathbf{W}(t)\right]^\mathsf{T} \left[\boldsymbol{\Phi}_{\mathrm{a}}(t, t)\right]^\mathsf{T} dt = \mathbf{W}(t)\, \boldsymbol{S}_w(t) \left[\mathbf{W}(t)\right]^\mathsf{T} dt \ , \tag{5.95}$$

$$\boldsymbol{Q}_\eta(t) \cong \boldsymbol{\Phi}_{\mathrm{a}}(t, t)\, \mathbf{G}(t)\, \boldsymbol{S}_\eta(t) \left[\mathbf{G}(t)\right]^\mathsf{T} \left[\boldsymbol{\Phi}_{\mathrm{a}}(t, t)\right]^\mathsf{T} dt = \mathbf{G}(t)\, \boldsymbol{S}_\eta(t) \left[\mathbf{G}(t)\right]^\mathsf{T} dt \ . \tag{5.96}$$

Therefore, for this research, these approximations are used in conjunction with Eq. (5.91) to propagate the augmented state covariance matrix $\mathcal{C}_{\mathrm{a}}$. The filter navigation state error covariance $\hat{\mathcal{P}}$ is propagated in a similar manner with Eqs. (3.89)–(3.94) on page 45 and F_x defined in Eq. (4.61) on page 69.

5.5.3 Covariance Correction

In general, discrete instantaneous state covariance corrections are associated with and used to account for impulsive translational or rotational maneuvers. In this research, there are no impulsive translational or rotational maneuvers. However, in order to resolve the fine-count partial-derivative dilemma encountered and discussed in the previous chapter [see § 4.4.3], the pseudo-state $\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}$ (i.e., the Euler rotation vector of the inertial-to-desired-body attitude quaternion $\hat{\boldsymbol{q}}_{i,\mathrm{com}}^{\hat{b}}$ command issued by guidance) was introduced and appended to the true and modified true state vectors. Then, every time PEG is called the issued attitude command is applied as an instantaneous augmented state covariance correction, and when guidance modes to fine count (and PEG is no longer called) no more covariance corrections are made, thereby properly capturing the effects of the constant attitude command on the covariance of the augmented state while eliminating the need for the partial derivatives of the attitude command during fine count.

The general form of a discrete instantaneous state correction at time t_j is written as

$$\boldsymbol{x}^{+c}[t_j] = \boldsymbol{x}^{-c}[t_j] + \boldsymbol{d}(\boldsymbol{x}^{-c}[t_j], t_j) \ , \tag{5.97}$$

where the correction $\boldsymbol{d} : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function of the state $\boldsymbol{x}^{-c}[t_j] \in \mathbb{R}^n$, and the superscripts $^{-c}$ and $^{+c}$ on the state $\boldsymbol{x}$ denote respectively just *before* and *after* state correction. When PEG is called there is an instantaneous change in the pseudo-state $\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}$, which can be

written in the same general form as Eq. (5.97),

$$\hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b},+c}[t_k] = \hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b},-c}[t_k] + \left\{ \hat{\boldsymbol{\theta}}_{\text{PEG}}\big(\hat{\boldsymbol{x}}^{-c}[t_k], t_k\big) - \hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b},-c}[t_k] \right\} \ , \tag{5.98}$$

where PEG is denoted here as $\hat{\boldsymbol{\theta}}_{\text{PEG}} : \mathbb{R}^{m'} \to \mathbb{R}^3$, which is a nonlinear function of the navigation states $\hat{\boldsymbol{x}}^{-c}[t_k]$ at time t_k [see § 3.2.2]. It is important to note that the pseudo-state $\hat{\boldsymbol{\theta}}_{i,\text{com}}^{\hat{b}}$ is the only state that changes, i.e., all other true and navigation states remain unchanged. Therefore, the following expressions respectively represent the true and navigation state corrections at time t_k,

$$\boldsymbol{x}^{+c}[t_k] = \boldsymbol{x}^{-c}[t_k] + \boldsymbol{d}\big(\boldsymbol{x}^{-c}[t_k], \hat{\boldsymbol{x}}^{-c}[t_k], t_k\big) \ , \tag{5.99}$$

$$\hat{\boldsymbol{x}}^{+c}[t_k] = \hat{\boldsymbol{x}}^{-c}[t_k] \ , \tag{5.100}$$

where the correction $\boldsymbol{d} : \mathbb{R}^{n'} \times \mathbb{R}^{m'} \to \mathbb{R}^{n'}$ is now a nonlinear function of not only the true states $\boldsymbol{x}^{-c}[t_k] \in \mathbb{R}^{n'}$ but also the navigation states $\hat{\boldsymbol{x}}^{-c}[t_k] \in \mathbb{R}^{m'}$ as well. Next, given that the true and navigation states at time t_k can be respectively written as

$$\boldsymbol{x}[t_k] = \bar{\boldsymbol{x}}[t_k] + \delta\boldsymbol{x}[t_k] \ , \tag{5.101}$$

$$\hat{\boldsymbol{x}}[t_k] = \bar{\mathbf{x}}[t_k] + \delta\hat{\boldsymbol{x}}[t_k] \ , \tag{5.102}$$

and the nominal true and nominal navigation state correction equations at time t_k are respectively

$$\bar{\boldsymbol{x}}^{+c}[t_k] = \bar{\boldsymbol{x}}^{-c}[t_k] + \boldsymbol{d}\big(\bar{\boldsymbol{x}}^{-c}[t_k], \bar{\mathbf{x}}^{-c}[t_k], t_k\big) \ , \tag{5.103}$$

$$\bar{\mathbf{x}}^{+c}[t_k] = \bar{\mathbf{x}}^{-c}[t_k] \ , \tag{5.104}$$

then it follows that Eqs. (5.99) and (5.100) can be linearized using the process outlined in § 4.2, yielding the following linear state correction expressions

$$\delta\boldsymbol{x}^{+c}[t_k] = \left\{ \boldsymbol{I}_{n \times n} + \boldsymbol{D}_x^{-c}[t_k] \right\} \delta\boldsymbol{x}^{-c}[t_k] + \boldsymbol{D}_{\hat{x}}^{-c}[t_k] \, \delta\hat{\boldsymbol{x}}^{-c}[t_k] \ , \tag{5.105}$$

$$\delta\hat{\boldsymbol{x}}^{+c}[t_k] = \delta\hat{\boldsymbol{x}}^{-c}[t_k] \ , \tag{5.106}$$

where $\boldsymbol{D}_x^{-c}[t_k] \in \mathbb{R}^{n \times n}$ and $\boldsymbol{D}_{\hat{x}}^{-c}[t_k] \in \mathbb{R}^{n \times m}$ are the Jacobians defined respectively as

$$\boldsymbol{D}_x^{-c}[t_k] \equiv \left. \frac{\partial \boldsymbol{d}\big(\boldsymbol{x}^{-c}[t_k], \hat{\boldsymbol{x}}^{-c}[t_k], t_k\big)}{\partial \boldsymbol{x}_{\mathrm{m}}[t_k]} \right|_{\bar{\boldsymbol{x}}, \bar{\mathbf{x}}} \tag{5.107}$$

and

$$\boldsymbol{D}_{\hat{x}}^{-c}[t_k] \equiv \left. \frac{\partial \boldsymbol{d}\big(\boldsymbol{x}^{-c}[t_k], \hat{\boldsymbol{x}}^{-c}[t_k], t_k\big)}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}[t_k]} \right|_{\bar{\boldsymbol{x}}, \bar{\mathbf{x}}} . \tag{5.108}$$

Recall that the inertial-to-body attitude quaternion $\boldsymbol{q}_i^b \in \mathbb{R}^4$ is the standard representation of the launch vehicle attitude. However, due to state covariance matrix singularity issues associated with the quaternion [54], a modified state vector approach is adopted and utilized to form the augmented state covariance correction equations. In this approach, the inertial-to-body attitude quaternions $\boldsymbol{q}_i^b, \hat{\boldsymbol{q}}_i^b \in \mathbb{R}^4$ respectively in the true state vector $\boldsymbol{x} \in \mathbb{R}^{n'}$, defined as

$$\boldsymbol{x} = \big(\boldsymbol{r}^i, \, \boldsymbol{v}^i, \, \boldsymbol{q}_i^b, \, \boldsymbol{\omega}_{ib}^b, \, m_b, \, \boldsymbol{p}, \, \hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}\big) \,, \tag{5.109}$$

and the navigation state vector $\hat{\boldsymbol{x}} \in \mathbb{R}^{m'}$, defined as

$$\hat{\boldsymbol{x}} = \big(\hat{\boldsymbol{r}}^i, \, \hat{\boldsymbol{v}}^i, \, \hat{\boldsymbol{q}}_i^b, \, \hat{\boldsymbol{p}}\big) \,, \tag{5.110}$$

are respectively replaced with the Euler rotation vectors $\boldsymbol{\theta}_i^b, \hat{\boldsymbol{\theta}}_i^b \in \mathbb{R}^3$, which represent the attitude of the body frame b with respect to the inertial frame i expressed in body coordinates, resulting in the modified true state vector $\boldsymbol{x}_{\mathrm{m}} \in \mathbb{R}^n$ (where $n = n' - 1$), as defined in Eq. (5.45), and the modified navigation state vector $\hat{\boldsymbol{x}}_{\mathrm{m}} \in \mathbb{R}^m$, as defined in Eq. (5.46). Consequently, the linearized state correction equations, given in Eqs. (5.105) and (5.106), are based upon the modified state vectors, such that $\delta\boldsymbol{x} \in \mathbb{R}^n$, $\delta\hat{\boldsymbol{x}} \in \mathbb{R}^m$, and the dimensions of the Jacobians $\boldsymbol{D}_x^{-c}$ and $\boldsymbol{D}_{\hat{x}}^{-c}$ have been defined accordingly. Since $\hat{\boldsymbol{\theta}}_{i,\mathrm{com}}^{\hat{b}}$ is the only state that changes, it follows that the Jacobians are simply

$$\boldsymbol{D}_x^{-c}[t_k] = \begin{bmatrix} \boldsymbol{0}_{n-3 \times n-3} & \boldsymbol{0}_{n-3 \times 3} \\ \boldsymbol{0}_{3 \times n-3} & -\boldsymbol{I}_{3 \times 3} \end{bmatrix} \tag{5.111}$$

and

$$\boldsymbol{D}_{\hat{x}}^{-c}[t_k] = \begin{bmatrix} \mathbf{0}_{n-3\times m} \\ \dfrac{\partial \hat{\boldsymbol{\theta}}_{\mathrm{PEG}}\big(\hat{\boldsymbol{x}}^{-c}[t_k], t_k\big)}{\partial \hat{\boldsymbol{x}}_{\mathrm{m}}[t_k]}\Bigg|_{\hat{\boldsymbol{x}}=\bar{\mathbf{x}}} \end{bmatrix} , \tag{5.112}$$

where the remaining partial derivatives are those of the guidance attitude command issued by PEG with respect to the navigation states $\hat{\boldsymbol{x}}$ at time t_k.

Lastly, the linearized state correction equations, given in Eqs. (5.105) and (5.106), can be combined using the definition of the augmented state vector $\mathbf{x}_{\mathrm{a}}$, defined in Eq. (5.3), to form the augmented state correction equation at time t_k,

$$\mathbf{x}_{\mathrm{a}}^{+c}[t_k] = \big[\boldsymbol{I}_{\ell\times\ell} + \mathbf{D}_{\mathrm{x_a}}[t_k]\big]\,\mathbf{x}_{\mathrm{a}}^{-c}[t_k] , \tag{5.113}$$

where

$$\mathbf{D}_{\mathrm{x_a}}[t_k] \equiv \begin{bmatrix} \boldsymbol{D}_x^{-c}[t_k] & \boldsymbol{D}_{\hat{x}}^{-c}[t_k] \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{bmatrix} . \tag{5.114}$$

Multiplying Eq. (5.113) by its transpose, applying the expectation operator $E\{\ \}$, and recognizing that $\boldsymbol{\mathcal{C}}_{\mathrm{a}}(t_k) = E\left\{\mathbf{x}_{\mathrm{a}}(t_k)\left[\mathbf{x}_{\mathrm{a}}(t_k)\right]^{\mathsf{T}}\right\}$, it follows that the *augmented state covariance correction equation* at time t_k is given by

$$\boldsymbol{\mathcal{C}}_{\mathrm{a}}^{+c}[t_k] = \big[\boldsymbol{I}_{\ell\times\ell} + \mathbf{D}_{\mathrm{x_a}}[t_k]\big]\,\boldsymbol{\mathcal{C}}_{\mathrm{a}}^{-c}[t_k]\,\big[\boldsymbol{I}_{\ell\times\ell} + \mathbf{D}_{\mathrm{x_a}}[t_k]\big]^{\mathsf{T}} . \tag{5.115}$$

5.5.4 Covariance Shaping

In order to simulate in linear covariance the discrete event that occurs when guidance modes from PEG to fine count, the nominal value of the modified true state vector $\bar{\boldsymbol{x}}_{\mathrm{m}}$, the nominal time derivative of the modified true state vector $\dot{\bar{\boldsymbol{x}}}_{\mathrm{m}}$, the event trigger Ψ, the event trigger Jacobian $\boldsymbol{\Psi}_{\hat{x}}$, and the mapping function Jacobian $\boldsymbol{M}_x$ must all be specified at the nominal event time $\bar{t}_{\mathrm{e}}$. Under the assumption that the modified true state vector $\boldsymbol{x}_{\mathrm{m}} \in \mathbb{R}^n$ is given by Eq. (5.45), it follows that the nominal value of the modified true state vector $\bar{\boldsymbol{x}}_{\mathrm{m}}$ at the nominal event time $\bar{t}_{\mathrm{e}}$ is given by

$$\bar{\boldsymbol{x}}_{\mathrm{m}}(\bar{t}_{\mathrm{e}}) = \big(\bar{\boldsymbol{r}}^i,\ \bar{\boldsymbol{v}}^i,\ \bar{\boldsymbol{\theta}}_i^b,\ \bar{\boldsymbol{\omega}}_{ib}^b,\ \bar{m}_b,\ \bar{\boldsymbol{p}},\ \bar{\boldsymbol{\theta}}_{i,\mathrm{com}}^b\big) , \tag{5.116}$$

and the corresponding vector of nominal time derivatives is given by

$$\dot{\bar{\boldsymbol{x}}}_{\mathrm{m}}(\bar{t}_{\mathrm{e}}) = \left(\dot{\bar{\boldsymbol{r}}}^{i},\ \dot{\bar{\boldsymbol{v}}}^{i},\ \dot{\bar{\boldsymbol{\theta}}}^{b}_{i},\ \dot{\bar{\boldsymbol{\omega}}}^{b}_{ib},\ \dot{\bar{m}}_{b},\ \dot{\bar{\boldsymbol{p}}},\ \dot{\bar{\boldsymbol{\theta}}}^{b}_{i,\mathrm{com}}\right)\ . \tag{5.117}$$

These nominal time derivatives of the modified true states at the nominal event time $\bar{t}_{\mathrm{e}}$ are computed with the following set of first-order differential equations:

$$\dot{\bar{\boldsymbol{r}}}^{i}(\bar{t}_{\mathrm{e}}) = \bar{\boldsymbol{v}}^{i}(\bar{t}_{\mathrm{e}})\ , \tag{5.118}$$

$$\dot{\bar{\boldsymbol{v}}}^{i}(\bar{t}_{\mathrm{e}}) = -\frac{\mu}{\left\|\bar{\boldsymbol{r}}^{i}(\bar{t}_{\mathrm{e}})\right\|^{2}}\bar{\mathbf{u}}^{i}_{r}(\bar{t}_{\mathrm{e}}) + \left[\bar{\boldsymbol{T}}^{b}_{i}(\bar{t}_{\mathrm{e}})\right]^{\mathsf{T}}\frac{T_{\mathrm{vac}}}{\bar{m}_{b}(\bar{t}_{\mathrm{e}})}\mathbf{b}^{b}_{1}\ , \tag{5.119}$$

$$\dot{\bar{\boldsymbol{\theta}}}^{b}(\bar{t}_{\mathrm{e}}) = \bar{\boldsymbol{\omega}}^{b}_{ib}(\bar{t}_{\mathrm{e}})\ , \tag{5.120}$$

$$\dot{\bar{\boldsymbol{\omega}}}^{b}_{ib}(\bar{t}_{\mathrm{e}}) = \left[\bar{\boldsymbol{J}}^{b}_{b}(\bar{t}_{\mathrm{e}})\right]^{-1}\left[\bar{\boldsymbol{m}}^{b}_{\mathrm{com}}(\bar{t}_{\mathrm{e}}) - \bar{\boldsymbol{\omega}}^{b}_{ib}(\bar{t}_{\mathrm{e}}) \times \bar{\boldsymbol{J}}^{b}_{b}(\bar{t}_{\mathrm{e}})\,\bar{\boldsymbol{\omega}}^{b}_{ib}(\bar{t}_{\mathrm{e}})\right]\ , \tag{5.121}$$

$$\dot{\bar{m}}_{b}(\bar{t}_{\mathrm{e}}) = -\beta\ , \tag{5.122}$$

$$\dot{\bar{\boldsymbol{p}}}(\bar{t}_{\mathrm{e}}) = \mathbf{0}_{p\times 1}\ , \tag{5.123}$$

$$\dot{\bar{\boldsymbol{\theta}}}^{b}_{i,\mathrm{com}}(\bar{t}_{\mathrm{e}}) = \bar{\boldsymbol{T}}^{b}_{i}(\bar{t}_{\mathrm{e}})\,\bar{\boldsymbol{\omega}}^{i}_{ib,\mathrm{com}}(\bar{t}_{\mathrm{e}})\ . \tag{5.124}$$

It is important to note that the nominal values of the true error parameter states $\bar{\boldsymbol{p}} \in \mathbb{R}^{p}$ are zero for all time t. For this reason, the corresponding nominal dynamics are set to zero as well.

The event trigger is defined to occur when the time-to-go t_{go} is equal to desired length of fine count t_{fc}, as given by

$$\Psi(\hat{\boldsymbol{x}}, t_{\mathrm{e}}) = t_{\mathrm{go}}\big[\hat{\boldsymbol{x}}(t_{\mathrm{e}})\big] - t_{\mathrm{fc}}\ , \tag{5.125}$$

where the time-to-go is written as $t_{\mathrm{go}}\big[\hat{\boldsymbol{x}}(t_{\mathrm{e}})\big]$ to denote that it is computed by PEG and therefore a function of the navigation states $\hat{\boldsymbol{x}}$ at the event time t_{e}. The Jacobian of the event trigger with respect to the navigation states at the nominal event time $\bar{t}_{\mathrm{e}}$ is therefore given by

$$\boldsymbol{\Psi}_{\hat{x}}(\bar{t}_{e}) = \left.\frac{\partial t_{\mathrm{go}}\big[\hat{\boldsymbol{x}}(t_{\mathrm{e}})\big]}{\partial \hat{\boldsymbol{x}}}\right|_{\bar{\mathbf{x}}}\ . \tag{5.126}$$

Due to the highly nonlinear nature of PEG, an analytic expression for the time-to-go t_{go} parameter in terms of the navigation states $\hat{\boldsymbol{x}}$ could not be obtained. For this reason, the partial derivatives of the event trigger were computed numerically using the methods described in § 4.4.3.

In the formulation of the event trigger theory, found in § 5.4, it was shown that the covariance of the augmented states at the nominal event time $\bar{t}_e$ is given by

$$\mathcal{C}_a^{+s}(\bar{t}_e) = \mathbf{\Phi}_s(\bar{t}_e)\, \mathcal{C}_a^{-s}(\bar{t}_e)\, \left[\mathbf{\Phi}_s(\bar{t}_e)\right]^\mathsf{T} , \tag{5.127}$$

where the *covariance shaping matrix* $\mathbf{\Phi}_s$ is given by

$$\mathbf{\Phi}_s(\bar{t}_e) = \begin{bmatrix} \mathbf{I}_{n\times n} & -\dot{\bar{\boldsymbol{x}}}(\bar{t}_e)\left[\mathbf{\Psi}_{\hat{x}}(\bar{t}_e)\,\boldsymbol{M}_x(\bar{t}_e)\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_e)\right]^{-1}\mathbf{\Psi}_{\hat{x}}(\bar{t}_e) \\ \mathbf{0}_{m\times n} & \mathbf{I}_{m\times m} - \boldsymbol{M}_x(\bar{t}_e)\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_e)\left[\mathbf{\Psi}_{\hat{x}}(\bar{t}_e)\,\boldsymbol{M}_x(\bar{t}_e)\,\dot{\bar{\boldsymbol{x}}}(\bar{t}_e)\right]^{-1}\mathbf{\Psi}_{\hat{x}}(\bar{t}_e) \end{bmatrix} , \tag{5.128}$$

and $\boldsymbol{M}_x$ is the mapping function Jacobian defined in Eq. (5.11).

For this research, the only discrete event of concern is the last guidance mode change or transition from PEG to fine count. It is important to note that although, from a Monte Carlo perspective, MECO takes place at varying times with respect to lift-off (t_0), it always occurs t_{fc} seconds after the sample event time t_e. Thus, it is a time driven event and covariance shaping is not needed.

5.5.5 Covariance Post-Processing

In a Monte Carlo simulation, if a variable of interest is not explicitly modeled as one of the system states (i.e., in the true or navigation state vectors) but can be expressed as a linear or nonlinear function of the system states (true or navigation), then the covariance of the variable of interest can be computed in a relatively straightforward manner. For example, let $\boldsymbol{x} \in \mathbb{R}^a$ be the vector of system states and $\boldsymbol{y} \in \mathbb{R}^b$ be the variable of interest defined by the nonlinear vector function

$$\boldsymbol{y}(t) = \boldsymbol{f}(\boldsymbol{x}, t) . \tag{5.129}$$

Using this expression, the covariance $\boldsymbol{P}_{yy}$ of the variable of interest at the time t_k, from N Monte Carlo samples, is then computed as follows

$$\begin{aligned} \boldsymbol{P}_{yy}(t_k) &= \frac{1}{N-1}\sum_{j=1}^{N}\left[\boldsymbol{y}_j(t_k) - \bar{\boldsymbol{y}}(t_k)\right]\left[\boldsymbol{y}_j(t_k) - \bar{\boldsymbol{y}}(t_k)\right]^\mathsf{T} \\ &= \frac{1}{N-1}\sum_{j=1}^{N}\left[\boldsymbol{f}(\boldsymbol{x}_j, t_k) - \boldsymbol{f}(\bar{\boldsymbol{x}}, t_k)\right]\left[\boldsymbol{f}(\boldsymbol{x}_j, t_k) - \boldsymbol{f}(\bar{\boldsymbol{x}}_j, t_k)\right]^\mathsf{T} , \end{aligned} \tag{5.130}$$

where ‾ above the variable denotes the nominal value. Note that this last equation can also be equivalently expressed using the expectation operator,

$$\boldsymbol{P}_{yy}(t_k) = E\Big\{\big[\boldsymbol{f}(\boldsymbol{x}, t_k) - \boldsymbol{f}(\bar{\boldsymbol{x}}, t_k)\big]\big[\boldsymbol{f}(\boldsymbol{x}, t_k) - \boldsymbol{f}(\bar{\boldsymbol{x}}, t_k)\big]^\mathsf{T}\Big\} . \tag{5.131}$$

In a linear covariance simulation, however, there is only the nominal value and covariance of the system states, thus the calculation of the covariance of the variable of interest requires a different approach. First, the nonlinear vector function is linearized about the nominal using a truncated first-order Taylor series expansion,

$$\boldsymbol{f}(\boldsymbol{x}, t_k) \cong \boldsymbol{f}(\bar{\boldsymbol{x}}, t_k) + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}} \big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big] . \tag{5.132}$$

Substituting this expression into Eq. (5.131) and canceling terms produces

$$\boldsymbol{P}_{yy}(t_k) = E\left\{\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}}\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]\right]\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}}\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]\right]^\mathsf{T}\right\} .$$

Moving the transpose inside the square brackets and reversing the order of the factors yields

$$\boldsymbol{P}_{yy}(t_k) = E\left\{\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}}\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]^\mathsf{T}\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}}\right]^\mathsf{T}\right\} .$$

The partial derivatives can now be moved out of the expectation operator because they are known nonstochastic quantities

$$\boldsymbol{P}_{yy}(t_k) = \frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}} E\Big\{\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]^\mathsf{T}\Big\}\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}}\right]^\mathsf{T} ,$$

and noting that by definition

$$\boldsymbol{P}_{xx}(t_k) = E\Big\{\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]\big[\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)\big]^\mathsf{T}\Big\} , \tag{5.133}$$

it follows that the covariance of the variable of interest can be computed as follows

$$\boldsymbol{P}_{yy}(t_k) = \frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}} \boldsymbol{P}_{xx}(t_k)\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, t_k)}{\partial \boldsymbol{x}}\bigg|_{\bar{\boldsymbol{x}}}\right]^\mathsf{T} . \tag{5.134}$$

Now that the theory has been introduced, it is applied several important variables of interest for the powered ascent problem. They are the five guidance target parameters: altitude, velocity magnitude, flight path angle, inclination, and right ascension of the ascending node. Two additional variables of interest, not used in this research but important for powered ascent flight through an atmosphere, are the inertial velocity magnitude and flight path angle relative to the planet, hereafter referred to simply as the relative velocity magnitude and relative flight path angle, respectively.

Planetocentric Altitude

The planetocentric altitude r_{alt} is given by

$$r_{\text{alt}} = \left\| \boldsymbol{r}^i \right\| - r_{\text{eq}} \, , \tag{5.135}$$

where $\boldsymbol{r}^i$ is the inertial position vector expressed in inertial coordinates, and r_{eq} is the mean equatorial radius of planet. Since the altitude is only a function of the inertial position, it follows that the associated variance σ_{alt}^2 can be computed as follows

$$\sigma_{\text{alt}}^2 = \left. \frac{\partial r_{\text{alt}}}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} \boldsymbol{P}_{rr} \left[\left. \frac{\partial r_{\text{alt}}}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} \right]^{\mathsf{T}} \, , \tag{5.136}$$

where $\boldsymbol{P}_{rr} \in \mathbb{R}^{3 \times 3}$ is the covariance of the inertial position vector and the corresponding partial derivative is given by

$$\left. \frac{\partial r_{\text{alt}}}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} = \left[\bar{\mathbf{u}}_r^i \right]^{\mathsf{T}} \, , \tag{5.137}$$

where $\bar{\mathbf{u}}_r^i$ is the unit vector of the nominal inertial position expressed in inertial coordinates.

Inertial Velocity Magnitude

The inertial velocity magnitude v is given by

$$v = \left\| \boldsymbol{v}^i \right\| \, , \tag{5.138}$$

where $\boldsymbol{v}^i$ is the inertial velocity vector expressed in inertial coordinates. The associated variance σ_v^2 can be computed as follows

$$\sigma_v^2 = \left.\frac{\partial v}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} \boldsymbol{P}_{vv} \left[\left.\frac{\partial v}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}}\right]^\mathsf{T} , \tag{5.139}$$

where $\boldsymbol{P}_{vv} \in \mathbb{R}^{3\times3}$ is the covariance of the inertial velocity vector and the corresponding partial derivative is given by

$$\left.\frac{\partial v}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} = \left[\bar{\mathbf{u}}_v^i\right]^\mathsf{T} ,$$

where $\bar{\mathbf{u}}_v^i$ is the unit vector of the nominal inertial velocity expressed in inertial coordinates.

Inertial Flight Path Angle

The inertial flight path angle (FPA) γ can be written as

$$\sin \gamma = \left[\mathbf{u}_r^i\right]^\mathsf{T}\mathbf{u}_v^i , \tag{5.140}$$

where $\mathbf{u}_r^i$ and $\mathbf{u}_v^i$ are respectively the unit vectors of the inertial position and velocity expressed in inertial coordinates. Since the flight path angle is a function of both the inertial position and velocity states, it follows that the associated variance σ_γ^2 can be computed as follows

$$\sigma_\gamma^2 = \left[\left.\frac{\partial \gamma}{\partial \boldsymbol{r}^i}\right|_{\bar{\boldsymbol{x}}} \quad \left.\frac{\partial \gamma}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} \right] \begin{bmatrix} \boldsymbol{P}_{rr} & \boldsymbol{P}_{rv} \\ \left[\boldsymbol{P}_{rv}\right]^\mathsf{T} & \boldsymbol{P}_{vv} \end{bmatrix} \left[\left.\frac{\partial \gamma}{\partial \boldsymbol{r}^i}\right|_{\bar{\boldsymbol{x}}} \quad \left.\frac{\partial \gamma}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} \right]^\mathsf{T} , \tag{5.141}$$

where $\boldsymbol{P}_{rv} \in \mathbb{R}^{3\times3}$ is the cross correlation covariance of the inertial position and velocity vectors and the corresponding partial derivatives are given by

$$\left.\frac{\partial \gamma}{\partial \boldsymbol{r}^i}\right|_{\bar{\boldsymbol{x}}} = \left\{ \frac{\left[\bar{\mathbf{u}}_v^i\right]^\mathsf{T}}{\left\|\bar{r}^i\right\|} \left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}_r^i \left[\bar{\mathbf{u}}_r^i\right]^\mathsf{T}\right) \right\} \frac{1}{\cos \bar{\gamma}} , \tag{5.142}$$

$$\left.\frac{\partial \gamma}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} = \left\{ \frac{\left[\bar{\mathbf{u}}_r^i\right]^\mathsf{T}}{\left\|\bar{v}^i\right\|} \left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}_v^i \left[\bar{\mathbf{u}}_v^i\right]^\mathsf{T}\right) \right\} \frac{1}{\cos \bar{\gamma}} . \tag{5.143}$$

Inclination

The inclination i can be written as

$$\cos i = \left[\mathbf{i}_3^i\right]^\mathsf{T} \mathbf{u}_h^i \ , \tag{5.144}$$

where $\mathbf{i}_3^i$ is the unit inertial frame basis vector, as defined in Appendix A.1, and $\mathbf{u}_h^i$ is the unit angular momentum vector of the launch vehicle, expressed in inertial coordinates respectively. The angular momentum vector $\boldsymbol{h}^i$ is defined as

$$\boldsymbol{h}^i = \boldsymbol{r}^i \times \boldsymbol{v}^i \ . \tag{5.145}$$

Since the inclination is a function of both the inertial position and velocity states, it follows that the associated variance σ_{inc}^2 can be computed as follows

$$\sigma_{\text{inc}}^2 = \left[\begin{array}{cc} \left.\dfrac{\partial i}{\partial \boldsymbol{r}^i}\right|_{\bar{\boldsymbol{x}}} & \left.\dfrac{\partial i}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} \end{array}\right] \left[\begin{array}{cc} \boldsymbol{P}_{rr} & \boldsymbol{P}_{rv} \\ \left[\boldsymbol{P}_{rv}\right]^\mathsf{T} & \boldsymbol{P}_{vv} \end{array}\right] \left[\begin{array}{cc} \left.\dfrac{\partial i}{\partial \boldsymbol{r}^i}\right|_{\bar{\boldsymbol{x}}} & \left.\dfrac{\partial i}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} \end{array}\right]^\mathsf{T} \ , \tag{5.146}$$

where the corresponding partial derivatives are given by

$$\left.\frac{\partial i}{\partial \boldsymbol{r}^i}\right|_{\bar{\boldsymbol{x}}} = \left\{ \frac{\left[\bar{\mathbf{i}}_3^i\right]^\mathsf{T}}{\left\|\bar{\boldsymbol{h}}^i(t)\right\|}\left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}_h^i\left[\bar{\mathbf{u}}_h^i\right]^\mathsf{T}\right)\left[-\bar{\boldsymbol{v}}^i\times\right]\right\} \frac{1}{-\sin\bar{i}} \ , \tag{5.147}$$

$$\left.\frac{\partial i}{\partial \boldsymbol{v}^i}\right|_{\bar{\boldsymbol{x}}} = \left\{ \frac{\left[\bar{\mathbf{i}}_3^i\right]^\mathsf{T}}{\left\|\bar{\boldsymbol{h}}^i\right\|}\left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}_h^i\left[\bar{\mathbf{u}}_h^i\right]^\mathsf{T}\right)\left[\bar{\boldsymbol{r}}^i\times\right]\right\} \frac{1}{-\sin\bar{i}} \ . \tag{5.148}$$

Right Ascension of the Ascending Node

The right ascension of the ascending (RAAN) Ω can be written as

$$\cos\Omega = \left[\mathbf{i}_1^i\right]^\mathsf{T} \mathbf{u}_n^i \ , \tag{5.149}$$

where $\mathbf{i}_1^i$ is the unit inertial frame basis vector, as defined in Appendix A.1, and $\mathbf{u}_n^i$ is the unit node vector, expressed in inertial coordinates respectively. The node vector $\boldsymbol{n}^i$ is defined as

$$\boldsymbol{n}^i = \mathbf{i}_3^i \times \boldsymbol{h}^i \ , \tag{5.150}$$

where $\mathbf{i}_3^i$ is the unit inertial frame basis vector and $\boldsymbol{h}^i$ is the angular momentum vector defined in Eq. (5.145). Note that if the second component of the node vector $n_2 < 0$, then the RAAN computed with Eq. (5.149) needs to be placed in the appropriate quadrant with $\Omega = 360° - \Omega$. Since the RAAN is a function of both the inertial position and velocity states, it follows that the associated variance σ_Ω^2 can be computed as follows

$$\sigma_\Omega^2 = \left[\left. \frac{\partial \Omega}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} \quad \left. \frac{\partial \Omega}{\partial \boldsymbol{v}^i} \right|_{\bar{\boldsymbol{x}}} \right] \left[\begin{array}{cc} \boldsymbol{P}_{rr} & \boldsymbol{P}_{rv} \\ [\boldsymbol{P}_{rv}]^\mathsf{T} & \boldsymbol{P}_{vv} \end{array} \right] \left[\left. \frac{\partial \Omega}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} \quad \left. \frac{\partial \Omega}{\partial \boldsymbol{v}^i} \right|_{\bar{\boldsymbol{x}}} \right]^\mathsf{T} , \tag{5.151}$$

where the corresponding partial derivatives are given by

$$\left. \frac{\partial \Omega}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} = \left\{ \frac{[\bar{\mathbf{i}}_1^i]^\mathsf{T}}{\|\bar{\boldsymbol{n}}^i\|} \left(\boldsymbol{I}_{3\times 3} - \bar{\mathbf{u}}_n^i \left[\bar{\mathbf{u}}_n^i\right]^\mathsf{T} \right) [\bar{\mathbf{i}}_3^i \times] [-\bar{\boldsymbol{v}}^i \times] \right\} \frac{1}{-\sin \bar{\Omega}} , \tag{5.152}$$

$$\left. \frac{\partial \Omega}{\partial \boldsymbol{v}^i} \right|_{\bar{\boldsymbol{x}}} = \left\{ \frac{[\bar{\mathbf{i}}_1^i]^\mathsf{T}}{\|\bar{\boldsymbol{n}}^i\|} \left(\boldsymbol{I}_{3\times 3} - \bar{\mathbf{u}}_n^i \left[\bar{\mathbf{u}}_n^i\right]^\mathsf{T} \right) [\bar{\mathbf{i}}_3^i \times] [\bar{\boldsymbol{r}}^i \times] \right\} \frac{1}{-\sin \bar{\Omega}} . \tag{5.153}$$

Relative Velocity Magnitude

The relative velocity magnitude v_{rel} is given by

$$v_{\text{rel}} = \left\| \boldsymbol{v}_{\text{rel}}^i \right\| , \tag{5.154}$$

where the relative velocity vector $\boldsymbol{v}_{\text{rel}}^i$, expressed in inertial coordinates, is defined as

$$\boldsymbol{v}_{\text{rel}}^i = \boldsymbol{v}^i - \boldsymbol{\omega}_{ip}^i \times \boldsymbol{r}^i , \tag{5.155}$$

and $\boldsymbol{\omega}_{ip}^i$ is the angular velocity vector of the planet expressed in inertial coordinates. Since the relative velocity magnitude is a function of both the inertial position and velocity states, it follows that the associated variance $\sigma_{v_{\text{rel}}}^2$ can be computed as follows

$$\sigma_{v_{\text{rel}}}^2 = \left[\left. \frac{\partial v_{\text{rel}}}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} \quad \left. \frac{\partial v_{\text{rel}}}{\partial \boldsymbol{v}^i} \right|_{\bar{\boldsymbol{x}}} \right] \left[\begin{array}{cc} \boldsymbol{P}_{rr} & \boldsymbol{P}_{rv} \\ [\boldsymbol{P}_{rv}]^\mathsf{T} & \boldsymbol{P}_{vv} \end{array} \right] \left[\left. \frac{\partial v_{\text{rel}}}{\partial \boldsymbol{r}^i} \right|_{\bar{\boldsymbol{x}}} \quad \left. \frac{\partial v_{\text{rel}}}{\partial \boldsymbol{v}^i} \right|_{\bar{\boldsymbol{x}}} \right]^\mathsf{T} , \tag{5.156}$$

where the corresponding partial derivatives are given by

$$\frac{\partial v_{\mathrm{rel}}}{\partial \boldsymbol{r}^i}\bigg|_{\bar{\boldsymbol{x}}} = -\left[\bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\right]^{\mathsf{T}}\left[\bar{\boldsymbol{\omega}}^i_p\times\right] \;, \tag{5.157}$$

$$\frac{\partial v_{\mathrm{rel}}}{\partial \boldsymbol{v}^i}\bigg|_{\bar{\boldsymbol{x}}} = \left[\bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\right]^{\mathsf{T}} \;, \tag{5.158}$$

and $\bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}$ is the nominal unit relative velocity vector expressed in inertial coordinates.

Relative Flight Path Angle

The relative flight path angle γ_{rel} can be written as

$$\sin\gamma_{\mathrm{rel}} = \left[\mathbf{u}^i_r\right]^{\mathsf{T}}\mathbf{u}^i_{v_{\mathrm{rel}}} \;, \tag{5.159}$$

where $\mathbf{u}^i_r$ and $\mathbf{u}^i_{v_{\mathrm{rel}}}$ are respectively the unit inertial position and unit Moon-relative velocity vectors expressed in inertial coordinates. Since the relative FPA is a function of both the inertial position and velocity states, it follows that the associated variance $\sigma^2_{\gamma_{\mathrm{rel}}}$ can be computed as follows

$$\sigma^2_{\gamma_{\mathrm{rel}}} = \left[\; \frac{\partial \gamma_{\mathrm{rel}}}{\partial \boldsymbol{r}^i}\bigg|_{\bar{\boldsymbol{x}}} \quad \frac{\partial \gamma_{\mathrm{rel}}}{\partial \boldsymbol{v}^i}\bigg|_{\bar{\boldsymbol{x}}} \;\right] \begin{bmatrix} \boldsymbol{P}_{rr} & \boldsymbol{P}_{rv} \\ \left[\boldsymbol{P}_{rv}\right]^{\mathsf{T}} & \boldsymbol{P}_{vv} \end{bmatrix} \left[\; \frac{\partial \gamma_{\mathrm{rel}}}{\partial \boldsymbol{r}^i}\bigg|_{\bar{\boldsymbol{x}}} \quad \frac{\partial \gamma_{\mathrm{rel}}}{\partial \boldsymbol{v}^i}\bigg|_{\bar{\boldsymbol{x}}} \;\right]^{\mathsf{T}} \;, \tag{5.160}$$

where the corresponding partial derivatives are given by

$$\frac{\partial \gamma_{\mathrm{rel}}}{\partial \boldsymbol{r}^i}\bigg|_{\bar{\boldsymbol{x}}} = \left\{ \frac{\left[\bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\right]^{\mathsf{T}}}{\|\bar{\boldsymbol{r}}^i\|}\left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}^i_r\left[\bar{\mathbf{u}}^i_r\right]^{\mathsf{T}}\right) - \frac{\left[\bar{\mathbf{u}}^i_r\right]^{\mathsf{T}}}{\|\bar{\boldsymbol{v}}^i_{\mathrm{rel}}\|}\left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\left[\bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\right]^{\mathsf{T}}\right)\left[\bar{\boldsymbol{\omega}}^i_p\times\right] \right\}\frac{1}{\cos\bar{\gamma}_{\mathrm{rel}}} \;, \tag{5.161}$$

$$\frac{\partial \gamma_{\mathrm{rel}}}{\partial \boldsymbol{v}^i}\bigg|_{\bar{\boldsymbol{x}}} = \left\{ \frac{\left[\bar{\mathbf{u}}^i_r\right]^{\mathsf{T}}}{\|\bar{\boldsymbol{v}}^i_{\mathrm{rel}}\|}\left(\boldsymbol{I}_{3\times3} - \bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\left[\bar{\mathbf{u}}^i_{v_{\mathrm{rel}}}\right]^{\mathsf{T}}\right) \right\}\frac{1}{\cos\bar{\gamma}_{\mathrm{rel}}} \;. \tag{5.162}$$

In summary, the purpose of this chapter was to formulate the linear covariance equations, thereby completing the development process of the linear covariance simulation. To this end, the augmented state vector and its associated covariance matrix were introduced, the event trigger theory was developed for the general case, and the methods for initializing, propagating, correcting, and shaping the augmented state covariance matrix were established. It is important to point out that the linear covariance equations presented herein constitute the first formulation and development of a linear covariance simulation for evaluation of a launch vehicle GN&C system in a

closed-loop setting during powered ascent.

Chapter 6

Launch Scenario & Nominal Ascent Profile

"A theory can be proved by experiment; but no path leads

from experiment to the birth of a theory."

– Albert Einstein

Now that all of the necessary theory and equations have been developed, the next step is to setup and define the details of the experiment that will be used to test the theory and demonstrate the capabilities of the analysis tool. Hence, the purpose of this chapter is to define and describe the launch scenario and accompanying nominal reference trajectory or nominal ascent profile. The launch scenario and nominal ascent profile designed for this dissertation are based upon the scenarios and ascent trajectories outlined in Sostaric and Merriam [48] and Fill [71].

6.1 Launch Scenario

The launch scenario consists of a launch from the surface of the Moon, where the the operating environment is the vacuum of space and the Moon is modeled as a spheroid with a 1738.09018 km radius, rotational rate of 1.52504146e-4 deg/s, and gravitational constant of 4902.777969 km^3/s^2. The launch site is the Aristarchus Plateau located at 26 deg latitude and 311 deg (east) longitude, and the target orbit is defined by a 1783.2 km semi-major axis, 0.0168 eccentricity, 145.765 deg inclination, and 356.768 deg RAAN. In addition, the first opportunity to launch into the target orbit corresponds to a *northerly* launch direction and 290.1 deg launch azimuth.

The launch vehicle is modeled as a rigid, SSTO vehicle with time-varying mass and mass MOI. The initial wet mass of the launch vehicle is 6100 kg, which consists of 3100 kg of dry mass and a fuel budget of 3000 kg, and the mass flow rate is a constant 8.16 kg/s. Moreover, the initial principle-axis mass MOI about the body x, y, and x axes are respectively 4540.0 $kg\,m^2$, 7491.6 $kg\,m^2$, and 8491.6 $kg\,m^2$. Thrust is provided by a single liquid-propellant engine with a 320 s specific impulse and a 25.6 kN (5755.1 lbf) nominal average thrust, in vacuum.
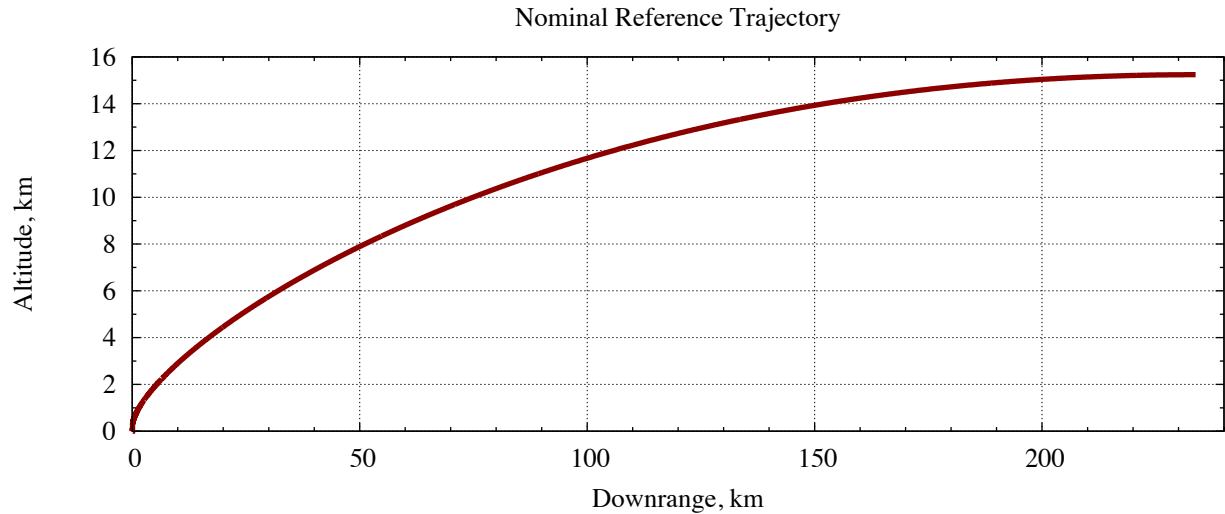
Figure 6.1: *Nominal reference trajectory.* Plot of the nominal planetocentric altitude versus nominal downrange position with respect to the launch site.

