CDGPS-Based Relative Navigation for Multiple Spacecraft

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 0 1 2004
LIBRARIES
AERO

by

Megan Leigh Mitchell

Bachelor of Science Aerospace Engineering
University of Texas at Austin, 2000

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

© Massachusetts Institute of Technology 2004. All rights reserved.

Author .....
Department of Aeronautics and Astronautics
May 17, 2004

Certified by .....
Jonathan P. How
Associate Professor
Thesis Supervisor

Accepted by .....
Edward M. Greitzer
Chairman, Department Committee on Graduate Students

CDGPS-Based Relative Navigation for Multiple Spacecraft

by

Megan Leigh Mitchell

Submitted to the Department of Aeronautics and Astronautics
on May 17, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science Aeronautics and Astronautics

Abstract

This thesis investigates the use of Carrier-phase Differential GPS (CDGPS) in relative navigation filters for formation flying spacecraft. This work analyzes the relationship between the Extended Kalman Filter (EKF) design parameters and the resulting estimation accuracies, and in particular, the effect of the process and measurement noises on the semimajor axis error. This analysis clearly demonstrates that CDGPS-based relative navigation Kalman filters yield good estimation performance without satisfying the strong correlation property that previous work had associated with "good" navigation filters. Several examples are presented to show that the Kalman filter can be forced to create solutions with stronger correlations, but these always result in larger semimajor axis errors. These linear and nonlinear simulations also demonstrated the crucial role of the process noise in determining the semimajor axis knowledge. More sophisticated nonlinear models were included to reduce the propagation error in the estimator, but for long time steps and large separations, the EKF, which only uses a linearized covariance propagation, yielded very poor performance. In contrast, the CDGPS-based Unscented Kalman relative navigation Filter (UKF) handled the dynamic and measurement nonlinearities much better and yielded far superior performance than the EKF. The UKF produced good estimates for scenarios with long baselines and time steps for which the EKF would diverge rapidly. A hardware-in-the-loop testbed that is compatible with the Spirent Simulator at NASA GSFC was developed to provide a very flexible and robust capability for demonstrating CDGPS technologies in closed-loop. This extended previous work to implement the decentralized relative navigation algorithms in real time.

Thesis Supervisor: Jonathan P. How
Title: Associate Professor

Acknowledgments

First I thank my advisor, Prof. Jonathan How, for his guidance, lessons, humor and patience throughout my time at MIT. This research was supported by the National Science Foundation Graduate Research Fellowship Program and NASA Grants #NAG3-2839, #NAG5-10440, #NCC5-704, and #NCC5-729. Collaboration and discussion with Dr. Oliver Montenbruck, Prof. Terry Alfriend, Rich Burns, and many other people at NASA GSFC has been helpful in steering my work. Many of my colleagues at MIT, including Louis Breger, Arthur Richards, Ian Garcia, and Ellis King, have been gracious in offering their insight into technical issues that arose in my work. The support of these and many other MIT friends has made this journey memorable and enjoyable. To my dear friends back in the great State of Texas, thank you for all the calls and letters. I am grateful for my parents, whose love and encouragement helped me get where I am, and for my brother, Stuart, who should know that he is an inspiration to me. Above all, I thank my Heavenly Father, for it was His strength, and not my own, that enabled me to complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis investigates the use of Carrier-phase Differential GPS (CDGPS) in relative navigation filters for formation flying spacecraft. Previous work successfully developed an Extended Kalman Filter for this purpose [1]. This thesis extends that work by exploring the performance limitations and by explicitly implementing the decentralized form of the navigation filter and executing it in real time. Fundamental performance limits of the filter are explored in an analysis of the relationship between the filter noises and the navigation accuracy. The Extended Kalman Filter is also replaced with an Unscented Kalman Filter that more accurately handles the nonlinear dynamics and measurement equations. Finally, closed loop navigation and control experiments conducted at NASA Goddard Space Flight Center (GSFC) advanced the CDGPS-based relative navigation algorithms closer to space flight.

1.1 Motivation for Formation Flying Missions

Flying satellites in autonomously controlled formations is desirable for many space science and military applications [2]. The distributed satellites provide platforms for astrophysical interferometric observation instruments, Earth mapping systems such as synthetic aperture radar, or just distributed Earth observation systems [3]. Flying a formation of satellites increases the ability to distribute scientific sensors over longer baselines, giving better resolution. Furthermore, the geometry of the sensors is not

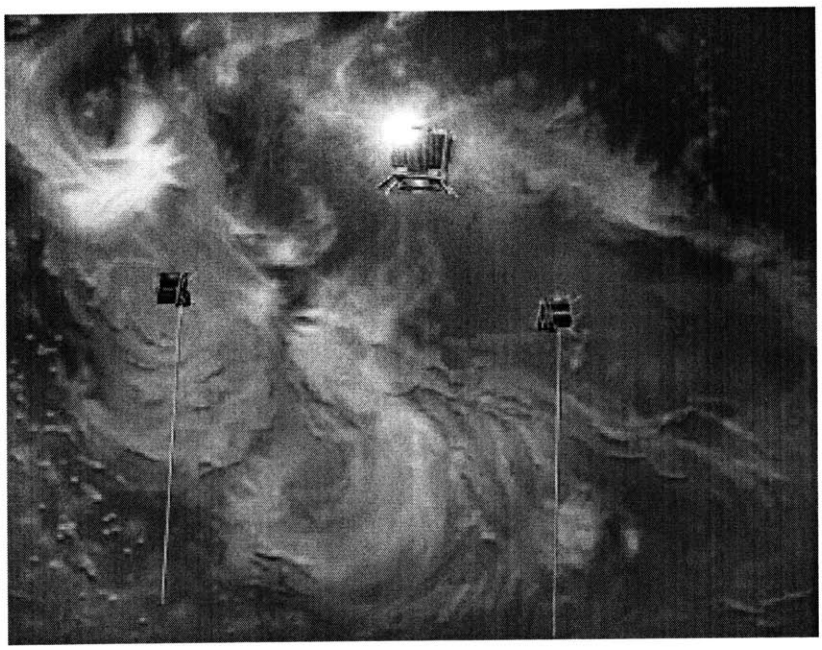

Fig. 1–1: Illustration of the proposed Orion formation flying mission

fixed by a physical structure, and can be changed with formation maneuvers as the need arises. Ref. [4] includes a list of some of the formation flying missions that have been proposed to take advantage of these features. One such mission, Orion, is a low-cost platform for demonstrating formation flying technologies and is shown in Fig. 1–1.

The dynamics of formation flying are based on the same mathematical models developed for rendezvous and station keeping missions [5]. Unlike these other missions, formation flying emphasizes cooperation between vehicles to meet fuel or maneuver time goals [5, 20]. Though relative navigation algorithms are typically formulated independently of the control objectives, accurate navigation estimates are required by the controller to make formation flying missions feasible [7].

A key step in precise formation flying is developing a sensor that can be used to accurately measure the **relative** positions and velocities of the vehicles in the fleet. Carrier-phase Differential GPS provides an ideal sensor for this relative navigation [8]. The following section highlights some of the recent work to develop CDGPS relative navigation filters for Low Earth Orbit (LEO) missions.

1.2 Previous Work on CDGPS-based Relative Navigation Filters

This thesis extends the work done by Busse, which presented the design and demonstration of a relative navigation filter based on CDGPS measurements [1, 9]. This filter was developed for use on formation flying satellites in LEO. An Extended Kalman Filter (EKF) that used nonlinear state propagation and nonlinear measurement update methods was shown to provide a highly accurate estimate. Adaptive techniques for process and measurement noise estimation were implemented as well, and shown to further improve the estimates. Busse demonstrated this algorithm with software and hardware simulations. The software simulations propagated a simple orbital dynamics truth model and used the measurement equations to simulate carrier phase measurements. The hardware simulations combined an Orion GPS receiver with a Spirent simulator at NASA GSFC, which generated the RF signals that the receiver would see in LEO. The stored receiver measurements were used to evaluate the filter off-line. The performance of this estimator, with position errors of $1-2$ cm and velocity errors of 0.5 mm/sec, matched or exceeded that of its contemporaries [1, 10, 11]. More recent advances in relative navigation have also been made by Leung and Montenbruck [12]. Extensive work to adapt the GPS receiver for use in LEO has resulted in noise levels of 0.5 mm for carrier phase measurements and 0.07 m/sec for Doppler measurements. When using a receiver with extremely low measurement noise, the relative navigation filter accuracy was reported as 1.5 mm for position and 0.005 mm/sec for velocity. This thesis used the same receivers as Busse, with a slightly revised tracking loop filter.

Formations in near-circular Low Earth orbits with baselines of a kilometer or less have been a starting point for much research in formation flying. However, future scientific goals may require formations in eccentric orbits or with larger baselines. These scenarios represent challenges for the standard LEO CDGPS-based navigation systems [13]. Ref. [14] uses a two-step filter to provide relative navigation for close, elliptical formations. The two-step filter was found to be superior to an iterated Ex-

tended Kalman filter when the initial error was large and when few measurements were available Ref. [15] conducted initial experiments using a CDGPS for large formations, between 1 and 500 km. Initial results show that, with an accurate dynamics model, CDGPS may still provide good navigation performance.

Coupling estimation algorithms with control algorithms in closed loop demonstrations is another aspect required to prove any navigation system. Some past research has worked towards this goal by incorporating pre-planned thrusts in off-line analysis [1]. Ref. [16] developed a closed loop simulation to test Lyapunov direct control methods that used the Spirent simulator in a study limited to two vehicles.

1.3 Contributions

One of the major goals of this thesis is to understand the fundamental relationships between the dynamic system, the measurements and the filter, as well as any inherent limits in filter performance. This knowledge can hopefully be used to highlight steps to improve formation flying navigation systems. This goal is addressed with several different experiments. Finally, a testbed was developed for demonstrating closed loop navigation and control algorithms.

Chapter 2 provides the basic estimator framework for the GPS measurements and the relative navigation filter. The research in Chapter 3 was motivated by a conjecture in the literature which suggested that the orbital GPS navigation results available to date (our CDGPS results included) may have a fundamental deficiency. The issue is with the semimajor axis knowledge, which is crucial for obtaining good control performance. Ref. [17] presents an analysis of the relationship between semimajor axis error, position and velocity error, and the correlation coefficient between position and velocity error. That analysis suggest that a "good" navigation filter would have a strong correlation (i.e. coefficient near -1) to reduce the semimajor axis error [18]. However, practical experience with CDGPS-based filters has shown this is seldom true, even when the accuracies appear to be very high (typical correlations ≈ -0.1). This chapter investigates this issue in detail, showing that the Kalman filter uses a

very different strategy for obtaining good estimates of the semimajor axis than those suggested in [18].

Chapter 3 also explores the relationship between filter accuracy and process & measurement noises. Both linear planar and nonlinear models are used to create plots similar to those in [17] that provide further insights on what types of noise improvements (*i.e.,* using better models to reduce process noise or lower the phase noise of the receiver to reduce the sensor noise) would improve the CDGPS filter accuracy.

Chapter 4 investigates the navigation accuracy in more detail by evaluating another variant of the Kalman Filter for nonlinear systems, called the Unscented Kalman Filter (UKF). The UKF was developed to provide better estimation results for nonlinear systems [19]. The chapter explores the limitations of an EKF and the benefits of the UKF as the estimation time step increases and the separation between the spacecraft grows. Both of these changes accentuate the nonlinearity in the system. Simulations using measurements from software models and from hardware tests at NASA GSFC both confirmed that the Unscented filter yields much better performance for large baselines or long time steps.

While the stand-alone performance of the navigation algorithm is certainly important, the final measure of success for an estimator is determined by its performance in a closed loop control scenario. A testbed was developed for the estimation algorithms described in this research to be connected to control algorithms described in Ref. [20]. The closed loop testing can be done with software-only or with hardware-in-the-loop. The software-only simulations are based on truth trajectory created with a high fidelity orbital propagator, and use simulated measurements created from the measurement equations and the truth trajectory. The hardware simulations, which replace the simulated measurements with a Spirent GPS signal simulator and Orion GPS receiver, were conducted at NASA GSFC. Chapter 5 describes the development of the testbed architecture required to support these closed loop simulations. This includes the Navigation Executive software that regulates the communication and executes the filter steps. Several simulations that illustrate the development of the

testbed are presented. While the testbed was created to demonstrate the estimation algorithms in this thesis and the control algorithms in Ref. [20], either component can be exchanged in future experiments. The Orion receiver could also be replaced to evaluate the performance of the EKF/UKF estimators with another receiver.

Chapter 2

A Kalman Filter with Relative Orbital Dynamics and CDGPS Measurements

In 1960, Robert Kalman introduced a new approach for minimum mean-square error filtering that used state space methods [21]. The Kalman Filter is a recursive scheme that propagates a current estimate of a state and the error covariance matrix of that state forward in time. The filter optimally blends the new information introduced by the measurements with old information embodied in the prior state with a Kalman gain matrix. The gain matrix balances uncertainty in the measurements with the uncertainty in the dynamics model [22]. The Kalman filter is guaranteed to be the best filter for a linear system with linear measurements [23]. However, few systems can be accurately modeled with linear dynamics. Shortly after its inception, improvements on the Kalman filter to handle nonlinear systems were proposed. One of the most popular choices, the Extended Kalman Filter, was used in previous work on relative navigation filters in LEO [1]. This chapter begins with a review of the discrete Extended Kalman Filter (EKF) and its application to the relative orbital navigation problem. Next, the mathematical descriptions of the system dynamics and measurement models that are used in the Kalman filter are presented. A review of the pertinent orbital mechanics equations leads to the definition of the system

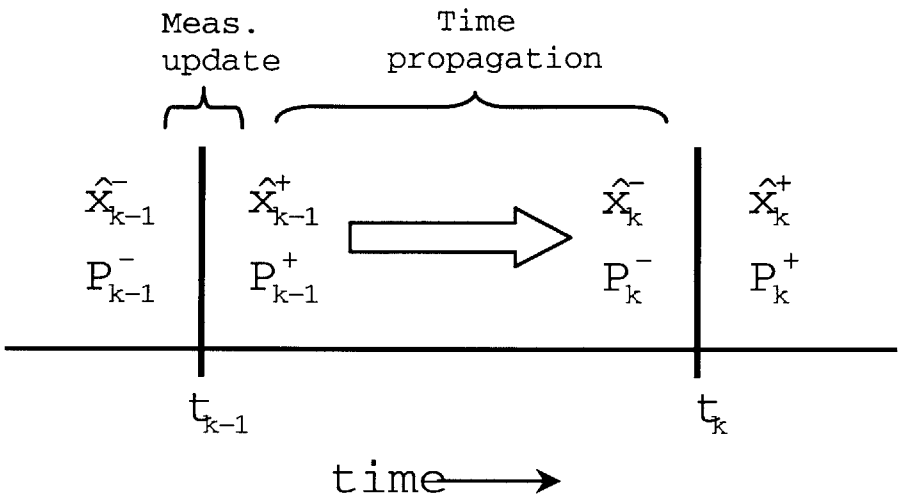

Fig. 2–1: Kalman Filter Process

dynamics model. The equations that relate Carrier Differential GPS measurements to the relative state are the basis for the measurement model in this Kalman Filter. The Kalman Filter, Orbital Dynamics, and GPS Measurements are discussed in this chapter and provide a base for the discussions in later chapters.