6.2 Nominal Ascent Profile

The nominal ascent profile begins with a 3 s vertical rise to an altitude of 11.6 m, during which the attitude of the launch vehicle remains constant with respect to the rotating Moon. This is followed by a 10 s pitch-over maneuver, which rotates the thrust direction from the vertical to the optimal heading and pitch angle for ascent. PEG takes over 13 s after lift-off and steers the vehicle into the desired target orbit. At 321.73 s after lift-off, PEG is terminated and guidance modes into fine count, where the inertial thrust command is held constant. MECO occurs 4 s later or 325.73 s after lift-off. Figure 6.1 shows the nominal altitude versus downrange position profile.

Nominal time history profiles for each of the five vehicle states (i.e., inertial position, inertial velocity, body attitude, body angular velocity, and wet mass) are shown in Figures 6.2–6.6. Observe that each figure contains a bar near the bottom that denotes the portions of the time line over which vertical rise (VR), pitch over (PO), and PEG occur. Moreover, in order to present a better view of the VR and PO segments of the nominal trajectory, the first 16 s of the trajectory are plotted on a different scale than the rest of the trajectory. Figure 6.4 (body y-axis) shows that the optimal pitch angle for ascent is approximately 50 deg off of the vertical, which therefore attests to the necessity of a pitch-over maneuver. The nominal wet mass at MECO is 3442.8 kg, thus leaving 342.8 kg in reserve, as shown in Figure 6.6. Figures 6.7 and 6.8 show the nominal time history profiles respectively of the principle-axis mass MOI and ACS torque command. Figures 6.9 and 6.10 show the nominal time history profiles respectively of the commanded body attitude and body

Table 6.1: *Nominal reference trajectory.* Guidance targets and simulation insertion errors.

Target Parameter	Target Value	Insertion Error
Planetocentric Altitude, r_{alt}	15.24 km	-0.2555639 m
Inertial Velocity Magnitude, v	1686.156 m/s	-0.0239552 m/s
Inertial Flight Path Angle, γ	0 deg	-0.0068248 deg
Inclination, i	145.765 deg	-0.0000252 deg
RAAN, Ω	356.786 deg	-0.0000709 deg

angular velocity from guidance along with the actual values for direct comparison. Figure **??** shows the nominal time history profile of the thrust acceleration.

Recall that the five terminal targets for PEG are planetocentric altitude, inertial velocity magnitude, inertial FPA, inclination, and RAAN. The values used in this research are given in Table 6.1 and corresponding nominal time history profiles are provided in Figures 6.11–6.15. In Figures 6.14 and 6.15, it would appear that the target values for the inclination and RAAN are achieved early on in the trajectory (i.e., around 50 s) rather than at the end or MECO. However, this is nothing more than an artifact of the scaling of the vertical axes in these two figures. What is not clearly shown is that the inclination and RAAN actually overshoot their targets slightly, approach the targets from above, and finally converge to their target values near MECO. Ideally, the terminal targets would be achieved perfectly in the nominal reference trajectory, i.e., the orbit insertion errors would nominally be zero. In reality, however, the simulation that generates the nominal reference trajectory is only able to get close to the target values. As such, the orbit insertion errors are generally nonzero and therefore provide a simple measure of how well the simulation is able to perform under nominal conditions. The simulation induced insertion errors for this research are given in Table 6.1, which indicate a well-tuned and reasonably accurate simulation.

Figure 6.2: *Nominal reference trajectory.* Nominal time history of the nominal inertial position vector.
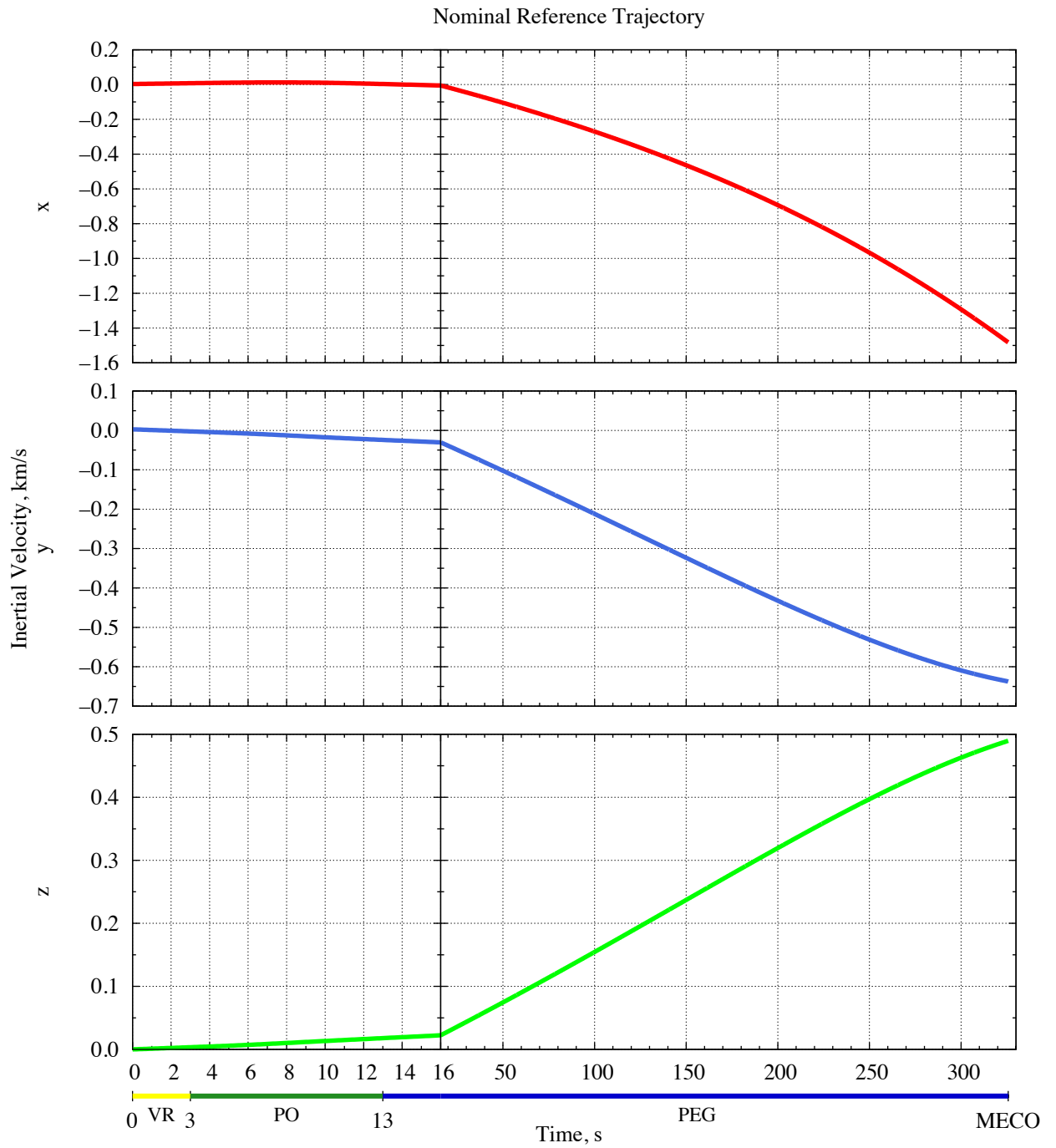
Figure 6.3: *Nominal reference trajectory.* Nominal time history of the nominal inertial velocity vector.
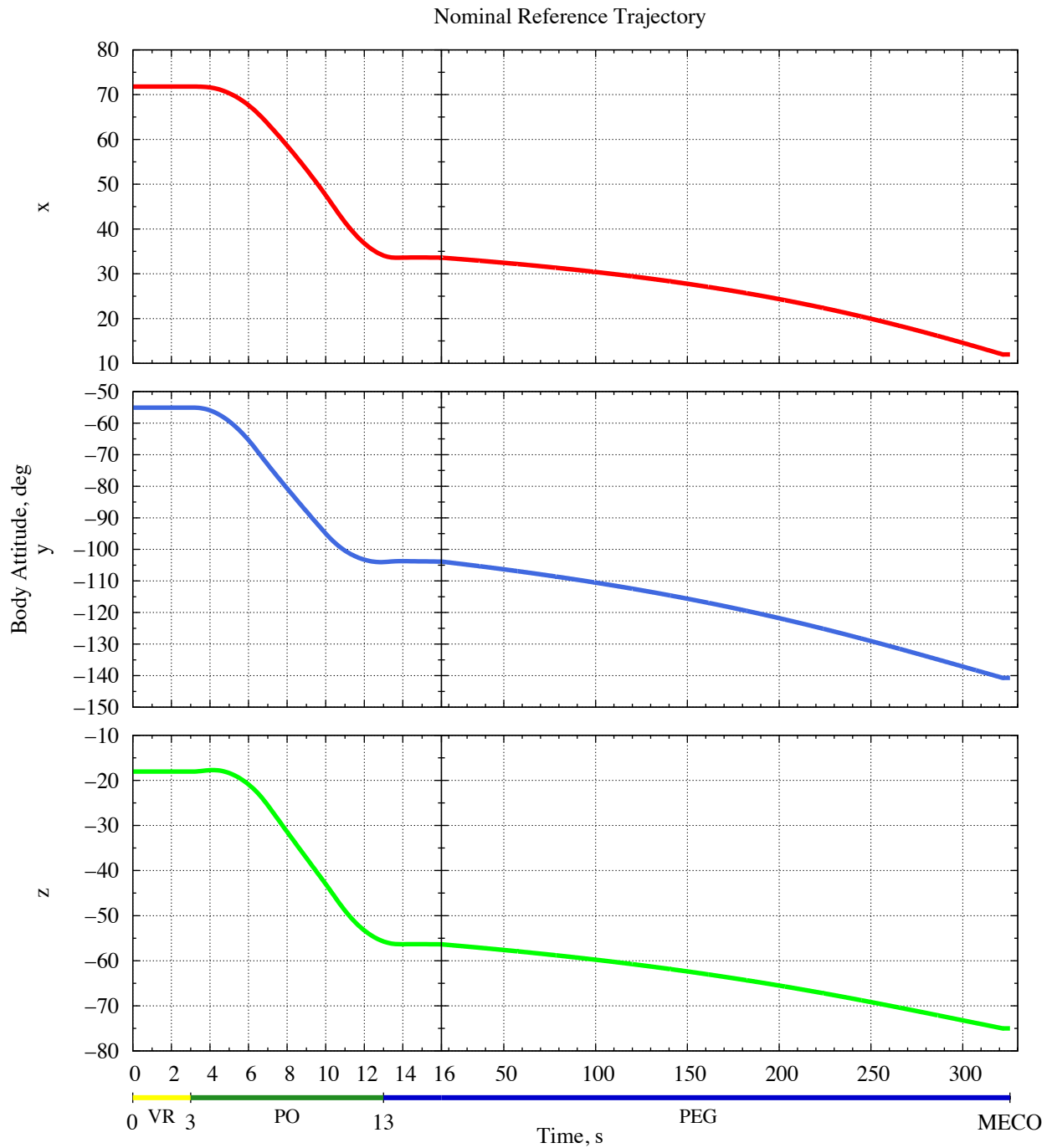
Figure 6.4: *Nominal reference trajectory.* Nominal time history of the nominal body attitude, represented by the three-angle Euler rotation vector, with respect to the inertial frame.
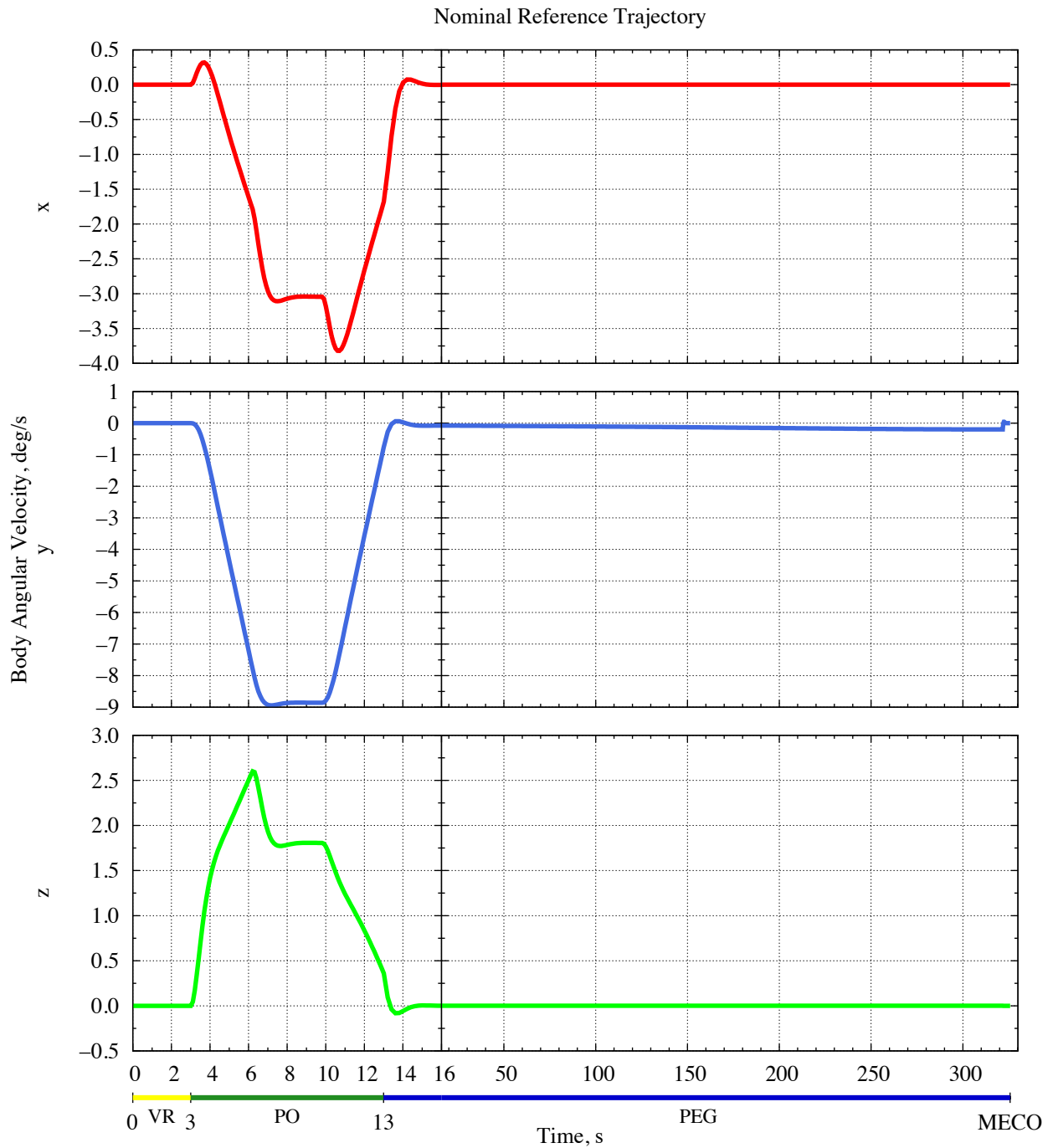
Figure 6.5: *Nominal reference trajectory.* Nominal time history of the nominal body angular velocity vector.
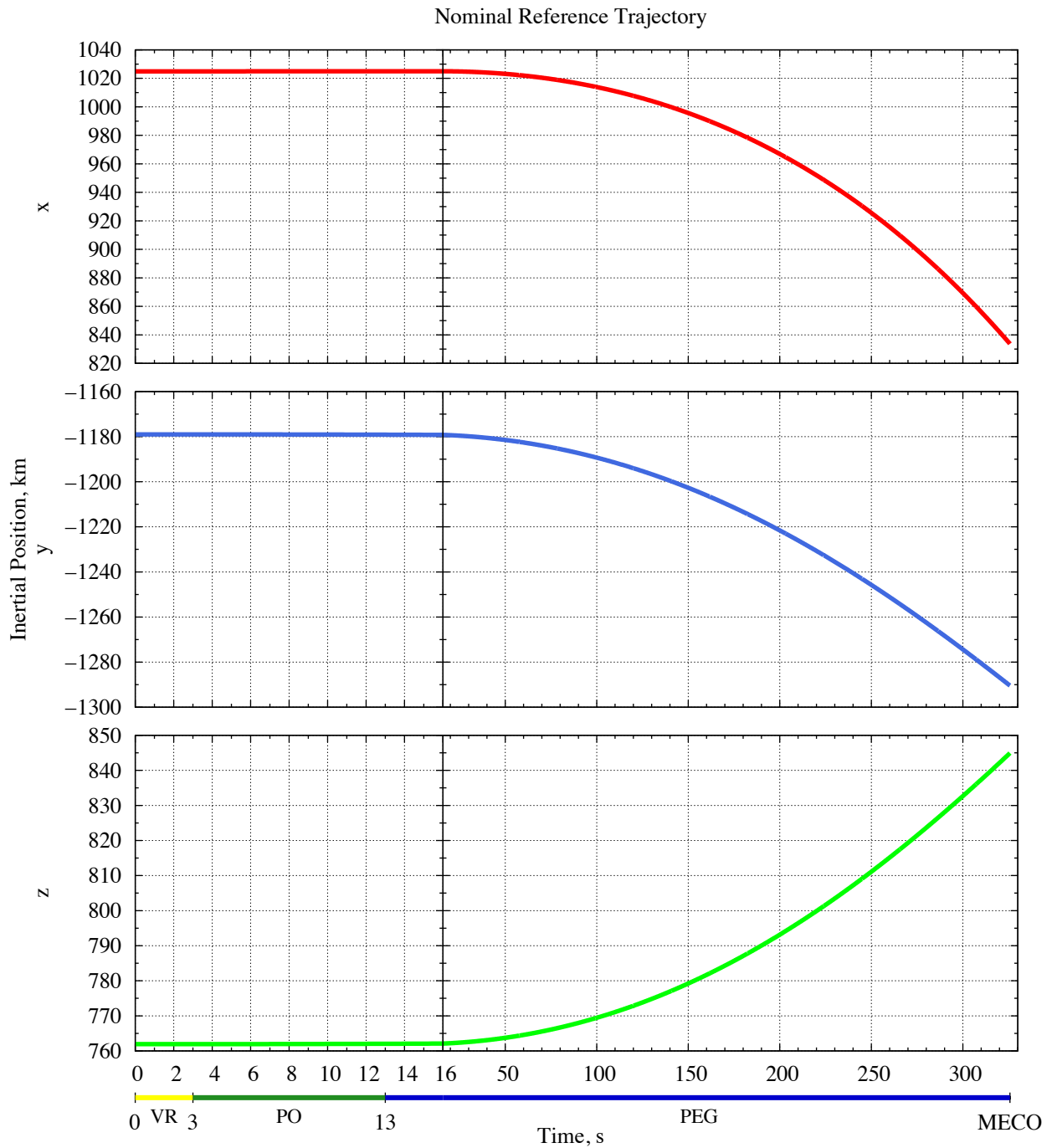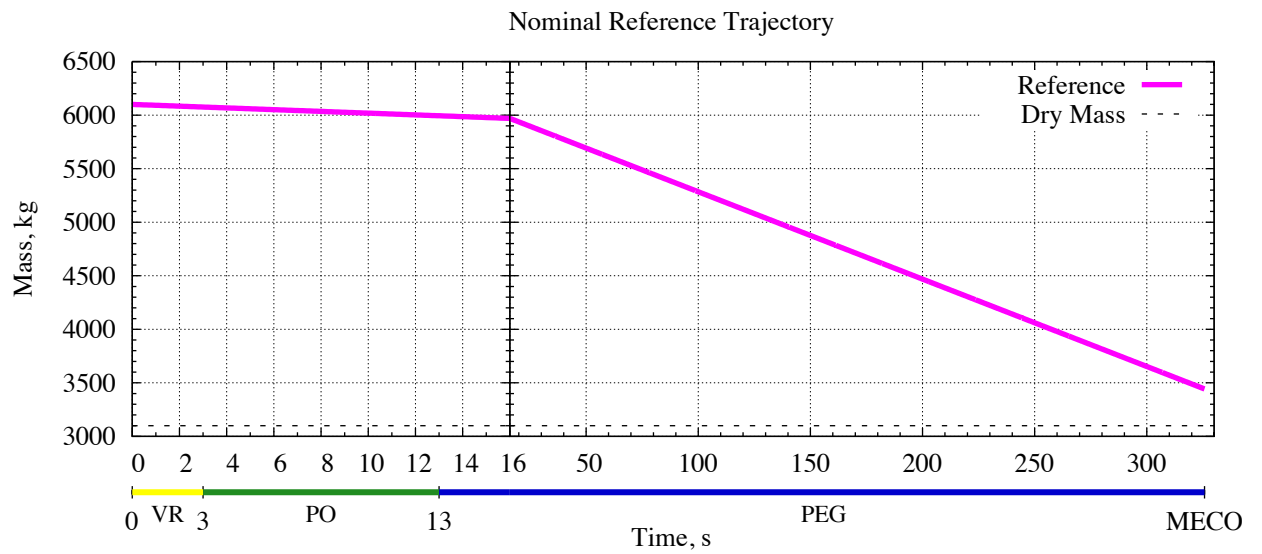
Figure 6.6: *Nominal reference trajectory.* Nominal time history of the nominal wet mass.
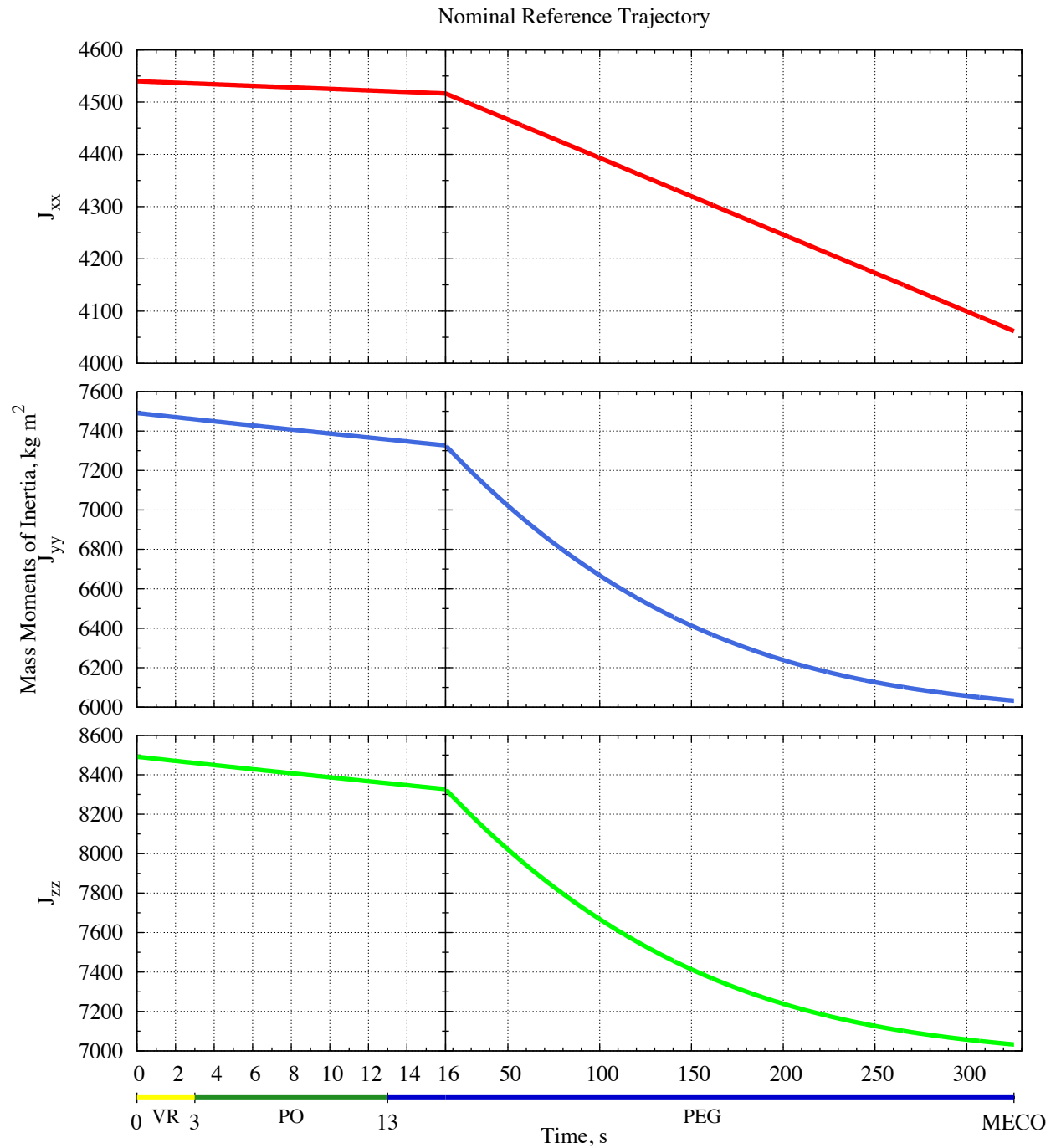
Figure 6.7: *Nominal reference trajectory.* Nominal time history of the nominal principle-axis mass moments-of-inertia (MOI).
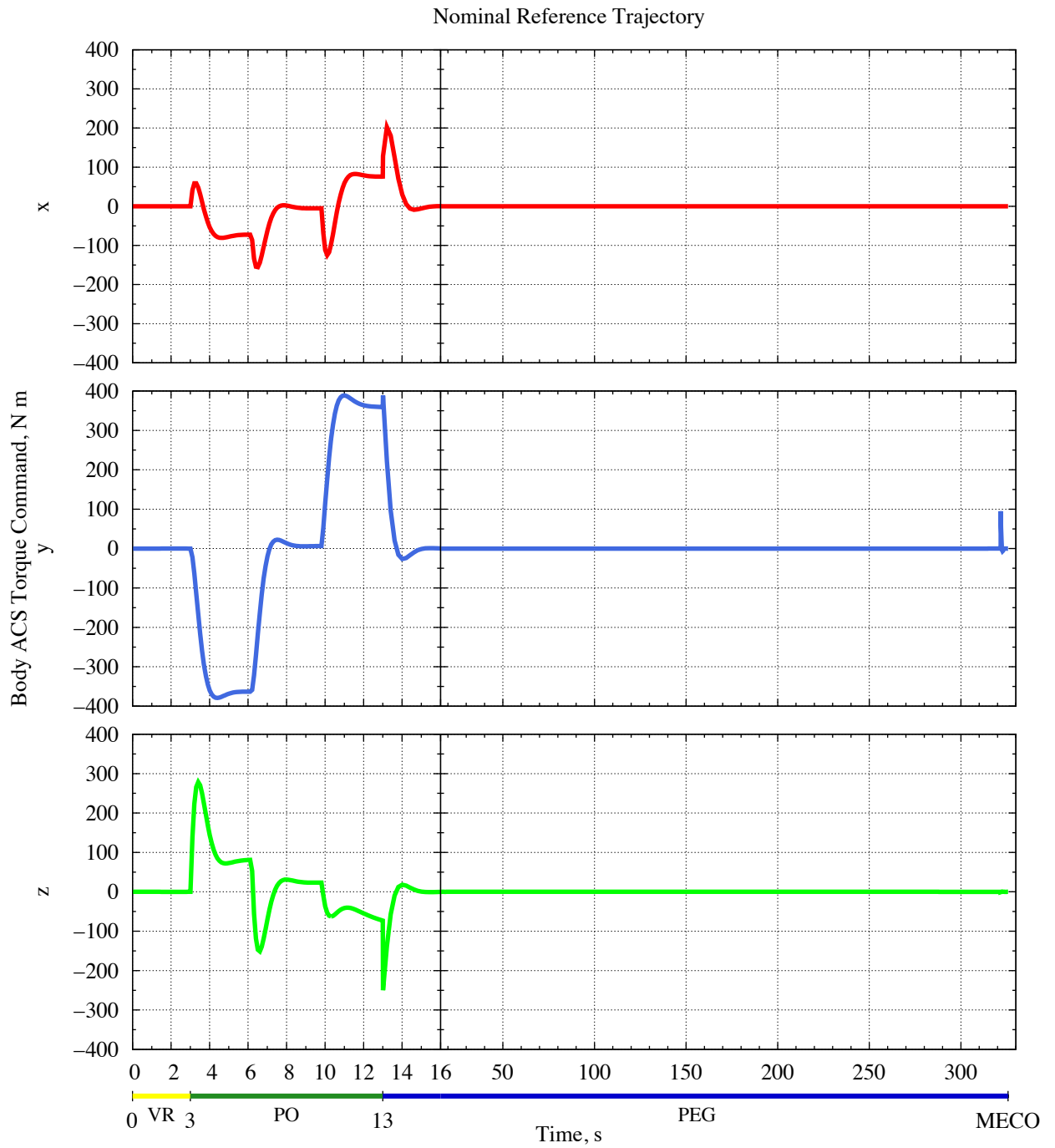
Figure 6.8: *Nominal reference trajectory.* Nominal time history of the nominal attitude control system (ACS) torque command.
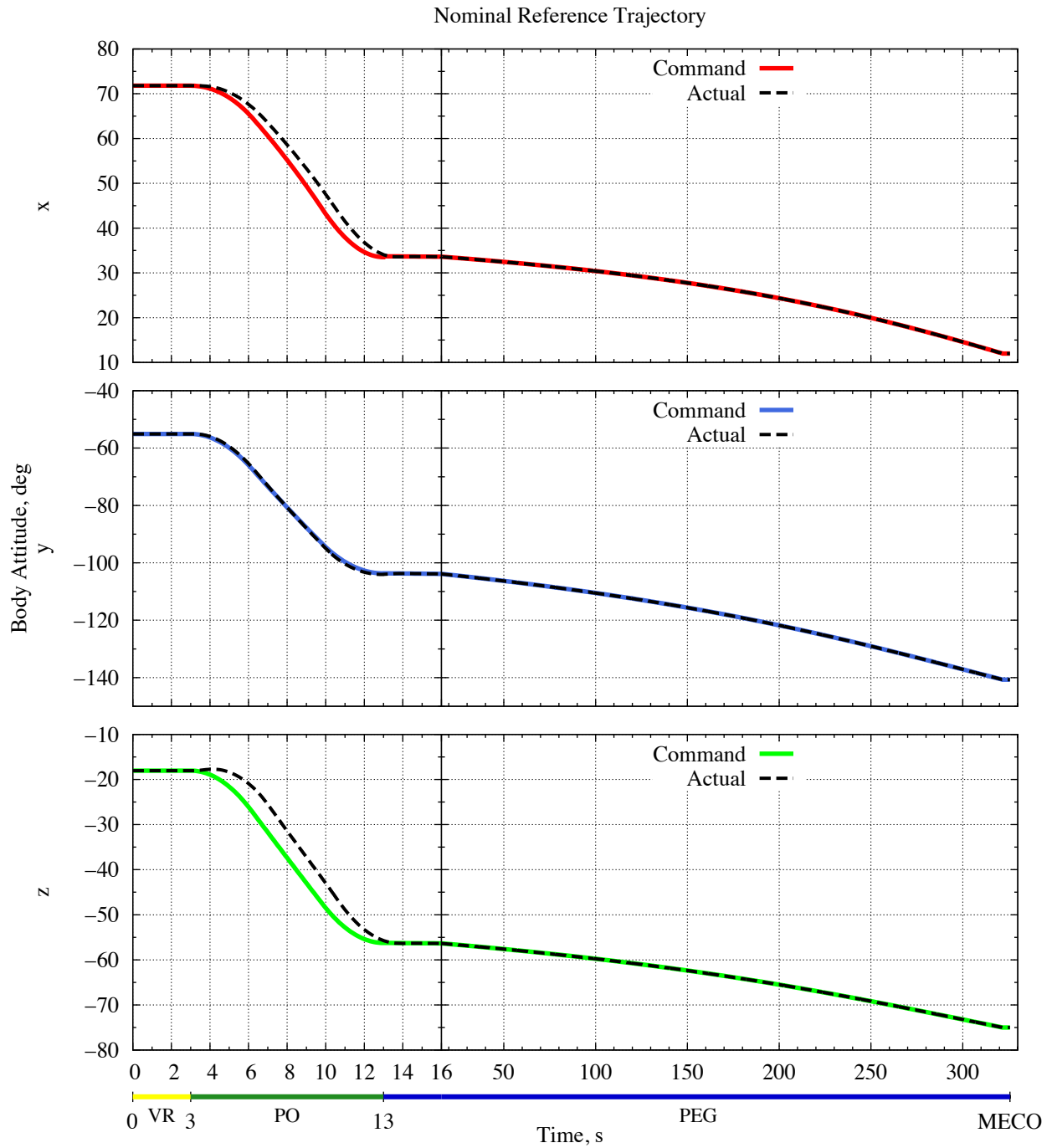
Figure 6.9: *Nominal reference trajectory.* Nominal time history comparison of the nominal commanded and nominal actual body attitude, represented by the three-angle Euler rotation vector, with respect to the inertial frame.
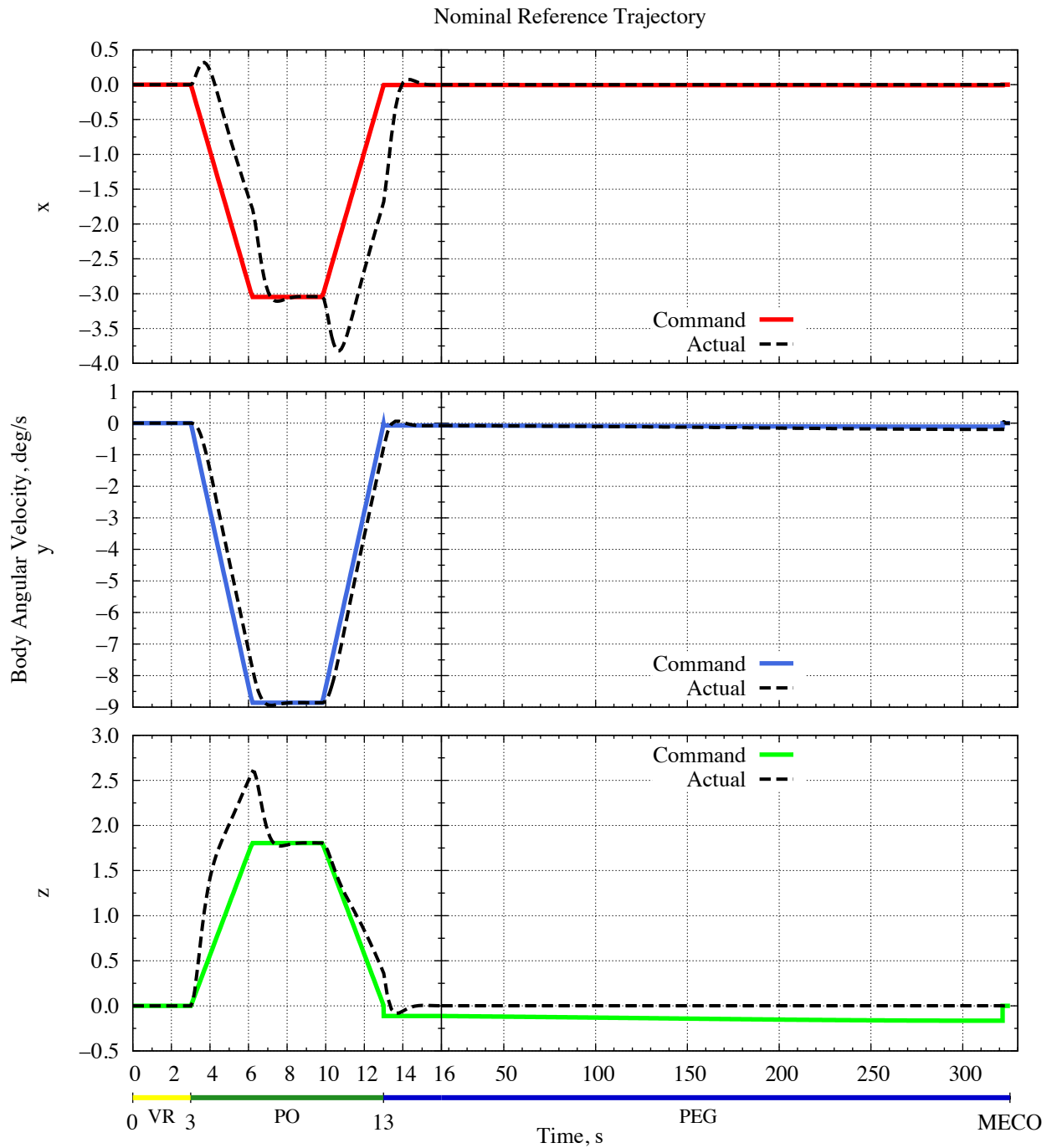
Figure 6.10: *Nominal reference trajectory.* Nominal time history comparison of the nominal commanded and nominal actual body angular velocity vectors.
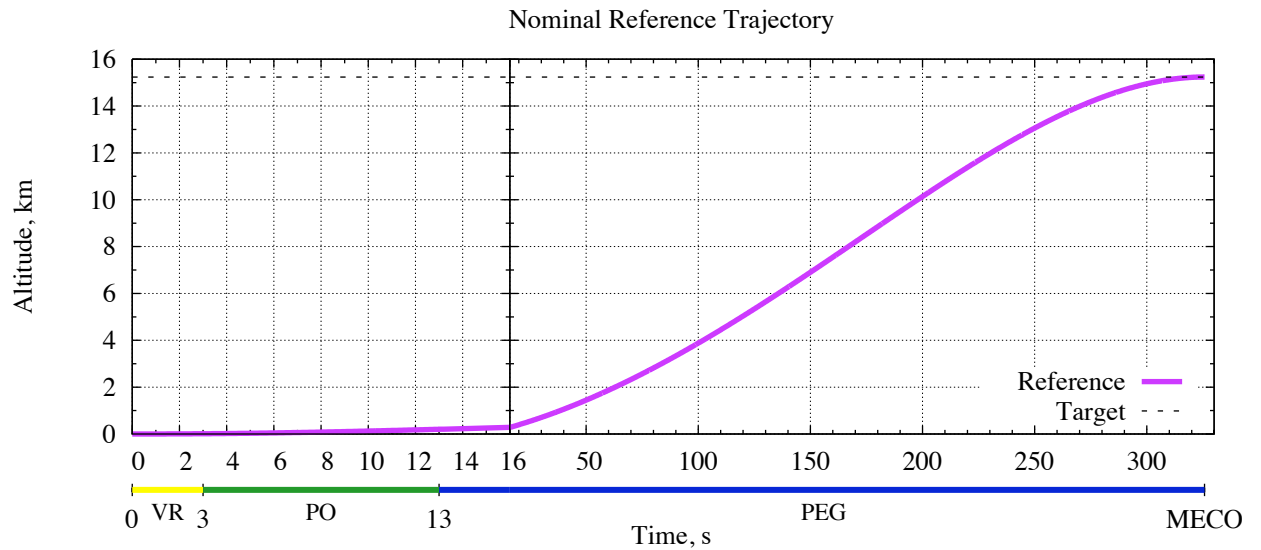
Figure 6.11: *Nominal reference trajectory.* Nominal time history of the nominal planetocentric altitude with a target value of 15.24 km.
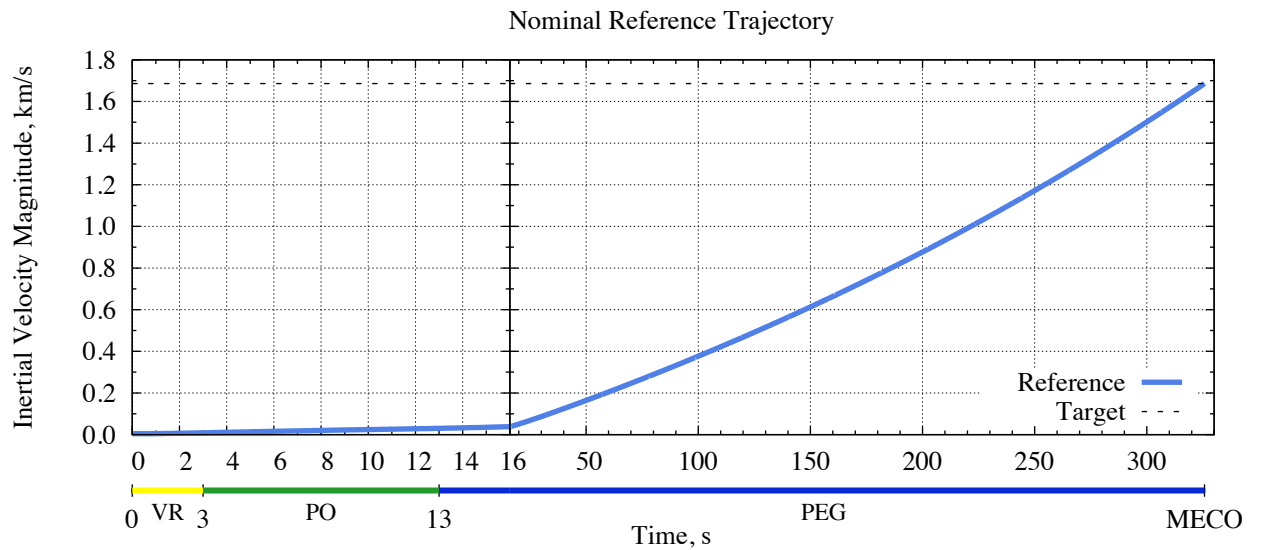
Figure 6.12: *Nominal reference trajectory.* Nominal time history of the nominal inertial velocity magnitude with a target value of 1686.156 m/s.
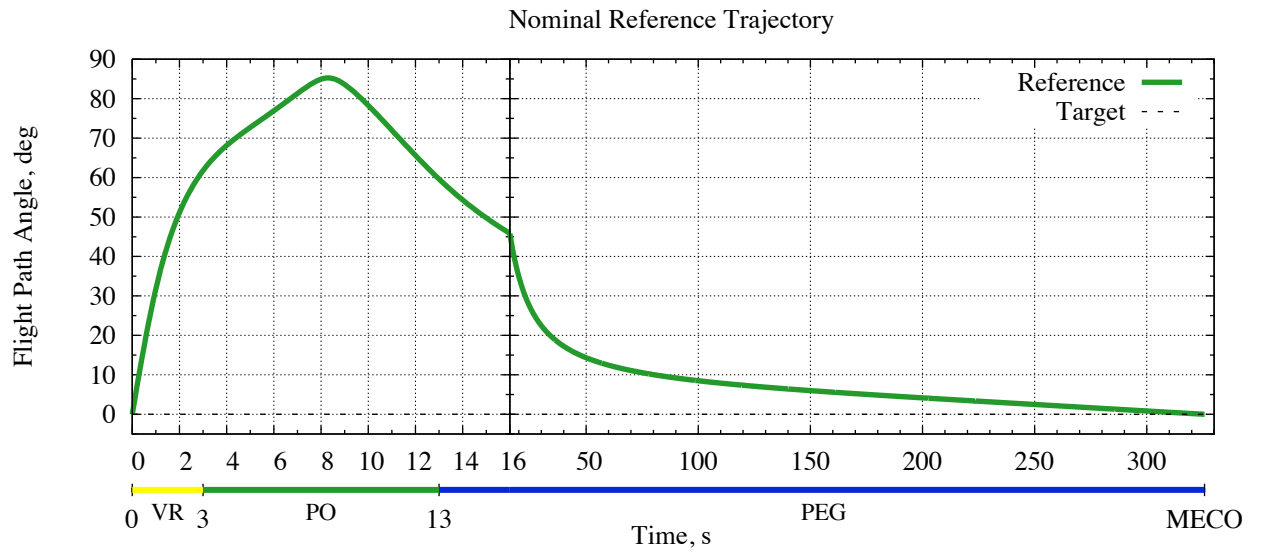
Figure 6.13: *Nominal reference trajectory.* Nominal time history of the nominal inertial flight path angle with a target value of 0 deg.
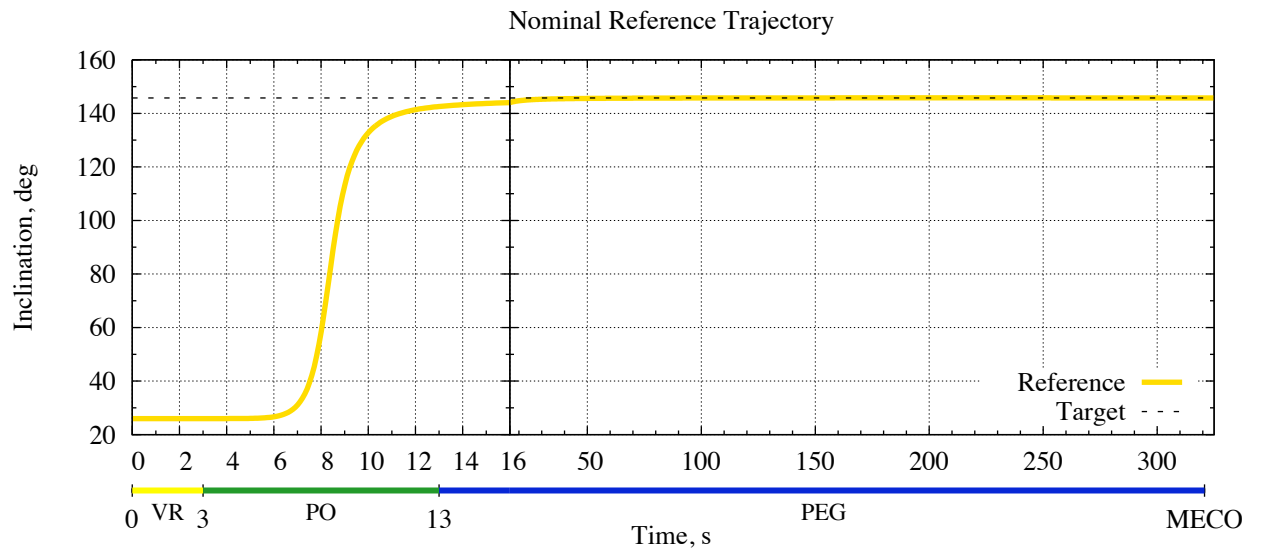
Figure 6.14: *Nominal reference trajectory.* Nominal time history of the nominal inclination angle with a target value of 145.765 deg.