2.1 The Extended Kalman Filter

This section presents the equations used in the discrete Extended Kalman Filter (EKF). The dynamics and measurements models used in the Kalman filter are discussed in subsequent sections. The discrete EKF is as a state estimator for systems whose state dynamics model, measurement model, or both may be nonlinear, as in Eqs. 2.1 and 2.6 [23]. The dynamics model provides the equations to propagate $\hat{\mathbf{x}}_k$, the estimate of the state $\mathbf{x}$ at time k, to time step $k+1$, producing $\hat{\mathbf{x}}_{k+1}$. The measurement model then incorporates the new sensor information to update this estimate, updating the a priori estimate $\hat{\mathbf{x}}^-_{k+1}$ to the a posteriori estimate, $\hat{\mathbf{x}}^+_{k+1}$. This process is illustrated in Fig. 2–1.

The continuous state $\mathbf{x}$ is governed by the dynamics

$$\dot{\mathbf{x}}(t_k) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t_k) + \mathbf{w}(t_k) \tag{2.1}$$

where $\mathbf{u}$ is a known control input, and $\mathbf{w}(t)$ is an additive white noise that models the error accumulated by uncertainty in the dynamics during the time step. The power spectral density of this zero mean, white noise process is

$$Q = E[\mathbf{w}(t)\ \mathbf{w}(t)^T] \tag{2.2}$$

To proceed, linear expressions for the dynamics and measurement equations must be formed. In general, this requires knowledge of the probability density function [23], but the EKF approximates the nonlinear function by expanding it in a Taylor series, at each time step, about the current estimate,

$$F_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \tag{2.3}$$

The dynamics are discretized with time step Δt by forming the state transition matrix,

$$\Phi_k = e^{F_k \Delta t} \tag{2.4}$$

The cumulative effect of the white noise process $\mathbf{w}(t)$ over the time step is captured in the discrete process noise covariance matrix

$$Q_k = \int_0^{\Delta t} e^{F_k \tau} Q (e^{F_k \tau})^T d\tau \tag{2.5}$$

The vector of measurements, $\mathbf{y}$,

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, t) + \mathbf{v}_k \tag{2.6}$$

is modeled as a nonlinear function of the state and time, with an additive white noise process $v(t)$ that accounts for uncertainty the sensors and their models. The

measurement noise covariance matrix is defined by

$$R_k = E[\mathbf{v}_k \ \mathbf{v}_k^T] \tag{2.7}$$

The nonlinear measurement equation is also linearized about the current estimate,

$$H_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \tag{2.8}$$

Because approximations must be made in the linearization, the EKF is a sub-optimal filter, in the sense that its stability and performance are not guaranteed. Fortunately, the dynamics of orbital motion are fairly simple, and the EKF can have very good performance in space navigation applications [22]. The discrete, linear representation of the system dynamics are

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} + \mathbf{u}_{k-1} \tag{2.9}$$

The confidence in the current estimate is captured in the state error covariance matrix, P,

$$P_k = \mathrm{E}[\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^T] = \mathrm{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T] \tag{2.10}$$

where $\tilde{\mathbf{x}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k$ is the estimation error. The first step in the EKF involves propagating the state and error covariance forward in time. Eq. 2.9 (with zero process noise) is used to propagate the state estimate. The error covariance is propagated forward using

$$P_k^- = \Phi_{k-1}P_{k-1}^+\Phi_{k-1}^T + Q_{k-1} \tag{2.11}$$

An alternate approach to the time propagation step involves using the nonlinear dynamics equations to propagate the state. A $4^{th}-$order Runge-Kutta integration scheme uses the nonlinear state dynamics equation

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t)) \quad \text{for } t = t_{k-1} \rightarrow t_k \tag{2.12}$$

to find $\hat{\mathbf{x}}_k$. The state covariance is still propagated with Eq. 2.11, so the state transition matrix Φ_{k-1} must be calculated regardless of whether the linear or nonlinear state propagation is chosen. Previous research found the nonlinear method of state propagation offers significant improvement as both vehicle separation and integration time were increased [1]. Except where noted, the state propagation in this research employs this nonlinear method.

The second step of the filter uses the measurement equation to update the a priori state $\hat{\mathbf{x}}_k^-$ to the a posteriori state $\hat{\mathbf{x}}_k^+$. When a measurement becomes available, the new information provided by the measurement and the previous information captured in the state estimate are combined to form an updated state estimate. The Kalman gain K is the blending gain matrix that is used to weight the importance of the old and new information. The optimum gain matrix is formulated by minimizing the trace of the a posteriori state error covariance matrix P_k^+, which essentially minimizes the estimation error vector at each time step [23]. The terms in the gain matrix equation include the previous state estimate, the linearized measurement matrix, and the expected noise of the new measurements,

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{2.13}$$

The nonlinear measurement equation is used to update the state estimate

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- - K_k(\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)) \tag{2.14}$$

Note that the computation of the gain matrix K_k requires the linear measurement matrix H_k. The covariance is updated after the measurement with

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \tag{2.15}$$

which is the Joseph form of the covariance update whose inherent symmetry makes it numerically stable [24]. This section has introduced the basic equations for the Extended Kalman Filter. The definition of the state $\mathbf{x}$ and the development of the state

dynamics equations is presented in Section 2.2. The definition of the measurement vector $\mathbf{y}$ and the equations that relate the measurement to the state are presented in Section 2.3.

2.2 System Dynamics

The relative state used in this work includes the quantities of interest for navigation and control, relative position and velocity of the vehicle, as well as several other quantities that included in the augmented state so the filter will function properly. These other quantities, associated with the use of CDGPS, include the clock offset, the clock drift rate, and a carrier phase bias for each GPS satellite that is tracked. The state vector used for relative navigation between two vehicles in this work is

$$\mathbf{x}_k = \begin{bmatrix} \Delta\mathbf{r}_{ij}(t_k) \\ \Delta b_{ij}(t_k) \\ \Delta\dot{\mathbf{r}}_{ij}(t_k) \\ \Delta\dot{b}_{ij}(t_k) \\ \Delta\beta_{ij}^1 \\ \vdots \\ \Delta\beta_{ij}^N \end{bmatrix} = \begin{bmatrix} \textit{position vector} \\ \textit{clock offset} \\ \textit{velocity vector} \\ \textit{clock drift} \\ \textit{carrier phase bias, channel 1} \\ \vdots \\ \textit{carrier phase bias, channel N} \end{bmatrix} \tag{2.16}$$

where the relative vectors are expressed in the ECEF frame. The relative position and velocity dynamics, addressed in Section 2.2.1, are defined by orbital mechanics equations. The dynamics equations for the clock states for the carrier biases are shown in Section 2.2.2 and Section 2.2.3, respectively.

2.2.1 Orbital Mechanics

This research is focused on using the Kalman Filter to provide relative navigation for formations of spacecraft. There are several methods for computing the relative position and velocity. The absolute states of the Leader and Follower vehicles could be

found and differenced. This has proved inadequate for close formations [1]. Another strategy, used here, calculates the relative state of Follower vehicle, referenced to the position of the Leader vehicle.

The absolute state is used in the process of obtaining GPS measurements and in functions external to navigation and control, so it is also estimated. There are many estimation techniques available to form an absolute state solution [25, 26, 27]. A standard EKF was used in this work to estimate the absolute state. The dynamics used for the absolute state estimation are presented here and are also used in the development of the relative dynamics equations.