Nominal Reference Trajectory
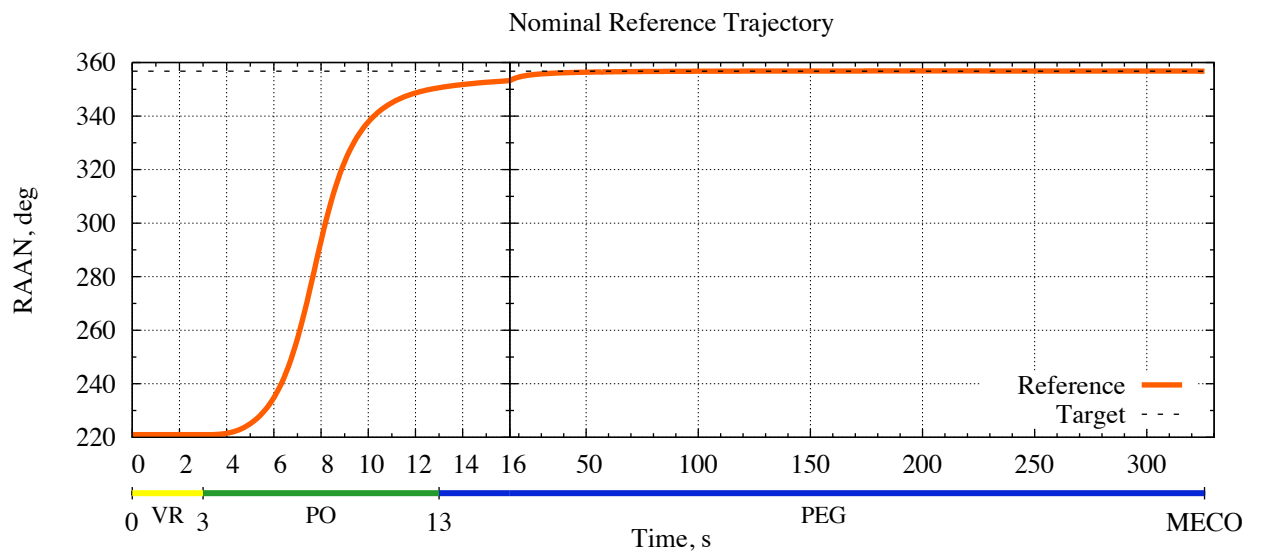
Figure 6.15: *Nominal reference trajectory.* Nominal time history of the nominal right ascension of the ascending node (RAAN) angle with a target value of 356.786 deg.

Chapter 7

Results & Discussion

"The study and knowledge of the universe would somehow

be lame and defective were no practical results to follow."

– Marcus Tullius Cicero

For this research, four studies were selected and performed using the two related but very different techniques, viz., Monte Carlo (MC) and linear covariance (LC). The first three studies are each focused on a specific group of states or parameters, i.e., vehicle states, sensor error parameters, and actuator error parameters. The purpose of these studies was to validate the implementation of the nonlinear models and algorithms in the MC simulation and the corresponding linearized models and algorithms in the LC simulation. The fourth and final study examines the performance of the entire GN&C system in the presence of sensor errors, actuator execution uncertainties, and random environment disturbances. The purpose of this study is to demonstrate the remarkable capabilities of the linear covariance simulation and analysis tool as it pertains to the thesis of the dissertation. Only the results from the fourth and final study are presented, compared, and discussed here.

The GN&C system performance study consists of three cases or levels of uncertainty: baseline, low cost, and high cost. The *baseline* case represents the nominal or typical uncertainty that one might expect for a lunar launch. The *low/high cost* cases, on the other hand, represent a tenfold *decrease/increase* in accuracy with respect to the baseline, which can in general be associated with the cost of the sensors, actuators, or other various system components. The purposes behind including the low and high cost cases are two-fold: 1) to push and test the capabilities of the linear covariance simulation beyond the nominal operating regime to see if there are areas in the trajectory where the linear approximation might no longer hold, and 2) to explore and quantify the effects (if any) that the accuracy/cost of sensors, actuators, or other system components as well as knowledge of the environment might have on system performance.

7.1 Performance Metrics

In this research, two key metrics are used to evaluate, measure, and quantify the performance of the launch vehicle GN&C system during powered ascent. They are 1) the covariance of the true state dispersions $\mathcal{D}$ and 2) the covariance of the true navigation state errors $\mathcal{P}$. Recall that the true state dispersions $\delta\boldsymbol{x}$ are defined as the difference between the true states $\boldsymbol{x}$ and the nominal states $\bar{\boldsymbol{x}}$, i.e., the deviation of the actual trajectory from the nominal reference trajectory. This deviation is caused by several contributing factors, including but not limited to actuator execution variability, GN&C algorithm selection and performance, navigation error, and random disturbances acting on the launch vehicle. The covariance of the true state dispersions is therefore a measure of how far the actual trajectory is expected to deviate from the nominal path, given the contributing factors previously mentioned. On the other hand, the true navigation state errors $\delta\boldsymbol{e}$ are defined as the difference between the true navigation states $\mathbf{x}$ and the navigation states $\hat{\boldsymbol{x}}$, i.e., the deviation of the actual trajectory from the estimated trajectory. This deviation is also caused by several contributing factors, including but not limited to sensor errors, system modeling errors, initial condition uncertainties, and errors in the quantities derived from imperfect measurements. The covariance of the true navigation state errors is therefore a measure of how far the navigation filter's estimate or knowledge of the actual trajectory is expected to deviate from the actual trajectory, given the contributing factors listed above.

7.1.1 Plots

Before proceeding with the presentation and discussion of the results, a brief introduction to the type of plots that will be used to present and compare the results from MC and LC analysis is in order. The most commonly used type of plot is the nominal time history of the $\pm 3\sigma$ bounds, where σ is the standard deviation or square root of the variance, and the term 'nominal' refers only to the time line or horizontal plot axis. An example is given in Figure 7.1, which shows the nominal time history of the planetocentric altitude true state dispersions $\pm 3\sigma$. In MC analysis, the time histories of the true state dispersions $\delta\boldsymbol{x}(t)$ and true navigation state errors $\delta\boldsymbol{e}(t)$ from N simulations are computed and collected. Figure 7.2 (top plot) shows the nominal time history of the planetocentric altitude true state dispersions from 150 MC simulations (or samples). From these N samples, the covariance of the true state dispersions $\mathcal{D}(t)$ and the true navigation state

Figure 7.1: *Sample results.* Nominal time history comparison of the planetocentric altitude true state dispersions $\pm 3\sigma$ from MC and LC analysis.

errors $\boldsymbol{\mathcal{P}}(t)$ are computed respectively using

$$\boldsymbol{\mathcal{D}}(t) = \frac{1}{N-1} \sum_{i=1}^{N} \delta \boldsymbol{x}_i(t) \left[\delta \boldsymbol{x}_i(t) \right]^{\mathsf{T}} , \qquad (7.1)$$

$$\boldsymbol{\mathcal{P}}(t) = \frac{1}{N-1} \sum_{i=1}^{N} \delta \boldsymbol{e}_i(t) \left[\delta \boldsymbol{e}_i(t) \right]^{\mathsf{T}} , \qquad (7.2)$$

and from the covariance matrices the associated $\pm 3\sigma$ bounds are obtained. The $\pm 3\sigma$ bounds are significant in that, for a normal or Gaussian distribution, they will contain 99.73% of the samples, as illustrated in Figure 7.2 (bottom plot). LC is able to generate the same statistical $\pm 3\sigma$ bounds, as shown in Figure 7.1, but with only **one** simulation run. The gap in the MC data near the end of the trajectory, which is most obvious in Figure 7.2, is due to the time-varying nature of the guidance mode change from PEG to fine count and the result of resetting the simulation clock (during post processing) to a common time $\bar{t}_{\mathrm{e}}$ for each sample run as the event is triggered, such that all subsequent results are then measured with respect to the common time $\bar{t}_{\mathrm{e}}$. This is key because doing so allows for direct comparison of the MC and LC time history results.

7.1.2 Fractional Error

It is important to recall and note that the statistical accuracy of MC analysis, i.e., the error in the estimate of the true standard deviation, is a function of and determined by the number of MC samples obtained. For example, the estimated or computed standard deviation from 100 MC

True State Dispersions

True State Dispersions
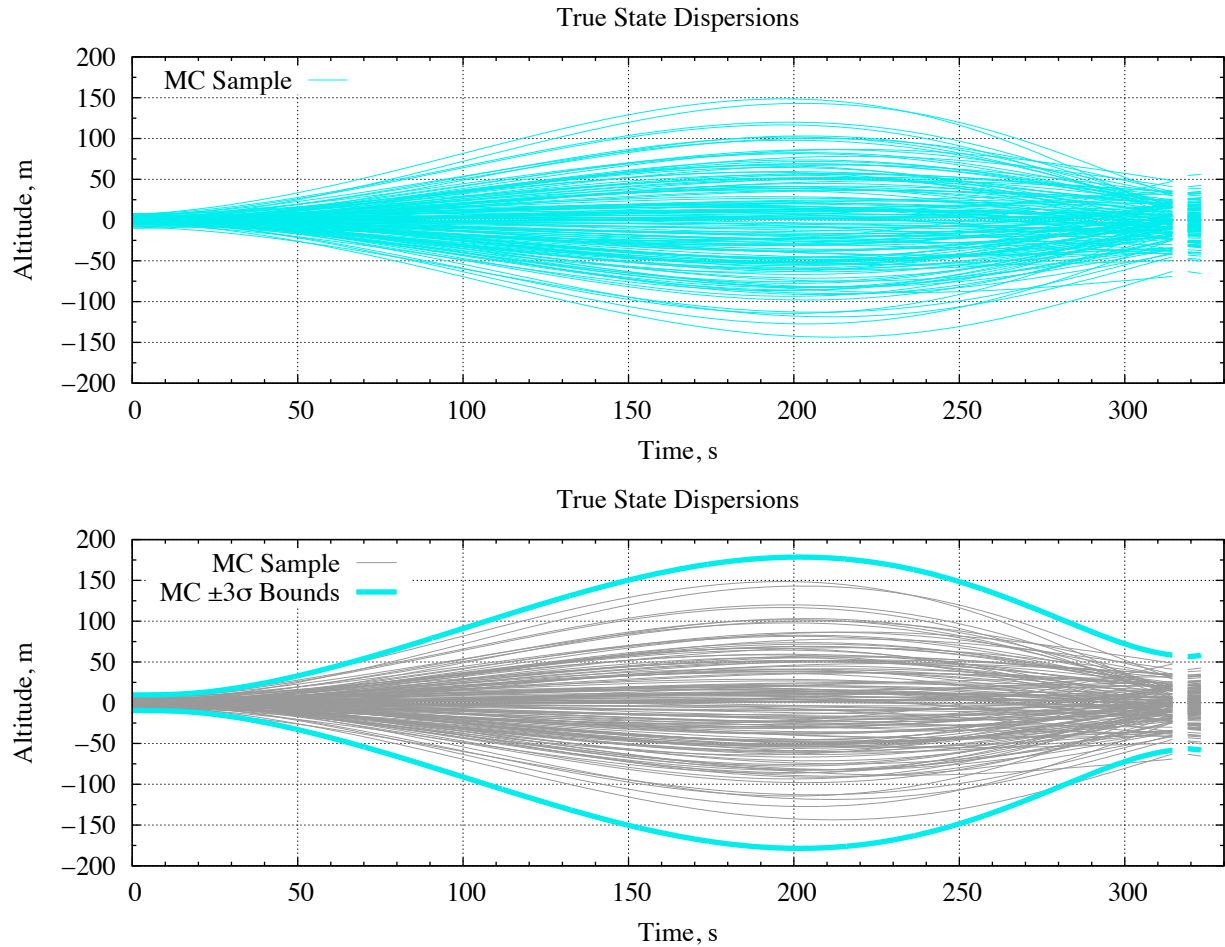
Figure 7.2: *Sample results.* Nominal time history of the planetocentric altitude true state dispersions. (*top*) Hair plot of 150 MC samples. (*bottom*) For a normal or Gaussian distribution, 99.73% of the MC samples will fall within the $\pm 3\sigma$ bounds.

samples would not be as accurate a representation of the true standard deviation as would that of 1000 MC samples. To quantify this, the *fractional error* ϵ_σ in the estimate of the true standard deviation is used, which is given by [84]

$$\epsilon_\sigma = \frac{1}{\sqrt{2(N-1)}} \, , \tag{7.3}$$

where N is the number of MC samples. Note that the fractional error is the same for any multiplier of the standard deviation σ (i.e., 1σ, 2σ, 3σ, etc.). This is due to the fact that any multiplier applied to the estimated value must also be applied to the true value, and since the fractional error is essentially a ratio it can be shown that the multipliers cancel one another. From the expression above, it follows that the fractional error for $N = 300$, 500, and 1000 is respectively 4.1%, 3.2%,
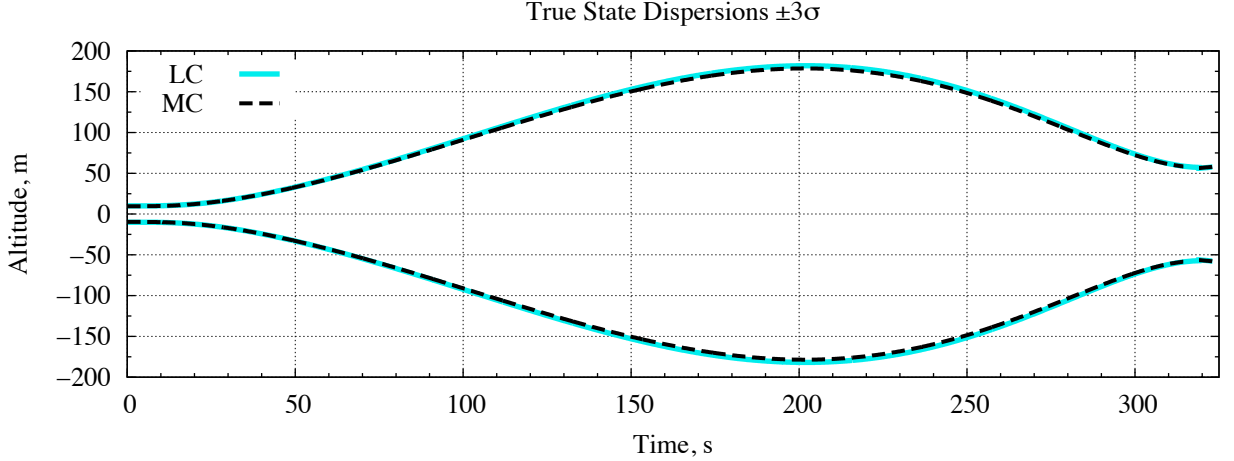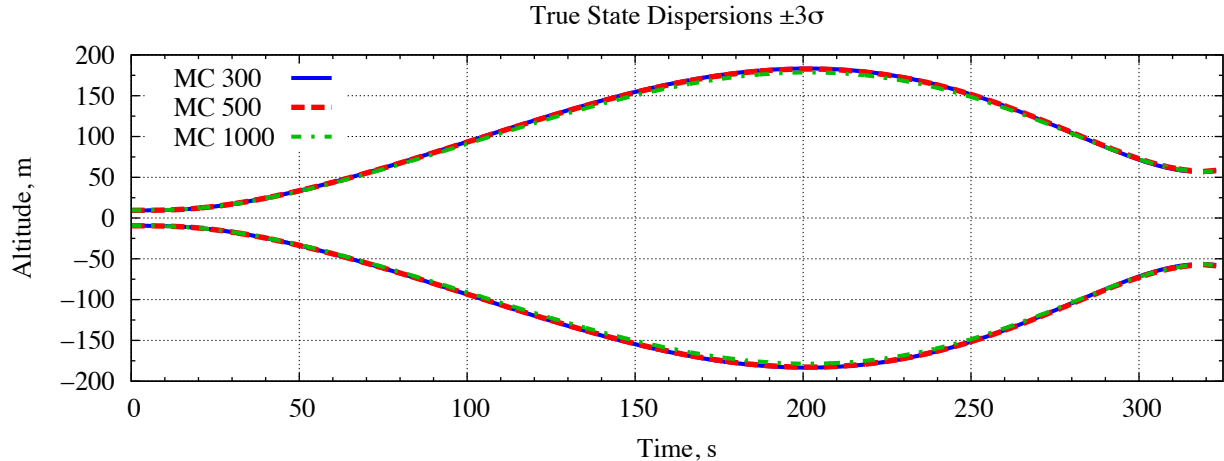
Figure 7.3: *Sample results.* Nominal time history comparison of the planetocentric altitude true state dispersions $\pm 3\sigma$ from 300, 500, and 1000 MC samples.

and 2.2%. There are two important things to take away from this: 1) there is only a 1% increase in accuracy by doubling the number of MC samples from 500 to 1000, and 2) the fractional error is never zero, not even for 100,000 MC samples. Now, it is one thing to talk about the fractional error associated with the number of sample runs and quite another to visualize graphically the impact that the number of runs has on the actual results. Figure 7.3 was generated for this purpose, which shows the nominal time history of the planetocentric altitude true state dispersions $\pm 3\sigma$ for 300, 500, and 1000 MC samples. In this example, it is readily observed that the three sets of results are nearly identical. Admittedly, this may not always be the case. However, after taking all of these things into consideration, it was decided that 500 MC samples and a corresponding fractional error of 3.2% would be sufficient (i.e., provide enough accuracy) for the needs of this research.

7.1.3 Normalized Percent Difference

When comparing the results from MC and LC analysis, it is of primary interest to this research to provide a quantitative measure of just how much the two sets of results differ. The most common method and the one selected for this research is the *normalized percent difference*, which in this case is MC normalized, as given by

$$\% \, \text{Diff.} = \frac{\left|\sigma_{\text{MC}} - \sigma_{\text{LC}}\right|}{\sigma_{\text{MC}}} * 100 \,, \tag{7.4}$$

where $|\cdot|$ denotes the absolute value operator and $*$ denotes the multiplication operator. It is
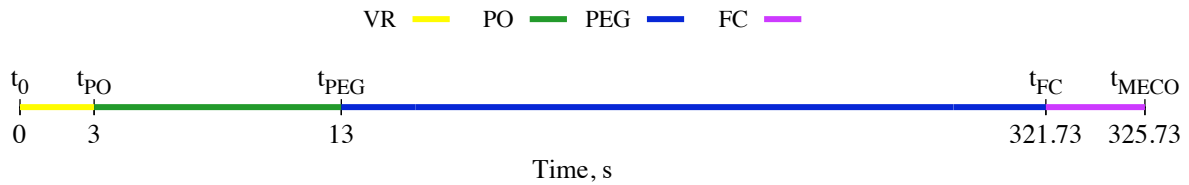
Figure 7.4: *Nominal reference trajectory.* Guidance modes and five key points along the nominal reference trajectory (NRT).

important to note that this method fails when either σ_{MC} or σ_{LC} is exactly zero. For example, if σ_{LC} is zero and σ_{MC} is nonzero, then the percent difference is 100%, regardless of the actual value of σ_{MC}. Furthermore, if σ_{MC} is zero then a division by zero occurs in Eq. (7.4), which makes the percent difference undefined, regardless of the value of σ_{LC}. This measure is hereafter computed at five key points along the trajectory, which are illustrated in Figure (7.4) and described in greater detail as follows:

- t_0 ($t = 0$ s) engine ignition, guidance in VR mode, guidance flying open-loop;

- t_{PO} ($t = 3$ s) guidance switches from VR to PO mode, guidance continues flying open-loop;

- t_{PEG} ($t = 13$ s) guidance switches from PO to PEG mode, guidance switches to closed-loop;

- t_{FC} ($\bar{t} = 321.73$ s) guidance switches from PEG to fine count (FC), event trigger occurs, guidance switches back to open-loop;

- t_{MECO} ($\bar{t} = 325.73$ s) MECO occurs, end of simulation.

7.2 GN&C System Performance Study

The purpose of this study is to examine the performance of the entire GN&C system in the presence of sensor errors, actuator execution variabilities, and random environment disturbances. As such, the uncertainty values in the initial conditions of all of the true states (i.e., vehicle states, sensor parameters, actuator parameters, and environment disturbance parameters) are considered. Since the inertial navigation filter operates unaided, i.e., no other measurements are available besides those from the IMU, it follows that only the inertial position, inertial velocity, and body attitude navigation states are estimated. This is because without additional sensor measurements the sensor and actuator error parameters are not observable.

7.2.1 Baseline Case

The 3σ uncertainty values in the initial conditions of the true states and initial knowledge of the navigation states employed in this *baseline* case of the GN&C system performance study are provided in Tables 7.1 and 7.2. Recall that the *baseline* case represents the nominal or typical uncertainty that one might expect for a lunar launch. For a launch vehicle that is initially sitting on the launch pad, it is perhaps more intuitive to define the uncertainty in the initial value of the inertial position state in terms of the planetographic coordinate system (i.e., north-east-down) and then later transform it to the inertial coordinate system. Hence, the 3σ uncertainty values associated with the initial inertial position state, specified in Tables 7.1 and 7.2, are expressed in planetographic coordinates rather than inertial coordinates. Moreover, it can be shown that due to the nature of the problem the initial inertial velocity state is highly correlated with the initial inertial position state. As a result, any uncertainty in the initial inertial velocity state is due solely to the uncertainty in the initial inertial position state. For example, an uncertainty of 100.0 m (3σ) in the north/south and east/west directions and 10.0 m (3σ) in the up/down directions of the inertial position corresponds to an uncertainty of 0.27 mm/s (3σ) in the north/south and east/west directions and 0 mm/s (3σ) in the up/down directions in the inertial velocity. Hence, the uncertainty values associated with the initial inertial velocity state, given in Tables 7.1 and 7.2, are not user-defined but rather the result of the values chosen for the initial inertial position state. Furthermore, it can be shown that the same applies to the initial body angular velocity state, i.e., it is highly correlated with the initial body attitude state error and thus the uncertainty in the initial angular velocity is solely determined by the uncertainty in the initial attitude of the launch vehicle. Lastly, the baseline 3σ value specified for the uncertainty in the initial wet mass is 1% of the initial nominal wet mass of the launch vehicle.

The 3σ values specified in Table 7.1 associated with the gyro and accelerometer sensor error parameters are consistent with those used in [85] and [12], whereas the 3σ values associated with the thruster and ACS actuator error parameters are based upon a 1% error or uncertainty, i.e., for a scale factor with units of parts-per-million (PPM), a 1% error corresponds to 1% of 1 million parts or 1×10^4 PPM. For a misalignment with units of radians, a 1% uncertainty corresponds to 1% of 1 rad or 0.01 rad. However, for the bias and noise, 1% of the nominal thrust and 1% of the nominal torque are respectively employed for the thruster and ACS. Furthermore, it is assumed

that the thruster bias will only affect the thrust vector along the nominal thrust direction or body x-axis. Accordingly, a value of zero is specified for the thruster bias in the body y- and z-axis directions. Lastly, it is assumed that all of the sensor error parameters and the majority of the actuator error parameters will be nearly constant throughout the ascent burn. Hence, the associated time constants τ are set to values much larger than the nominal ascent burn or 1×10^8 s. The notable exception is the thruster misalignment, which could conceivably vary more rapidly due to fluctuations in the exhaust velocity vector. Thus, the thruster misalignment time constant is set to a conservative value of 10 s.

The translational and rotational acceleration amplitudes listed in Table 7.1 correspond to the amplitudes of the sinusoidal disturbance models defined in Eqs. (3.10) and (3.11) on page 20, which represent a crude attempt at modeling the effects of fuel slosh on the vehicle dynamics. The actual amplitudes and frequencies of the sinusoidal disturbances associated with fuel slosh are dependent upon many factors, e.g., fuel mass, fuel tank geometry, fuel damping mechanisms, vehicle mass, moments-of-inertia, and attitude control law design. It is thus impossible to know exactly what the frequencies and amplitudes will be for the lunar ascent vehicle at this time. However, LC analysis provides a way for the GN&C engineer to evaluate the GN&C system performance over a range of frequencies and amplitudes in a fraction of the time that it would take using MC techniques. In this research, a single disturbance frequency $\omega_{\mathrm{od}} = 2.33$ rad/s (0.37 Hz) is considered, which is consistent with those employed in [86] and derived from the following expression expression for a cylindrical fuel tank geometry [87]

$$\omega_{\mathrm{od}} = \sqrt{\frac{2\,g}{r}} \,, \tag{7.5}$$

where $g = 1.6\,\mathrm{m/s}^2$ is the acceleration due to gravity (in this case, that of the moon) and $r = 0.6$ m is the radius of the cylindrical fuel tank. A simple pendulum model, based upon [88], was used to generate the 3σ values associated with the translational and rotational acceleration amplitudes specified in Table 7.1.

Launch Vehicle States

The launch vehicle's ascent trajectory is completely represented by the five launch vehicle states: inertial position, inertial velocity, body attitude, body angular velocity, and wet mass. Figures 7.5–7.14 show the nominal time history results from MC and LC analysis of the true

Table 7.1: *GN&C system performance study, baseline case.* Summary of true states and associated initial conditions 3σ.

Error Source		3σ Uncertainty		τ, s
Vehicle States	Position	100.0 m	(north/south, east/west)	—
		10.0 m	(up/down)	—
	Velocity	0.27 mm/s	(north/south, east/west)	—
		0 mm/s	(up/down)	—
	Attitude	1.0 deg	(per axis)	—
	Angular Velocity	2.4×10^{-6} deg/s	(x axis)	—
		1.2×10^{-6} deg/s	(y axis)	—
		2.7×10^{-6} deg/s	(z axis)	—
	Mass	60.0 kg		—
Gyro	Scale Factor	4.8 PPM	(per axis)	1×10^8
	Misalignment	60.0 arcsec	(per axis)	1×10^8
	Bias	0.06 deg/hr	(per axis)	1×10^8
	Noise	1.5×10^{-4} deg/$\sqrt{\text{s}}$	(per axis)	—
Accelerometer	Scale Factor	198.0 PPM	(per axis)	1×10^8
	Misalignment	60.0 arcsec	(per axis)	1×10^8
	Bias	90.0 μg	(per axis)	1×10^8
	Noise	5.1×10^{-4} m/s$\sqrt{\text{s}}$	(per axis)	—
Thruster	Scale Factor	1×10^4 PPM	(per axis)	1×10^8
	Misalignment	0.01 rad	(per axis)	10.0
	Bias	256.0 N	(x axis)	1×10^8
		0 N	(y axis, z axis)	—
	Noise	256.0 N/$\sqrt{\text{s}}$	(per axis)	—
ACS	Scale Factor	1×10^4 PPM	(per axis)	1×10^8
	Misalignment	0.01 rad	(per axis)	1×10^8
	Bias	4.45 N-m	(per axis)	1×10^8
	Noise	4.45 N-m/$\sqrt{\text{s}}$	(per axis)	—
Translational Acceleration	Amplitudes 1, 2	0.03 m/s^2	(per axis)	1×10^8
Rotational Acceleration	Amplitudes 1, 2	0.015 rad/s^2	(per axis)	1×10^8

Table 7.2: *GN&C system performance study, baseline case.* Summary of true and filter navigation states and associated initial conditions 3σ.

Error Source		3σ Uncertainty		τ, s
Vehicle States	Position	100.0 m	(north/south, east/west)	—
		10.0 m	(up/down)	—
	Velocity	0.27 mm/s	(north/south, east/west)	—
		0 mm/s	(up/down)	—
	Attitude	0.1 deg	(per axis)	—

state dispersions $\pm 3\sigma$ and true navigation state errors $\pm 3\sigma$ associated with the five launch vehicle states.[1] The true state dispersions are a measure of how far the actual flight path is expected to deviate from the nominal flight path, whereas the true navigation state errors are a measure of how far the navigation filter's estimate (or knowledge) of the actual trajectory is expected to deviate from the actual trajectory. Hence, the corresponding $\pm 3\sigma$ bounds respectively represent the *performance envelope* of the GN&C system and navigation filter. Furthermore, the position and velocity, attitude and angular velocity, and wet mass true state dispersions are also known as *trajectory control dispersions*, *attitude control dispersions*, and *propellant dispersions*, respectively.

All of the figures associated with the first four launch vehicle states (i.e., position, velocity, attitude, and angular velocity) contain plots of the x, y, and z components as well as a plot of the *root-sum-square* (RSS) of the components, which represents the *total true state dispersions* or *total true navigation state errors* of a given state. In order to provide a quick and easy way to visually compare these results, the vertical axes of the x, y, and z results are plotted on the same scale, and the vertical axes of the true navigation state error results are plotted on the same scales as the corresponding true state dispersion results. Occasionally, important transients or trends are inadvertently masked by the scaling of the vertical or horizontal axes of a given plot and are therefore somewhat difficult to discern or recognize. Figures 7.9 and 7.12 are a good example of this. Consequently, Figures 7.10 and 7.13 have been added, which provide a zoomed-in view of the first and last 16 s of Figures 7.9 and 7.12, respectively.

It is important to note that the gap in the MC data, near the end of the trajectory, is due to

[1]Note that there are no true navigation state error results for the body angular velocity and wet mass states. This is due to them not being estimated by the navigation filter, as explained in § 3.2.1.

the time-varying nature of the guidance mode change from PEG to fine count event and the result of resetting the simulation clock (during post processing) for each sample run to a common time $\bar{t}_e$ (i.e., the nominal time of the event) as the event is triggered, such that all subsequent results are then measured with respect to the common time $\bar{t}_e$. This allows the MC and LC results to be plotted together. The abrupt change in the LC true state dispersion results, towards the end of the trajectory, is also due to the guidance mode change from PEG to fine count event and the result of applying covariance shaping to the augmented state covariance matrix at the nominal time of the event, as explained in § 5.5.4. Notice that, as expected, the covariance shaping has no affect on the LC true navigation state error results.

It is observed in Figures 7.5–7.14 that the MC and LC results match exceptionally well. In many of the plots, the two sets of results are practically indistinguishable. In order to get at better sense of just how much they actually differ, the normalized percent difference was computed at five key points along the trajectory, as explained in § 7.1.3. Tables 7.3–7.6 list the MC and LC 3σ values and corresponding normalized percent difference for the true state dispersions and true navigation state errors (denoted respectively as $\delta\boldsymbol{x}$ and $\delta\boldsymbol{e}$) associated with the five launch vehicle states. Recall that the percent difference measure fails when either the MC or LC σ value is exactly zero. Such a failure occurs at time t_0 in the z-component of the inertial velocity true state dispersions and true navigation state errors, as shown in Table 7.4, where the LC values are exactly zero, thereby resulting in a percent difference of 100%. The corresponding MC values were originally zero but were made nonzero because the inertial position and inertial velocity states are highly correlated, such that several of the eigenvalues of the initial true state dispersion and true navigation state error covariance matrices were negative. This caused problems in the MC simulation state initialization routine because the states of the system are initialized with the eigenvalues and eigenvectors of the covariance matrices and the routine requires that the eigenvalues be nonnegative. To circumvent this problem, some additional variance was added to the diagonal elements of the inertial velocity state covariance until the negative eigenvalues disappeared.

Orbit Insertion

Orbit insertion dispersion results at MECO for the burn time, guidance target parameters, and propellant expenditure are provided in Table 7.7, which includes the nominal, MC and LC 3σ,

True State Dispersions ±3σ – GN&C System Performance Study, Baseline Case

Figure 7.5: *GN&C system performance study, baseline case.* Nominal time history comparison of the inertial position true state dispersions ±3σ from MC and LC analysis.

Figure 7.6: *GN&C system performance study, baseline case.* Nominal time history comparison of the inertial position true navigation state errors $\pm 3\sigma$ from MC and LC analysis.

Figure 7.7: *GN&C system performance study, baseline case.* Nominal time history comparison of the inertial velocity true state dispersions ±3σ from MC and LC analysis.

Figure 7.8: *GN&C system performance study, baseline case.* Nominal time history comparison of the inertial velocity true navigation state errors $\pm 3\sigma$ from MC and LC analysis.

Figure 7.9: (1 of 2) *GN&C system performance study, baseline case.* Nominal time history comparison of the body attitude true state dispersions ±3σ from MC and LC analysis.
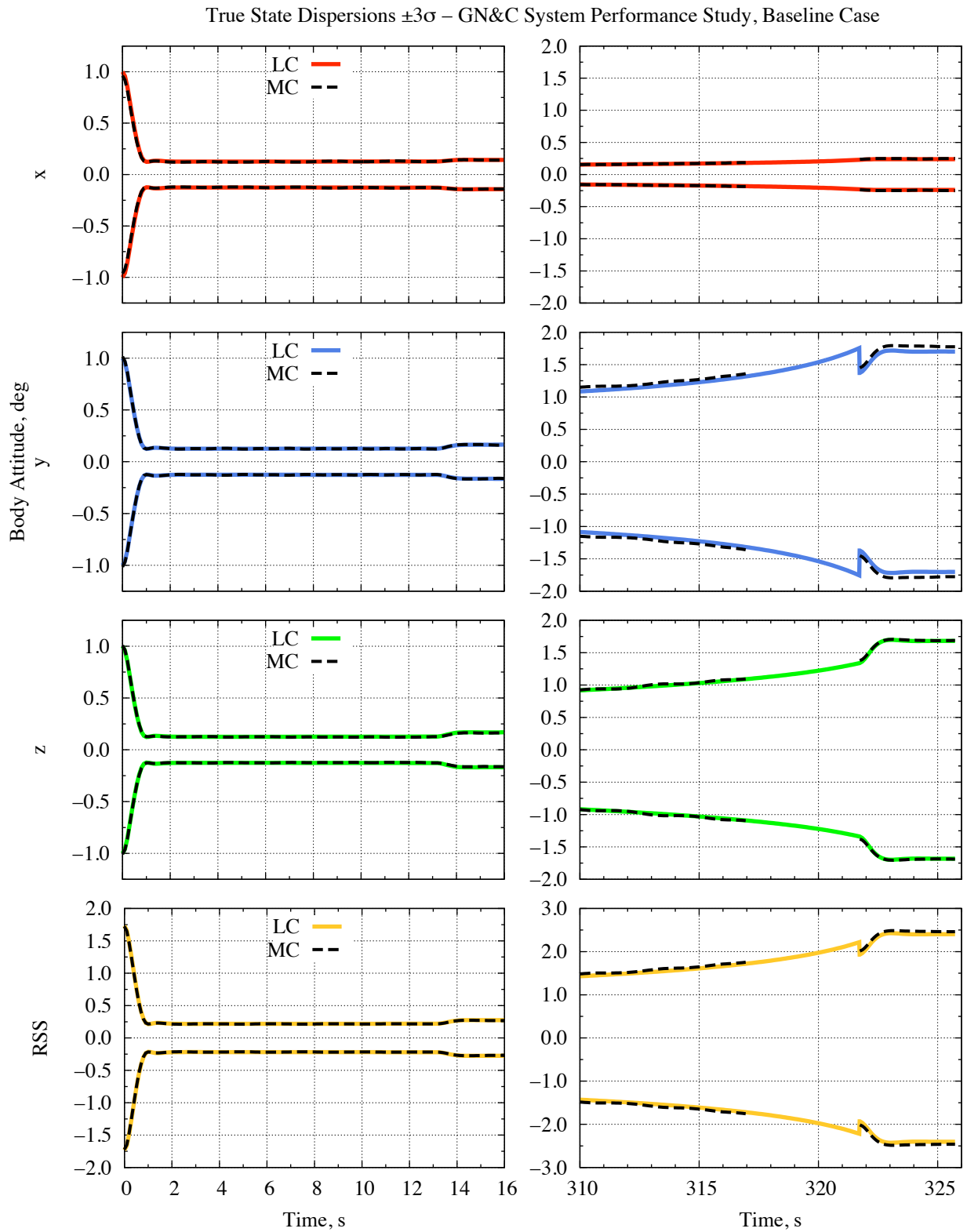
Figure 7.10: (2 of 2) *GN&C system performance study, baseline case.* Nominal time history comparison of the body attitude true state dispersions $\pm 3\sigma$ from MC and LC analysis. Zoomed-in view of the first and last 16 s.
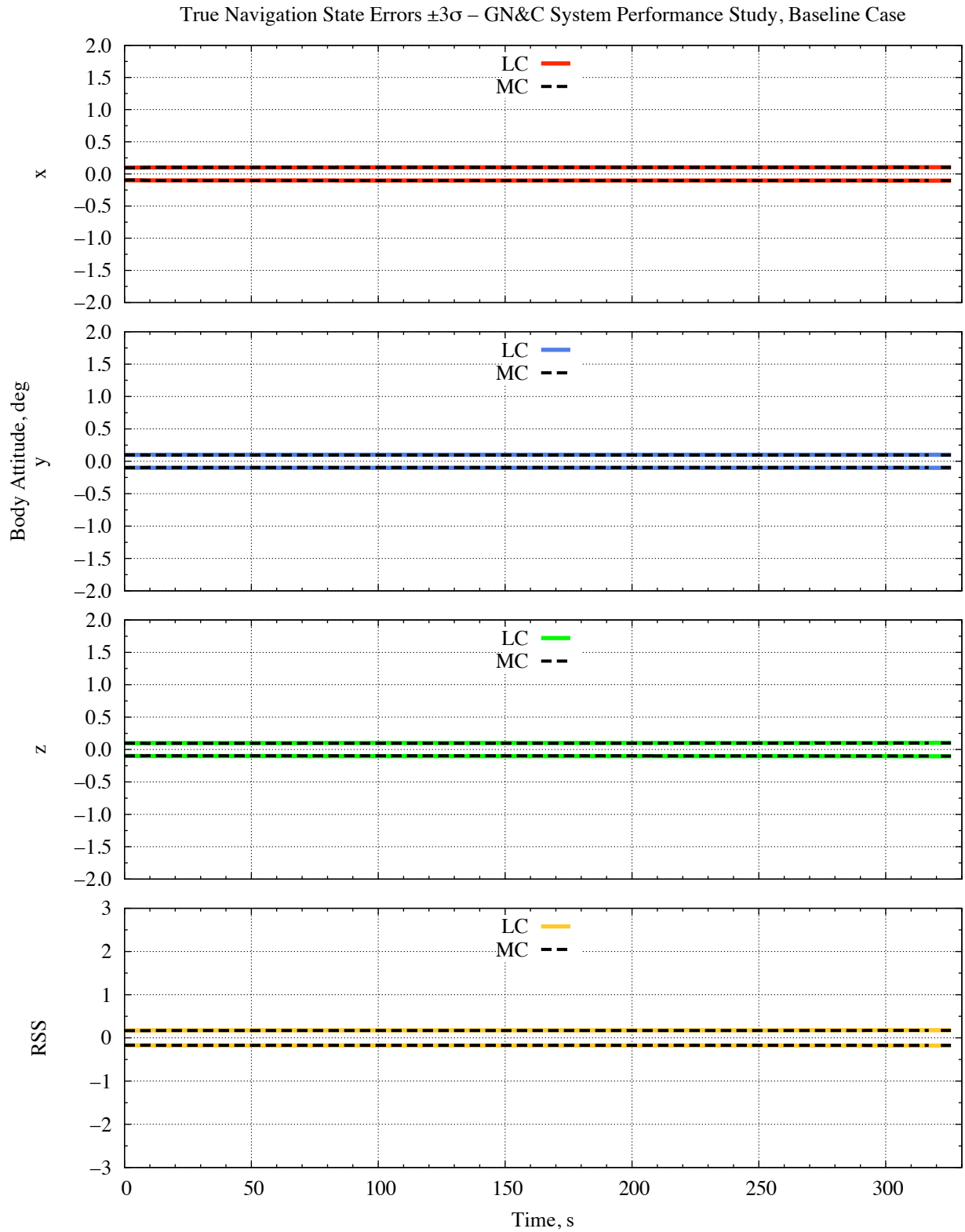
Figure 7.11: *GN&C system performance study, baseline case.* Nominal time history comparison of the body attitude true navigation state errors ±3σ from MC and LC analysis.
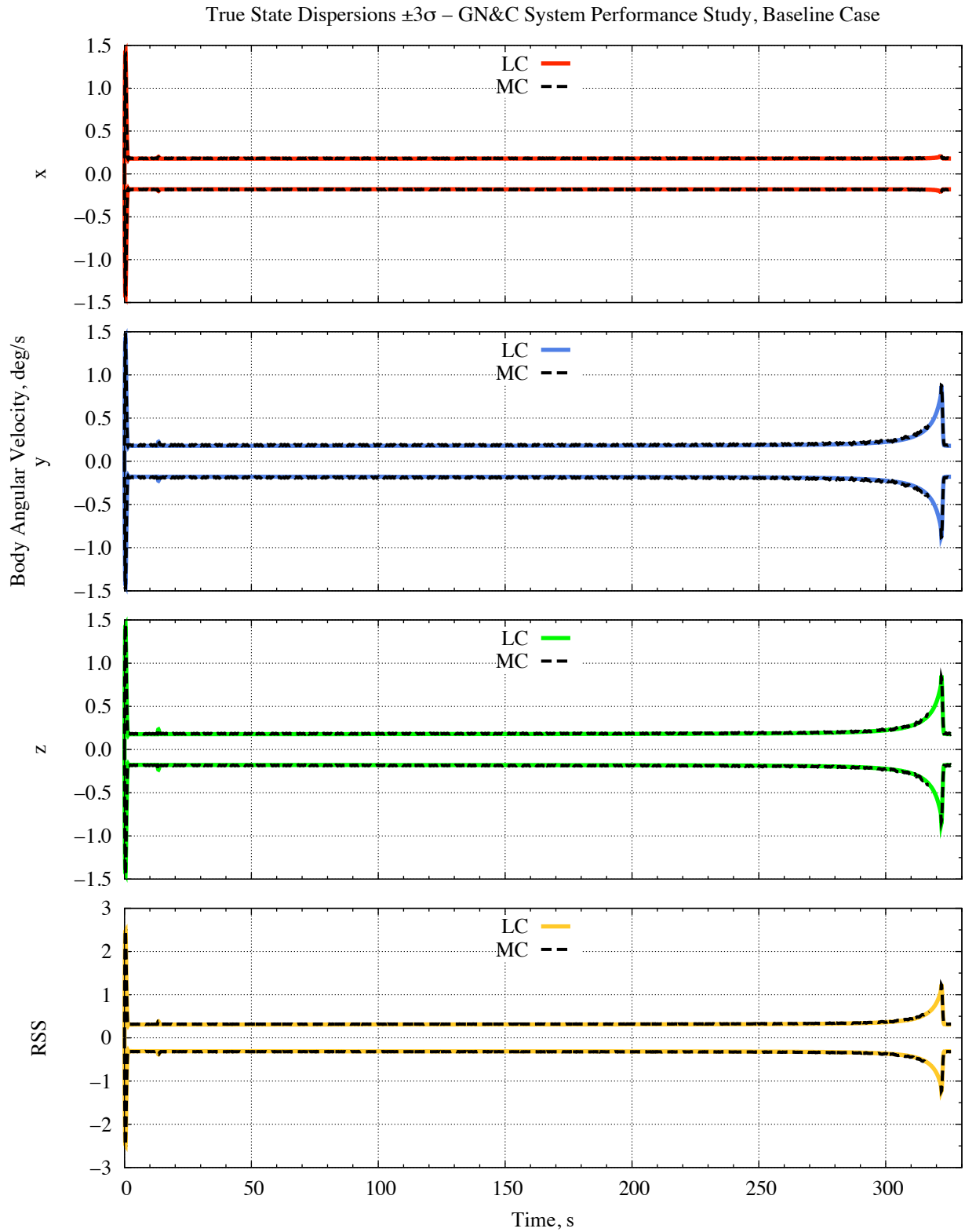
Figure 7.12: (1 of 2) *GN&C system performance study, baseline case.* Nominal time history comparison of the body angular velocity true state dispersions $\pm 3\sigma$ from MC and LC analysis.

True State Dispersions ±3σ – GN&C System Performance Study, Baseline Case
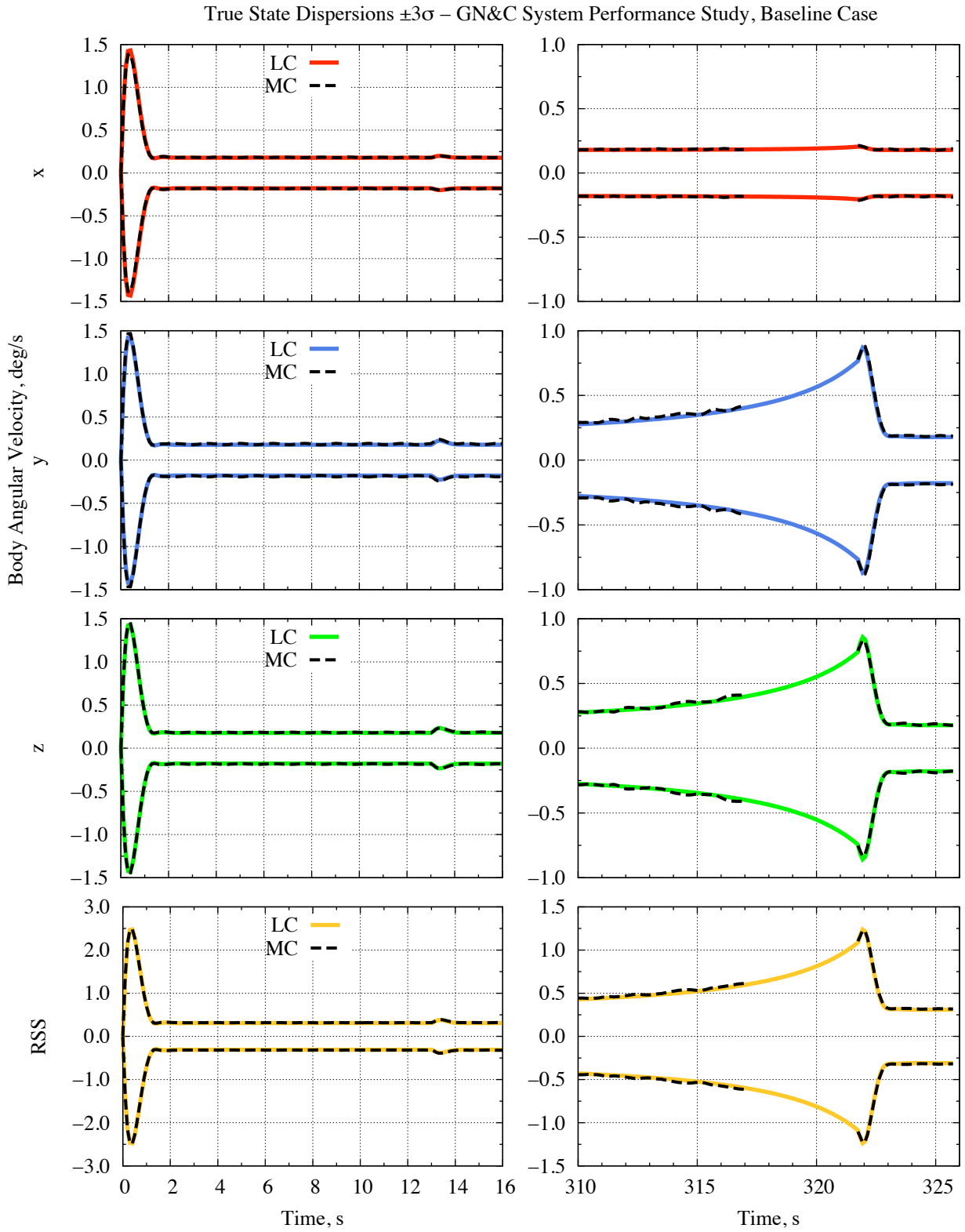
Figure 7.13: (2 of 2) *GN&C system performance study, baseline case.* Nominal time history comparison of the body angular velocity true state dispersions ±3σ from MC and LC analysis. Zoomed-in view of the first and last 16 s.

True State Dispersions ±3σ – GN&C System Performance Study, Baseline Case
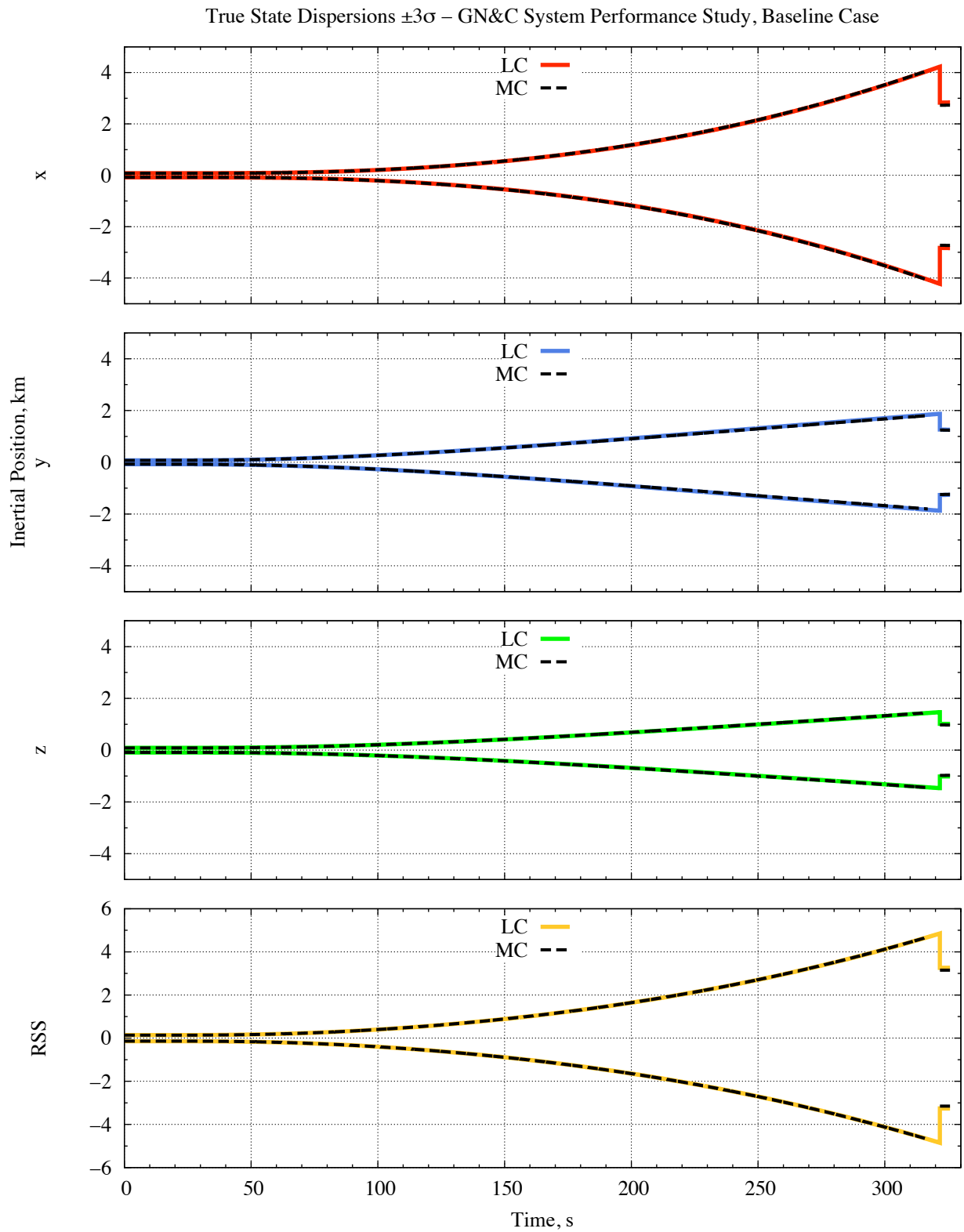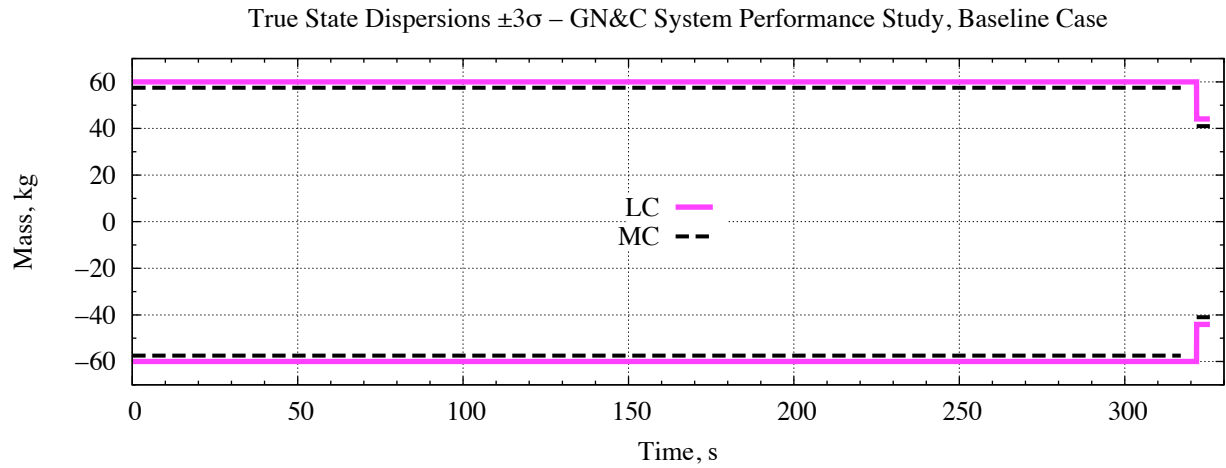
Figure 7.14: *GN&C system performance study, baseline case.* Nominal time history comparison of the wet mass true state dispersions ±3σ from MC and LC analysis.
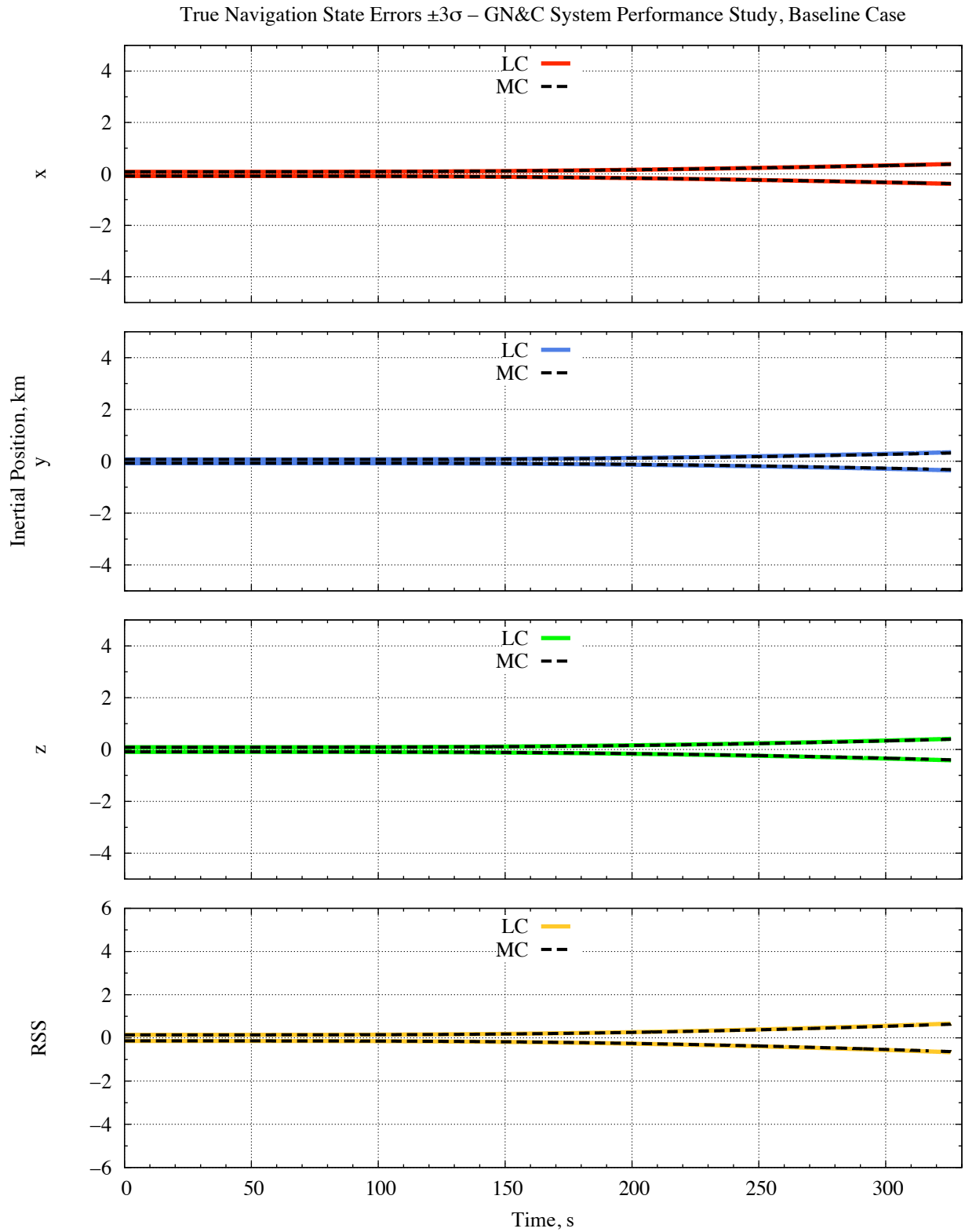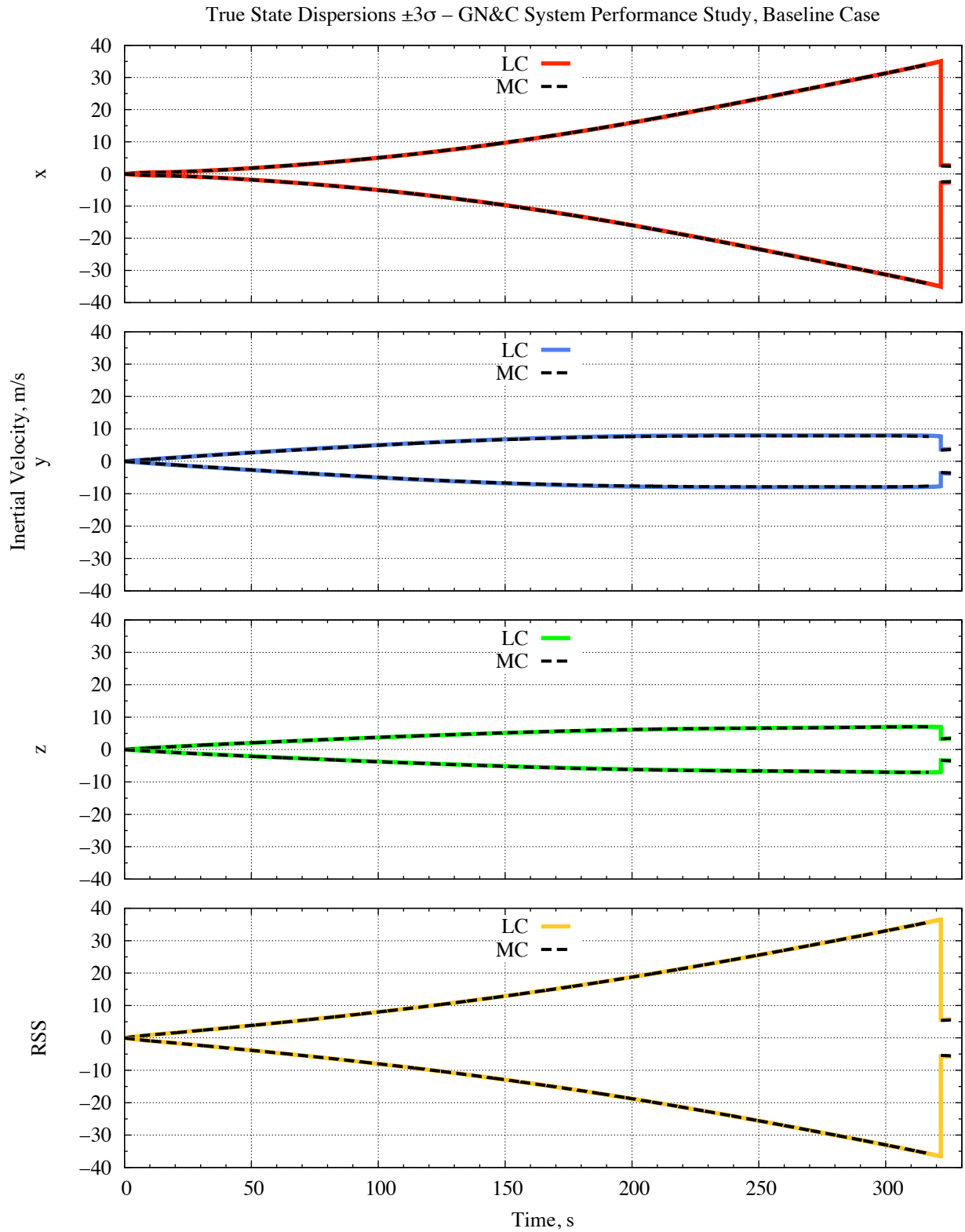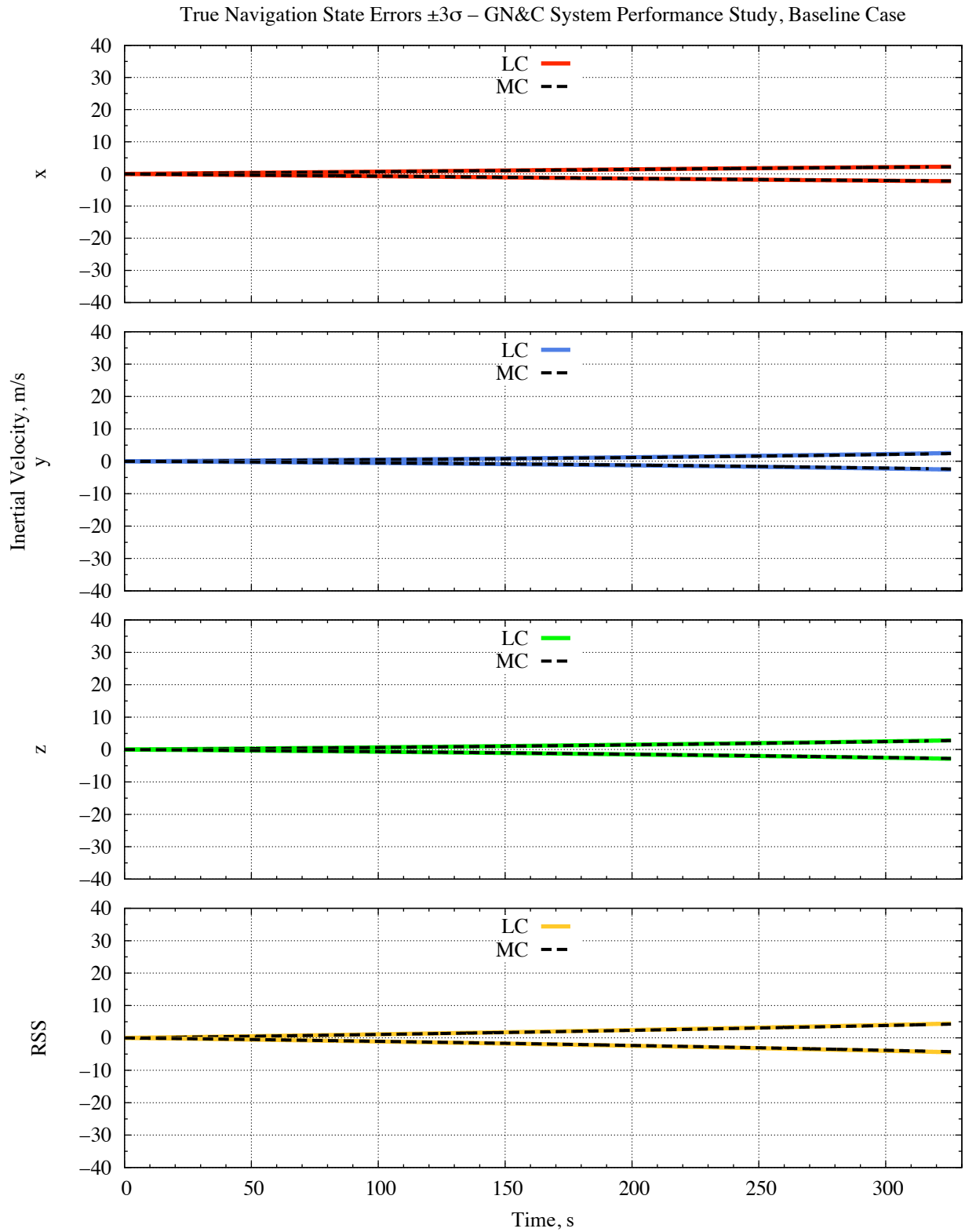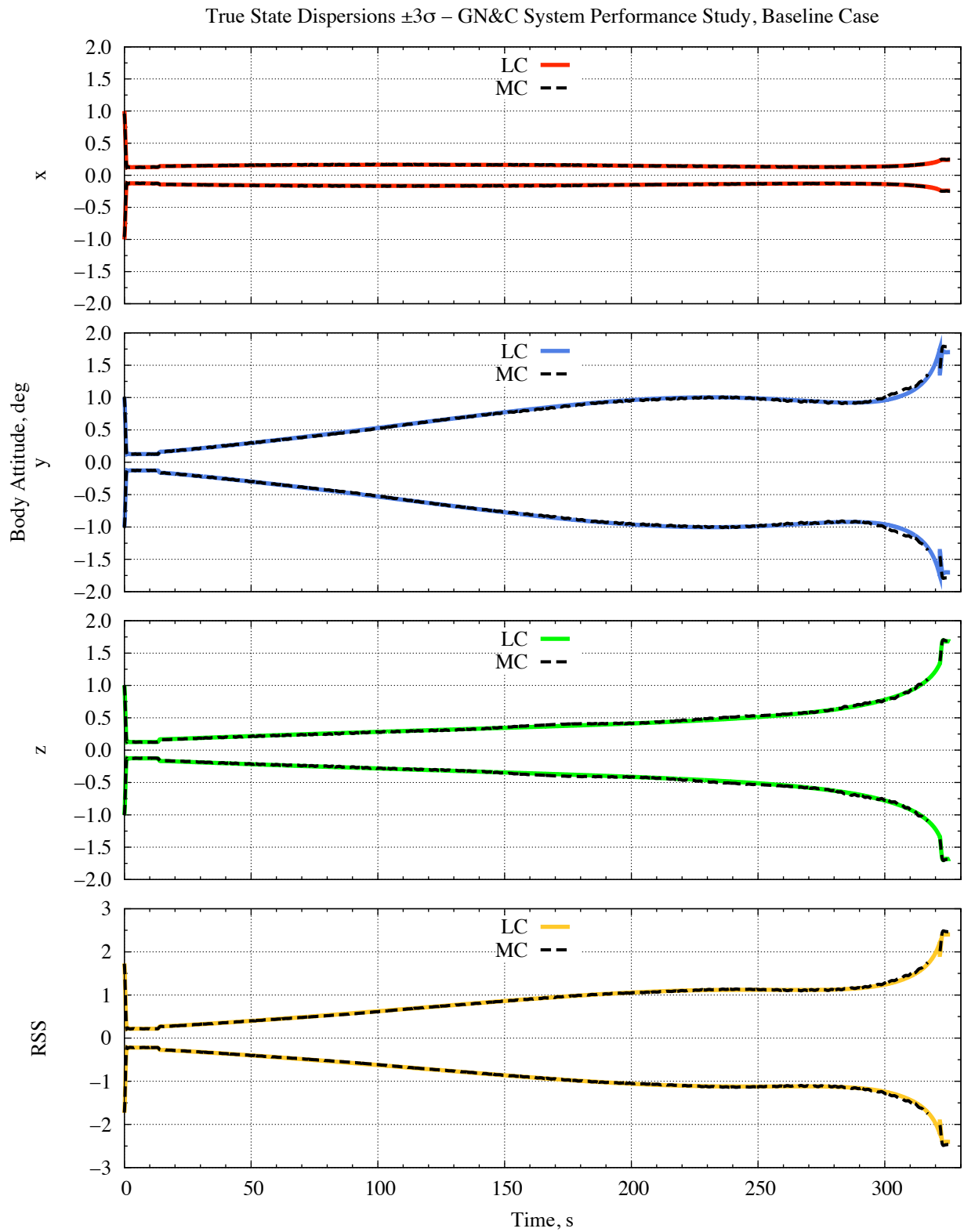
Table 7.3: *GN&C system performance study, baseline case.* Inertial position true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

| | | Inertial Position, m | | | | | | | |
| | | x | | y | | z | | RSS | |
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	76.5139	83.1256	72.2200	69.8368	86.0061	88.3215	135.894	139.956
	LC	80.9800	80.9800	73.7883	73.7883	89.9862	89.9862	141.774	141.774
	% Diff.	5.83691	2.58120	2.17167	5.65827	4.62776	1.88485	4.32732	1.29918
t_{PO}	MC	76.4975	83.1252	72.2287	69.8363	85.9966	88.3216	135.883	139.956
	LC	80.9798	80.9793	73.7891	73.7885	89.9863	89.9859	141.775	141.774
	% Diff.	5.85942	2.58155	2.16028	5.65923	4.63938	1.88430	4.33570	1.29909
t_{PEG}	MC	76.4506	83.1254	72.7686	69.8192	85.9392	88.3210	136.108	139.947
	LC	81.0681	80.9739	73.9558	73.7857	90.0755	89.9808	141.969	141.766
	% Diff.	6.03989	2.58829	1.63154	5.68113	4.81298	1.87924	4.30553	1.29990
t_{FC}	MC	2724.81	369.228	1246.26	310.802	974.219	387.150	3150.69	618.718
	LC	2827.36	376.281	1263.88	338.509	1017.47	403.158	3259.85	647.080
	% Diff.	3.76346	1.91033	1.41410	8.91467	4.44003	4.13489	3.46451	4.58396
t_{MECO}	MC	2730.04	377.724	1238.60	319.835	971.918	397.630	3151.49	634.885
	LC	2834.44	385.010	1256.21	348.217	1015.40	413.934	3262.39	663.950
	% Diff.	3.82412	1.92897	1.42180	8.87384	4.47408	4.10042	3.51890	4.57791

Table 7.4: *GN&C system performance study, baseline case.* Inertial velocity true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Inertial Velocity, m/s							
		x		y		z		RSS	
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	1.922E-04	1.860E-04	2.037E-04	2.212E-04	1.442E-07	1.442E-07	2.801E-04	2.890E-04
	LC	1.964E-04	1.964E-04	2.155E-04	2.155E-04	0	0	2.916E-04	2.916E-04
	% Diff.	2.17167	5.61161	5.79480	2.55494	100.000	100.000	4.10416	0.90718
t_{PO}	MC	0.17439	0.01780	0.18356	0.01676	0.17203	0.01973	0.30610	0.03142
	LC	0.17939	0.01831	0.18898	0.01672	0.16521	0.02029	0.30852	0.03204
	% Diff.	2.86679	2.84600	2.95358	0.19191	3.96535	2.80829	0.79082	1.97603
t_{PEG}	MC	0.48917	0.08542	0.76516	0.05419	0.64038	0.07430	1.11123	0.12551
	LC	0.49048	0.08795	0.77509	0.05453	0.63437	0.07759	1.11524	0.12934
	% Diff.	0.26748	2.96188	1.29837	0.63686	0.93880	4.43845	0.36052	3.05444
t_{FC}	MC	2.47664	2.17462	3.47787	2.32887	3.34016	2.71709	5.42088	4.18750
	LC	2.78981	2.23014	3.69733	2.49802	3.33579	2.78677	5.70796	4.35657
	% Diff.	12.6452	2.55278	6.31023	7.26331	0.13069	2.56453	5.29581	4.03741
t_{MECO}	MC	2.38235	2.19500	3.60462	2.37558	3.48493	2.76541	5.55100	4.25545
	LC	2.63675	2.25103	3.80199	2.54762	3.43458	2.83502	5.76228	4.42661
	% Diff.	10.67832	2.55260	5.47540	7.24213	1.44465	2.51723	3.80620	4.02198

Table 7.5: *GN&C system performance study, baseline case.* Body attitude true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Body Attitude, deg							
		x		y		z		RSS	
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	0.96090	0.09521	1.01445	0.09728	1.00744	0.09808	1.72261	0.16777
	LC	1.00000	0.10000	1.00000	0.10000	1.00000	0.10000	1.73205	0.17321
	% Diff.	4.06942	5.03166	1.42418	2.79305	0.73897	1.96277	0.54828	3.23829
t_{PO}	MC	0.12366	0.09519	0.12678	0.09730	0.12596	0.09807	0.21732	0.16777
	LC	0.12590	0.10000	0.12573	0.10000	0.12570	0.10000	0.21785	0.17321
	% Diff.	1.80757	5.05244	0.82846	2.77597	0.20035	1.96925	0.24228	3.24134
t_{PEG}	MC	0.12708	0.10261	0.12342	0.09609	0.12629	0.09525	0.21755	0.16981
	LC	0.12733	0.10143	0.12617	0.10022	0.12719	0.10152	0.21979	0.17504
	% Diff.	0.20068	1.14762	2.22604	4.28998	0.71279	6.58533	1.02868	3.07945
t_{FC}	MC	0.23694	0.10178	1.45061	0.09638	1.37834	0.10027	2.01500	0.17235
	LC	0.22995	0.10357	1.37229	0.10040	1.34120	0.10352	1.93258	0.17755
	% Diff.	2.94903	1.75828	5.39912	4.17120	2.69502	3.23832	4.09049	3.01868
t_{MECO}	MC	0.24995	0.10177	1.77577	0.09642	1.68850	0.10027	2.46310	0.17236
	LC	0.23919	0.10358	1.70073	0.10041	1.68073	0.10352	2.40303	0.17756
	% Diff.	4.30552	1.77592	4.22576	4.13761	0.46020	3.24554	2.43889	3.01696

Table 7.6: *GN&C system performance study, baseline case.* (*left*) Body angular velocity and (*right*) wet mass true state dispersions 3σ and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Body Angular Velocity, deg/s			
		x	y	z	RSS
		$\delta\mathbf{x}, 3\sigma$	$\delta\mathbf{x}, 3\sigma$	$\delta\mathbf{x}, 3\sigma$	$\delta\mathbf{x}, 3\sigma$
t_0	MC	2.4102E-06	1.1746E-06	2.7236E-06	3.8219E-06
	LC	2.3923E-06	1.1668E-06	2.6617E-06	3.7642E-06
	% Diff.	0.743645	0.661167	2.27334	1.50967
t_{PO}	MC	0.183478	0.178958	0.186340	0.316880
	LC	0.178453	0.177796	0.177710	0.308282
	% Diff.	2.73869	0.649236	4.63169	2.71335
t_{PEG}	MC	0.180536	0.191801	0.179001	0.318469
	LC	0.178623	0.177966	0.177832	0.308549
	% Diff.	1.05968	7.21322	0.653027	3.11488
t_{FC}	MC	0.215020	0.776454	0.752452	1.10241
	LC	0.204769	0.765876	0.745710	1.08838
	% Diff.	4.76735	1.36241	0.895915	1.27190
t_{MECO}	MC	0.187651	0.185243	0.177300	0.317747
	LC	0.178896	0.178189	0.178033	0.308951
	% Diff.	4.66576	3.80798	0.413412	2.76824

		Mass, kg
		$\delta\mathbf{x}, 3\sigma$
t_0	MC	57.4629
	LC	60.0000
	% Diff.	4.41514
t_{PO}	MC	57.4629
	LC	60.0000
	% Diff.	4.41514
t_{PEG}	MC	57.4629
	LC	60.0000
	% Diff.	4.41514
t_{FC}	MC	40.9902
	LC	44.0551
	% Diff.	7.47724
t_{MECO}	MC	40.9902
	LC	44.0551
	% Diff.	7.47724

and normalized percent difference values. With the exception of the inertial velocity magnitude, all of the normalized percent difference values for the orbit insertion dispersion results are less than 4%. One thing that the normalized percent difference does not take into account is the nominal value. In the case of the inertial velocity magnitude, the difference in the 3σ values from the two sets of results is only 0.17454 m/s, which is fairly insignificant when compared to the nominal value of 1686.13 m/s.

7.2.2 High Cost Case

The purpose of the high cost case is to explore what effects the accuracy/cost of the sensors, actuators, or other system components and knowledge of the system and environment might have on the performance of the GN&C system. The 3σ uncertainty values in the initial conditions of the true state and initial knowledge of the navigation states employed in this *high cost* case are simply those specified in Tables 7.1 and 7.2, respectively, *decreased* by a factor of 10.

Table 7.7: *GN&C system performance study, baseline case.* Orbit insertion dispersion results at MECO.

Parameter	Nominal	MC, 3σ	LC, 3σ	% Diff.
Burn Time, s	325.730	4.76313	4.81843	1.16115
Planetocentric Altitude, m	15239.7	398.687	409.697	2.76146
Inertial Velocity Magnitude, m/s	1686.13	1.26162	1.08708	13.8350
Flight Path Angle, deg	$-6.8248\mathrm{e}{-03}$	$9.6515\mathrm{e}{-02}$	$9.9958\mathrm{e}{-02}$	3.56713
Inclination, deg	145.765	$6.5890\mathrm{e}{-03}$	$6.8274\mathrm{e}{-03}$	3.61803
RAAN, deg	356.786	0.148147	0.151769	2.44455
Propellant Expenditure, kg	2657.22	40.9902	44.0551	7.47724

Launch Vehicle States

Figures 7.15–7.24 show the nominal time history results from MC and LC analysis of the true state dispersions $\pm3\sigma$ and true navigation state errors $\pm3\sigma$ associated with the five launch vehicle states: inertial position, inertial velocity, body attitude, body angular velocity, and wet mass. Figures 7.20 and 7.23 provide a zoomed-in view of the first and last 16 s of Figures 7.19 and 7.22, respectively. It is observed in all of the nominal time history results that the MC and LC results are nearly indistinguishable. To quantify just how much the two sets of results actually differ, the normalized percent difference was computed at five key points along the trajectory, as explained in § 7.1.3. Tables 7.8–7.11 list the MC and LC 3σ values and corresponding normalized percent difference for the true state dispersions and true navigation state errors (denoted respectively as $\delta\boldsymbol{x}$ and $\delta\boldsymbol{e}$) associated with the five launch vehicle states.

Orbit Insertion

Orbit insertion dispersion results at MECO for the burn time, guidance target parameters, and propellant expenditure are provided in Table 7.12, which includes the nominal, MC and LC 3σ, and normalized percent difference values. With the exception of the inertial velocity magnitude, all of the normalized percent difference values for the orbit insertion dispersion results are less than 9%. In the case of the inertial velocity magnitude, the difference in the 3σ values from the two sets of results is only 0.027481 m/s, which is clearly negligible when compared to the nominal value of 1686.13 m/s.

Figure 7.15: *GN&C system performance study, high cost case.* Nominal time history comparison of the inertial position true state dispersions $\pm 3\sigma$ from MC and LC analysis.

Figure 7.16: *GN&C system performance study, high cost case.* Nominal time history comparison of the inertial position true navigation state errors ±3σ from MC and LC analysis.

Figure 7.17: *GN&C system performance study, high cost case.* Nominal time history comparison of the inertial velocity true state dispersions ±3σ from MC and LC analysis.

True Navigation State Errors ±3σ – GN&C System Performance Study, High Cost Case

Figure 7.18: *GN&C system performance study, high cost case.* Nominal time history comparison of the inertial velocity true navigation state errors ±3σ from MC and LC analysis.

Figure 7.19: (1 of 2) *GN&C system performance study, high cost case.* Nominal time history comparison of the body attitude true state dispersions $\pm 3\sigma$ from MC and LC analysis.

Figure 7.20: (2 of 2) *GN&C system performance study, high cost case.* Nominal time history comparison of the body attitude true state dispersions $\pm3\sigma$ from MC and LC analysis. Zoomed-in view of the first and last 16 s.

Figure 7.21: *GN&C system performance study, high cost case.* Nominal time history comparison of the body attitude true navigation state errors $\pm 3\sigma$ from MC and LC analysis.
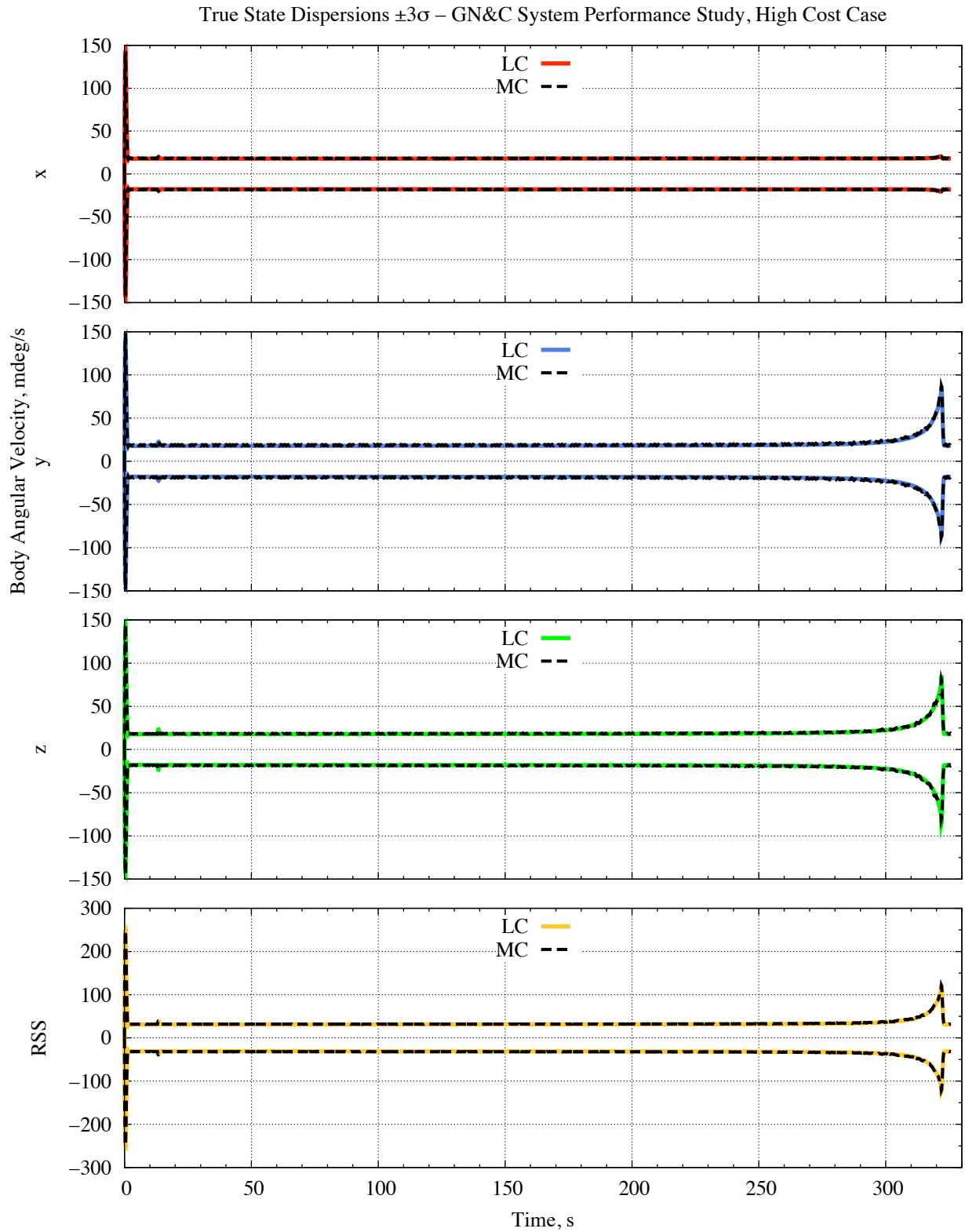
Figure 7.22: (1 of 2) *GN&C system performance study, high cost case.* Nominal time history comparison of the body angular velocity true state dispersions ±3σ from MC and LC analysis.