Absolute Orbital Motion

Newton's law of gravitation, Eq. 2.17, refines Kepler's observations that planets and other orbiting bodies follow conic section paths [28]. The motion of an orbiting body around a central massive body, *i.e.*, a satellite around the Earth, is governed by

$$\ddot{\mathbf{r}} = -\frac{\mu \mathbf{r}}{\|\mathbf{r}\|^3} \tag{2.17}$$

where $\mathbf{r}$ is the position vector of the orbiting body in the ECI frame, and $\mu = 3.986 \times 10^{14} \frac{m^3}{sec^2}$ is the gravitational parameter of the Earth. The reference frames used in these discussion are defined fully in Appendix A. In the case of an Earth orbiting satellite, there are additional perturbations that may be modeled

$$\ddot{\mathbf{r}} = -\frac{\mu \mathbf{r}}{\|\mathbf{r}\|^3} + \mathbf{a}_{J2} + \mathbf{a}_D + \mathbf{a}_B + \mathbf{a}_{SRP} \tag{2.18}$$

where the additional acceleration terms, $\mathbf{a}_{J2}$, $\mathbf{a}_D$, $\mathbf{a}_B$, and $\mathbf{a}_{SRP}$, model perturbation forces due to a non-spherical earth, aerodynamic drag, 3rd-body effects, and solar radiation pressure.

The acceleration term $\mathbf{a}_{J2}$ is a first order model of the nonuniform gravitational field that is caused by the oblateness and nonuniform composition of the earth [29, 30]. The oblateness results from the centripetal forces that forces mass outward at the

equator. The nonuniform distribution of mass throughout the earth leads to the nonuniform gravitational field. The deviation from a point mass gravitational model is captured by modeling different spherical harmonics. The J_2 term, modeling a zonal harmonic that represents the Earths equatorial bulge, accounts for the strongest perturbation in the Earth's gravitational field [31]. Higher order harmonics terms are not included in the Kalman filters used in this research. The J_2 term is used in some of the Kalman filer variants examined in this research [32, 33], and is quantified

$$\mathbf{a}_{J2} = -\frac{3}{2}\frac{J_2 \mu R_e}{r^5}\begin{bmatrix} \left(1-\frac{5z^2}{r^2}\right)x \\ \left(1-\frac{5z^2}{r^2}\right)y \\ \left(3-\frac{5z^2}{r^2}\right)z \end{bmatrix} \tag{2.19}$$

where the ECI position vector $\mathbf{r} = [\ x \quad y \quad z\ ]^T$ and has a length of r. The average radius of the Earth, R_e, is 6378.14 km, and the first zonal coefficient due to the Earth's oblateness, J_2, has a value of 0.0010826269 [28].

The acceleration term $\mathbf{a}_D$ accounts for aerodynamics forces. These forces are caused by the interaction between the atmosphere, whose density is a strong function of altitude and sun exposure, and the surface of the vehicle [28]. The drag term is more significant than the lift force on the blunt body of the spacecraft, so typically only the drag is modeled. The magnitude of the drag force is dependent on the density of the atmosphere, and the drag coefficient, which is determined by the fontal area of the satellite and the in-track velocity of the satellite. Drag has more of an effect on satellites in LEO than satellites in higher orbits, but is generally small. It is not included in this research.

The term $\mathbf{a}_B$ accounts for the force due to the gravitational field of additional 3^{rd} body masses, such as the Earth's moon. This force is greater for orbits with high inclination and large eccentricity. The primary orbit this research addresses is a near-circular Low Earth Orbit. The 3rd-body perturbation is not modeled in this research, but may become important for missions with eccentric orbits.

The acceleration term $\mathbf{a}_{SRP}$ models the pressure force on the spacecraft generated

by solar radiation. The effect of this forces is greater for satellites in high orbits. Solar radiation pressure is not modeled in this research.

An equivalent representation of the Cartesian state is given by a set of orbital elements. The orbital elements are a set of angles and other parameters that describe the orbital ellipse, its orientation with respect to an inertial reference frame, and the position of the spacecraft in the ellipse. A good discussion of orbital elements and the conversion of between Cartesian coordinates and orbital elements is found in Ref. [28]. Several of the elements and associated parameters appear in the discussion of the relative navigation filters in this research, and are thus defined here. The orbital period, τ, is

$$\tau = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{2.20}$$

where a is the semimajor axis of the orbital ellipse. The mean motion of the orbit, n, is

$$n = \sqrt{\frac{\mu}{a^3}} = \frac{2\pi}{\tau} \tag{2.21}$$

The orbital energy, related to the velocity and height of the satellite, and the semimajor axis a of the orbit, is

$$\frac{\dot{\mathbf{r}}^2}{2} - \frac{\mu}{\|\mathbf{r}\|} = -\frac{\mu}{2a} \tag{2.22}$$

Relative Orbital Motion

The relative position and velocity terms in the state vector are governed by relative orbital dynamics equations, which can be derived from the absolute orbital dynamics equations. The relative position between vehicles i and j is shown in Fig. 2–2. The relative position vector, $\Delta\mathbf{r}_{ij}$, is defined as the difference between the absolute position vectors of vehicles i and j,

$$\Delta\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i \tag{2.23}$$

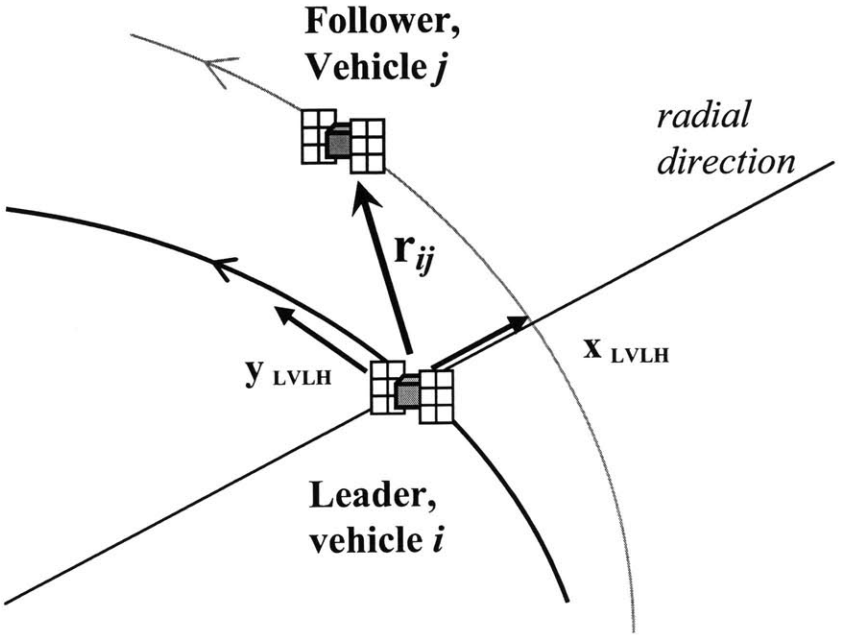

Fig. 2–2: Relative Position Between Leader and Follower Vehicles

It follows that the relative acceleration is defined by the difference of the two absolute accelerations, with thrust accelerations accounted for with the term δu_{ij},

$$\Delta\ddot{\mathbf{r}}_{ij} = \ddot{\mathbf{r}}_j - \ddot{\mathbf{r}}_i + \Delta u_{ij} \tag{2.24}$$

Substituting Equations 2.17 and 2.23 into 2.24 gives an expression for the relative dynamics with perturbation terms in the ECI frame,

$$\Delta\ddot{\mathbf{r}}_{ij} = \frac{\mu}{\|\mathbf{r}_i\|^3}\left[\mathbf{r}_i - \frac{\|\mathbf{r}_i\|^3(\mathbf{r}_i + \Delta\mathbf{r}_{ij})}{\sqrt{\|\mathbf{r}_i\|^2 + 2\mathbf{r}_i \cdot \Delta\mathbf{r}_{ij} + \|\Delta\mathbf{r}_{ij}\|^2)^3}}\right] + \Delta u_{ij} \tag{2.25}$$

This expression includes the absolute position vector $\mathbf{r}_{ij}$ of the reference vehicle. The GPS constellation and the GPS navigation message use the ECEF reference frame, so ECEF is a natural reference frame for navigation filters using GPS measurements. To transform Eq. 2.26 from the ECI frame to the ECEF frame, a correction term

accounting for the Coriolis effect is added to the dynamics equation,

$$\Delta\ddot{\mathbf{r}}_{ij} = \frac{\mu}{\|\mathbf{r}_i\|^3}\left[\mathbf{r} - \frac{\|\mathbf{r}_i\|^3(\mathbf{r}_i + \Delta\mathbf{r}_{ij})}{\sqrt{\|\mathbf{r}_i\|^2 + 2\mathbf{r}_i \cdot \Delta\mathbf{r}_{ij} + \|\Delta\mathbf{r}_{ij}\|^2)^3}}\right] + \Delta u_{ij} + C_{ECEF} + \Delta\mathbf{a}_{J2} + \Delta\mathbf{a}_D + \Delta\mathbf{a}_B + \Delta\mathbf{a}_{SRP} \tag{2.26}$$

where the differential disturbance perturbations have been added, and

$$C_{ECEF} = 2\Omega_e \times \Delta\dot{\mathbf{r}}_{ij} + \Omega_e \times \Omega_e \times \Delta\mathbf{r}_{ij} \tag{2.27}$$

where $\Omega_e = 7.292 \times 10^{-5}\frac{rad}{sec}$ is the rotation rate of the Earth about its axis. Other more sophisticated methods for performing the ECI to ECEF rotation are described in Ref. [28], but the propagation time in the Kalman filter is typically very short, so Eq. 2.27 is usually sufficient. The Jacobian F of the nonlinear relative dynamics equations in Eq. 2.26 is required for the linear state propagation scheme and for the covariance propagation. The linear dynamics matrix for the position and velocity states is given as

$$F = \begin{bmatrix} F_{pp} & F_{pv} \\ F_{vp} & F_{vv} \end{bmatrix} \tag{2.28}$$

where the position and velocity partitions are

$$F_{pp} = \mathbf{0}_{3\times3} \tag{2.29}$$

$$F_{pv} = \mathbf{I}_{3\times3} \tag{2.30}$$

$$F_{vp} = \frac{\mu}{r^3}\begin{bmatrix} -1 + 3\frac{x^2}{r^2} - \frac{r^3}{\mu}\Omega_e^2 & 3\frac{xy}{r^2} & 3\frac{xz}{r^2} \\ 3\frac{yx}{r^2} & -1 + 3\frac{y^2}{r^2} - \frac{r^3}{\mu}\Omega_e^2 & 3\frac{yz}{r^2} \\ 3\frac{zx}{r^2} & 3\frac{zy}{r^2} & -1 + 3\frac{z^2}{r^2} \end{bmatrix} \tag{2.31}$$

$$F_{vv} = \begin{bmatrix} 0 & 2\Omega_e & 0 \\ -2\Omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.32}$$

The expressions for relative state dynamics expressed in the LVLH frame are

also considered. The LVLH dynamics are not used for the state propagation, but are useful for the discussions found in Chapter 3. These equations follow from the assumption that the spacecraft are in close proximity and are in near-circular orbits. The differential perturbation forces are combined into a single acceleration term, $\mathbf{f}$. The linearized relative orbital dynamics for circular orbits expressed in the LVLH frame, also called Hill's or Clohessy-Wiltshire equations [34, 28], are

$$\ddot{x} - 2n\dot{y} - 3n^2 x = f_x \tag{2.33}$$

$$\ddot{y} + 2n\dot{x} = f_y \tag{2.34}$$

$$\ddot{z} + n^2 z = f_z \tag{2.35}$$

These equations highlight the relative motion in the radial (x), in-track (y), and cross track directions (z) in the LVLH frame (see Appendix A). The coupling between the radial and in-track directions is apparent in these equations. The out-of-plane motion is modeled as a simple harmonic oscillator. Note that it is well known that the accuracy of these equations degrades rapidly as the separation between vehicles increases or as eccentricity is introduced to the reference orbit.

2.2.2 Clock Dynamics

CDGPS navigation techniques require that time be known with a high degree of accuracy. The clocks on the local receivers are relatively low quality and unstable, so a clock offset and a clock drift rate must be estimated by including each in the Kalman filter state definition. The dynamics of the clock offset from GPS time, b and the clock drift rate, $\dot{b}$, are modeled as

$$\begin{bmatrix} \dot{b} \\ \ddot{b} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b \\ \dot{b} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_b \tag{2.36}$$

When the single differences are performed between two vehicles, the relative clock dynamics retain only the differential white noise term,

$$\begin{bmatrix} \Delta\dot{b}_{ij} \\ \Delta\ddot{b}_{ij} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta b_{ij} \\ \Delta\dot{b}_{ij} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_{\Delta b} \tag{2.37}$$

Thus, in the state propagation step, the clock model will contribute nothing to the state transition matrix, but will introduce terms in the noise covariance model.

2.2.3 Carrier Phase Biases

Each GPS carrier phase measurement includes a bias term. This bias is treated as a constant that must be estimated. A description of the carrier bias is found in Section 2.3.2. The differential biases are modeled as constants,

$$\Delta\dot{\beta}_{ij}^{m} = 0 \tag{2.38}$$

2.2.4 System Dynamics Summary

The full relative state is defined as

$$\mathbf{x}_k = \begin{bmatrix} \Delta\mathbf{r}_{ij}(t_k) \\ \Delta b_{ij}(t_k) \\ \Delta\dot{\mathbf{r}}_{ij}(t_k) \\ \Delta\dot{b}_{ij}(t_k) \\ \Delta\beta_{ij}^{1} \\ \vdots \\ \Delta\beta_{ij}^{N} \end{bmatrix} \tag{2.39}$$

The carrier biases, $\Delta\beta_{ij}^{m}$, are constant and are ignored during the state propagation step. A truncated vector which excludes the biases $\Delta\beta_{ij}^{1}, \ldots, \Delta\beta_{ij}^{N}$ is propagated.

The linearized dynamics of the position, velocity, and clock states are modeled as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} F_{pp} & 0 & F_{pv} & 0 \\ 0 & 0 & 0 & 1 \\ F_{vp} & 0 & F_{vv} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \mathbf{w} + \mathbf{u} \tag{2.40}$$

where

$$\mathbf{w} = \begin{bmatrix} \mathbf{0}_{1\times 4} & \mathbf{w}_{\Delta \mathbf{r}} & w_{\Delta b} \end{bmatrix}^T \tag{2.41}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{0}_{1\times 4} & \Delta \mathbf{u}_{ij} & 0 \end{bmatrix}^T \tag{2.42}$$

The dynamics and process noise models are then discretized as discussed in Section 2.1.

2.3 Measurement Update: Carrier Differential GPS

2.3.1 A Review of Global Positioning System Basics

The NAVSTAR Global Positioning System (GPS) is a space based, radio navigation system developed, owned, and operated by the United States Department of Defense. The GPS satellites transmit signals on two carrier frequencies. The civilian L1 frequency, 1575.42 MHz, carries a pseudo-random code for timing and contains a navigation message with ephemeris data [35]. GPS positioning is based on the principle of trilateration, which is the process of ranging off at least three objects with known position to determine a local position. The clocks that are used in the GPS ranging process are low quality, so the time is added as a fourth dimension. Because of the time uncertainty, the four required ranges are not true measures of position, and are thus called pseudoranges.

The standard method of obtaining a pseudorange involves using the navigation information on the code message. Code based pseudorange measurements typically

produce differential accuracies of several meters, which are not sufficient for formation flying missions. Carrier techniques offer much higher accuracy by calculating pseudoranges from the phase measurement of the RF carrier wave. If carrier measurements from a mobile receiver and a base station are differenced (forming a CDGPS measurement), the motion can be observed highly accurately. If the base is also moving, as in the case of Leader and Follower vehicles in a satellite formation, the CDGPS observable leads to relative position and velocity [25]. CDGPS measurements are used in this research to provide highly accurate relative navigation. The following section develops the CDGPS based pseudorange equations.