True State Dispersions ±3σ – GN&C System Performance Study, High Cost Case
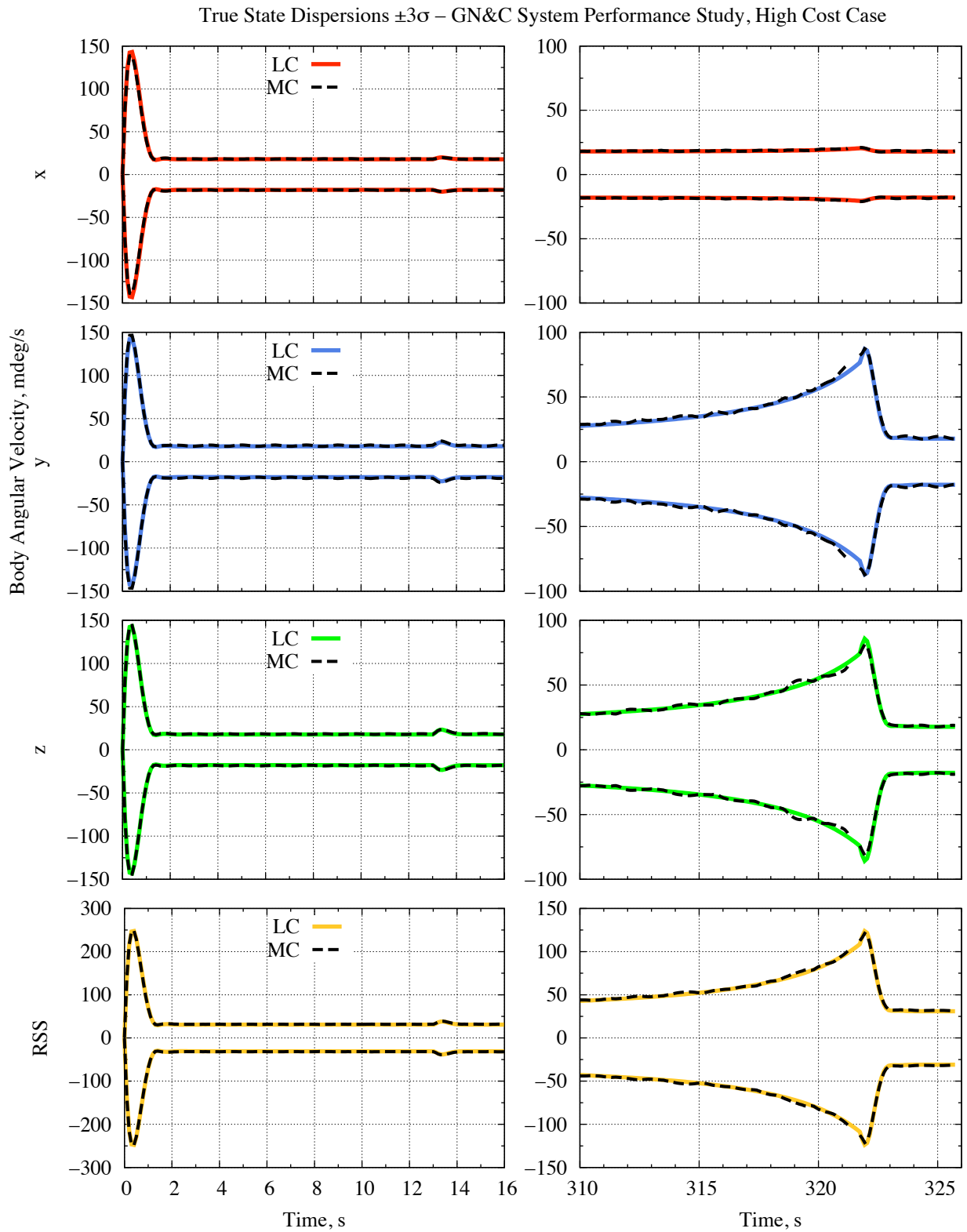
Figure 7.23: (2 of 2) *GN&C system performance study, high cost case.* Nominal time history comparison of the body angular velocity true state dispersions ±3σ from MC and LC analysis. Zoomed-in view of the first and last 16 s.

True State Dispersions ±3σ – GN&C System Performance Study, High Cost Case
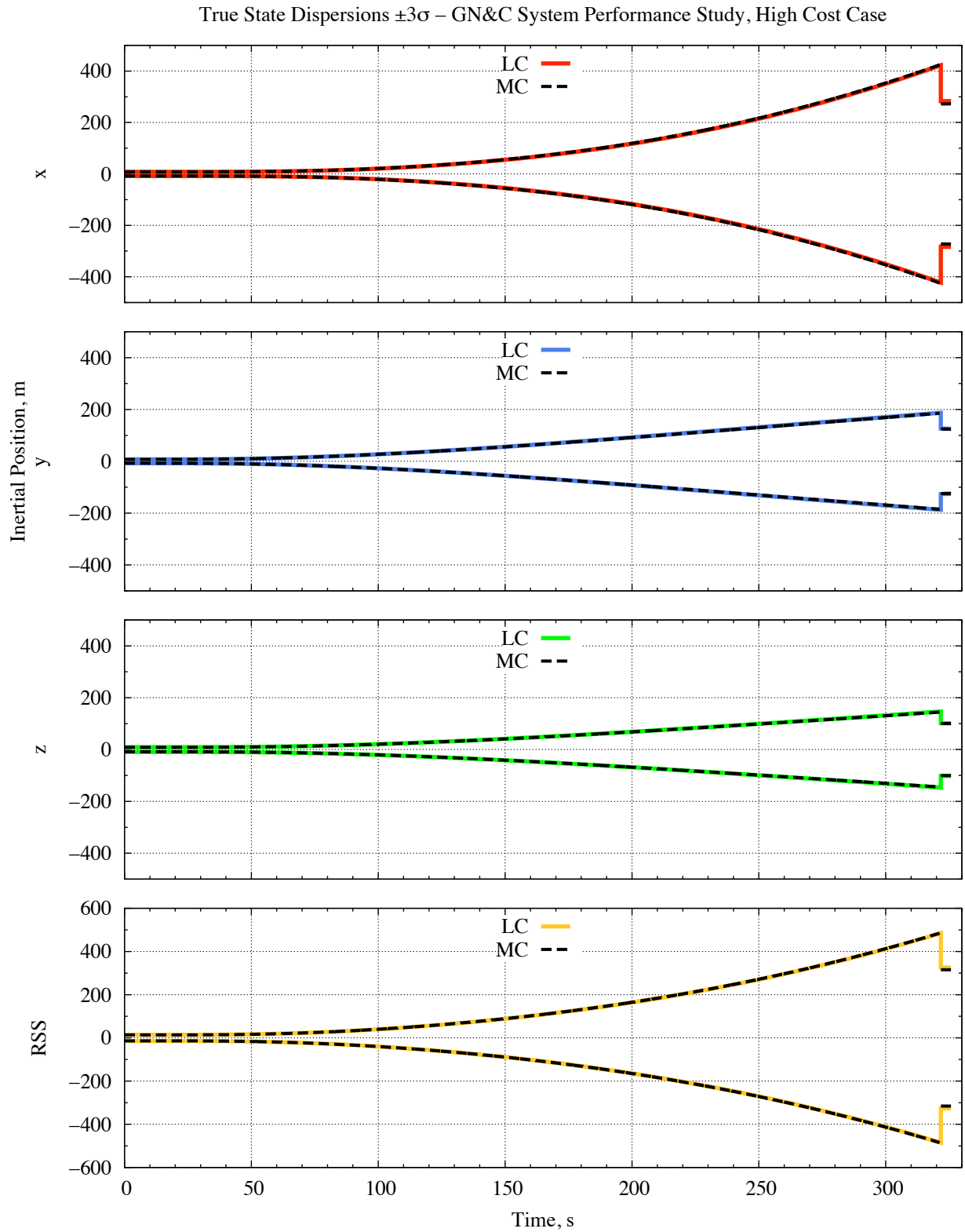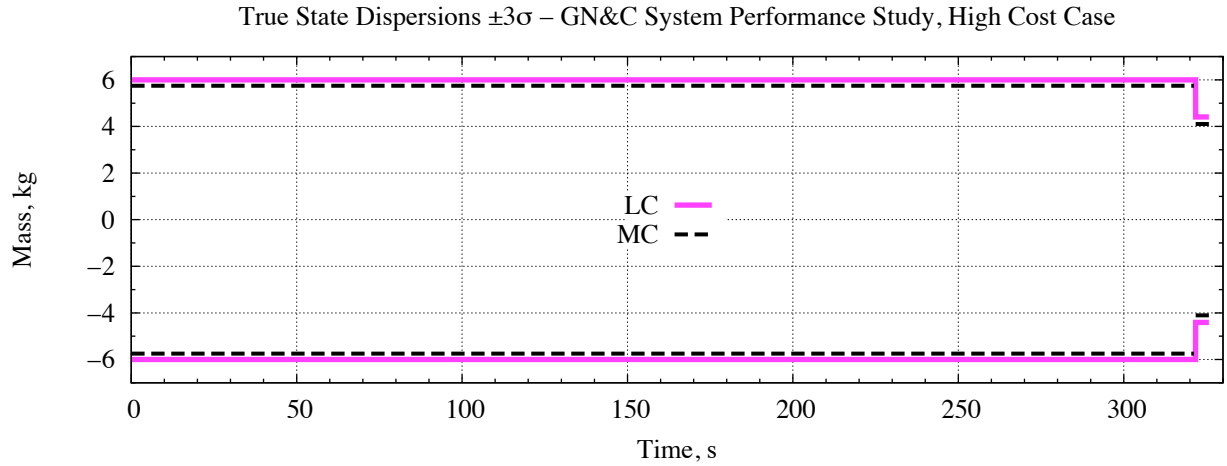

Figure 7.24: *GN&C system performance study, high cost case.* Nominal time history comparison of the wet mass true state dispersions ±3σ from MC and LC analysis.

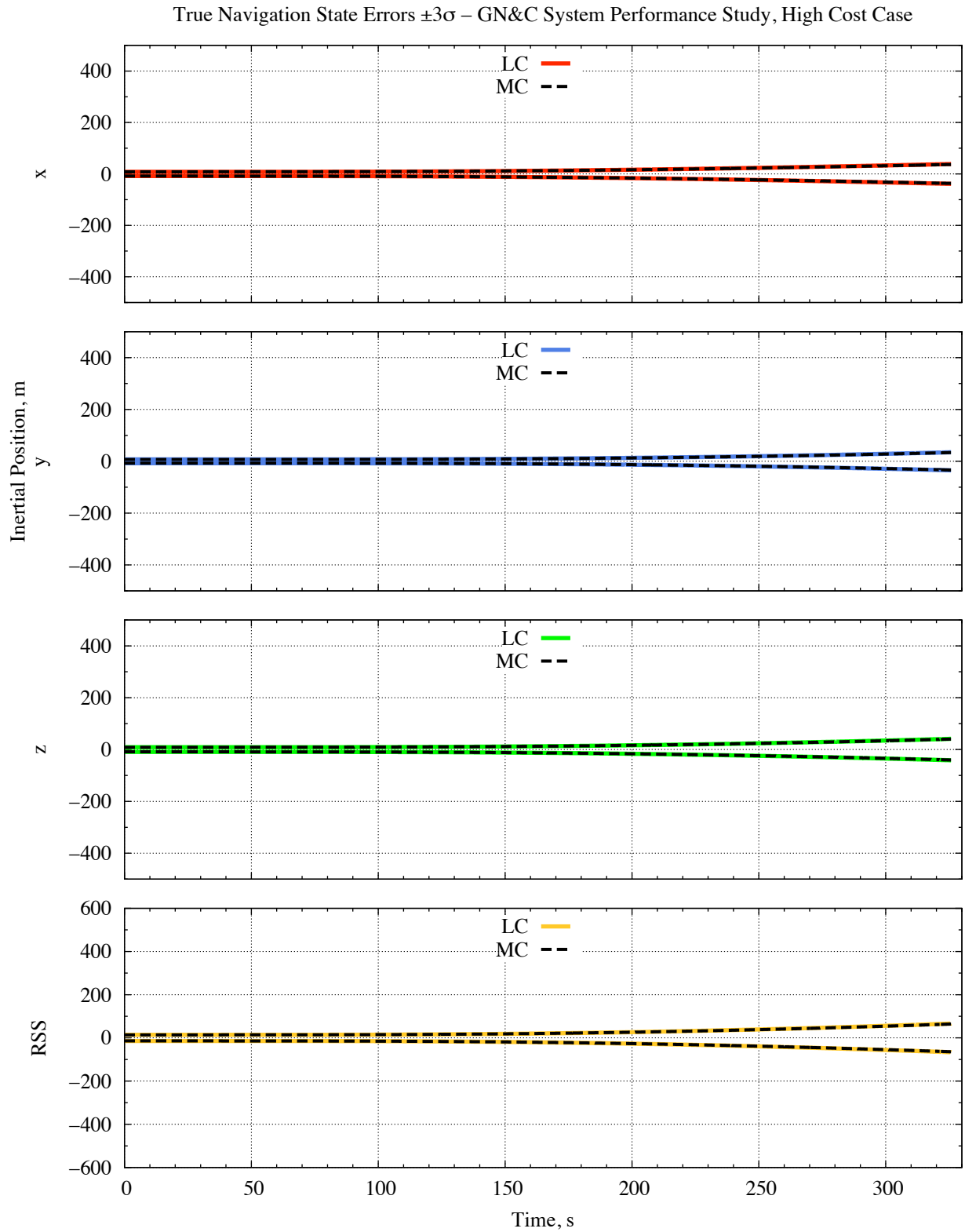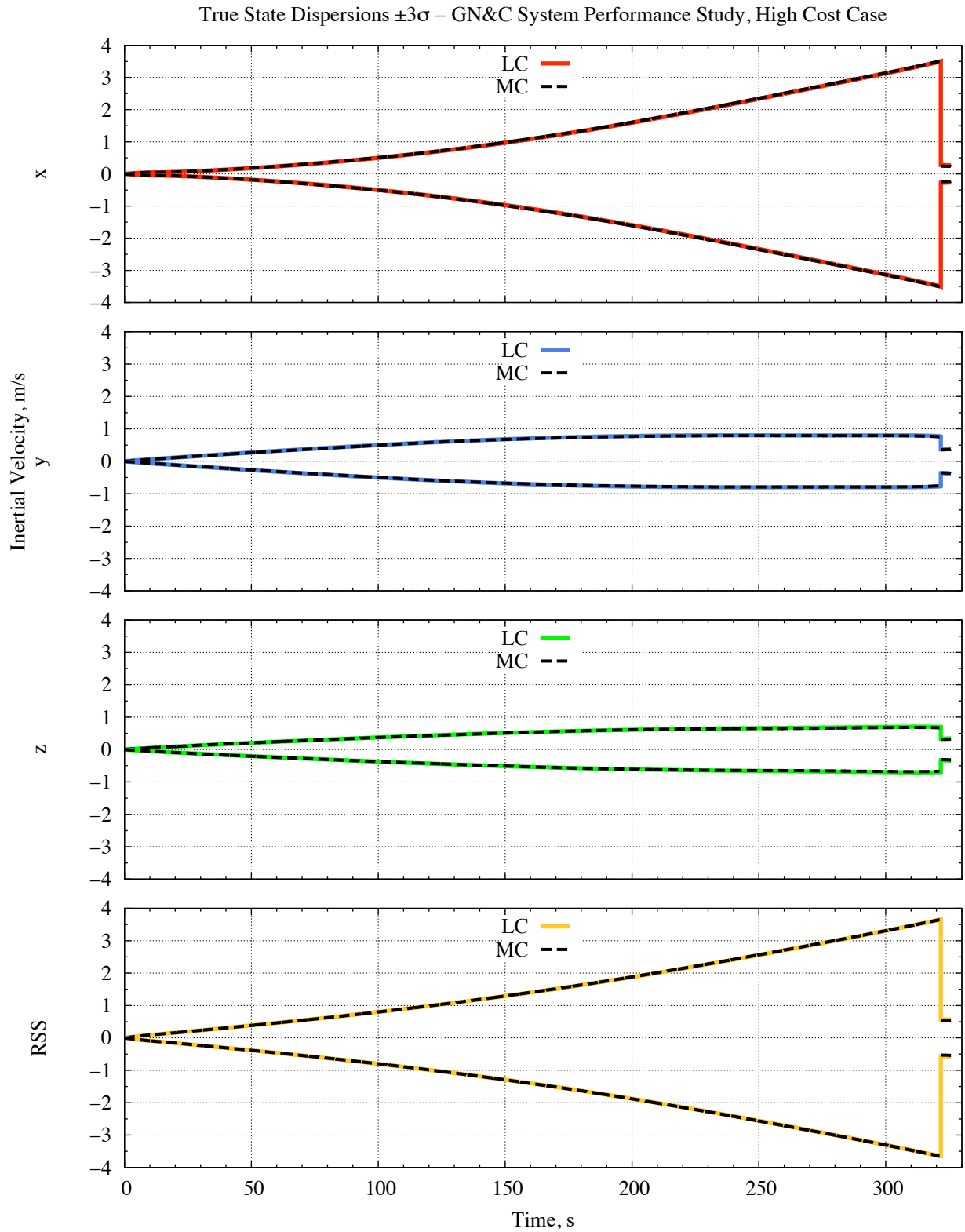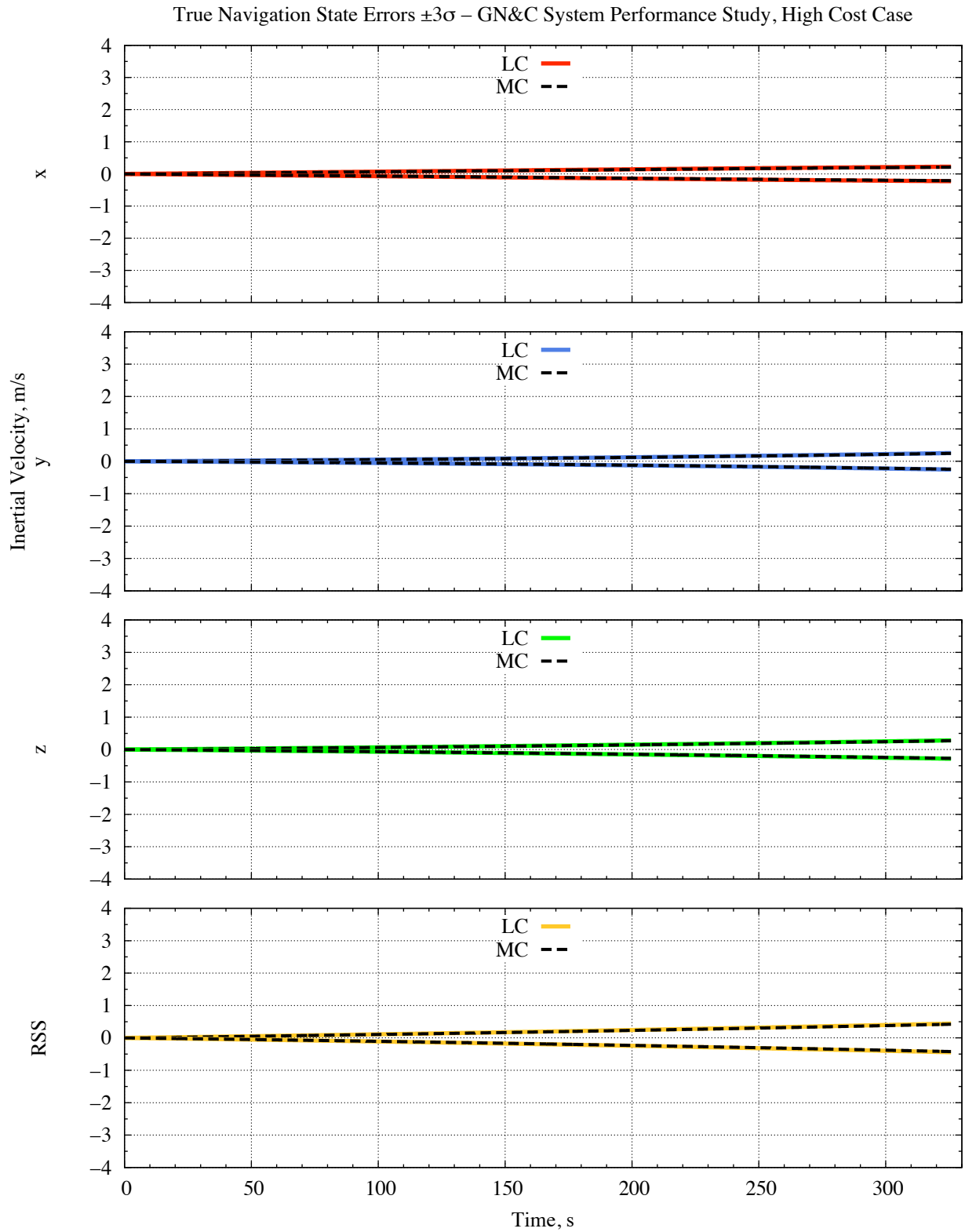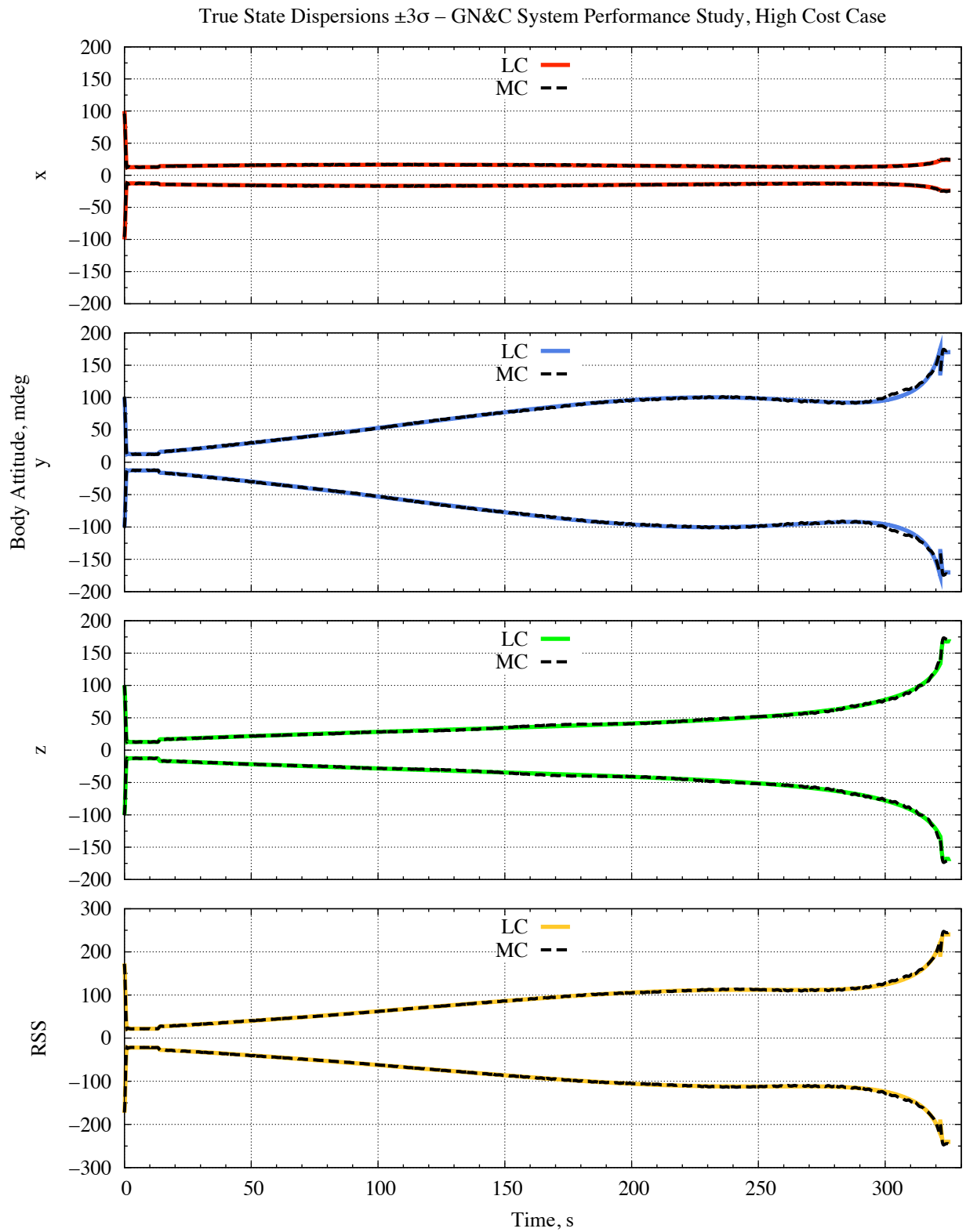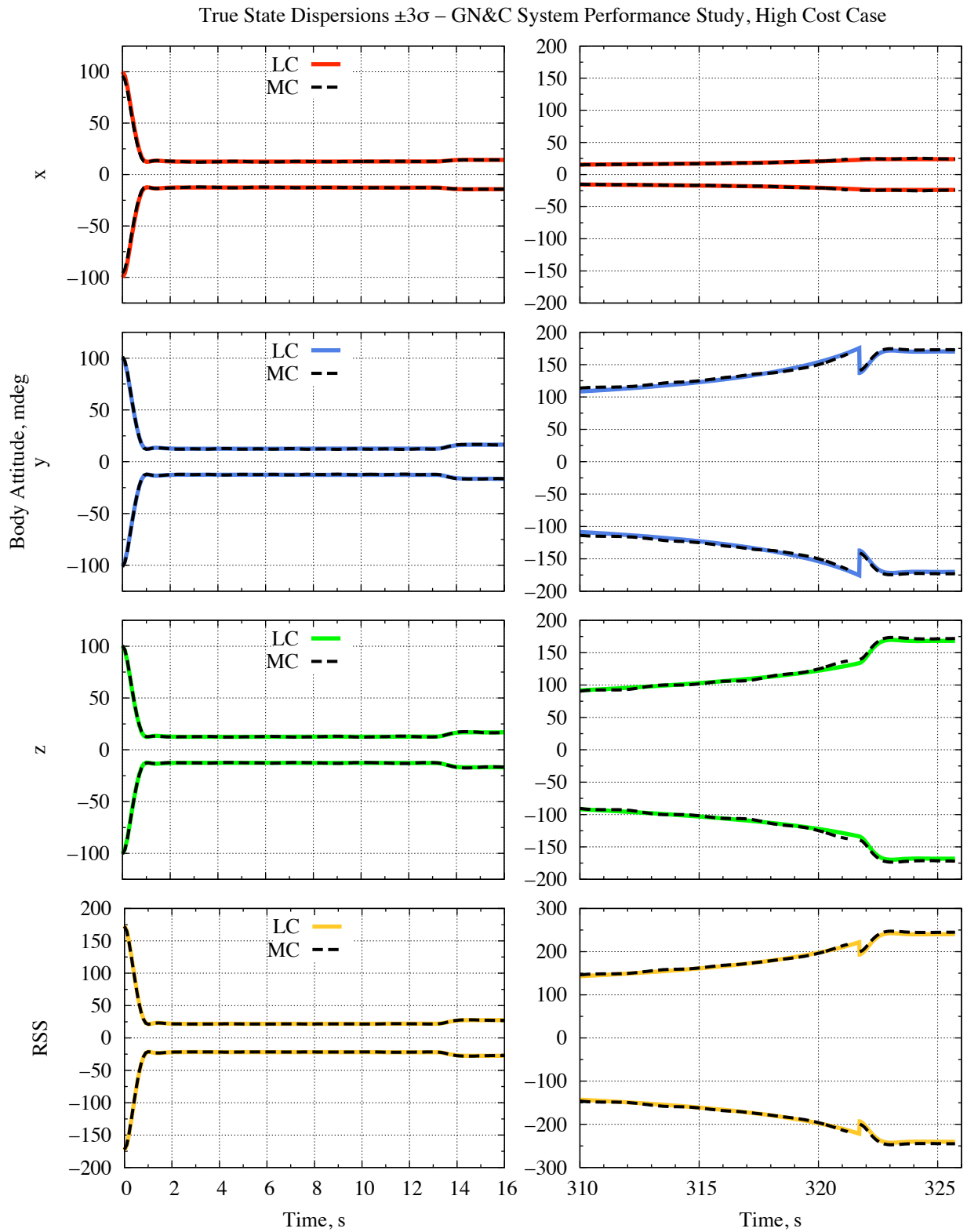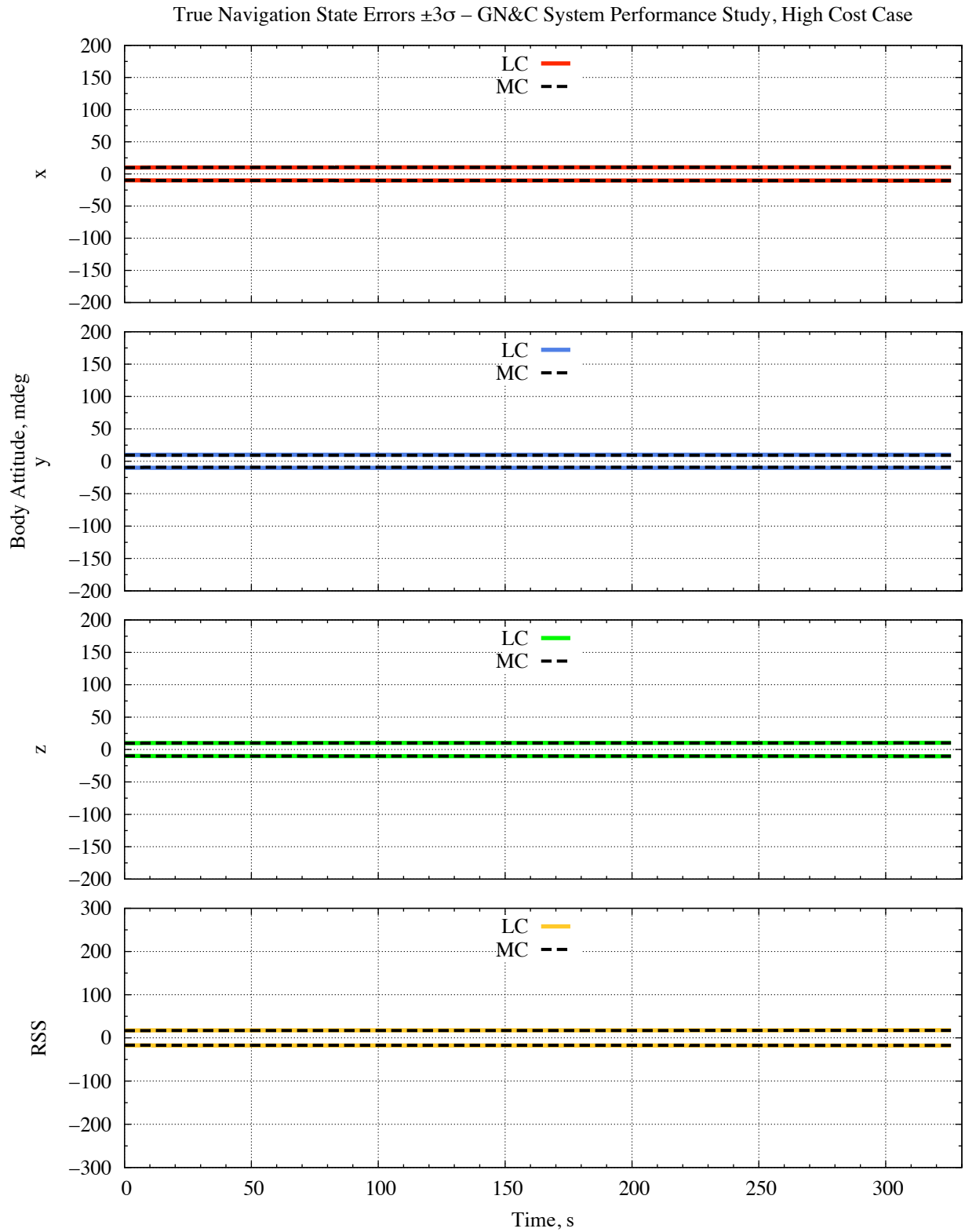Table 7.8: *GN&C system performance study, high cost case.* Inertial position true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Inertial Position, m							
		x		y		z		RSS	
		$\delta \mathbf{x}$, 3σ	$\delta \mathbf{e}$, 3σ	$\delta \mathbf{x}$, 3σ	$\delta \mathbf{e}$, 3σ	$\delta \mathbf{x}$, 3σ	$\delta \mathbf{e}$, 3σ	$\delta \mathbf{x}$, 3σ	$\delta \mathbf{e}$, 3σ
t_0	MC	7.65141	8.31255	7.22199	6.98365	8.60061	8.83217	13.5894	13.9956
	LC	8.09800	8.09800	7.37883	7.37883	8.99862	8.99862	14.1774	14.1774
	% Diff.	5.83671	2.58111	2.17173	5.65873	4.62771	1.88463	4.32726	1.29923
t_{PO}	MC	7.64979	8.31267	7.22290	6.98379	8.59966	8.83219	13.5884	13.9958
	LC	8.09798	8.09793	7.37891	7.37885	8.99863	8.99859	14.1775	14.1774
	% Diff.	5.85876	2.58334	2.15987	5.65671	4.63939	1.88399	4.33538	1.29770
t_{PEG}	MC	7.64576	8.31579	7.27765	6.98585	8.59373	8.83308	13.6115	13.9992
	LC	8.10681	8.09739	7.39558	7.37857	9.00755	8.99808	14.1969	14.1766
	% Diff.	6.03015	2.62633	1.62042	5.62167	4.81528	1.86799	4.30016	1.26730
t_{FC}	MC	271.997	36.0266	124.573	32.8187	100.489	39.2962	315.593	62.6034
	LC	282.736	37.6281	126.388	33.8509	101.747	40.3158	325.985	64.7080
	% Diff.	3.94816	4.44533	1.45672	3.14497	1.25221	2.59456	3.29276	3.36181
t_{MECO}	MC	272.505	36.8396	123.812	33.7495	100.306	40.3276	315.673	64.2067
	LC	283.444	38.5010	125.621	34.8217	101.540	41.3934	326.239	66.3950
	% Diff.	4.01444	4.50977	1.461531	3.17683	1.230935	2.64295	3.34712	3.40815

Table 7.9: *GN&C system performance study, high cost case.* Inertial velocity true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Inertial Velocity, m/s							
		x		y		z		RSS	
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	1.922E-05	1.859E-05	2.037E-05	2.212E-05	1.667E-07	1.667E-07	2.801E-05	2.890E-05
	LC	1.964E-05	1.964E-05	2.155E-05	2.155E-05	0	0	2.916E-05	2.916E-05
	% Diff.	2.17173	5.65018	5.81129	2.56870	100.000	100.000	4.11094	0.912418
t_{PO}	MC	1.738E-02	1.713E-03	1.842E-02	1.650E-03	1.720E-02	1.977E-03	3.061E-02	3.092E-03
	LC	1.794E-02	1.831E-03	1.890E-02	1.672E-03	1.652E-02	2.029E-03	3.085E-02	3.204E-03
	% Diff.	3.22100	6.89632	2.58081	1.35305	3.93922	2.63155	0.779606	3.60037
t_{PEG}	MC	4.854E-02	8.211E-03	7.691E-02	5.405E-03	6.384E-02	7.572E-03	1.111E-01	1.241E-02
	LC	4.905E-02	8.795E-03	7.751E-02	5.453E-03	6.344E-02	7.759E-03	1.115E-01	1.293E-02
	% Diff.	1.05027	7.10611	0.782609	0.886728	0.627874	2.48088	0.370583	4.23684
t_{FC}	MC	0.243737	0.208322	0.354733	0.240019	0.310632	0.267221	0.530788	0.415228
	LC	0.278981	0.223014	0.369733	0.249802	0.333579	0.278677	0.570796	0.435657
	% Diff.	14.4600	7.05266	4.22847	4.07571	7.38730	4.28702	7.53756	4.91993
t_{MECO}	MC	0.232603	0.210266	0.368734	0.244744	0.320621	0.271829	0.541171	0.421903
	LC	0.263675	0.225103	0.380199	0.254762	0.343458	0.283502	0.576228	0.442661
	% Diff.	13.3581	7.05624	3.10946	4.09365	7.12275	4.29443	6.47797	4.92011

Table 7.10: *GN&C system performance study, high cost case.* Body attitude true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Body Attitude, deg							
		x		y		z		RSS	
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	9.594E-02	9.778E-03	1.015E-01	9.450E-03	1.007E-01	9.835E-03	1.722E-01	1.678E-02
	LC	1.000E-01	1.000E-02	1.000E-01	1.000E-02	1.000E-01	1.000E-02	1.732E-01	1.732E-02
	% Diff.	4.23414	2.27501	1.50901	5.82045	0.718639	1.67650	0.574686	3.20948
t_{PO}	MC	1.215E-02	9.775E-03	1.243E-02	9.451E-03	1.262E-02	9.833E-03	2.149E-02	1.678E-02
	LC	1.259E-02	1.000E-02	1.257E-02	1.000E-02	1.257E-02	1.000E-02	2.178E-02	1.732E-02
	% Diff.	3.57687	2.30196	1.11299	5.81401	0.417935	1.69450	1.38631	3.22299
t_{PEG}	MC	1.283E-02	1.020E-02	1.215E-02	9.268E-03	1.240E-02	1.002E-02	2.158E-02	1.704E-02
	LC	1.273E-02	1.014E-02	1.262E-02	1.002E-02	1.272E-02	1.015E-02	2.198E-02	1.750E-02
	% Diff.	0.759876	0.526872	3.86598	8.13072	2.58495	1.28007	1.82773	2.72365
t_{FC}	MC	2.446E-02	1.042E-02	1.408E-01	9.280E-03	1.396E-01	1.025E-02	1.998E-01	1.732E-02
	LC	2.300E-02	1.036E-02	1.372E-01	1.004E-02	1.341E-01	1.035E-02	1.933E-01	1.776E-02
	% Diff.	5.99837	0.621765	2.55342	8.19999	3.93915	0.964295	3.27865	2.53443
t_{MECO}	MC	2.421E-02	1.042E-02	1.729E-01	9.280E-03	1.719E-01	1.026E-02	2.450E-01	1.732E-02
	LC	2.392E-02	1.036E-02	1.701E-01	1.004E-02	1.681E-01	1.035E-02	2.403E-01	1.776E-02
	% Diff.	1.19217	0.625399	1.64405	8.19303	2.23670	0.935799	1.93090	2.52106

Table 7.11: *GN&C system performance study, high cost case.* (*left*) Body angular velocity and (*right*) wet mass true state dispersions 3σ and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Body Angular Velocity, deg/s						Mass, kg
		x	y	z	RSS			
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{x}$, 3σ			$\delta\mathbf{x}$, 3σ
t_0	MC	2.4098E-07	1.1766E-07	2.7218E-07	3.8210E-07	t_0	MC	5.74629
	LC	2.3923E-07	1.1668E-07	2.6617E-07	3.7642E-07		LC	6.00000
	% Diff.	0.724186	0.828975	2.20926	1.48497		% Diff.	4.41514
t_{PO}	MC	1.8351E-02	1.7888E-02	1.8639E-02	3.1688E-02	t_{PO}	MC	5.74629
	LC	1.7845E-02	1.7780E-02	1.7771E-02	3.0828E-02		LC	6.00000
	% Diff.	2.75454	0.608702	4.65615	2.71465		% Diff.	4.41514
t_{PEG}	MC	1.8071E-02	1.9181E-02	1.7902E-02	3.1858E-02	t_{PEG}	MC	5.74629
	LC	1.7862E-02	1.7797E-02	1.7783E-02	3.0855E-02		LC	6.00000
	% Diff.	1.15584	7.21581	0.663495	3.14937		% Diff.	4.41514
t_{FC}	MC	2.1094E-02	8.2051E-02	7.4169E-02	0.112599	t_{FC}	MC	4.10670
	LC	2.0477E-02	7.6588E-02	7.4571E-02	0.108838		LC	4.40551
	% Diff.	2.92472	6.65886	0.541780	3.33940		% Diff.	7.27626
t_{MECO}	MC	1.8408E-02	1.7451E-02	1.8863E-02	3.1610E-02	t_{MECO}	MC	4.10670
	LC	1.7890E-02	1.7819E-02	1.7803E-02	3.0895E-02		LC	4.40551
	% Diff.	2.81440	2.10599	5.61657	2.26174		% Diff.	7.27626

Table 7.12: *GN&C system performance study, high cost case.* Orbit insertion dispersion results at MECO.

Parameter	Nominal	MC, 3σ	LC, 3σ	% Diff.
Burn Time, s	325.730	0.476812	0.481843	1.05520
Planetocentric Altitude, m	15239.7	38.3075	40.9697	6.94960
Inertial Velocity Magnitude, m/s	1686.13	0.136189	0.108708	20.1790
Flight Path Angle, deg	$-6.8248e-03$	9.2014e-03	9.9958e-03	8.63334
Inclination, deg	145.765	6.5013e-03	6.8274e-03	5.01655
RAAN, deg	356.786	1.4290e-02	1.5177e-02	6.20754
Propellant Expenditure, kg	2657.22	4.10670	4.40551	7.27626

7.2.3 Low Cost Case

The purposes of the low cost case are to test the capabilities of the LC simulation beyond the nominal operating regime to see if there are areas in the trajectory where the linear approximation might no longer hold, and to explore what effects the accuracy/cost of the sensors, actuators, or other system components and knowledge of the system and environment might have on the performance of the GN&C system. The 3σ uncertainty values in the initial conditions of the true state and initial knowledge of the navigation states used in this *low cost* case study are simply those specified in Tables 7.1 and 7.2, respectively, *increased* by a factor of 10.

Launch Vehicle States

Figures 7.25–7.34 show the nominal time history results from MC and LC analysis of the true state dispersions $\pm3\sigma$ and true navigation state errors $\pm3\sigma$ associated with the five launch vehicle states: inertial position, inertial velocity, body attitude, body angular velocity, and wet mass. Figures 7.30 and 7.33 provide a zoomed-in view of the first and last 16 s of Figures 7.29 and 7.32, respectively. It is observed in all of the nominal time history results that the MC and LC results are nearly indistinguishable. To quantify just how much the two sets of results actually differ, the normalized percent difference was computed at five key points along the trajectory, as explained in § 7.1.3. Tables 7.13–7.16 list the MC and LC 3σ values and corresponding normalized percent difference for the true state dispersions and true navigation state errors (denoted respectively as $\delta\boldsymbol{x}$ and $\delta\boldsymbol{e}$) associated with the five launch vehicle states.

Orbit Insertion

Orbit insertion dispersion results at MECO for the burn time, guidance target parameters, and propellant expenditure are provided in Table 7.17, which includes the nominal, MC and LC 3σ, and normalized percent difference values. All of the normalized percent difference values for the orbit insertion dispersion results are less than 11%.

7.2.4 Case Comparison

Figures 7.35–7.38 show the nominal time history results from LC analysis of the total true state dispersions 3σ and total true navigation state errors 3σ associated with the first four launch vehicle states (i.e., inertial position, inertial velocity, body attitude, and body angular velocity)

Figure 7.25: *GN&C system performance study, low cost case.* Nominal time history comparison of the inertial position true state dispersions ±3σ from MC and LC analysis.

Figure 7.26: *GN&C system performance study, low cost case.* Nominal time history comparison of the inertial position true navigation state errors $\pm 3\sigma$ from MC and LC analysis.

Figure 7.27: *GN&C system performance study, low cost case.* Nominal time history comparison of the inertial velocity true state dispersions $\pm 3\sigma$ from MC and LC analysis.

Figure 7.28: *GN&C system performance study, low cost case.* Nominal time history comparison of the inertial velocity true navigation state errors ±3σ from MC and LC analysis.

Figure 7.29: (1 of 2) *GN&C system performance study, low cost case.* Nominal time history comparison of the body attitude true state dispersions ±3σ from MC and LC analysis.

Figure 7.30: (2 of 2) *GN&C system performance study, low cost case.* Nominal time history comparison of the body attitude true state dispersions ±3σ from MC and LC analysis. Zoomed-in view of the first and last 16 s.