In addition to the code and carrier pseudoranges, a Doppler measurement,which can be related to velocity, is available from the GPS receiver. The Doppler measurement is created inside the receiver by differencing carrier phase measurements. Because this is not a truly independent measurement, previous research has found that adding Doppler measurements does not significantly improve the state estimate [1]. For this reason, the Doppler measurements are not included in this discussion.

2.3.2 Raw GPS Measurements

The code based pseudorange is used to calculate the absolute state. The code phase, ρ, is

$$\rho_i^m = \|\mathbf{r}^{m_i} - \mathbf{r}_i\| + b_i + B^{m_i} + I_i^m + \nu_\rho \tag{2.43}$$

where $\|\mathbf{r}^{m_i} - \mathbf{r}_i\|$ represents the true range between where the vehicle i is at the measurement time and the GPS satellite m at the transmission time. Offset errors in the clock of vehicle i and the GPS satellite m are captured in the terms b_i and B^{m_i}. The unmodeled (and unknowable) phenomenon that affect the code phase measurement are included in the noise term, ν_ρ. The term I_i^m models the delay imposed on the signal by the ionosphere. This term is modeled as

$$I_i^m = \frac{82.1 \times TEC}{F_c^2 \times \sqrt{\sin^2 \gamma_i^m + 0.076} + \sin \gamma_i^m} \tag{2.44}$$

where TEC is the total electron count on the atmosphere, a varying quantity influenced by, among other things, local solar illumination and sunspot activity. The frequency of the electromagnetic, F_c, and the elevation angle of the GPS satellite m with respect to vehicle i, γ_i^m, both influence the path delay caused by the ionosphere.

The carrier phase pseudorange, similarly relating range, clock states, and ionospheric delay, is

$$\phi_i^m = \|\mathbf{r}^{m_i} - \mathbf{r}_i\| + b_i + B^{m_i} + \beta_i^m - I_i^m + \nu_\phi \tag{2.45}$$

A carrier phase noise term, ν_ϕ in Eq. 2.45, replaces the code noise from Eq. 2.43. The difference in the effect of wave delay seen by the carrier and group delay seen by the code is reflected in the carrier pseudorange in Eq. 2.45 that has an ionospheric delay term that opposite in sign.

The additional term β_i^m introduced in the carrier pseudorange is a carrier phase bias. The bias is required to deal with an integer ambiguity in the phase measurement. The distance between the GPS satellite and the vehicle can be expressed as the sum of the carrier phase ϕ, and an integer multiple k of the carrier wavelength λ,

$$d = \phi + k\lambda$$

where $\lambda \approx 19.2$ cm. The distance viewed as being measured in units of carrier wavelengths, the fractional part of the distance, which is the carrier phase measurement, is known very accurately. The part of the distance that is covered by the integer multiple of wavelengths cannot be determined immediately from the information in the carrier phase measurement. Fortunately, there are a number of techniques available to determine this integer number. A passive technique called *kinematic positioning* was used in this research. As the GPS constellation and the spacecraft move relative to each other, the range measurements will change, but the bias remains constant [1]. With measurements collected over time, the biases are then observable and can be estimated. While this technique results in a longer startup time, it is quite simple and the biases do not change after the initial startup period. When new GPS satellites

enter the antenna's field of view, the biases in their measurements can be determined very quickly. Another advantage to this approach is that because the bias estimates are not necessarily required to be integers, the bias estimate can include constant errors, such as those potentially introduced by an antenna line bias or the correlator inside the receiver.

The line-of-sight (LOS) vector is a unit vector whose origin is vehicle i and points towards GPS satellite m,

$$los_i^m = \frac{\mathbf{r}^{m_i} - \mathbf{r}_i}{\|\mathbf{r}^{m_i} - \mathbf{r}^i\|} \tag{2.46}$$

The vectors in the LOS equation refer to the positions of vehicle i at the time of measurement and the GPS satellite m at the time of signal transmission. The measurement matrix H includes the LOS vectors for each GPS satellite tracked,

$$H_{LOS} = \begin{bmatrix} los_i^1 & 1 \\ \vdots & \\ los_i^N & 1 \end{bmatrix} \tag{2.47}$$

The Geometric Dilution of Precision (GDOP), indicates the distribution of satellites,

$$\text{GDOP} = \sqrt{\texttt{trace}\ [(H_{LOS}^T H_{LOS})^{-1}]} \tag{2.48}$$

A low GDOP indicates good GPS satellite coverage, which means that measurements are available in all directions, providing good observability of the state. Conversely, a large GDOP indicates poor coverage and may result in degraded estimates.

2.3.3 Differential Carrier Phase Measurements and Relative State

When two vehicles in close proximity track the same GPS satellites, the measurement for GPS satellite m taken by vehicle i will see many of the same errors as the measurement taken by vehicle j. If these measurements are differenced, then the errors cancel to a large degree. This is the crux of the advantage of Carrier Differential GPS

(CDGPS). The carrier differential phase is defined as

$$\Delta\phi_{ij}^{m} = \phi_{j}^{m} - \phi_{i}^{m} \tag{2.49}$$

where ϕ_i^m and ϕ_j^m are the raw carrier phases from GPS satellite m measured by vehicles i and j. This difference is formed for each GPS satellite commonly tracked by both vehicles. Substituting the Eq. 2.45 into this difference yields an expression for the carrier differential phase measurement,

$$\Delta\phi_{ij}^{m} = \|\mathbf{r}^{m_i} - \mathbf{r}_i\| - \|\mathbf{r}^{m_j} - \mathbf{r}_j\| + \Delta\beta_{ij}^{m} + \Delta b_{ij} + \Delta B_{ij}^{m} + \Delta I_{ij}^{m} + \nu_{\Delta\phi} \tag{2.50}$$

The carrier differential phase can be expressed explicitly as a function of the relative state, as defined in Eq. 2.23,

$$\Delta\phi_{ij}^{m} = \|\mathbf{r}^{m_i} - \mathbf{r}_i\| - \|\mathbf{r}^{m_j} - (\mathbf{r}_i + \Delta\mathbf{r}_{ij})\| + \Delta\beta_{ij}^{m} + \Delta b_{ij} + \Delta B_{ij}^{m} + \Delta I_{ij}^{m} + \nu_{\Delta\phi} \tag{2.51}$$

As in the equations for relative orbital mechanics, the relative carrier phase measurement equation retains the absolute state of the reference vehicle. Also, the error terms introduced for the raw carrier phase measurements have become differential terms. If the vehicles are close, it is reasonable to assume that the terms modeling the GPS satellite clock error and the ionospheric delay cancel,

$$\Delta\phi_{ij}^{m} = \|\mathbf{r}^{m_i} - \mathbf{r}_i\| - \|\mathbf{r}^{m_j} - (\mathbf{r}_i + \Delta\mathbf{r}_{ij})\| + \Delta\beta_{ij}^{m} + \Delta b_{ij} + \nu_{\Delta\phi} \tag{2.52}$$

2.3.4 A Summary of Measurement Equations

The measurement vector, $\mathbf{y}$, contains all the measurements that are used in the Kalman filter. The GPS receiver used in this research tracks up to 12 GPS satellites, so the measurement vector

$$\mathbf{y}_k = \begin{bmatrix} \Delta\phi_{ij}^{1}(t_k) \\ \vdots \\ \Delta\phi_{ij}^{N}(t_k) \end{bmatrix} \tag{2.53}$$

may include as many as $N = 12$ differential carrier phase measurements. Given an estimate of the relative state, the nonlinear measurement equation is

$$\hat{\mathbf{y}}_k = \begin{bmatrix} \|\hat{\mathbf{r}}^{1_i} - \hat{\mathbf{r}}_i\| - \|\hat{\mathbf{r}}^{1_j} - (\hat{\mathbf{r}}_i + \Delta\mathbf{r}_{ij})\| + \Delta b_{ij} + \Delta\beta_{ij}^1 + \Delta\hat{B}_{ij}^1 \\ \vdots \\ \|\hat{\mathbf{r}}^{N_i} - \hat{\mathbf{r}}_i\| - \|\hat{\mathbf{r}}^{N_j} - (\hat{\mathbf{r}}_i + \Delta\mathbf{r}_{ij})\| + \Delta b_{ij} + \Delta\beta_{ij}^N + \Delta\hat{B}_{ij}^N \end{bmatrix} \tag{2.54}$$

$$+ \begin{bmatrix} \Delta\hat{I}_{ij}^1(\hat{\mathbf{r}}_i, \Delta\hat{\mathbf{r}}_{ij}, \hat{\mathbf{r}}^m) + \nu_{\Delta\phi} \\ \vdots \\ \Delta\hat{I}_{ij}^N(\hat{\mathbf{r}}_i, \Delta\hat{\mathbf{r}}_{ij}, \hat{\mathbf{r}}^m) + \nu_{\Delta\phi} \end{bmatrix}$$

$$= \hat{h}_k(\hat{\mathbf{x}}_k^-) + \nu \tag{2.55}$$

and the associated Jacobian is

$$H_k = \begin{bmatrix} LOS_{(N\times 3)} & \mathbf{1}_{(N\times 1)} & \mathbf{0}_{(N\times 3)} & \mathbf{0}_{(N\times 1)} & \mathbf{I}_{(N\times N)} \end{bmatrix} \tag{2.56}$$

2.4 Summary

The EKF for CDGPS-based relative navigation described in this chapter was developed and tested by Busse [1], and is the basis for much of the work in this thesis. Chapter 3 investigates the relationships between measurement and process noises and filter performance for this filter. Busse's algorithm was structured to allow real-time, decentralized execution, but he did not implement it that way. Chapter 5 describes the adaptation of this algorithm in a testbed for closed loop navigation and control algorithms.

Chapter 3

Noise and Navigation Accuracy for CDGPS Filters

Effective design and analysis of a navigation filter must include a detailed evaluation of the filter's performance. Understanding the factors driving the performance and the elements of the system that limit navigation accuracies provides useful information for future improvements. Changing the various elements of the estimator, such as the sensor or dynamics model, can provide new insights into the steps necessary to improve performance. This chapter performs a key step in the estimator evaluation by investigating the relationship between the Kalman filter design parameters and the resulting navigational accuracies, and in particular, the roles of the process and measurement noises and their effect on semimajor axis error. Accurate knowledge of semimajor axis error is a dominate factor for control system performance, and thus is functionally the most important navigation parameter [6].

The chapter begins with a review of how semimajor axis error relates to the other navigational errors in our CDGPS-based Kalman filter. The design parameters available in the Kalman filter are reviewed prior to the exploration of their effects. This investigation begins with a linear model that excludes errors introduced by absolute state error, ionosphere, clocks, and carrier phase bias. The goal is to use the linear model to gain insights into the fundamental behavior of the filter before adding other real-world effects.

The linear model is first used to demonstrate that navigation accuracy degrades when the problem statement is changed to force a high correlation between the position and velocity estimates. This provides a counterexample to a previous conjecture that suggests semimajor axis error can effectively be canceled when there is high correlation between position and velocity error [18]. The example is further used to explore how the filter design parameters affect the navigation accuracy. Finally, the mapping between the simplified, linear simulations and full CDGPS-driven simulations is shown to validate conclusions drawn from the linear analysis.

3.1 Relating Navigational Errors to Semimajor Axis Error

In formation flying missions, accurate knowledge of the difference in semimajor axes, or equivalently, the difference in orbital energy, between the vehicles in the formation is important [7], [17], [18]. A difference in semimajor axes means that the two vehicles have different orbital periods and thus they will drift out of formation unless considerable control effort is applied [6].

The output of the CDGPS Kalman filter includes the relative formation state in a Local Vertical Local Horizontal (LVLH) reference frame, which is defined in Appendix A. Understanding the relationship between position and velocity accuracies and semimajor axis accuracy is key to evaluating the output of this type of filter. While Ref. [18] develops the navigation error analysis from absolute state relations, the results can be reformulated for the relative case. The relative navigation error equations, shown below, relate semimajor axis error to position and velocity errors. Note that this discussion is limited to circular reference orbits. The semimajor axis, a, of vehicle i is

$$\frac{1}{a_i} = \frac{2}{r_i} - \frac{v_i^2}{\mu} \tag{3.1}$$

where r and v are the position and velocity magnitudes in the Earth Centered Inertial (ECI) reference frame, and μ is the gravitational constant of the Earth. Eq. 3.1 is

used to find the difference in semimajor axes of vehicles i and j,

$$\Delta a_{ij} \approx 2(r_j - r_i) + \frac{2}{n}(v_j - v_i) \tag{3.2}$$

where the vehicles are assumed to be in circular orbits and close to each other. The relative position and velocity differences in Eq. 3.2 are the differences in the magnitudes of position and velocity of the two vehicles. If the two vehicles are close and in circular orbits, a reasonable approximation is to assume that the position difference is in the radial direction and the velocity difference is in the in-track direction [34]. The radial, in-track, and cross-track directions define the x, y and z axes of the LVLH reference frame. The relative dynamics in this LVLH reference frame are described by Hill's equations [34],

$$\ddot{x} - 2n\dot{y} - 3n^2 x = f_x \tag{3.3}$$

$$\ddot{y} + 2n\dot{x} = f_y \tag{3.4}$$

$$\ddot{z} + n^2 z = f_z \tag{3.5}$$

The force-free solution to Hill's equations is

$$x(t) = \frac{\dot{x}_0}{n}\sin nt - \left(\frac{2\dot{y}_0}{n} + 3x_0\right)\cos nt + \left(\frac{2\dot{y}_0}{n} + 4x_0\right) \tag{3.6}$$

$$y(t) = \frac{2\dot{x}_0}{n}\cos nt + \left(\frac{4\dot{y}_0}{n} + 6x_0\right)\sin nt + \left(y_0 - \frac{2\dot{x}_0}{n}\right) - (3\dot{y}_0 + 6nx_0)t \tag{3.7}$$

$$z(t) = z_0 \cos nt + \frac{\dot{z}_0}{n}\sin nt \tag{3.8}$$

In terms of relative radial position and in-track velocity, x and $\dot{y}$, in the Hill's reference frame, the difference in semimajor axes (the ij subscript is omitted) is approximately given by

$$\Delta a \approx 2\left(2x + \frac{\dot{y}}{n}\right) = -(6nx + 3\dot{y})\left(\frac{-2}{3n}\right) \tag{3.9}$$

The differential semimajor axis is directly related to the secular drift term in the solution to Hill's equations, $-(6nx + 3\dot{y})t$, by a factor of $-\frac{2}{3n}$. If the difference in

semimajor axes is zero, there will be no secular drift between the spacecraft. The standard deviation of the differential semimajor axis estimate, $\sigma_{\Delta a}$, follows directly, as in [18],

$$\sigma_{\Delta a} = 2\sqrt{4\sigma_x^2 + \frac{4}{n}\rho_{x\dot{y}}\sigma_x\sigma_{\dot{y}} + \frac{1}{n^2}\sigma_{\dot{y}}^2} \tag{3.10}$$