Figure 7.31: *GN&C system performance study, low cost case.* Nominal time history comparison of the body attitude true navigation state errors $\pm 3\sigma$ from MC and LC analysis.
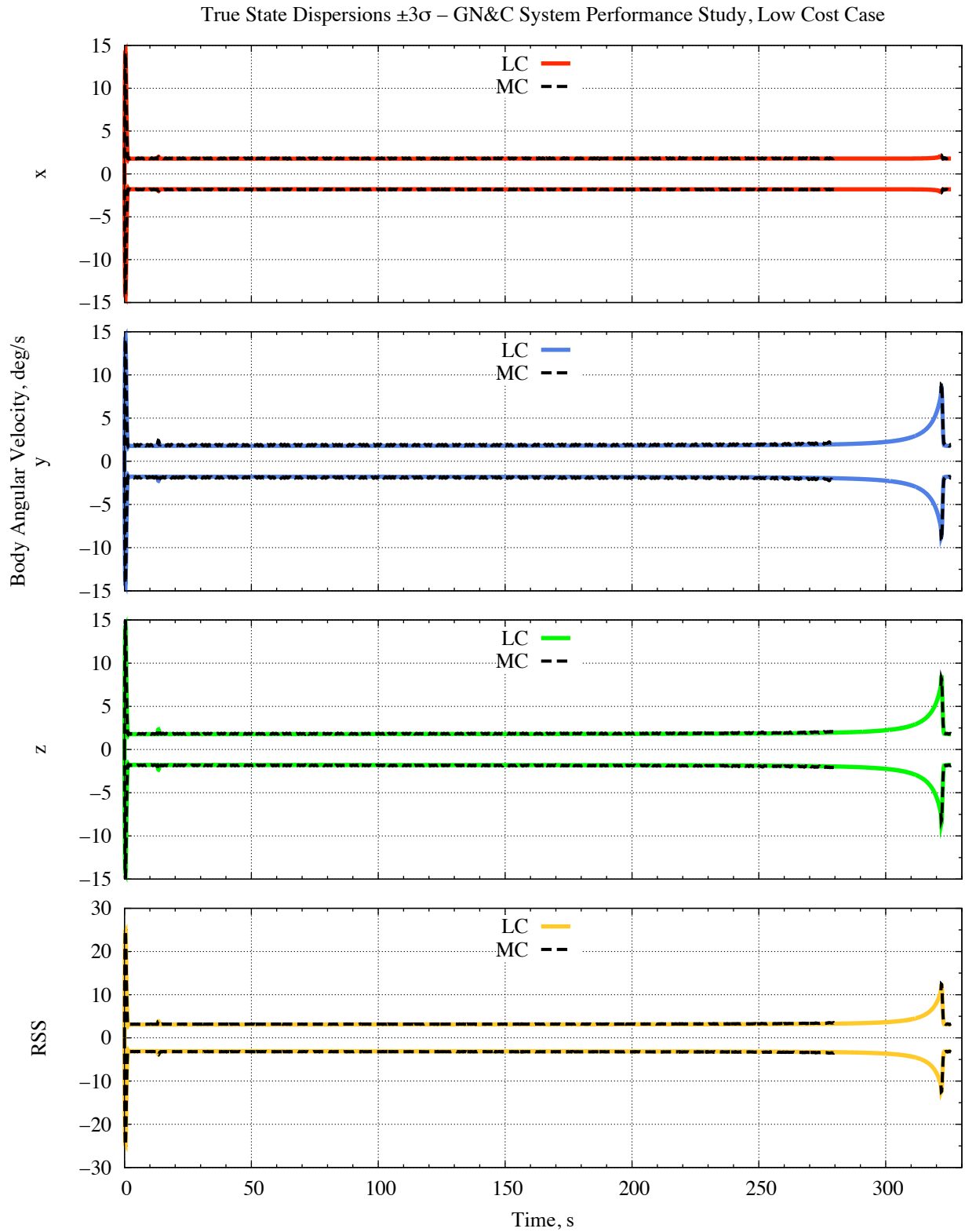
Figure 7.32: (1 of 2) *GN&C system performance study, low cost case.* Nominal time history comparison of the body angular velocity true state dispersions ±3σ from MC and LC analysis.
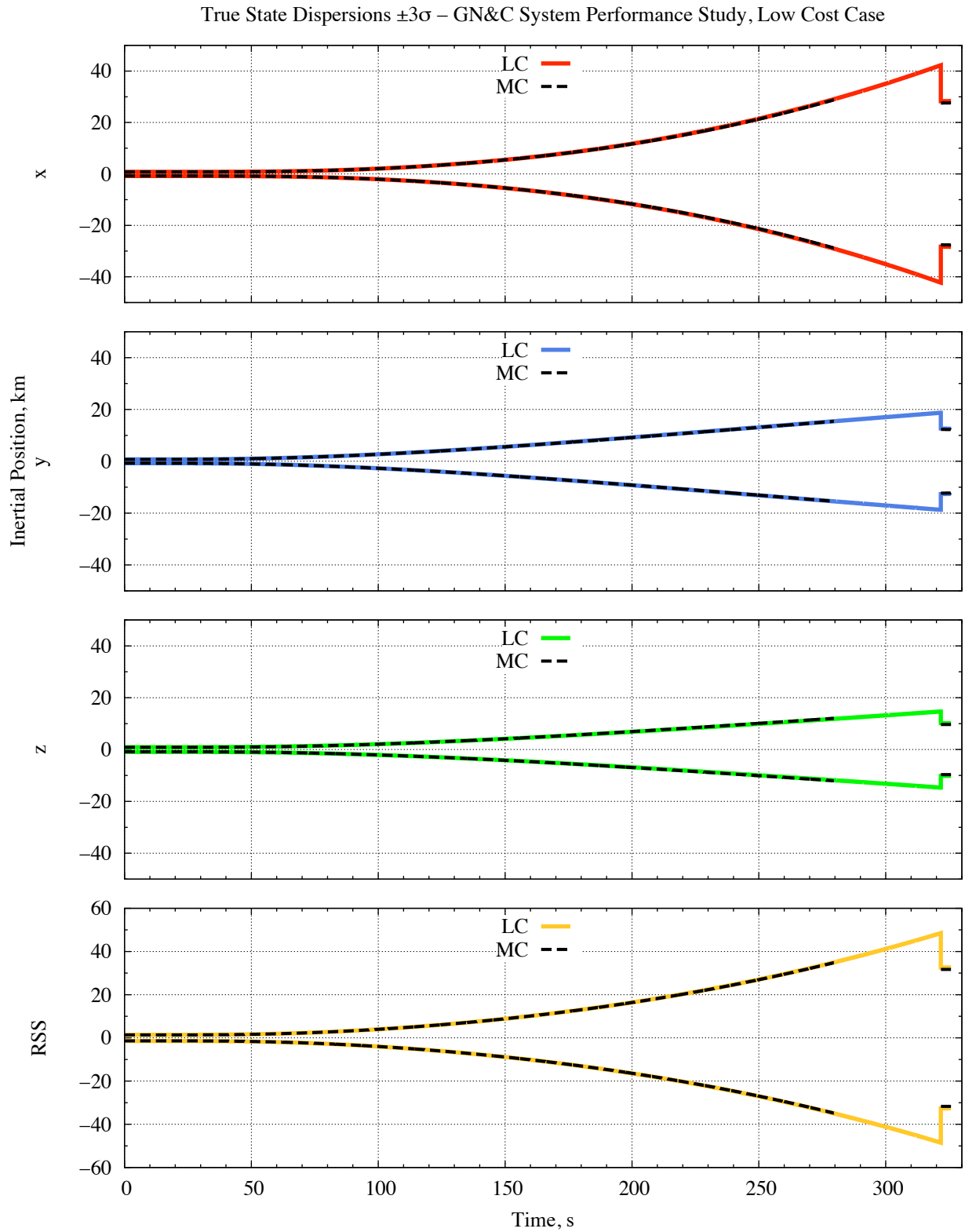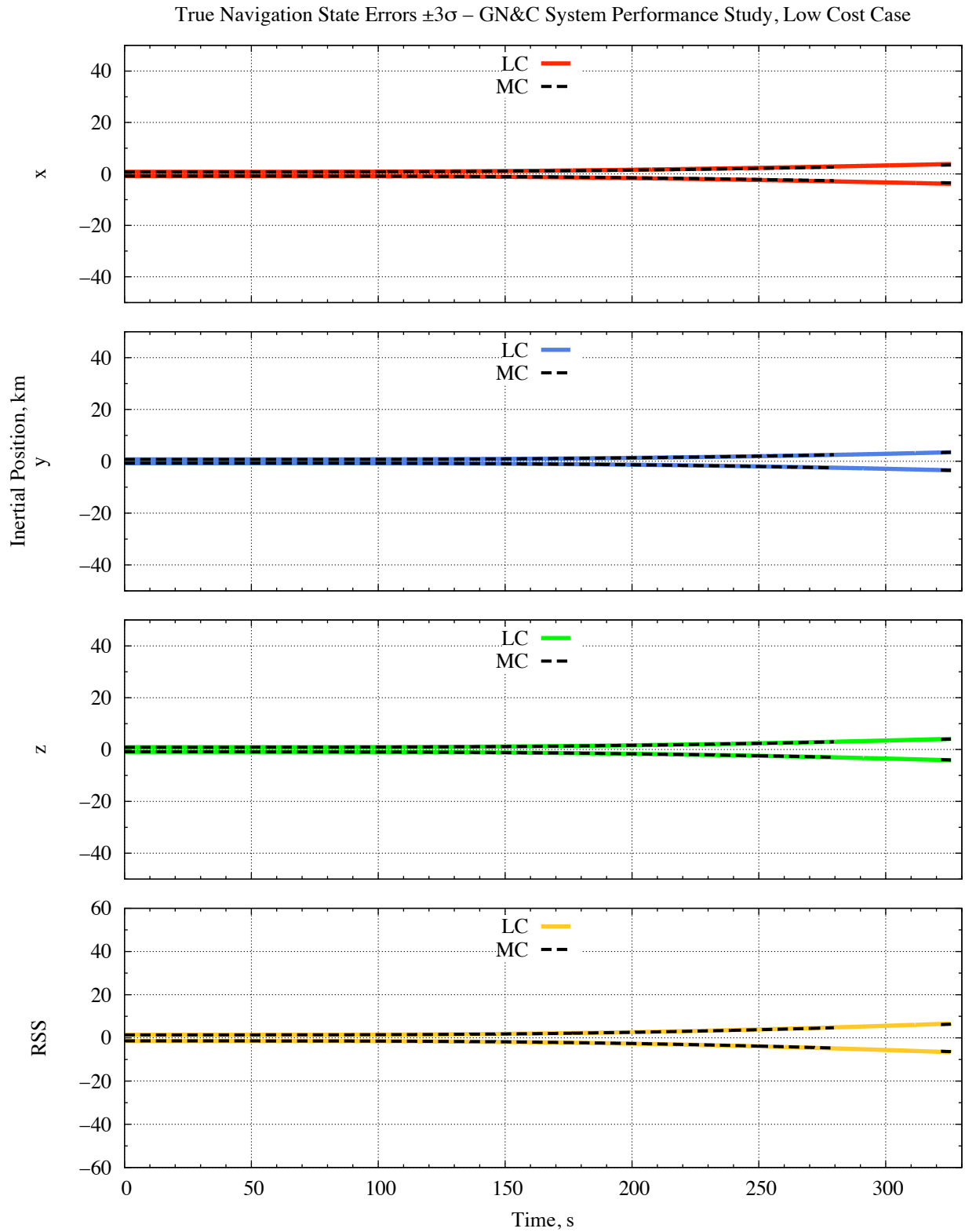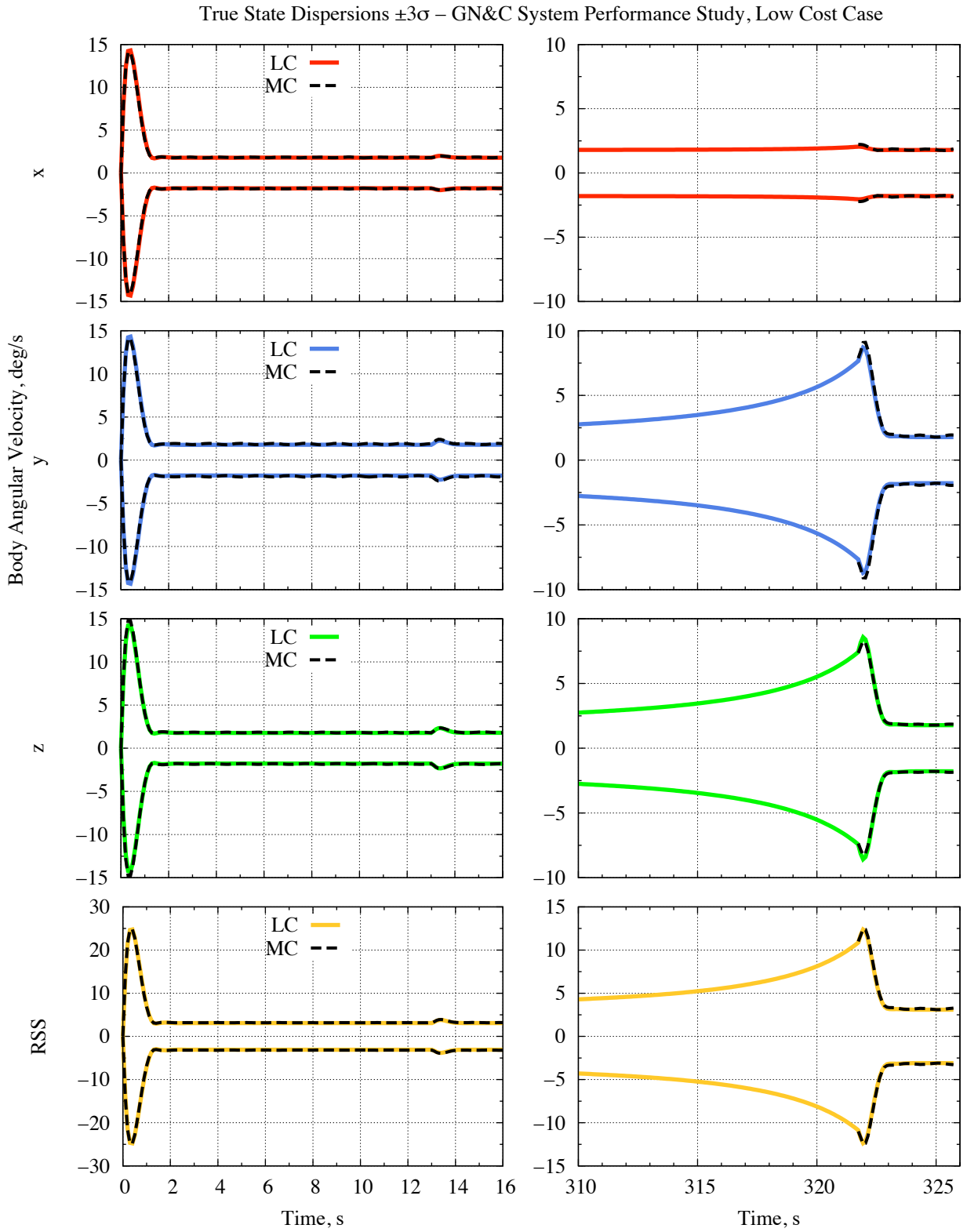
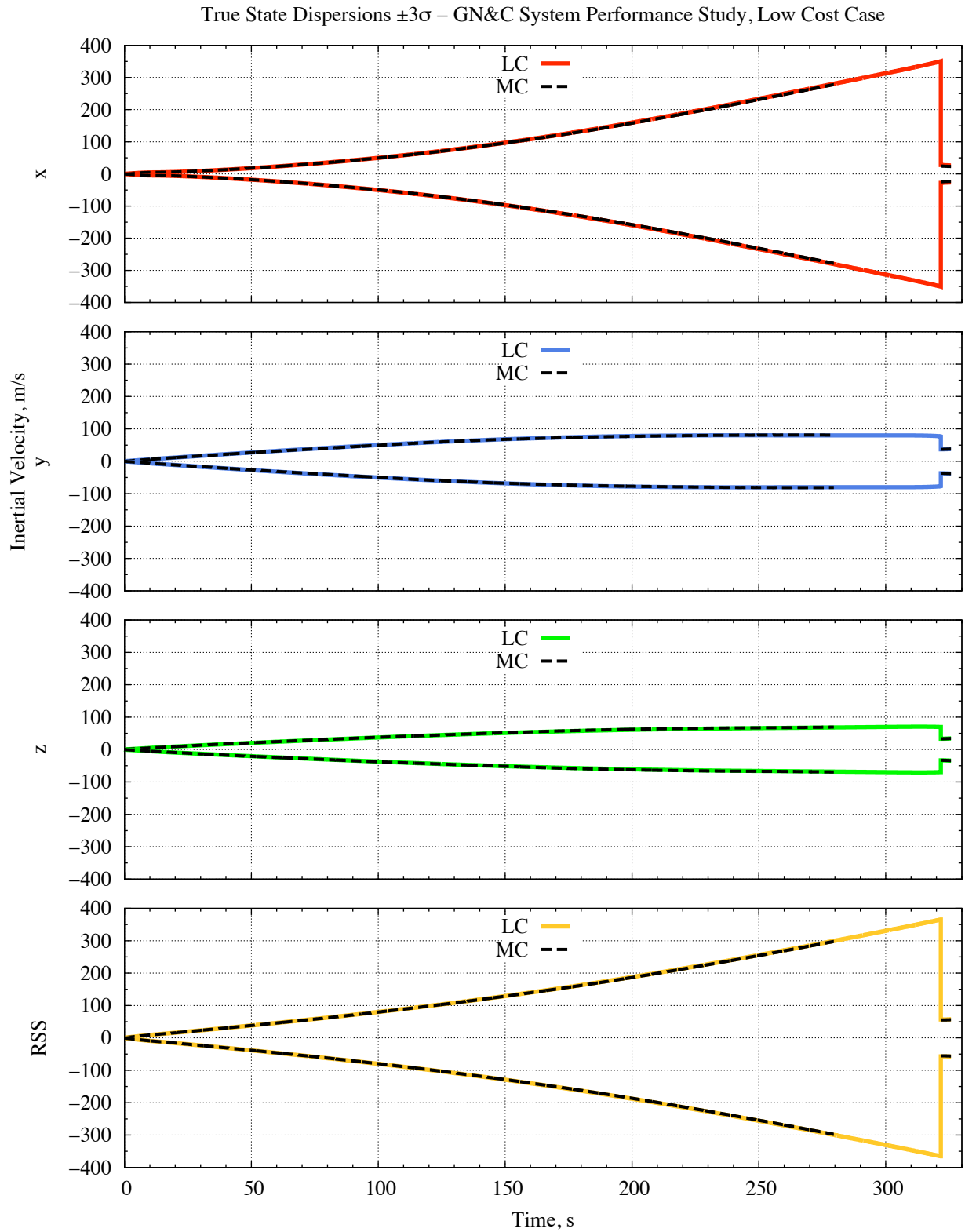Figure 7.33: (2 of 2) *GN&C system performance study, low cost case.* Nominal time history comparison of the body angular velocity true state dispersions $\pm 3\sigma$ from MC and LC analysis. Zoomed-in view of the first and last 16 s.
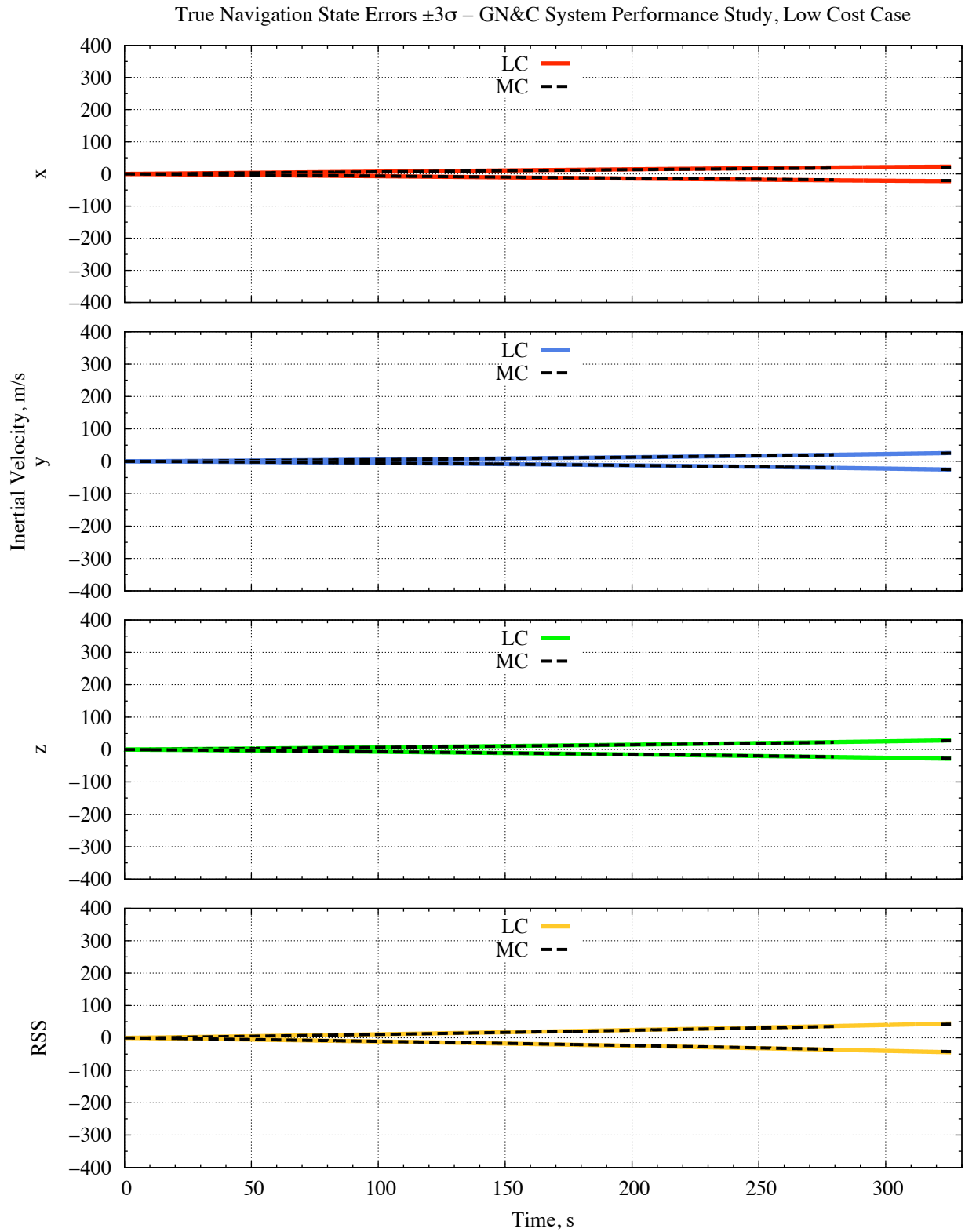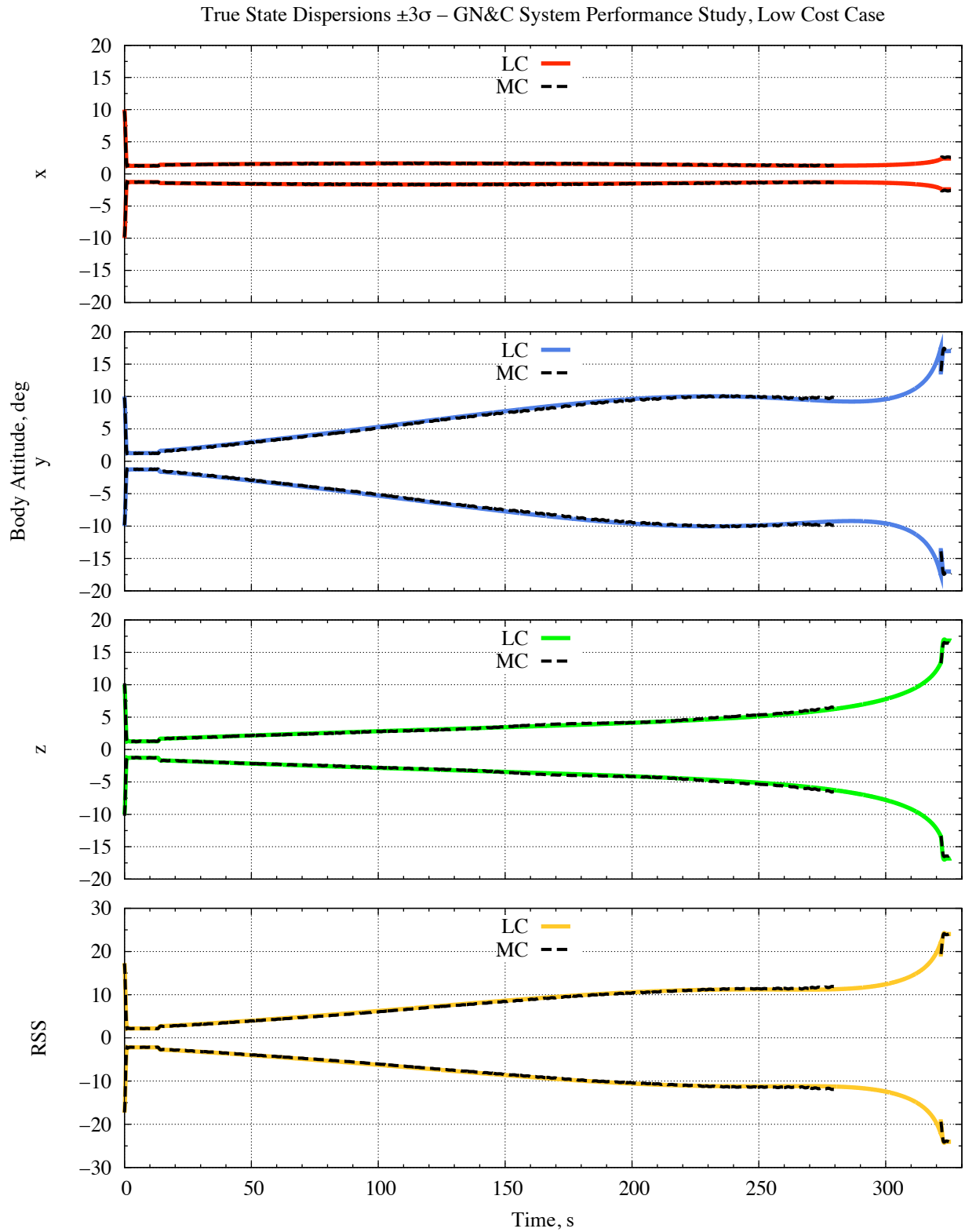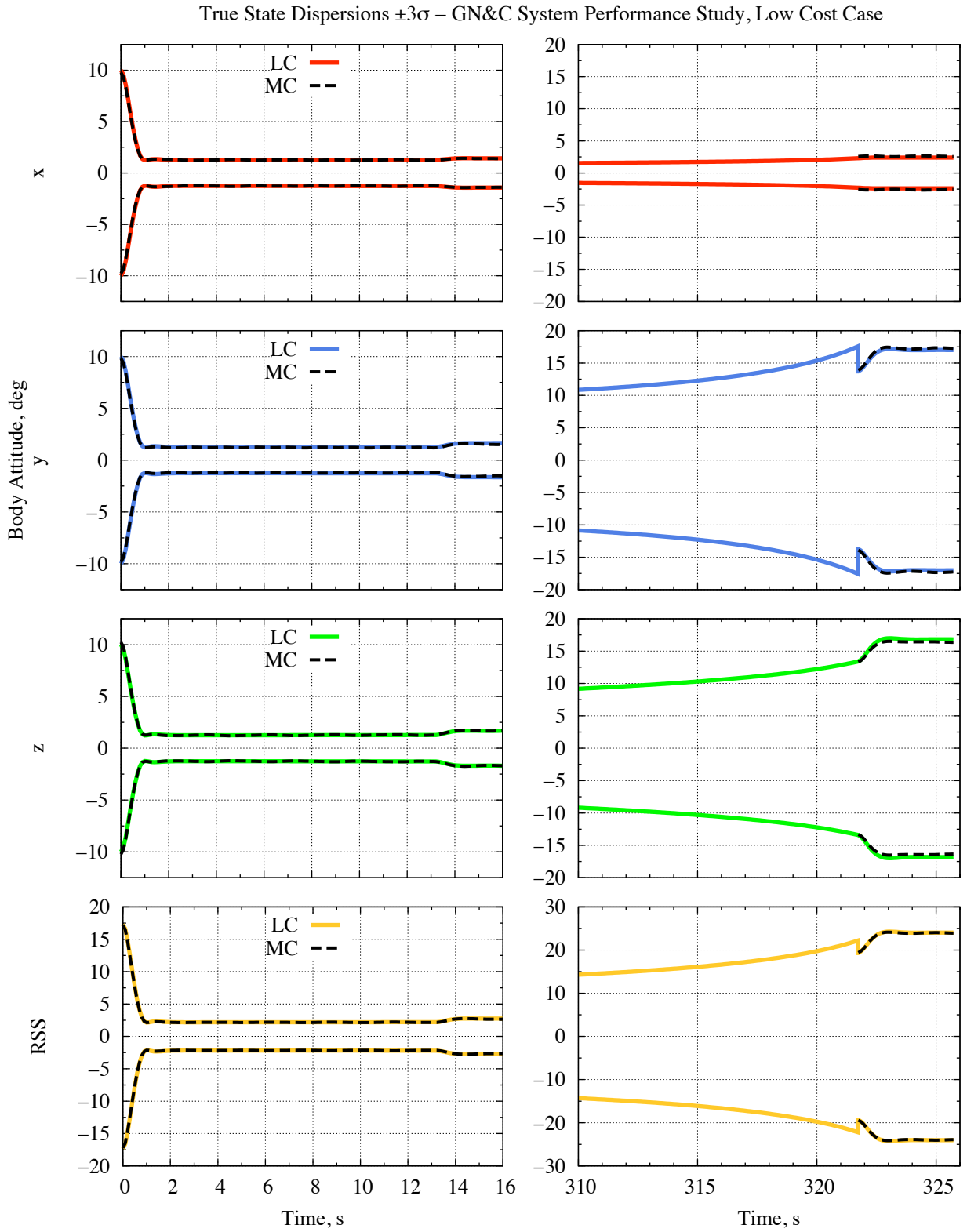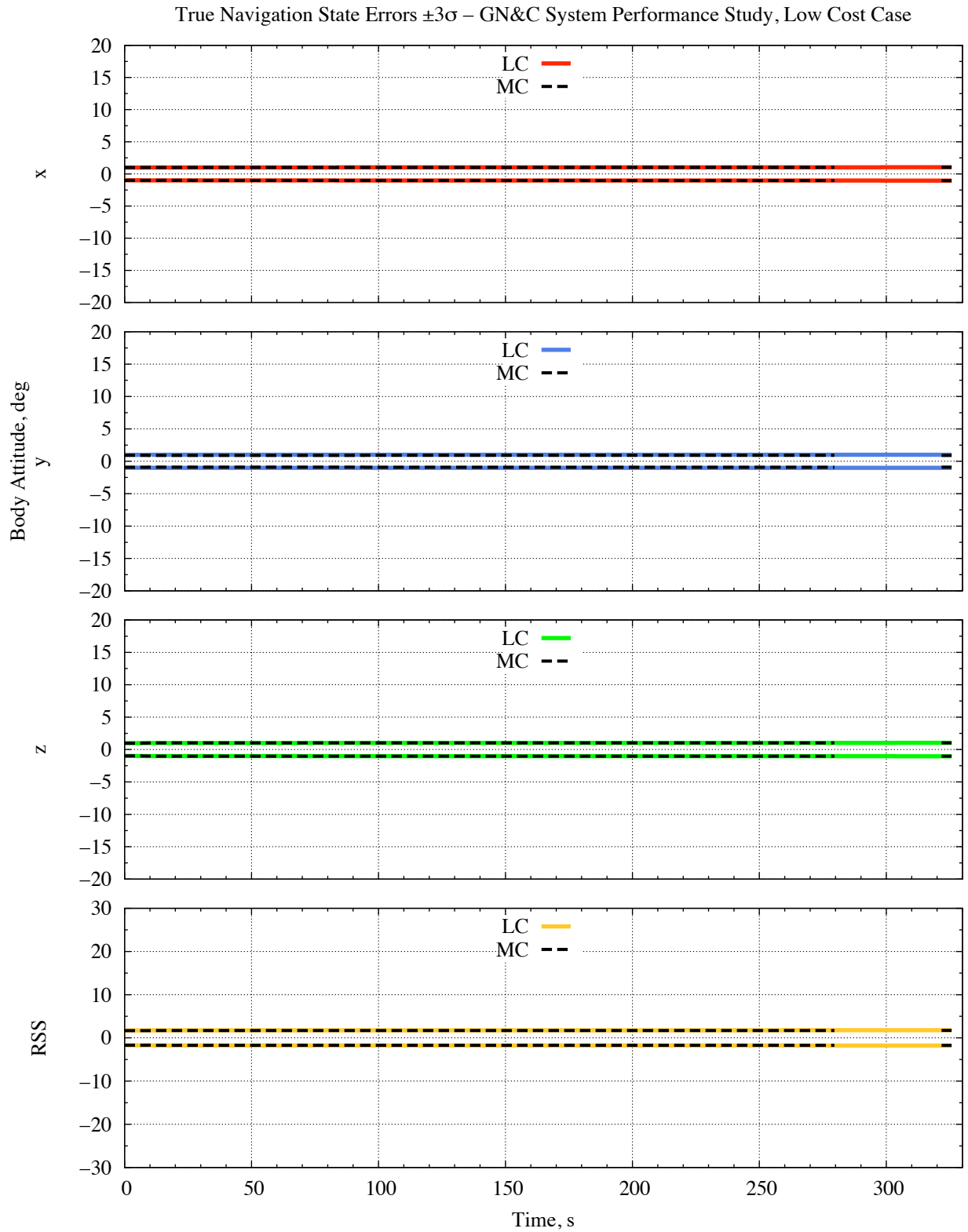
True State Dispersions ±3σ – GN&C System Performance Study, Low Cost Case
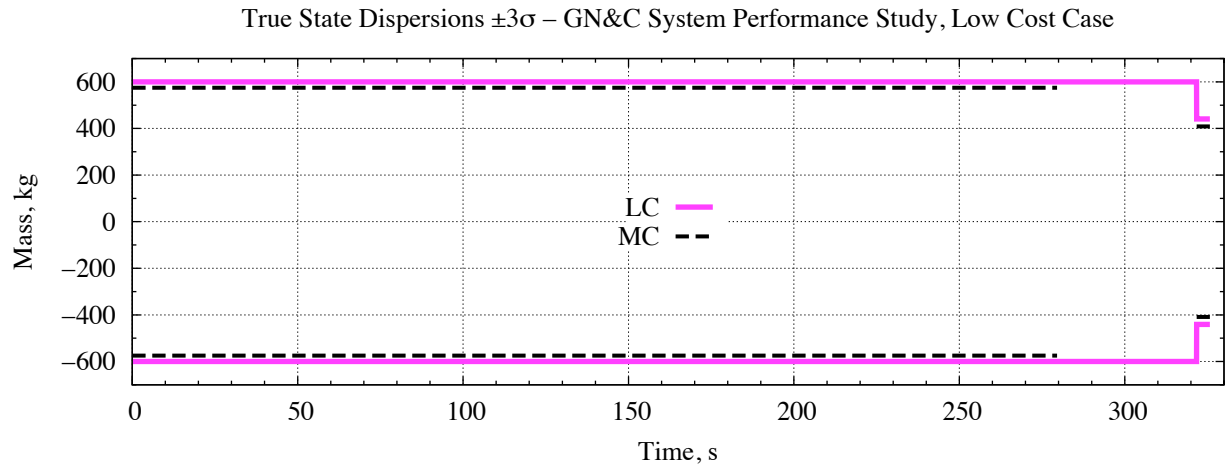
Figure 7.34: *GN&C system performance study, low cost case.* Nominal time history comparison of the wet mass true state dispersions ±3σ from MC and LC analysis.

Table 7.13: *GN&C system performance study, low cost case.* Inertial position true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Inertial Position, m							
		x		y		z		RSS	
		δx, 3σ	δe, 3σ	δx, 3σ	δe, 3σ	δx, 3σ	δe, 3σ	δx, 3σ	δe, 3σ
t_0	MC	765.136	831.260	722.200	698.374	860.062	883.209	1358.94	1399.56
	LC	809.800	809.800	737.883	737.883	899.862	899.862	1417.74	1417.74
	% Diff.	5.83734	2.58161	2.17167	5.65734	4.62762	1.88560	4.32740	1.29910
t_{PO}	MC	764.913	831.233	722.330	698.378	859.963	883.219	1358.82	1399.56
	LC	809.798	809.793	737.891	737.885	899.863	899.859	1417.75	1417.74
	% Diff.	5.86796	2.57936	2.15427	5.65695	4.63970	1.88402	4.33676	1.29923
t_{PEG}	MC	764.055	830.820	727.747	698.426	859.331	883.420	1360.83	1399.46
	LC	810.681	809.739	739.558	737.857	900.755	899.808	1419.69	1417.66
	% Diff.	6.10238	2.53736	1.62295	5.64572	4.82050	1.85511	4.32538	1.30057
t_{FC}	MC	27553.6	3401.98	12311.4	3378.65	9689.92	3906.64	31696.4	6184.70
	LC	28273.6	3762.81	12638.8	3385.09	10174.7	4031.58	32598.5	6470.80
	% Diff.	2.61318	10.6067	2.65951	0.190617	5.00335	3.19804	2.84589	4.62583
t_{MECO}	MC	27610.2	3480.97	12231.9	3474.31	9662.27	4009.54	31706.5	6345.43
	LC	28344.4	3850.10	12562.1	3482.17	10154.0	4139.34	32623.9	6639.50
	% Diff.	2.65915	10.6041	2.69981	0.225983	5.08940	3.23723	2.89330	4.63429

Table 7.14: *GN&C system performance study, low cost case.* Inertial velocity true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Inertial Velocity, m/s							
		x		y		z		RSS	
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	1.922E-03	1.859E-03	2.036E-03	2.213E-03	1.317E-05	1.317E-05	2.800E-03	2.890E-03
	LC	1.964E-03	1.964E-03	2.155E-03	2.155E-03	0	0	2.916E-03	2.916E-03
	% Diff.	2.17167	5.65301	5.86447	2.60981	100.000	100.000	4.13928	0.889205
t_{PO}	MC	1.75833	0.170911	1.84167	0.158675	1.69014	0.198354	3.05615	0.306157
	LC	1.79388	0.183067	1.88976	0.167241	1.65208	0.202873	3.08522	0.320375
	% Diff.	2.02202	7.11279	2.61170	5.39823	2.25216	2.27837	0.951308	4.64402
t_{PEG}	MC	4.90396	0.816996	7.67053	0.530353	6.32532	0.768767	11.0858	1.24087
	LC	4.90480	0.879501	7.75090	0.545341	6.34371	0.775941	11.1524	1.29345
	% Diff.	0.01708	7.65061	1.04774	2.82606	0.290677	0.933254	0.600550	4.23705
t_{FC}	MC	24.5465	20.3564	37.1457	24.5783	33.1968	26.6995	55.5371	41.6093
	LC	27.8981	22.3014	36.9733	24.9802	33.3579	27.8677	57.0796	43.5657
	% Diff.	13.6543	9.55481	0.464153	1.63528	0.485227	4.37527	2.77752	4.70172
t_{MECO}	MC	23.2176	20.5480	37.9419	25.0559	34.2015	27.1481	56.1105	42.2734
	LC	26.3675	22.5103	38.0199	25.4762	34.3458	28.3502	57.6228	44.2661
	% Diff.	13.56679	9.54959	0.205651	1.67747	0.421915	4.42807	2.695323	4.71383

Table 7.15: *GN&C system performance study, low cost case.* Body attitude true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Body Attitude, deg							
		x		y		z		RSS	
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{e}$, 3σ
t_0	MC	9.77717	1.00616	9.82049	0.929096	10.2322	0.968407	17.2259	1.67732
	LC	10.0000	1.00000	10.0000	1.00000	10.0000	1.00000	17.3205	1.73205
	% Diff.	2.27913	0.612134	1.82793	7.63154	2.26895	3.26241	0.549138	3.26329
t_{PO}	MC	1.23251	0.999850	1.25292	0.930107	1.25310	0.973565	2.15850	1.67709
	LC	1.25897	1.00000	1.25726	1.00000	1.25704	1.00000	2.17850	1.73206
	% Diff.	2.14679	0.015371	0.346564	7.51488	0.314501	2.71562	0.92631	3.27757
t_{PEG}	MC	1.25115	0.998978	1.19579	0.919623	1.27312	1.02054	2.14852	1.69857
	LC	1.27332	1.01432	1.26166	1.00215	1.27185	1.01525	2.19790	1.75039
	% Diff.	1.77221	1.53564	5.50876	8.97451	0.099690	0.518354	2.29832	3.05076
t_{FC}	MC	2.57219	1.03599	13.8880	0.920011	13.3303	1.02256	19.4214	1.72202
	LC	2.29953	1.03572	13.7229	1.00404	13.4120	1.03519	19.3258	1.77551
	% Diff.	10.6005	0.026274	1.18895	9.13401	0.612374	1.23426	0.492275	3.10642
t_{MECO}	MC	2.54970	1.03764	17.2580	0.917612	16.3489	1.02282	23.9088	1.72188
	LC	2.39186	1.03575	17.0073	1.00408	16.8073	1.03522	24.0303	1.77557
	% Diff.	6.19075	0.181648	1.45277	9.42371	2.80368	1.21170	0.508383	3.11803

Table 7.16: *GN&C system performance study, low cost case.* (*left*) Body angular velocity and (*right*) wet mass true state dispersions 3σ and normalized percent difference from MC and LC analysis at five key points along the trajectory.

		Body Angular Velocity, deg/s			
		x	y	z	RSS
		$\delta\mathbf{x}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{x}$, 3σ	$\delta\mathbf{x}$, 3σ
t_0	MC	2.4483E-05	1.1942E-05	2.6824E-05	3.8230E-05
	LC	2.3923E-05	1.1668E-05	2.6617E-05	3.7642E-05
	% Diff.	2.28457	2.29181	0.773281	1.53833
t_{PO}	MC	1.84023	1.79785	1.85905	3.17408
	LC	1.78453	1.77796	1.77710	3.08282
	% Diff.	3.02686	1.10656	4.40805	2.87522
t_{PEG}	MC	1.79565	1.91959	1.78869	3.17940
	LC	1.78623	1.77966	1.77832	3.08549
	% Diff.	0.52450	7.28944	0.579476	2.95369
t_{FC}	MC	2.21825	7.87991	7.35698	11.0063
	LC	2.04769	7.65876	7.45710	10.8838
	% Diff.	7.68853	2.80651	1.36090	1.11266
t_{MECO}	MC	1.87228	1.94386	1.85081	3.27254
	LC	1.78896	1.78189	1.78033	3.08951
	% Diff.	4.45030	8.33242	3.80787	5.59284

		Mass, kg
		$\delta\mathbf{x}$, 3σ
t_0	MC	574.629
	LC	600.000
	% Diff.	4.41514
t_{PO}	MC	574.629
	LC	600.000
	% Diff.	4.41514
t_{PEG}	MC	574.629
	LC	600.000
	% Diff.	4.41514
t_{FC}	MC	409.043
	LC	440.551
	% Diff.	7.70297
t_{MECO}	MC	409.043
	LC	440.551
	% Diff.	7.70297

Table 7.17: *GN&C system performance study, low cost case.* Orbit insertion dispersion results at MECO.

Parameter	Nominal	MC, 3σ	LC, 3σ	% Diff.
Burn Time, s	325.730	47.8166	48.1843	0.769103
Planetocentric Altitude, m	15239.7	3745.53	4096.97	9.38294
Inertial Velocity Magnitude, m/s	1686.13	12.0836	10.8708	10.0368
Flight Path Angle, deg	$-6.8248\mathrm{e}{-03}$	0.928371	0.999583	7.67068
Inclination, deg	145.765	0.670425	0.682743	1.83741
RAAN, deg	356.786	1.48380	1.51769	2.28369
Propellant Expenditure, kg	2657.22	409.043	440.551	7.70297

Figure 7.35: *GN&C system performance study.* Nominal time history comparison of the inertial position RSS (*top*) true state dispersions 3σ and (*bottom*) true navigation state errors 3σ for the baseline, high cost, and low cost cases from LC analysis .

for the baseline, high cost, and low cost cases. Figure 7.39 shows the nominal time history results from LC analysis of the wet mass true state dispersions 3σ for the baseline, high cost, and low cost cases. Notice that in order to properly view the LC results from all three cases, it was necessary to plot the vertical axes on a log-scale. It is observed that the results from the low and high cost cases are almost exactly a factor of 10 larger and smaller than the baseline case results. In all three cases, the normalized percent difference was computed at five key points along the trajectory, as explained in § 7.1.3. Tables 7.18–7.22 provide the MC and LC 3σ values and corresponding normalized percent difference for the total true state dispersions and total true navigation state errors (denoted respectively as $\delta\boldsymbol{x}$ and $\delta\boldsymbol{e}$) associated with the five launch vehicle states for the baseline, high cost, and low cost cases.
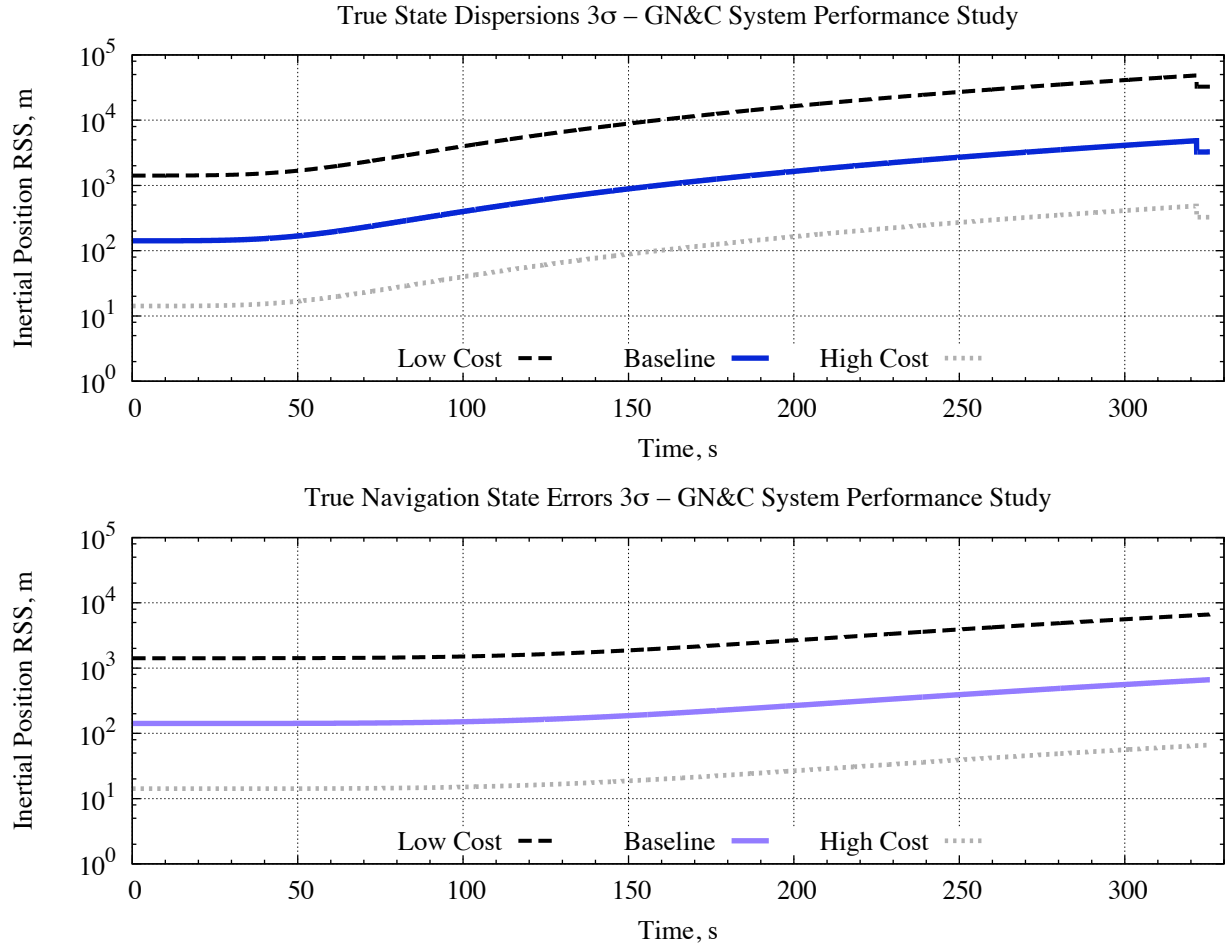
Figure 7.36: *GN&C system performance study.* Nominal time history comparison of the inertial velocity RSS (*top*) true state dispersions 3σ and (*bottom*) true navigation state errors 3σ for the baseline, high cost, and low cost cases from LC analysis .

Figure 7.37: *GN&C system performance study.* Nominal time history comparison of the body attitude RSS (*top*) true state dispersions 3σ and (*bottom*) true navigation state errors 3σ for the baseline, high cost, and low cost cases from LC analysis .

Figure 7.38: *GN&C system performance study.* Nominal time history comparison of the body angular velocity RSS true state dispersions 3σ for the baseline, high cost, and low cost cases from LC analysis .

True State Dispersions 3σ – GN&C System Performance Study
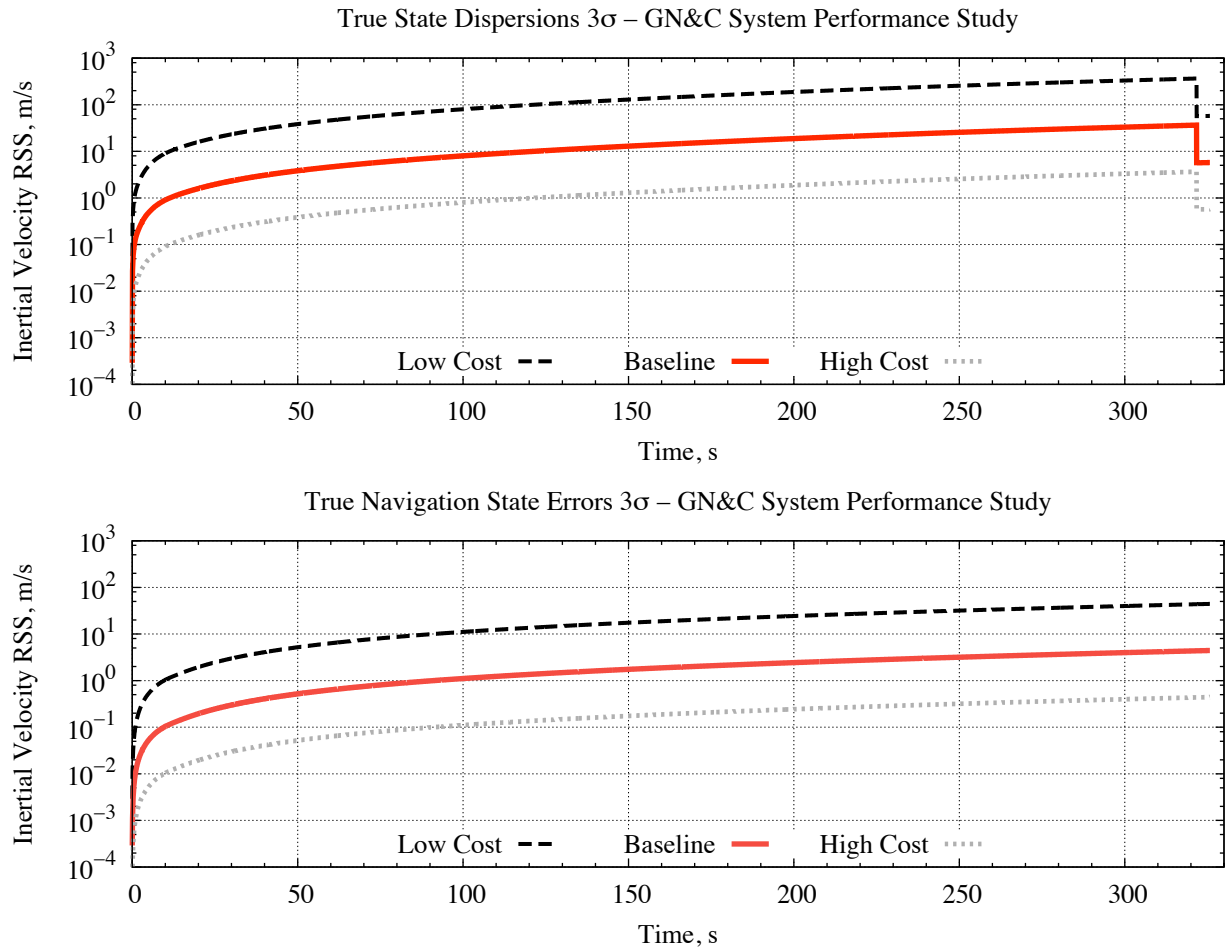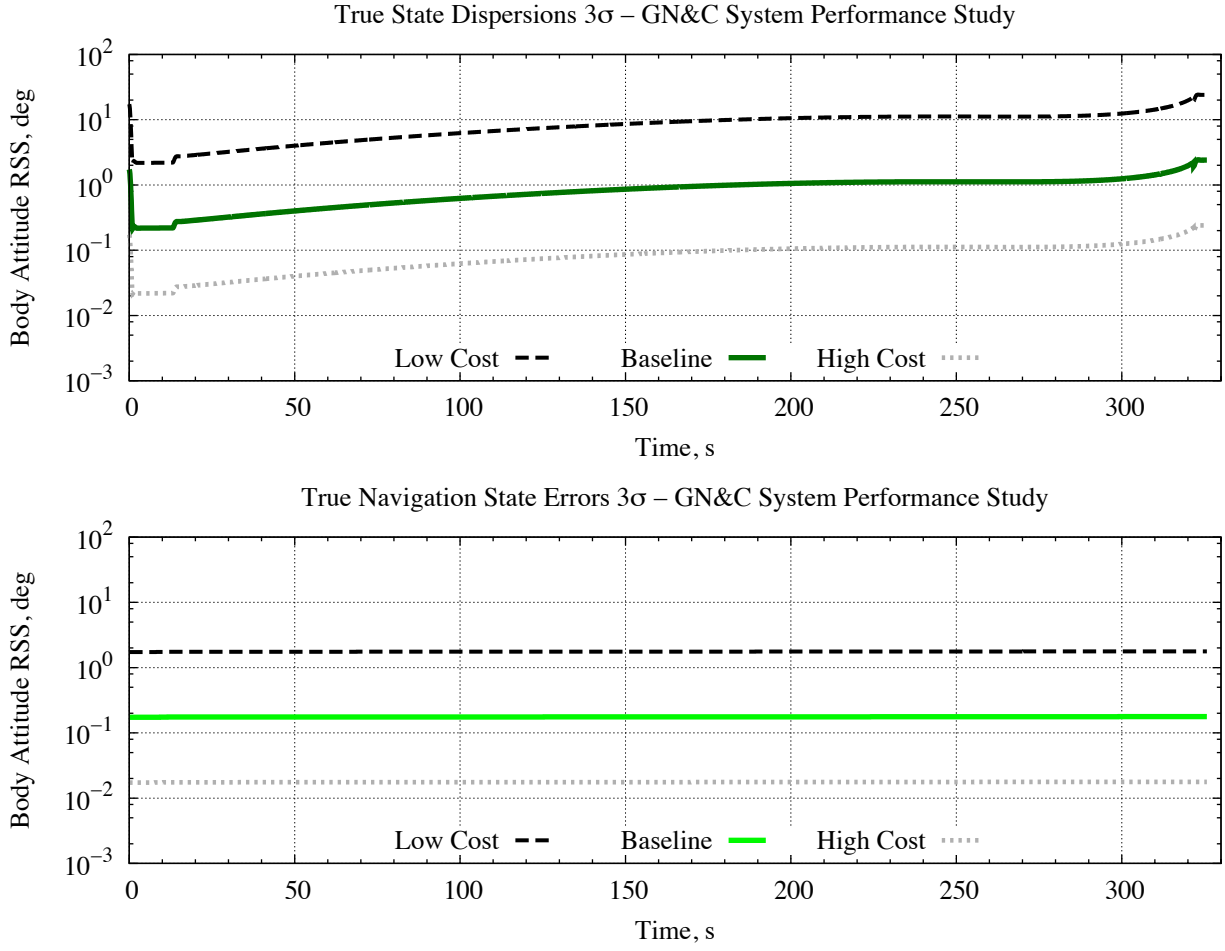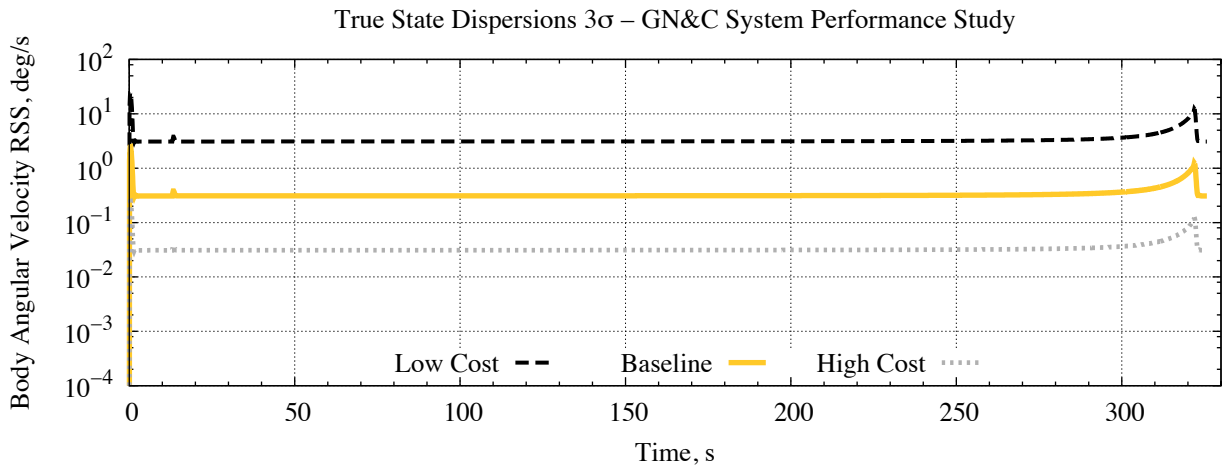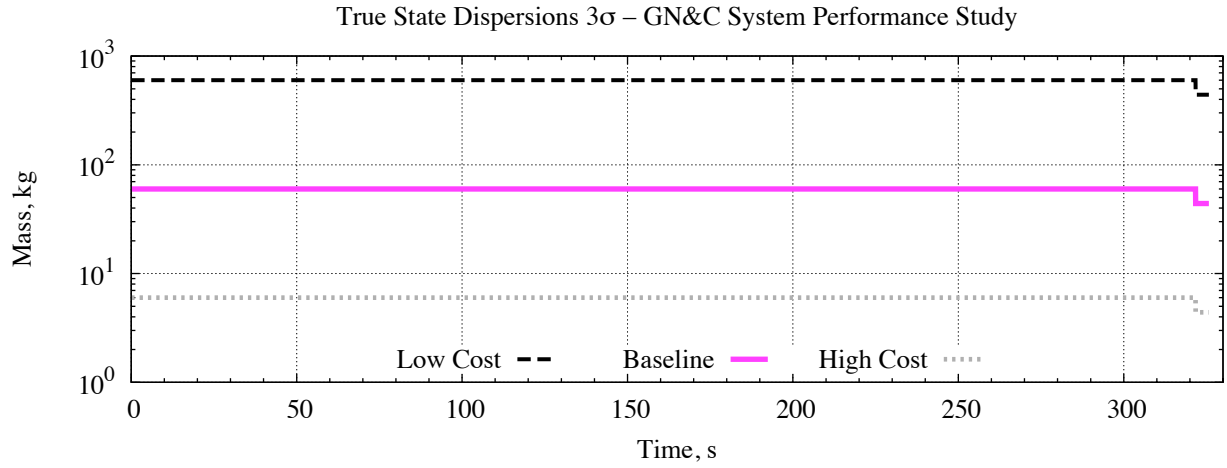

Figure 7.39: *GN&C system performance study.* Nominal time history comparison of the wet mass true state dispersions 3σ for the baseline, high cost, and low cost cases from LC analysis .

Table 7.18: *GN&C system performance study.* Inertial position RSS true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference for the baseline, high cost, and low cost cases from MC and LC analysis at five key points along the trajectory.

		Inertial Position RSS, m					
		$\delta\mathbf{x}$, 3σ			$\delta\mathbf{e}$, 3σ		
		High Cost	Baseline	Low Cost	High Cost	Baseline	Low Cost
t_0	MC	13.5894	135.894	1358.94	13.9956	139.956	1399.56
	LC	14.1774	141.774	1417.74	14.1774	141.774	1417.74
	% Diff.	4.32726	4.32732	4.32740	1.29923	1.29918	1.29910
t_{PO}	MC	13.5884	135.883	1358.82	13.9958	139.956	1399.56
	LC	14.1775	141.775	1417.75	14.1774	141.774	1417.74
	% Diff.	4.33538	4.33570	4.33676	1.29770	1.29909	1.29923
t_{PEG}	MC	13.6115	136.108	1360.83	13.9992	139.947	1399.46
	LC	14.1969	141.969	1419.69	14.1766	141.766	1417.66
	% Diff.	4.30016	4.30553	4.32538	1.26730	1.29990	1.30057
t_{FC}	MC	315.593	3150.69	31696.4	62.6034	618.718	6184.70
	LC	325.985	3259.85	32598.5	64.7080	647.080	6470.80
	% Diff.	3.29276	3.46451	2.84589	3.36181	4.58396	4.62583
t_{MECO}	MC	315.673	3151.49	31706.5	64.2067	634.885	6345.43
	LC	326.239	3262.39	32623.9	66.3950	663.950	6639.50
	% Diff.	3.34712	3.51890	2.89330	3.40815	4.57791	4.63429

Table 7.19: *GN&C system performance study.* Inertial velocity RSS true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference for the baseline, high cost, and low cost cases from MC and LC analysis at five key points along the trajectory.

| | | Inertial Velocity RSS, m/s | | | | | |
| | | $\delta\mathbf{x}$, 3σ | | | $\delta\mathbf{e}$, 3σ | | |
		High Cost	Baseline	Low Cost	High Cost	Baseline	Low Cost
t_0	MC	2.801E-05	2.801E-04	2.800E-03	2.890E-05	2.890E-04	2.890E-03
	LC	2.916E-05	2.916E-04	2.916E-03	2.916E-05	2.916E-04	2.916E-03
	% Diff.	4.11094	4.10416	4.13928	0.912418	0.90718	0.889205
t_{PO}	MC	3.0614E-02	0.306101	3.05615	3.0924E-03	3.1417E-02	0.306157
	LC	3.0852E-02	0.308522	3.08522	3.2038E-03	3.2038E-02	0.320375
	% Diff.	0.779606	0.790819	0.951308	3.60037	1.97603	4.64402
t_{PEG}	MC	0.111112	1.11123	11.0858	1.2409E-02	0.125511	1.24087
	LC	0.111524	1.11524	11.1524	1.2934E-02	0.129345	1.29345
	% Diff.	0.370583	0.36052	0.600550	4.23684	3.05444	4.23705
t_{FC}	MC	0.530788	5.42088	55.5371	0.415228	4.18750	41.6093
	LC	0.570796	5.70796	57.0796	0.435657	4.35657	43.5657
	% Diff.	7.53756	5.29581	2.77752	4.91993	4.03741	4.70172
t_{MECO}	MC	0.541171	5.55100	56.1105	0.421903	4.25545	42.2734
	LC	0.576228	5.76228	57.6228	0.442661	4.42661	44.2661
	% Diff.	6.47797	3.80620	2.69532	4.92011	4.02198	4.71383

Table 7.20: *GN&C system performance study.* Body attitude RSS true state dispersions 3σ, true navigation state errors 3σ, and normalized percent difference for the baseline, high cost, and low cost cases from MC and LC analysis at five key points along the trajectory.

| | | Body Attitude RSS, deg | | | | | |
| | | $\delta\mathbf{x}$, 3σ | | | $\delta\mathbf{e}$, 3σ | | |
		High Cost	Baseline	Low Cost	High Cost	Baseline	Low Cost
t_0	MC	0.172215	1.72261	17.2259	1.6782E-02	0.167772	1.67732
	LC	0.173205	1.73205	17.3205	1.7321E-02	0.173205	1.73205
	% Diff.	0.574686	0.548281	0.549138	3.20948	3.23829	3.26329
t_{PO}	MC	2.1487E-02	0.217323	2.15850	1.6780E-02	0.167768	1.67709
	LC	2.1785E-02	0.217850	2.17850	1.7321E-02	0.173206	1.73206
	% Diff.	1.38631	0.242276	0.926306	3.22299	3.24134	3.27757
t_{PEG}	MC	2.1584E-02	0.217552	2.14852	1.7040E-02	0.169810	1.69857
	LC	2.1979E-02	0.219790	2.19790	1.7504E-02	0.175039	1.75039
	% Diff.	1.82773	1.02868	2.29832	2.72365	3.07945	3.05076
t_{FC}	MC	0.199809	2.01500	19.4214	1.7316E-02	0.172348	1.72202
	LC	0.193258	1.93258	19.3258	1.7755E-02	0.177551	1.77551
	% Diff.	3.27865	4.09049	0.492275	2.53443	3.01868	3.10642
t_{MECO}	MC	0.245034	2.46310	23.9088	1.7319E-02	0.172357	1.72188
	LC	0.240303	2.40303	24.0303	1.7756E-02	0.177557	1.77557
	% Diff.	1.93090	2.43889	0.508383	2.52106	3.01696	3.11803

Table 7.21: *GN&C system performance study.* Body angular velocity RSS true state dispersions 3σ and normalized percent difference for the baseline, high cost, and low cost cases from MC and LC analysis at five key points along the trajectory.

| | | Body Angular Velocity RSS, deg/s | | |
| | | $\delta\mathbf{x}$, 3σ | | |
		High Cost	Baseline	Low Cost
t_0	MC	3.8210E-07	3.8219E-06	3.8230E-05
	LC	3.7642E-07	3.7642E-06	3.7642E-05
	% Diff.	1.48497	1.50967	1.53833
t_{PO}	MC	3.1688E-02	0.316880	3.17408
	LC	3.0828E-02	0.308282	3.08282
	% Diff.	2.71465	2.71335	2.87522
t_{PEG}	MC	3.1858E-02	0.318469	3.17940
	LC	3.0855E-02	0.308549	3.08549
	% Diff.	3.14937	3.11488	2.95369
t_{FC}	MC	0.112599	1.10241	11.0063
	LC	0.108838	1.08838	10.8838
	% Diff.	3.33940	1.27190	1.11266
t_{MECO}	MC	3.1610E-02	0.317747	3.27254
	LC	3.0895E-02	0.308951	3.08951
	% Diff.	2.26174	2.76824	5.59284

Table 7.22: *GN&C system performance study.* Wet mass true state dispersions 3σ and normalized percent difference for the baseline, high cost, and low cost cases from MC and LC analysis at five key points along the trajectory.

| | | Mass, kg | | |
| | | $\delta\mathbf{x}$, 3σ | | |
		High Cost	Baseline	Low Cost
t_0	MC	5.74629	57.4629	574.629
	LC	6.00000	60.0000	600.000
	% Diff.	4.41514	4.41514	4.41514
t_{PO}	MC	5.74629	57.4629	574.629
	LC	6.00000	60.0000	600.000
	% Diff.	4.41514	4.41514	4.41514
t_{PEG}	MC	5.74629	57.4629	574.629
	LC	6.00000	60.0000	600.000
	% Diff.	4.41514	4.41514	4.41514
t_{FC}	MC	4.10670	40.9902	409.043
	LC	4.40551	44.0551	440.551
	% Diff.	7.27626	7.47724	7.70297
t_{MECO}	MC	4.10670	40.9902	409.043
	LC	4.40551	44.0551	440.551
	% Diff.	7.27626	7.47724	7.70297

Chapter 8

Conclusions and Future Research

"By prevailing over all obstacles and distractions, one

may unfailingly arrive at his chosen goal or destination."

– Christopher Columbus

In recent years there has been considerable effort to develop alternative methods or techniques that are capable of achieving equivalent results to those of Monte Carlo methods, but in a fraction of the time. One of these alternative techniques, linear covariance analysis, has been successfully applied in this research to create an accurate and fast ascent GN&C analysis tool. This research has shown that linear covariance analysis provides the accuracy required for preliminary ascent GN&C system design. Statistical results from linear covariance analysis are generally within 10% of Monte Carlo results, and in most cases the differences are less than 5%. This is an excellent result given the many complex nonlinearities that are embedded in the ascent GN&C problem.

However, the real value of this tool lies in its speed. CPU times were collected using the `tic/toc` functions in MATLAB R2010a on a MacBook laptop with Mac OS X 10.5.8, 2.4 GHz Intel Core 2 Duo processor, and 2 GB 667 MHz DDR2 SDRAM. The linear covariance data was generated for the baseline, high cost, and low cost cases in 357.108 s (5 min 57.108 s), 357.657 s, and 357.089 s, respectively. The Monte Carlo data, for the baseline case, was generated with an average of 740.369 s (12 min 20.369 s) per sample run and a standard deviation of 4.0872 s. The total time for 500 sample runs is approximately 370184.5 s (4 days 6 hrs 49 min 44.5 s). Therefore, the linear covariance simulation is 1036.62 times faster than the Monte Carlo simulation.

To achieve this result, there were several challenging theoretical and technical research problems that needed to be solved. The first was the implementation of a closed-loop ascent guidance algorithm in a linear covariance analysis environment. Ascent guidance algorithms such as the Space Shuttle PEG are very complex, highly nonlinear, and iterative in nature. As a result, an algorithm like PEG poses a substantial challenge to a linear covariance simulation. In this research,

the linearization of this guidance law was successfully accomplished using a very careful numerical approach and a complex-step derivative technique for computing numerical partial derivatives.

A second major challenge was to determine how to address the fine-count partial-derivative dilemma. During fine count, the last commanded inertial thrust direction computed by PEG is held constant, which corresponds to a constant attitude command and a zero attitude rate command. This represents an in-flight guidance mode change that is not easily described in a linear covariance context. The solution was to first represent the guidance mode change as an event trigger, and then to introduce a pseudo-state for the attitude-hold command. While the theory for the event trigger is well known, the addition of a pseudo-state was a new and key element of this research. The pseudo-state effectively collected and captured the correlations between the attitude-hold command and all system uncertainties. Then, during the final seconds of flight, these correlations are used to effectively propagate the final rocket position and velocity errors based on the variance of the attitude-hold command and the variance of the associated pseudo-state. This was a key important development of this research.

The continuation of this research is important and the next steps to be taken can be divided into two groups: detailed ascent design models, and system design optimization. In the area of detailed linear covariance ascent design models what is needed most are models for a stochastic atmosphere and models for ascent vehicle staging. For the atmosphere model, an important step forward will be to convert a know stochastic atmosphere model, e.g., Earth Global Reference Atmospheric Model (GRAM), into a covariance-based atmosphere model. For vehicle staging, event triggers will need to be implemented for each stage. The addition of these two models will significantly increase the range of ascent guidance/launch vehicle applications that can be studied using linear covariance techniques.

The second area of potential future research exploits the speed of linear covariance analysis to design and optimize ascent GN&C systems. In the research just presented, only three system configurations were evaluated: the baseline (or nominal), high cost, and low cost. In a more detailed future analysis, a full spectrum/range of actuators, sensors, and disturbance models will need to be considered. By exploiting the efficiency of linear covariance analysis, it becomes possible to imbed the linear covariance tool inside a Monte Carlo analysis where random samples/combinations of sensors and actuators are selected. For each sample, key mission performance metrics can

be compared with mission requirements to determine suitable, cost-effective, hardware system configurations.

Finally, it is also feasible to develop analytical and/or numerical optimization tools that utilize linear covariance techniques. Once again, it is the efficiency of the linear covariance analysis that makes this research topic feasible. The goal is to create a stochastic optimization tool that receives mission requirements as input and provides optimal hardware configurations that meet those mission requirements as output.

Clearly, there is considerably more work to be done. The research presented in this dissertation has made the first important steps towards implementing the complexities of ascent GN&C analysis techniques in a linear covariance analysis environment. Future developments in ascent GN&C system design and optimization will be able to build upon the results of this research.

References

[1] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Vol. 1, Navtech Book & Software Store, Arlington, VA, 1994.

[2] Merel, M. H. and Mullin, F. J., "Analytic Monte Carlo Error Analysis," *Journal of Spacecraft and Rockets*, Vol. 5, No. 11, Nov 1968, pp. 1304–1308.

[3] Johnson, R. A. and Bhattacharyya, G. K., *Statistics: Principles and Methods*, John Wiley & Sons, Inc., Hoboken, NJ, 4th ed., 2001.

[4] Geller, D. K., "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, Nov–Dec 2006, pp. 1404–1414.

[5] Gelb, A., Kasper, Jr., J. F., Nash, Jr., R. A., Price, C. F., and Sutherland, Jr., A. A., *Applied Optimal Estimation*, Massachusetts Institute of Technology Press, Cambridge, MA, 1974, pp. 229–276, Chap. 7.

[6] Moch, T. K., "Determination of Launch Vehicle Trajectory Dispersions," *Journal of Spacecraft and Rockets*, Vol. 3, No. 11, Nov 1966, pp. 1682–1685.

[7] Chen, C.-T., *Linear System Theory and Design*, Oxford University Press, New York, NY, 3rd ed., 1999.

[8] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.

[9] Geller, D. K., "Analysis of the Relative Attitude Estimation and Control Problem for Satellite Inspection and Orbital Rendezvous," *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, Apr–Jun 2007, pp. 195–214.

[10] Woffinden, D. C. and Geller, D. K., "Relative Angles-Only Navigation and Pose Estimation for Autonomous Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, Sep–Oct 2007, pp. 1455–1469.

[11] Geller, D. K., "Orbital Rendezvous: When Is Autonomy Required?" *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 4, Jul–Aug 2007, pp. 974–981.

[12] Geller, D. K. and Christensen, D. P., "Linear Covariance Analysis for Powered Lunar Descent and Landing," *Journal of Spacecraft and Rockets*, Vol. 46, No. 6, Nov–Dec 2009, pp. 1231–1248.

[13] Christensen, D. P., *Terrain-Relative and Beacon-Relative Navigation for Lunar Powered Descent and Landing*, Master's thesis, Utah State University, Logan, UT, 2009.

[14] Stastny, N. B. and Geller, D. K., "Autonomous Optical Navigation at Jupiter: A Linear Covariance Analysis," *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, Mar–Apr 2008, pp. 290–298.

[15] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, Jan–Feb 2009, pp. 102–111.

[16] Stastny, N. B., *Autonomous Optical Navigation at Jupiter: A Linear Covariance Analysis*, Master's thesis, Utah State University, Logan, UT, 2006.

[17] Stoll, J. C., *Performance Analysis of a GPS Interferometric Attitude Determination System for a Gravity Gradient Stabilized Spacecraft*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[18] Carpenter, J. R. and Markley, F. L., "Generalized Linear Covariance Analysis," *The F. Landis Markley Astrodynamics Symposium*, No. AAS 08-273, American Astronautical Society, Cambridge, MA, Jun–Jul 2008.

[19] Markley, F. L., Seidewitz, E., and Nicholson, M., "A General Model for Attitude Determination Error Analysis," *NASA Conference: Flight Mechanics/Estimation Theory Symposium*, No. N89-15935, Greenbelt, MD, 1988, pp. 3–25.

[20] Markley, F. L., Seidewitz, E., and Deutschmann, J., "Attitude Determination Error Analysis: General Model and Specific Application," *Proceedings of the International Symposium*, Space Dynamics, Toulouse, France, Nov 1989, pp. 251–266.

[21] Visser, B. L., *An Application of Linear Covariance Analysis to the Design of Responsive Near-Rendezvous Missions*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 2007.

[22] Hassannia, J. H., *Navigation Performance and Integrity Monitoring for Ballistic Missiles Using All-In-View GPS*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1994.

[23] Gingiss, A. J., *Navigation Analysis of Earth-Moon Libration Point Missions*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1992.

[24] Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, Burlington, MA, 2004, pp. 387–434, Chap. 6.

[25] Tang, G.-J., Luo, Y.-Z., and Li, H.-Y., "Optimal Robust Linearized Impulsive Rendezvous," *Aerospace Science and Technology*, Vol. 11, 2007, pp. 563–569.

[26] Wolfe, C. A., "Linear Covariance Analysis and Sensitivity to a Class of Error Sources for Suboptimal Recursive Filters," *IEEE Conference on Decision and Control: 16th Symposium on Adaptive Processes and a Special Symposium on Fuzzy Set Theory and Applications*, Vol. 16, Institute of Electrical and Electronics Engineers, Dec 1977, pp. 206–209.

[27] Park, H. W., Lee, J. G., and Park, C. G., "Covariance Analysis of Strapdown INS Considering Gyrocompass Characteristics," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 31, No. 1, Jan 1995, pp. 320–328.

[28] Gossner, J. R., *An Analytic Method of Propagating a Covariance Matrix to a Maneuver Condition for Linear Covariance Analysis During Rendezvous*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1991.

[29] Moesser, T. J., *Guidance and Navigation Linear Covariance Analysis for Lunar Powered Descent*, Master's thesis, Utah State University, Logan, UT, 2010.

[30] Park, S.-Y., "Launch Vehicle Trajectories with a Dynamic Pressure Constraint," *Journal of Spacecraft and Rockets*, Vol. 35, No. 6, Nov–Dec 1998, pp. 765–773.

[31] Dukeman, G. A., *Closed-Loop Nominal and Abort Atmospheric Ascent Guidance for Rocket-Powered Launch Vehicles*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA, 2005.

[32] Lu, P., Sun, H., and Tsai, B., "Closed-Loop Endoatmospheric Ascent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, Mar–Apr 2003, pp. 283–294.

[33] Calise, A. J., Melamed, N., and Lee, S., "Design and Evaluation of a Three-Dimensional Optimal Ascent Guidance Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, Nov–Dec 1998, pp. 867–875.

[34] Rao, P. P., Sutter, B. M., and Hong, P. E., "Six-Degree-of-Freedom Trajectory Targeting and Optimization for Titan Launch Vehicles," *Journal of Spacecraft and Rockets*, Vol. 34, No. 3, May–Jun 1997, pp. 341–346.

[35] Hanson, J. M., Shrader, M. W., and Cruzen, C. A., "Ascent Guidance Comparisons," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 1994-3568, American Institute of Aeronautics and Astronautics, Scottsdale, AZ, Aug 1994.

[36] Leung, M. S. K. and Calise, A. J., "Hybrid Approach to Near-Optimal Launch Vehicle Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 5, Sep–Oct 1994, pp. 881–888.

[37] Hanson, J. M., Shrader, M. W., and Chang, H., "Guidance and Dispersion Studies of National Launch System Ascent Trajectories," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 1992-4306, American Institute of Aeronautics and Astronautics, Hilton Head Island, SC, Aug 1992.

[38] Vittal, R. V. and Bhat, M. S., "An Explicit Closed-Loop Guidance for Launch Vehicles," *Acta Astronautica*, Vol. 25, No. 3, 1991, pp. 119–129.

[39] Sinha, S. K. and Shrivastava, S. K., "Optimal Explicit Guidance of Multistage Launch Vehicle Along Three-Dimensional Trajectory," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, May–Jun 1990, pp. 394–403.

[40] Wie, B., Du, W., and Whorton, M., "Analysis and Design of Launch Vehicle Flight Control Systems," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 2008-6291, American Institute of Aeronautics and Astronautics, Honolulu, HI, Aug 2008.

[41] Seywald, H. and Cliff, E. M., "Neighboring Optimal Control Based Feedback Law for the Advanced Launch System," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, Nov–Dec 1994, pp. 1154–1162.

[42] Schleich, W. T., "Shuttle Vehicle Configuration Impact on Ascent Guidance and Control," *Journal of Guidance, Control, and Dynamics*, Vol. 7, No. 3, May–Jun 1984, pp. 338–346.

[43] Du, W., Wie, B., and Whorton, M., "Dynamic Modeling and Flight Control Simulation of a Large Flexible Launch Vehicle," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 2008-6620, American Institute of Aeronautics and Astronautics, Honolulu, HI, Aug 2008.

[44] Miele, A., Wang, T., and Melvin, W. W., "Optimization and Acceleration Guidance of Flight Trajectories in a Windshear," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, Jul–Aug 1987, pp. 368–377.

[45] Pearson, S. D., Vaughan, W. W., Batts, G. W., and Jasper, G. L., "Natural Terrestrial Environment Important to Advanced Launch Vehicle Design and Development," *35th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 1997-0186, American Institute of Aeronautics and Astronautics, Reno, NV, Jan 1997.

[46] Pearson, S. D., Jasper, G. L., Vaughan, W. W., and Batts, G. W., "Atmosphereic Flight Mechanics and Natural Terrestrial Environment Issues," *35th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 1997-0534, American Institute of Aeronautics and Astronautics, Reno, NV, Jan 1997.

[47] Dukeman, G. A. and Hill, A. D., "Rapid Trajectory Optimization For the ARES I Launch Vehicle," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 2008-6288, American Institute of Aeronautics and Astronautics, Honolulu, HI, Aug 2008.

[48] Sostaric, R. R. and Merriam, R. S., "Lunar Ascent and Rendezvous Trajectory Design," *31st Annual AAS Rocky Mountain Guidance and Control Conference*, No. AAS 08-066, American Astronautical Society, Breckenridge, CO, Feb 2008.

[49] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, Mar–Apr 1998, pp. 193–207.

[50] Shaver, D. A. and Hull, D. G., "Advanced Launch System Trajectory Optimization Using Suboptimal Control," *AIAA Guidance, Navigation and Control Conference, August 20–22, 1990, Portland, OR*, No. AIAA 1990-3413, American Institute of Aeronautics and Astronautics, Washington, DC, 1990, pp. 892–901.

[51] Seywald, H. and Cliff, E. M., "Optimal Rocket-Powered Ascent Study," *8th IEEE 1989 American Control Conference*, Institute of Electrical and Electronics Engineers, Pittsburgh, PA, Jun 1989.

[52] Morton, B., "Dispersion Analysis of Propulsion System Performance in Ascent-to-Orbit Missions," *AIAA/ASME/SAE/ASEE 25th Joint Propulsion Conference*, No. AIAA 1989-2342, American Institute of Aeronautics and Astronautics, Monterey, CA, Jul 1989.

[53] Walter, U., *Astronautics*, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2008, pp. 119–121, Chap. 6.

[54] Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sep–Oct 1982, pp. 417–429.

[55] Pittelkau, M. E., "Rotation Vector in Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, Nov–Dec 2003, pp. 855–860.

[56] Rao, A. V., *Dynamics of Particles and Rigid Bodies: A Systematic Approach*, Cambridge University Press, Cambridge, MA, 2006.

[57] Zipfel, P. H., *Modeling and Simulation of Aerospace Vehicle Dynamics*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Reston, VA, 2000.

[58] Sutton, G. P. and Biblarz, O., *Rocket Propulsion Elements*, John Wiley & Sons Inc., New York, NY, 7th ed., 2001, pp. 27–31, Chap. 2.

[59] Gates, C. R., "A Simplified Model of Midcourse Maneuver Execution Errors," Technical Report 32-504, Jet Propulsion Lab, Pasadena, CA, Oct 1963.

[60] Lawrence, A., *Modern Inertial Technology: Navigation, Guidance, and Control*, Springer-Verlag New York, Inc., New York, NY, 2nd ed., 1998, pp. 17–19, Chap. 1.

[61] Britting, K. R., *Inertial Navigation Systems Analysis*, Wiley-Interscience — a Division of John Wiley & Sons, Inc., New York, NY, 1971.

[62] Webster, J. G., *The Measurement, Instrumentation, and Sensors Handbook*, CRC Press, Boca Raton, FL, 1999, pp. 17–26, Chap. 17.

[63] Lawrence, A., *Modern Inertial Technology: Navigation, Guidance, and Control*, Springer-Verlag New York, Inc., New York, NY, 2nd ed., 1998, pp. 25–42, Chap. 2.

[64] Carpenter, J. R., "Delivery of Inertial Measurement Unit Model to Mars Surveyor 2001 Atmospheric Flight Team," Technical memorandum, National Aeronautics and Space Administration, Houston, TX, Sep 26, 1997.

[65] Jazwinski, A. H., *Stochastic Processes and Filtering Theory*, Dover Publications, Inc., Mineola, NY, 2007.

[66] Crassidis, J. L. and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Applied Mathematics and Nonlinear Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2004, p. 151, Chap. 3.

[67] Crain, T. P. and Bishop, R. H., "Mars Entry Navigation: Atmospheric Interface Through Parachute Deploy," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, No. AIAA 2002-4501, American Institute of Aeronautics and Astronautics, Monterey, CA, Aug 5-8, 2002.

[68] Woffinden, D. C., *Angles-Only Navigation for Autonomous Orbital Rendezvous*, Ph.D. thesis, Utah State University, Logan, UT, 2008.

[69] Lu, P., Zhang, L., and Sun, H., "Ascent Guidance for Responsive Launch: A Fixed-Point Approach," *AIAA Guidance, Navigation and Control Conference and Exhibit*, No. AIAA 2005-6453, American Institute of Aeronautics and Astronautics, San Francisco, CA, Aug 2005.

[70] McHenry, R. L., Brand, T. J., Long, A. D., Cockrell, B. F., and Thibodeau III, J. R., "Space Shuttle Ascent Guidance, Navigation, and Control," *The Journal of the Astronautical Sciences*, Vol. 27, No. 1, Jan–Mar 1979, pp. 1–38.

[71] Fill, T. J., "Lunar Landing and Ascent Trajectory Guidance Design for the Autonomous Landing and Hazard Avoidance Technology (ALHAT) Program," *20th AAS/AIAA Space Flight Mechanics Meeting*, No. AAS 10-257, San Diego, CA, Feb 14–17 2010.

[72] Jaggers, R. F., "An Explicit Solution to the Exoatmospheric Powered Flight Guidance and Trajectory Optimization Problem for Rocket Propelled Vehicles," *Proceedings of the AIAA 1977 Guidance and Control Conference*, Vol. AIAA 77-1051, American Institute of Aeronautics and Astronautics, Reston, VA, 1977, pp. 566–578.

[73] Cherry, G. W., "A General, Explicit, Optimizing Guidance Law for Rocket-Propelled Spaceflight," *AIAA/ION Astrodynamics Guidance and Control Conference*, No. AIAA 64-638, American Institute of Aeronautics and Astronautics, Los Angeles, CA, Aug 24–26, 1964.

[74] Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths Mathematical Texts, Butterworths, London, United Kingdom, 1963.

[75] Sidi, M. J., *Spacecraft Dynamics & Control: A Practical Engineering Approach*, Cambridge University Press, New York, NY, 2006.

[76] Markley, F. L., *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, Boston, MA, 1988, pp. 410–428, Chap. 12.

[77] Riegsecker, D., "Quaternion Algebra for Two Quaternion Definitions," Technical memorandum, Charles Stark Draper Laboratory, Cambridge, MA, Jul 17, 2003.

[78] Wie, B., *Space Vehicle Dynamics and Control*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Reston, VA, 1998.

[79] Woffinden, D. C., *On-Orbit Satellite Inspection: Navigation and Δv Analysis*, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[80] Chapra, S. C. and Canale, R. P., *Numerical Methods for Engineers*, McGraw-Hill, New York, NY, 4th ed., 2005.

[81] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, Sep 2003, pp. 245–262.

[82] Squire, W. and Trapp, G., "Using Complex Variables to Estimate Derivatives of Real Functions," *SIAM Review*, Vol. 40, No. 1, Mar 1998, pp. 110–112.

[83] Lear, W. M., "Kalman Filtering Techniques," Technical Report JSC-20688, National Aeronautics and Space Administration, Houston, TX, Sep 1985.

[84] Taylor, J. R., *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, Sausalito, CA, 2nd ed., 1997.

[85] Christensen, D. and Geller, D. K., "Terrain-Relative and Beacon-Relative Navigation for Lunar Powered Descent and Landing," *32nd Annual AAS Rocky Mountain Guidance and Control Conference*, No. AAS 2009-, American Astronautical Society, Breckenridge, CO, Jan 30 - Feb 4, 2009.

[86] Lee, A. Y., Strahan, A., Tanimoto, R., and Casillas, A., "Preliminary Characterization of the Altair Lunar Lander Slosh Dynamics and Some Implications for the Thrust Vector Control Design," *AIAA Guidance, Navigation and Control Conference*, No. AIAA 2010-7721, American Institute of Aeronautics and Astronautics, Toronto, Ontario Canada, Aug 2–5, 2010.

[87] Abramson, H. N., *The Dynamic Behavior of Liquids in Moving Containers with Applications to Space Vehicle Technology*, No. NASA SP-106, National Aeronautics and Space Administration, Washington, DC, 1966.

[88] Krishnaswamy, K. and Bugajski, D., "Inversion Based Multibody Control - Launch Vehicle with Fuel Slosh," *AIAA Guidance, Navigation and Control Conference*, No. AIAA 2005-6149, American Institute of Aeronautics and Astronautics, San Francisco, CA, Aug 15–18, 2005.

Appendices

Appendix A

Reference Frames and Coordinate Systems

"Tous les problemes de geometrie se peuvent facilement reduire a

tels termes, qu'il n'est besoin par apres que de connaitre la

longeur de quelques lignes droites, pour les construire."

– René Descartes

Ascent guidance, inertial navigation, and attitude determination and control, all of which are subjects of importance in this work, require the precise definition of a number of reference frames and their associated coordinate systems and coordinate transformations. As such, the purpose of this chapter is to present and define the various frames of reference, coordinate systems, and coordinate tranformations used in this study.

A.1 Reference Frames

Using the notions that a *reference frame* is "a perspective from which observations are made regarding the motion of a system" [56, pp. 28] and that reference frames and coordinate systems are two different and distinct entities [56, 57], three frames of reference of primary importance to modeling the motion of aerospace vehicles in three-dimensional Euclidean space are considered, viz. planet-centered inertial (J2000), planet fixed, and body fixed.

A.1.1 Planet-Centered Inertial (J2000) Frame

The *planet-centered inertial (J2000) reference frame*, denoted with i and hereafter referred to as simply the *inertial frame*, is located at the center of the planet. Its orientation is fixed with respect to the stars, as defined by the J2000 epoch and the basis vectors $\mathbf{i}_1$, $\mathbf{i}_2$, and $\mathbf{i}_3$. The first basis vector $\mathbf{i}_1$ lies in the equatorial plane and points in the direction of the *vernal equinox*,[1] denoted by

---

[1]The vernal equinox refers to both a point of reference and a time or epoch. The point is the intersection of the ecliptic (path of the Sun on the celestial sphere) and the celestial equator. In other words, the point where the Sun crosses the celestial equator in the spring. The most common epoch used today is the J2000 system. The term 'vernal' is derived from the Latin word *ver*, meaning spring, and 'equinox' is derived from the Latin words *aequus*, meaning equal, and *nox*, meaning night.
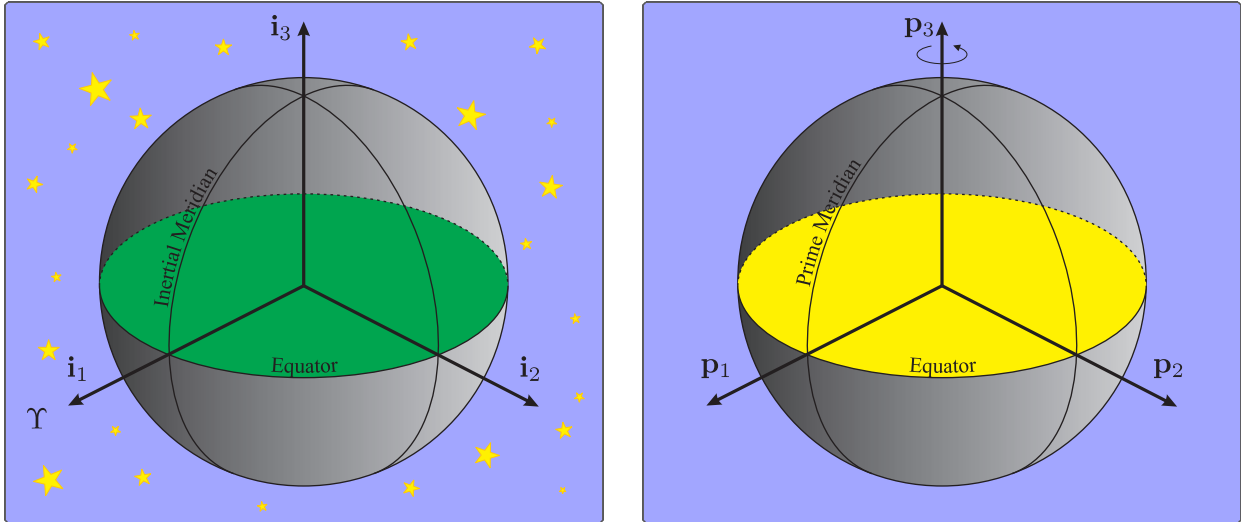
Figure A.1: Primary frames of reference: (*left*) Inertial reference frame. (*right*) Planet frame.

Υ. The third basis vector $\mathbf{i}_3$ points along the planet's axis of rotation (or angular velocity vector), and $\mathbf{i}_2$ (also in the equatorial plane) completes the right-handed orthogonal set, as illustrated in Figure A.1.

A.1.2 Planet-Fixed Frame

The *planet-fixed frame*, denoted with p and hereafter referred to as simply the *planet frame*, is collocated with the inertial frame, i.e., its origin is also at the center of the planet. Its orientation, however, is fixed with the planet as it rotates and defined by the basis vectors $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$. The first basis vector $\mathbf{p}_1$ lies in the equatorial plane and points in the direction of the intersection of the prime meridian and the equator. The third basis vector $\mathbf{p}_3$ points along the planet's axis of rotation (or angular velocity vector), and $\mathbf{p}_2$ (also in the equatorial plane) completes the right-handed orthogonal set, as depicted in Figure A.1.

A.1.3 Body Fixed Frame

The *body-fixed frame*, denoted with b and hereafter referred to as simply the *body frame*, is located with its origin at the center of mass of the vehicle. Its orientation is aligned with the principle axes of the moment of inertia tensor and defined by the basis vectors $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$. The first basis vector $\mathbf{b}_1$ points out the nose of the vehicle, $\mathbf{b}_2$ points out the right side of the vehicle in the plane of symmetry, and $\mathbf{b}_3$ (also in the plane of symmetry) completes the right-handed orthogonal set, as illustrated in Figure A.2.
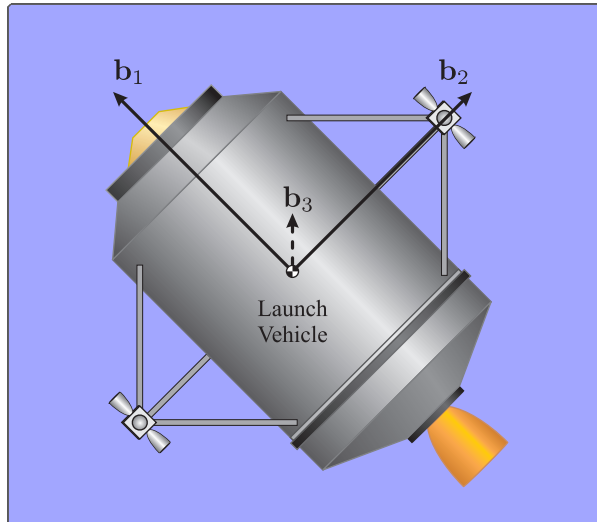
Figure A.2: Primary frame of reference: Body frame.

A.2   Coordinate Systems

Using the notion that a *coordinate system* "provides a way to measure the motion of the system within a particular reference frame" [56, pp. 27], four coordinate systems are considered, viz. inertial, planet, body, and planetographic coordinate systems.

A.2.1   Inertial Coordinate System

The *inertial coordinate system*, denoted with a superscript i, is the preferred coordinate system of the inertial frame, i.e., the coordinate axes 1^i, 2^i, and 3^i are chosen to be aligned with the orthogonal triad $\mathbf{i}_1$, $\mathbf{i}_2$, and $\mathbf{i}_3$, respectively.

A.2.2   Planet Coordinate System

The *planet coordinate system*, denoted with a superscript p, is the preferred coordinate system of the planet frame, i.e., the coordinate axes 1^p, 2^p, and 3^p are chosen to be aligned with the orthogonal triad $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, respectively. The inertial and planet coordinate systems are related by a single rotation about the 3^i, 3^p axes through the angle Ξ, as illustrated in Figure A.3. Thus the inertial-to-planet transformation matrix $\boldsymbol{T}_i^p$ is given by [57, pp. 71]

$$\boldsymbol{T}_i^p = \begin{bmatrix} \cos\Xi & \sin\Xi & 0 \\ -\sin\Xi & \cos\Xi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.1}$$

A.2.3   Body Coordinate System

The *body coordinate system*, denoted with a superscript b, is the preferred coordinate system of the body frame, i.e., the coordinate axes 1^b, 2^b, and 3^b are chosen to be aligned with the orthogonal triad $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, respectively. The relationship between the inertial and body coordinate systems (or attitude of the vehicle) is generally represented with the attitude quaternion, which can be thought of as a single rotation about a special axis called an eigenaxis. However, the quaternion parameterization of the inertial-to-body transformation is not discussed or given here, but can be found in §{quaternions}. Alternatively, the inertial and body coordinate frames can be related by three successive rotations through the so-called Euler angles: yaw ψ, pitch ϑ, and roll φ. It can be shown that the Euler angle parameterization ($\psi \rightarrow \vartheta \rightarrow \varphi$ sequence) of the inertial-to-body transformation matrix $\boldsymbol{T}_i^b$ is given by [57, pp. 74-75]

$$\boldsymbol{T}_i^b = \begin{bmatrix} \cos\psi\cos\vartheta & \sin\psi\cos\vartheta & -\sin\vartheta \\ \cos\psi\sin\vartheta\sin\varphi - \sin\psi\cos\varphi & \sin\psi\sin\vartheta\sin\varphi + \cos\psi\cos\varphi & \cos\vartheta\sin\varphi \\ \cos\psi\sin\vartheta\cos\varphi + \sin\psi\sin\varphi & \sin\psi\sin\vartheta\cos\varphi - \cos\psi\sin\varphi & \cos\vartheta\cos\varphi \end{bmatrix} . \quad (A.2)$$

A.2.4   Planetographic Coordinate System

In order to navigate on the surface of the planet, a grid consisting of lines of longitude and latitude is usually specified. *Longitude*, denoted by λ, is divided into $\pm180°$ and measured from the prime meridian with the positive direction to the east and negative to the west. *Latitude*, denoted by ϕ, is divided in $\pm90°$ and measured from the equator with the positive direction to the north and negative to the south.

With these definitions in place, the *planetographic coordinate system*, denoted with a superscript g, is defined such that the 1^g axis points north, the 3^g axis points to the center of the planet, and the 2^g axis points east thereby completing the right-handed coordinate system. Note that if the planet is approximated as a sphere (not a spheroid), the latitude previously defined becomes the *planetocentric latitude* and the planetographic coordinate system is relabeled the *planetocentric coordinate system*.[2] At a specific point on the surface of the planet, with longitude λ and latitude ϕ, the planet and planetographic coordinate systems, as depicted in Figure A.3, are related by three

---

[2]The planetocentric coordinate system is also referred to in the literature as the *topocentric* coordinate system, *topocentric equatorial* coordinate system, and a whole host of other variations.
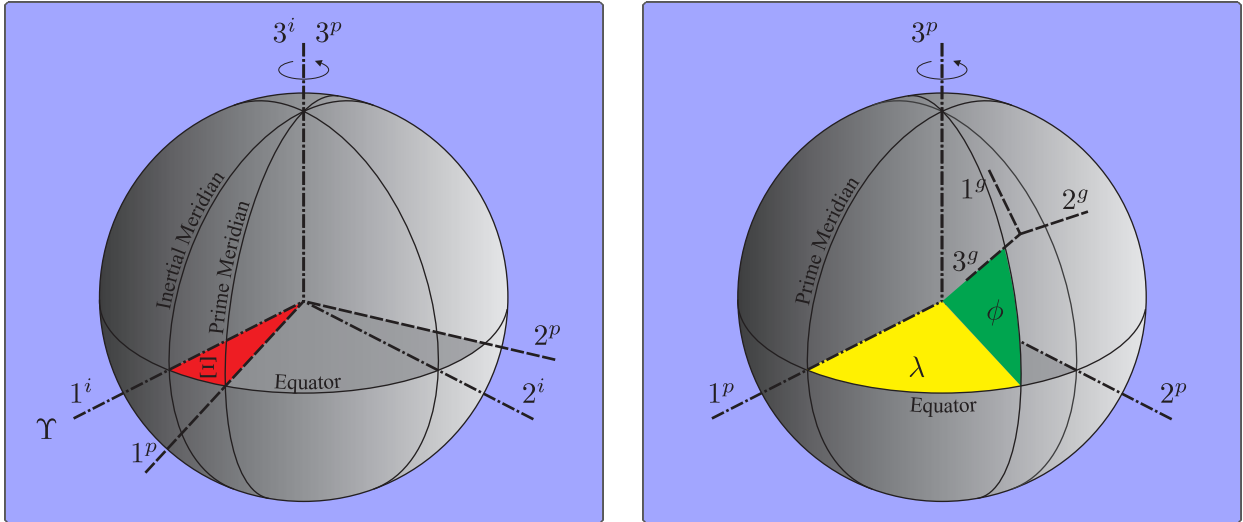
Figure A.3: Coordinate systems relationships: (*left*) Inertial and planet coordinate systems. (*right*) Planet and planetographic coordinate systems.

successive rotations (not shown). Thus, the planet-to-planetographic transformation matrix $\boldsymbol{T}_p^g$ is given by [57, pp. 72-73]

$$
\boldsymbol{T}_p^g = \begin{bmatrix}
-\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\
-\sin\lambda & \cos\lambda & 0 \\
-\cos\phi\cos\lambda & -\cos\phi\sin\lambda & -\sin\phi
\end{bmatrix} .
\tag{A.3}
$$

Appendix B

Complex-Step Derivative

B.1 Customized Functions

The intrinsic MATLAB functions `norm` and `atan2`, both of which are employed by the guidance algorithm, are not defined to handle complex variables. Since the functions are intrinsic to MATLAB and cannot be edited, it became necessary to develop and create customized functions that are able to handle both real and complex variables. The details of how the customized functions were coded are given below.

B.1.1 norm

The MATLAB `norm` function computes several vector and matrix norms, e.g., 1-norm, 2-norm, infinity norm, Frobenius norm, and Euclidean norm. For this research, only the Euclidean norm or Euclidean 'length' of a vector $\boldsymbol{x} \in \mathbb{C}^a$ is considered, which can be written as

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\mathsf{T}\boldsymbol{x}} \ . \tag{B.1}$$

In MATLAB, this can be written as follows:

```matlab
1  function xmag = cnorm(x)
2  % CNORM(X) is the Euclidean norm (i.e., length or magnitude) of a vector X,
3  % where X is comprised of real or complex elements.
4  %
5  % See also CATAN2
6  %
7  % ASSUMPTIONS:
8  % 1. The MATLAB function sqrt() can properly handle complex values.
9
10 % Compute Euclidean norm
11 xmag = sqrt(x.'*x);
```

where the `sqrt` function is able to handle complex values and the array transpose operator `.'` denotes the nonconjugate transpose or the transpose that does not take the complex conjugate of the variable on which it is operating.

B.1.2 atan2

The MATLAB `atan2` function computes the four-quadrant inverse tangent over the interval $[-\pi, \pi]$, but is not able to handle complex variables. The MATLAB `atan` function, on the other hand, is limited to the interval $[-\pi/2, \pi/2]$ but able to handle complex variables. Hence, a four-quadrant inverse tangent function `catan2`, that is able to handle complex variables, can be written in terms of the `atan` function as follows:

```matlab
1  function theta = catan2(y,x)
2  % CATAN2(Y,X) computes the four quadrant inverse tangent for real and
3  % complex values of Y and X.
4  %
5  % See also CNORM
6  %
7  % ASSUMPTIONS:
8  % 1. The MATLAB function atan() can properly handle complex values.
9  if real(x) > 0
10     theta = atan(y/x);
11 elseif real(x) < 0
12     if real(y) >= 0
13         theta = pi + atan(y/x);
14     else
15         theta = -pi + atan(y/x);
16     end
17 else
18     if real(y) > 0
19         theta = pi/2;
20     elseif real(y) < 0
21         theta = -pi/2;
22     else
23         theta = 0;
24     end
```

```
25   end
```

B.2 Quaternions

Perhaps the best way to illustrate how to properly apply the complex-step derivative (CSD) method to vector functions or algorithms with quaternions is through a series of examples. Let the nonlinear vector function that we desire to linearize be given by

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \ , \tag{B.2}$$

where $\boldsymbol{y} \in \mathbb{R}^{a'}$ and $\boldsymbol{f} : \mathbb{R}^{b'} \to \mathbb{R}^{a'}$ is a nonlinear function of $\boldsymbol{x} \in \mathbb{R}^{b'}$. Moreover, the input and output variables can be respectively written in terms of their scalar elements $\boldsymbol{y} = (y_1, y_2, \ldots, y_{a'})$ and $\boldsymbol{x} = (x_1, x_2, \ldots, x_{b'})$. Following the linearization process outlined in § 4.2, the resulting linear vector function is given by

$$\delta \boldsymbol{y} = \boldsymbol{F}_x \, \delta \boldsymbol{x} \ , \tag{B.3}$$

where the Jacobian $\boldsymbol{F}_x \in \mathbb{R}^{a' \times b'}$ can be written as

$$\boldsymbol{F}_x = \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|_{\bar{\boldsymbol{x}}} = \left[\left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1} \right|_{\bar{\boldsymbol{x}}} \quad \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_2} \right|_{\bar{\boldsymbol{x}}} \quad \cdots \quad \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_{b'}} \right|_{\bar{\boldsymbol{x}}} \right] \ . \tag{B.4}$$

To evaluate the first partial derivative $\left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1} \right|_{\bar{\boldsymbol{x}}}$ in the above Jacobian using the CSD method, the vector function $\boldsymbol{f}$ is evaluated with the complex-valued input variable given by

$$\boldsymbol{x} = \begin{bmatrix} \bar{x}_1 + ih \\ \bar{x}_2 + i0 \\ \vdots \\ \bar{x}_{b'} + i0 \end{bmatrix} \ , \tag{B.5}$$

where the perturbation step h has been added to the imaginary part of the first scalar element and the horizontal bar $^-$ denotes the nominal value of the states x_k ($k = 1, 2, \ldots, b'$). The resulting

complex-valued output variable is therefore given by

$$
\boldsymbol{y} = \begin{bmatrix} \bar{y}_1 + i\Delta y_1 \\ \bar{y}_2 + i\Delta y_2 \\ \vdots \\ \bar{y}_{a'} + i\Delta y_{a'} \end{bmatrix},
\tag{B.6}
$$

where the derivative information, Δy_l ($l = 1, 2, \ldots, a'$), is captured in the imaginary part. Hence, it follows that the first partial derivative in Eq. (B.4) is given by

$$
\left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1} \right|_{\bar{\boldsymbol{x}}} = \frac{1}{h} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_{a'} \end{bmatrix}.
\tag{B.7}
$$

This process is then repeated for the remaining partial derivatives.

B.2.1 Input Quaternion Case

Consider now the case where the input variable contains a quaternion. Therefore, let the first four elements of input variable $\boldsymbol{x}$ be the attitude quaternion $\boldsymbol{q} = (q_1, q_2, q_3, q_4)$, such that $\boldsymbol{x} = (q_1, q_2, q_3, q_4, \ldots, x_{b'})$. Recall that due to state covariance matrix singularity issues associated with the quaternion, the attitude quaternion $\boldsymbol{q} \in \mathbb{R}^4$ is replaced with the Euler rotation vector $\boldsymbol{\theta} \in \mathbb{R}^3$, resulting in a modified input variable $\boldsymbol{x}_{\mathrm{m}} \in \mathbb{R}^b$ (where $b = b' - 1$), such that $\boldsymbol{x}_{\mathrm{m}} = (\theta_1, \theta_2, \theta_3, \ldots, x_b)$. As a result, Eq. (B.3) is now derived using the modified input variable $\boldsymbol{x}_{\mathrm{m}}$ and the dimensions of the Jacobian in Eq. (B.4) have changed, such that $\boldsymbol{F}_x \in \mathbb{R}^{a' \times b}$ is given by

$$
\boldsymbol{F}_x = \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{m}}} \right|_{\bar{\boldsymbol{x}}} = \begin{bmatrix} \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1} \right|_{\bar{\boldsymbol{x}}} & \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_2} \right|_{\bar{\boldsymbol{x}}} & \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_3} \right|_{\bar{\boldsymbol{x}}} & \cdots & \left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_b} \right|_{\bar{\boldsymbol{x}}} \end{bmatrix}.
\tag{B.8}
$$

In order to evaluate the first partial derivative $\left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1} \right|_{\bar{\boldsymbol{x}}}$ in the above Jacobian using the CSD method, the perturbation step h must be added to the input attitude quaternion. However, the perturbation step cannot be added to a quaternion in the same manner as a vector. This is due to the fact that the quaternion must always be unitary, i.e., have a norm of 1. Hence, the perturbation step h is

added in the following manner: First, a complex-valued perturbation quaternion $\delta\boldsymbol{q} \in \mathbb{C}^4$ is formed as follows

$$\delta\boldsymbol{q} = \begin{bmatrix} -\delta\boldsymbol{\theta}/2 \\ 1 + i0 \end{bmatrix} , \tag{B.9}$$

where the pertubation step h has been added to imaginary part of the first scalar element of $\delta\boldsymbol{\theta} \in \mathbb{C}^3$,

$$\delta\boldsymbol{\theta} = \begin{bmatrix} 0 + ih \\ 0 + i0 \\ 0 + i0 \end{bmatrix} . \tag{B.10}$$

Next, the perturbed input attitude quaternion $\boldsymbol{q} \in \mathbb{C}^4$ is formed as follows

$$\boldsymbol{q} = \delta\boldsymbol{q} \otimes \left[\bar{\boldsymbol{q}}\right]^{-1} , \tag{B.11}$$

where $\otimes$ denotes quaternion multiplication and $\bar{\boldsymbol{q}} \in \mathbb{C}^4$ is simply the nominal input attitude quaternion given by

$$\bar{\boldsymbol{q}} = \begin{bmatrix} \bar{q}_1 + i0 \\ \bar{q}_2 + i0 \\ \bar{q}_3 + i0 \\ \bar{q}_4 + i0 \end{bmatrix} . \tag{B.12}$$

Note that the real part of the perturbation quaternion $\delta\boldsymbol{q}$ is the identity quaternion. This is important because the real part of the perturbed input attitude quaternion $\boldsymbol{q}$ in Eq. (B.11) needs to be equal to the real part of the nominal attitude quaternion $\bar{\boldsymbol{q}}$ defined in Eq. (B.12). The only way for this to happen is to have the real part of the perturbation quaternion $\delta\boldsymbol{q}$ equal to the identity quaternion. Now the vector function $\boldsymbol{f}$ is evaluated with the complex-valued input variable given by

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \bar{x}_5 + i0 \\ \vdots \\ \bar{x}_{b'} + i0 \end{bmatrix} , \tag{B.13}$$

where $\boldsymbol{q}$ is the perturbed input attitude quaternion from Eq. (B.11). The resulting complex-valued output variable $\boldsymbol{y}$ is therefore given by Eq. (B.6), where the derivative information is captured in

the imaginary part. It follows then that the first partial derivative in Eq. (B.8) is given by

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1}\bigg|_{\bar{\boldsymbol{x}}} = \frac{1}{h}\begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_{a'} \end{bmatrix}. \tag{B.14}$$

Since only the first three partial derivatives correspond to the input attitude quaternion, this process is then repeated for the next two partial derivatives, but with the perturbation step h being added respectively to the imaginary parts of the second and third scalar elements of $\delta\boldsymbol{\theta}$. The remaining partial derivatives are then computed using the process outlined previously for the vector input variable.

B.2.2 Output Quaternion Case

Lastly, consider the case where the output variable also contains a quaternion. Therefore, let the first four elements of the output variable $\boldsymbol{y}$ be the attitude quaternion $\boldsymbol{\varrho} = (\varrho_1, \varrho_2, \varrho_3, \varrho_4)$, such that $\boldsymbol{y} = (\varrho_1, \varrho_2, \varrho_3, \varrho_4, \ldots, y_{a'})$. Recall that due to state covariance matrix singularity issues associated with the quaternion, the output attitude quaternion $\boldsymbol{\varrho} \in \mathbb{R}^4$ is replaced with the Euler rotation vector $\boldsymbol{\vartheta} \in \mathbb{R}^3$, resulting in a modified output variable $\boldsymbol{y}_{\mathrm{m}} \in \mathbb{R}^a$ (where $a = a' - 1$), such that $\boldsymbol{y}_{\mathrm{m}} = (\vartheta_1, \vartheta_2, \vartheta_3, \ldots, y_a)$. As a result, the linearized vector function in Eq. (B.3) is now derived using the modified input and output variables $\boldsymbol{y}_{\mathrm{m}}$ and $\boldsymbol{x}_{\mathrm{m}}$, and the dimensions of the Jacobian $\boldsymbol{F}_x$ have changed, such that $\boldsymbol{F}_x \in \mathbb{R}^{a \times b}$ as given by

$$\boldsymbol{F}_x = \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{m}}}\bigg|_{\bar{\boldsymbol{x}}} = \begin{bmatrix} \dfrac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1}\bigg|_{\bar{\boldsymbol{x}}} & \dfrac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_2}\bigg|_{\bar{\boldsymbol{x}}} & \dfrac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_3}\bigg|_{\bar{\boldsymbol{x}}} & \cdots & \dfrac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_b}\bigg|_{\bar{\boldsymbol{x}}} \end{bmatrix}. \tag{B.15}$$

To evaluate the first partial derivative $\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1}\big|_{\bar{\boldsymbol{x}}}$ in the above Jacobian using the CSD method, the perturbation step h is added to the complex-valued input variable $\boldsymbol{x}$ using the process outlined above for the input attitude quaternion. The resulting complex-valued output variable is therefore

given by

$$\boldsymbol{y} = \begin{bmatrix} \bar{\varrho}_1 + i\Delta\varrho_1 \\ \bar{\varrho}_2 + i\Delta\varrho_2 \\ \bar{\varrho}_3 + i\Delta\varrho_3 \\ \bar{\varrho}_4 + i\Delta\varrho_4 \\ \vdots \\ \bar{y}_{a'} + i\Delta y_{a'} \end{bmatrix}, \tag{B.16}$$

where the derivative information is captured in the imaginary part. From this expression it is observed that the perturbed output attitude quaternion $\boldsymbol{\varrho} \in \mathbb{C}^4$ is given by

$$\boldsymbol{\varrho} = \begin{bmatrix} \bar{\varrho}_1 + i\Delta\varrho_1 \\ \bar{\varrho}_2 + i\Delta\varrho_2 \\ \bar{\varrho}_3 + i\Delta\varrho_3 \\ \bar{\varrho}_4 + i\Delta\varrho_4 \end{bmatrix}. \tag{B.17}$$

It is important to point out, however, that the desired partial derivative $\left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1} \right|_{\bar{\boldsymbol{x}}}$ actually contains the partial derivative of the output rotation vector $\boldsymbol{\vartheta}$, not the output attitude quaternion $\boldsymbol{\varrho}$. In other words, the desired partial derivative can be written as

$$\left. \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \theta_1} \right|_{\bar{\boldsymbol{x}}} = \begin{bmatrix} \frac{\partial \boldsymbol{\vartheta}}{\partial \theta_1} \\ \vdots \\ \frac{\partial y_a}{\partial \theta_1} \end{bmatrix}. \tag{B.18}$$

Hence, the derivative information in the imaginary part of the output attitude quaternion $\boldsymbol{\varrho}$ needs to be manipulated so as to yield a perturbed output rotation vector $\delta\boldsymbol{\vartheta}$. This is done by first forming the nominal output attitude quaternion $\bar{\boldsymbol{\varrho}} \in \mathbb{C}^4$ as follows

$$\bar{\boldsymbol{\varrho}} = \begin{bmatrix} \bar{\varrho}_1 + i0 \\ \bar{\varrho}_2 + i0 \\ \bar{\varrho}_3 + i0 \\ \bar{\varrho}_4 + i0 \end{bmatrix}. \tag{B.19}$$

Next, the output perturbation quaternion $\delta\boldsymbol{\varrho} \in \mathbb{C}^4$ is computed as follows

$$\delta\boldsymbol{\varrho} = \boldsymbol{\varrho} \otimes \left[\bar{\boldsymbol{\varrho}}\right]^{-1} = \begin{bmatrix} -\delta\boldsymbol{\vartheta}/2 \\ 1 + i0 \end{bmatrix}, \qquad \text{(B.20)}$$

where the resulting perturbed output rotation vector $\delta\boldsymbol{\vartheta} \in \mathbb{C}^3$ is given by

$$\delta\boldsymbol{\vartheta} = \begin{bmatrix} 0 + i\Delta\vartheta_1 \\ 0 + i\Delta\vartheta_2 \\ 0 + i\Delta\vartheta_3 \end{bmatrix}. \qquad \text{(B.21)}$$

Note that the real part of the output perturbation quaternion $\delta\boldsymbol{\varrho}$ needs to be the identity quaternion so that the derivative information is contained in only the first three elements of the imaginary part. It follows then that the first partial derivative in Eq. (B.15) is given by

$$\left.\frac{\partial\boldsymbol{f}(\boldsymbol{x})}{\partial\theta_1}\right|_{\bar{\boldsymbol{x}}} = \frac{1}{h} \begin{bmatrix} \Delta\vartheta_1 \\ \Delta\vartheta_2 \\ \Delta\vartheta_3 \\ \vdots \\ \Delta y_{a'} \end{bmatrix}, \qquad \text{(B.22)}$$

where the first four elements in the imaginary part of the perturbed output variable $\boldsymbol{y}$, given in Eq. (B.16), have been replaced with the imaginary part of the perturbed output rotation vector $\delta\boldsymbol{\vartheta}$ given in Eq. (B.21). This process is then repeated for the remaining partial derivatives in the Jacobian $\boldsymbol{F}_x$, defined in Eq. (B.15).

Curriculum Vitae

Michael Benjamin Rose

Work Experience

Senior GN&C Engineer 04/26/2010 – 07/31/2011
Emergent Space Technologies, Denver, CO

- Developed, tested, and validated simulation model of barometric altimeter for NASA/Lockheed Martin Multipurpose Crew Vehicle (MCV).
- Team Lead for AFRL Phase 1 SBIR titled "Data Fusion Strategies for Distributed Tracking of Space Objects" and AFRL Phase 1 STTR titled "Precision Tracking of Resident Space Objects with Covariance Realism."
- Developed an apparent magnitude measurement model, which utilizes a bidirectional reflectance distribution function (BRDF) to compute the amount of reflected light that a ground-based electro-optical sensor would detect from a sun-illuminated object in orbit about the Earth.

Linear Covariance Analysis Intern 06/01/2009 – 08/21/2009
Draper Laboratory, Houston, TX

- Developed, incorporated, and tested new attitude control capabilities in linear covariance tool for orbital rendezvous.
- Programmed novel tool for GNC analysis of powered ascent using linear covariance techniques.
- Documented and presented results to Draper/NASA Crew Exploration Vehicle (CEV) group.

Education

Utah State University, Logan, UT

- **Ph.D.** in Mechanical & Aerospace Engineering, May 2012
  - Dissertation: "Statistical Methods for Launch Vehicle Guidance, Navigation, and Control (GN&C) System Design and Analysis"
  - Emphasis: Guidance, Navigation, Control, Linear Covariance Analysis
  - GPA: 3.96 *Summa Cum Laude*, while working 20-30 hours per week.

Brigham Young University, Provo, UT

- **M.S.** in Physics, August 2006
  - Thesis: "A Search For Low-Amplitude Variability Among Population I Main Sequence Stars"
  - Emphasis: Astrophysics
  - GPA: 3.51, while working 20-30 hours per week.

- **B.S.** in Physics and Astronomy, April 2004

  – Minor in Mathematics

  – Thesis: "Improving Aperture Photometry Reduction Techniques"

  – Emphasis: Astrophysics, Computer Science, Mathematics

  – GPA: 3.34, while working 20-30 hours per week.

Snow College, Ephraim, UT

- **A.A.** in Physics, May 2000

  – Graduated *Cum Laude*

  – Emphasis: Physics, Chemistry, Mathematics

Computer Skills

Matlab	C	Mac	Word
Simulink	C++	Linux	Excel
Stateflow	Java	STK	PowerPoint
Maple	HTML	IRAF	LaTeX

Awards and Achievements

- John A. Gast Scholarship, Brigham Young University, 2003 – 2004
- Carl Joseph Christensen Scholarship, Brigham Young University, 2002 – 2003
- Academic Achievement Awards in Physics, Mathematics, and Chemistry, Snow College, 1999 – 2000
- Honorable Discharge, United States Marine Corps, 1998
- Eagle Scout, Boy Scouts of America, 1991

Publications and Conference Proceedings

Professional

- Woffinden, D.C., **Rose, M.B.**, Geller, D.K., "Trigger Angle Targeting For Orbital Rendezvous," *The Journal of the Astronautical Sciences*, Vol. 56, No. 4, Oct–Dec 2008.

- Geller, D.K., **Rose, M.B.**, Woffinden, D.C., "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, Jan–Feb 2009, pp. 102–111.

- **Rose, M.B.**, Hintz, E.G., "A Search For Low-Amplitude Variability In Six Open Clusters Using The Robust Median Statistic," 2007, The Astronomical Journal, 134, 2067.

- Hintz, E.G., **Rose, M.B.**, Bush, T.C., Maxwell, A.A., " Establishing Observational Baselines for Two δ Scuti Variables: V966 Herculis and V1438 Aquilae," 2006, The Astronomical Journal, 132, 393.

- Hintz, E.G., Bush, T.C., **Rose, M.B.**, "Monitoring Three Less-studied δ Scuti Variables: GW Ursae Majoris, BO Lyncis, and AN Lyncis," 2005, The Astronomical Journal, 130, 2876.

- Hintz, E.G., **Rose, M.B.**, "Variable Stars in the Field of NGC 6882/6885: The Case of V381 Vulpeculae and V382 Vulpeculae," 2005, Publications of the Astronomical Society of the Pacific, 117.

Conference Proceedings

- **Rose, M.B.**, Geller, D.K., "Linear Covariance Techniques for Powered Ascent," *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, Paper No. AIAA-2010-8175, Aug. 2-5, 2010.

- Woffinden, D.C., **Rose, M.B.**, Geller, D.K., "Trigger Angle Targeting For Orbital Rendezvous," 2008, February 1-6, 31st Annual American Astronautical Society Guidance and Control Conference, Breckinridge, CO, AAS 08-63.

- Geller, D.K., **Rose, M.B.**, Woffinden, D.C., "Event Triggers in Linear Covariance Analysis With Applications to Orbital Rendezvous Analysis," 2008, February 1-6, 31st Annual American Astronautical Society Guidance and Control Conference, Breckinridge, CO, AAS 08-69.

- McNamara, D.H., **Rose, M.B.**, Brown, P.J., Ketcheson, D.I., Maxwell, J.E., Smith, K.M., Woolley, R.C., "The Luminosities of Horizontal Branches and RR Lyrae Stars in Globular Clusters," 2004, Astronomical Society of the Pacific Conference Series, 310, 525.

Abstracts/Posters

- Hintz, E.G., **Rose, M.B.**, "The Search for Low Amplitude Pulsational Variable Stars in Six Open Clusters," 2006, Bulletin of the American Astronomical Society, 209, 16512.

- **Rose, M.B.**, and Hintz, E.G., "An Examination of Main Sequence Stars within the Instability Strip for Open Clusters Over a Range of Ages," 2004, Bulletin of the American Astronomical Society, 205, 2203.

- Hintz, E.G., **Rose, M.B.**, "Narrow Band Photometry of the Calcium H and K lines for Stars in Open Clusters," 2004, Bulletin of the American Astronomical Society, 205, 2210.

- **Rose, M.B.**, and Hintz, E.G., "Improving Variable Star Light Curves from a Campus Observatory," 2003, Bulletin of the American Astronomical Society, 203, 806.

- Hintz, E.G., Joner, M.D., **Rose, M.B.**, "An Emission-Line delta Scuti? The case of V382 Vul and Other Variables in the Field of NGC 6882/5," 2003, Bulletin of the American Astronomical Society, 203, 804.

- **Rose, M.B.**, and Hintz, E.G., "An Examination of the delta Scuti Variable V1438 Aquilae," 2002, Bulletin of the American Astronomical Society, 201, 11703.

Research Experience

Utah State University, Logan, UT

Graduate Research Assistant 2008–2010, to Dr. David Geller in developing "A Novel Hybrid Tool For Closed-Loop GNC Analysis of Launch Vehicle Ascent Trajectories Using Linear Covariance and Monte Carlo Techniques." Developed a novel trajectory control and navigation analysis tool for powered ascent, which is capable of rapid trade-space analysis and is designed to ultimately reduce turnaround time for launch vehicle design, mission planning, and redesign work. It is streamlined to quickly determine trajectory and attitude control dispersions, propellant dispersions, orbit insertion dispersions, navigation errors, and their sensitivities to sensor errors, actuator uncertainties, and random disturbances. The tool incorporates both Monte Carlo and linear covariance analysis techniques.

Graduate Research Assistant 2007, to Dr. David Geller in "Linear Covariance Analysis With Applications to Orbital Rendezvous." Built nominal trajectory design tools, co-developed theory behind event driven maneuvers in linear covariance analysis, expanded capabilities of closed-loop 6 degree-of-freedom linear covariance analysis tool, studied effects of elevation angle triggers on delta-v and positions dispersions. Presented results to sponsor NASA Johnson Space Center and at the 31st Annual American Astronautical Society Guidance and Control Conference, Breckinridge, CO.

Graduate Research Assistant 2007, to Dr. David Geller and Dr. Rees Fullmer in "USU's Simulink Library of Sensor and Actuator Models." Designed, developed, and tested high fidelity models of a single reaction wheel and a four-wheel system with and without momentum management. Project funded by Space Dynamics Laboratory.

Graduate Research Assistant 2006, to Dr. David Geller in "Monte Carlo Simulation of Autonomous Orbital Rendezvous." Used MATLAB's Simulink and Stateflow to expand the capabilities of a closed-loop 6 degree-of-freedom Monte Carlo simulation tool containing the various sensors, actuators, GN&C flight algorithms, and dynamics associated with autonomous orbital rendezvous. Coded angles-only navigation filter (extended Kalman filter) in C, compared output with Matlab M-code implementation, and evaluated CPU time requirements. Presented results to sponsor Lockheed Martin.

Brigham Young University, Provo, UT

Graduate Research Assistant 2004–2006, to Dr. Eric Hintz in "An Examination of Variable Star Populations in Nearby Open Clusters." Observed 6 open clusters in northern hemisphere over two years using 16-inch David Derrick Telescope and 1.8-meter at Dominion Astronomical Observatory (DAO) in Victoria, BC. Incorporated data for 4 additional clusters from undergraduate thesis projects. Co-designed, coded, and tested differential photometry package for use on clusters of stars. Presented preliminary findings at American Astronomical Society's 205th Meeting, and final results at American Astronomical Society's 209th Meeting. Published final results and findings in The Astronomical Journal.

Research Assistant 2003–2004, to Dr. Eric Hintz in "Optimization of Variable Star Photometric Measurements." Collaborated with professor on new methods of variable star data reduction, designed new script for tracking objects across multiple image frames, worked jointly on design of software package to be implemented into the Image Reduction and Analysis Facility

(IRAF), wrote initial version of software in IRAF shell language, applied new methods to several variable stars, acquired optimal light curves and measurable time of maximum light, and presented findings at American Astronomical Society's 203rd Meeting.

Research Assistant 2002–2004, to Dr. Eric Hintz and Mike Joner in "Spectroscopic Monitoring of Delta Scuti Stars." Wrote IRAF scripts to make the data reduction process much easier, traveled to Dominion Astronomical Observatory (DAO) in Victoria, BC to operate the 1.2-meter and 1.8-meter telescopes, reduced data for several nights of observing, used Kurucz models in coalition with Spectrum to obtain spectroscopic models to compare with our observed stars, and calculated and plotted radial velocity curves for V1438 Aquilae and other variable stars.

Research Assistant 2002–2004, to Dr. Eric Hintz in "An Examination of the Delta Scuti Variable V1438 Aquilae." Observed the star on 5 nights during the fall of 2002 and 4 nights in 2003 using the 16-inch David Derrick Telescope on campus, reduced the data in IRAF, acquired light curves, calculated times of maximum light, reduced spectroscopic data obtained at Dominion Astronomical Observatory, obtained radial velocity curve, and presented findings at the American Astronomical Society's 201st Meeting.

Research Assistant 2002–2003, to Dr. D. Harold McNamara in "Age-Dating Galactic Globular Clusters Using Current Isochrones and Models." Collected and plotted isochrone models from Yonsei-Yale and Vandenberg, gathered data for 47 globular clusters from literature, plotted data with models, analyzed and interpreted results, collaborated with professor and others to write a publishable article, published article in the Astronomical Society of the Pacific Conference Series.

Teaching Experience

Brigham Young University, Provo, UT

Lab Instructor 2004–2006, to Dr. Bryan Peterson in "Experiments in Electricity, Magnetism, Optics, and Radiation." Instruct proper use of and reveal troubleshooting methods for lab equipment, assist students in understanding physical concepts and experimental results, and give grades for preparation and participation.

Teaching Assistant 2003, to Dr. Eric Hintz in "Introduction to Applied Physics." Grade all homework assignments and exams, create solution keys to homework problems, and give feedback to professor on concepts not generally understood by students.

Research Instructor 2002–2004, to Dr. Eric Hintz in "Observational Astronomy and Advanced Observational Astrophysics." Instruct and oversee students on correct observing procedures, demonstrate proper use of data acquisition software and reduction methods using the image reduction and analysis facility, assist students with analyzing data results and preparing poster presentations, and give assistance and positive feedback on research papers.

Lab Instructor 2002–2003, to Dr. Bryan Peterson and Dr. Dallin Durfee in "Experiments in Mechanics and Thermodynamics." Instruct proper use of and reveal troubleshooting methods for lab equipment, assist students in understanding physical concepts and experimental results, and give grades for preparation and participation.

Teaching Assistant 2002–2003, to Dr. Lawrence Rees and Dr. David Allred in "Electricity and Magnetism: Theory and Applications." Assist in evaluation of homework and exams, oversee student appeals to homework scores, and provide feedback to professor on concepts not generally understood by students.

Lab Instructor 2001–2004, to Dr. J. Ward Moody, Dr. D. Harold McNamara, Dr. Eric Hintz, Dr. Benjamin Taylor, and Professor Mike Joner in "Descriptive Astronomy." Instruct three sections of the planetarium lab per week, administer and grade quizzes given in lab, provide help sessions, and assist students with their observing projects using the astronomy group 12-inch Schmidt telescope.

Snow College, Ephraim, UT

Student Teacher/Teaching Assistant 1998–2000, for Professor Jackie Young in "Elementary French I/II" and "Intermediate French I/II." Instructed first and second year courses, administered homework, exams, and special projects, and tutored students.