The parameters σ_x, $\sigma_{\dot{y}}$, and $\rho_{x\dot{y}}$ are derived from the error covariance matrix for the relative LVLH state estimate, $\hat{\mathbf{x}}$,

$$\hat{\mathbf{x}} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{3.11}$$

with estimation error $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$, which is assumed to be unbiased, $E[\tilde{\mathbf{x}}] = \mathbf{0}$, and have the covariance matrix of

$$E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y & \rho_{x\dot{x}}\sigma_x\sigma_{\dot{x}} & \rho_{x\dot{y}}\sigma_x\sigma_{\dot{y}} \\ \rho_{yx}\sigma_y\sigma_x & \sigma_y^2 & \rho_{y\dot{x}}\sigma_y\sigma_{\dot{x}} & \rho_{y\dot{y}}\sigma_y\sigma_{\dot{y}} \\ \rho_{\dot{x}x}\sigma_{\dot{x}}\sigma_x & \rho_{\dot{x}y}\sigma_{\dot{x}}\sigma_y & \sigma_{\dot{x}}^2 & \rho_{\dot{x}\dot{y}}\sigma_{\dot{x}}\sigma_{\dot{y}} \\ \rho_{\dot{y}x}\sigma_{\dot{y}}\sigma_x & \rho_{\dot{y}y}\sigma_{\dot{y}}\sigma_y & \rho_{\dot{y}\dot{x}}\sigma_{\dot{y}}\sigma_{\dot{x}} & \sigma_{\dot{y}}^2 \end{bmatrix} \tag{3.12}$$

Note that if the radial position and in-track velocity are linearly correlated ($\rho_{x\dot{y}} = -1$), the expression for semimajor axis variance, from Eq. 3.10, reduces to

$$\sigma_{\Delta a} = 2\sqrt{4\sigma_x^2 - \frac{4}{n}\sigma_x\sigma_{\dot{y}} + \frac{1}{n^2}\sigma_{\dot{y}}^2} = 2\sqrt{\left(2\sigma_x^2 - \frac{1}{n}\sigma_{\dot{y}}\right)^2} \tag{3.13}$$

If the position and velocity error are linearly correlated and satisfy

$$\sigma_{\dot{y}} = 2n\sigma_x \tag{3.14}$$

then the position and velocity errors cancel and there is **no** error in the semimajor axis estimate. In other words, the two requirements for zero semimajor axis variance

are:

$$\rho_{x\dot{y}} = -1 \quad \text{and} \quad \sigma_{\dot{y}} = 2n\sigma_x$$

which will subsequently be referred to as the correlation and balance requirements. In the examples, the balance is quantified with the *balance index*,

$$\texttt{bal} = \left| 1 - \frac{2n\sigma_x}{\sigma_{\dot{y}}} \right| \tag{3.15}$$

which should be zero when the balance requirement is met.

Note that if, instead, we have $\rho_{x\dot{y}} = 0$, then the expression for semimajor axis error reduces to

$$\sigma_{\Delta a} = 2\sqrt{4\sigma_x^2 + \frac{1}{n^2}\sigma_{\dot{y}}^2} \tag{3.16}$$

and in this case, $\sigma_{\Delta a}$ will not be zero unless both σ_x and $\sigma_{\dot{y}}$ are zero, which is not realistic.

The relationship between σ_x, $\sigma_{\dot{y}}$, $\rho_{x\dot{y}}$, and $\sigma_{\Delta a}$ is illustrated in Fig. 3–1 [18]. The x and y axes of the plot are the standard deviations of the position and velocity estimation errors. Contours of constant semimajor axis are shown on the figure. Each contour is associated with a value of $\rho_{x\dot{y}}$ in addition to a level of $\sigma_{\Delta a}$; several values of $\rho_{x\dot{y}}$ are shown for each level of $\sigma_{\Delta a}$. The diagonal line indicates where $\sigma_{\dot{y}} = 2n\sigma_x$. Along the diagonal, the lines of constant semimajor axis experience a "bump" that increases in size as the correlation tends towards -1. This bump corresponds to increasing cancelation between the error in x and $\dot{y}$ that results from increasing correlation in these errors. Essentially, if the errors have high correlation and the proper balance, the higher error levels can be tolerated with the same resulting semimajor axis. Each point on the graph corresponds to a unique set of σ_x and $\sigma_{\dot{y}}$. However, many points on the graph are intersected by more than one contour of constant semimajor axis. It is the correlation that determines the specific contour on which the system lies.

The analysis presented above was considered when exploring strategies to improve our CDGPS filter. Any navigation system that is not on the "bump" and does not

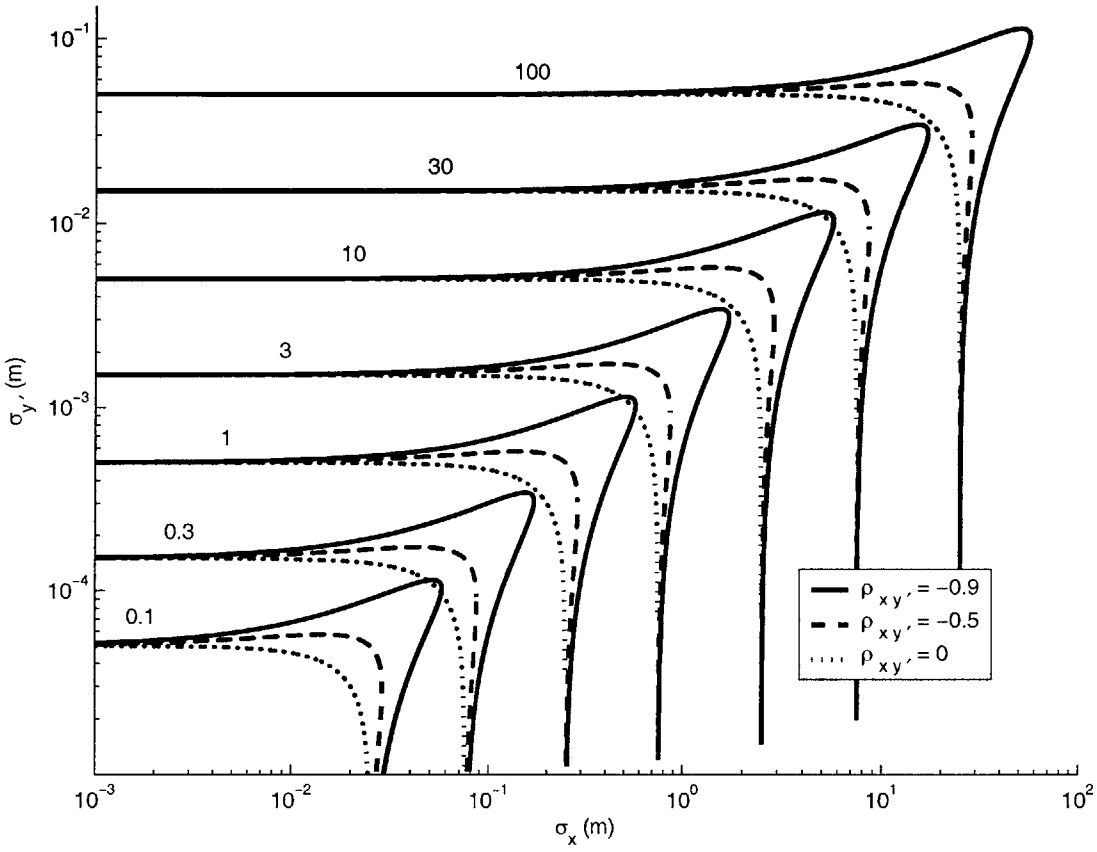

Fig. 3–1: Families of contours of constant semimajor axis are shown on axes of position and velocity accuracy. For each level of semimajor axis, the contour corresponding to three levels of correlation are shown [18].

have high correlation is not taking full advantage of the boost in semimajor axis knowledge that might otherwise be enjoyed. Thus, making changes to meet the balance and correlation requirements should result in improvements to semimajor axis error. Unfortunately, the Kalman filter does not allow independent control over σ_x, $\sigma_{\dot{y}}$, and $\rho_{x\dot{y}}$. Further discussion of the relationship between typical values of σ_x and $\sigma_{\dot{y}}$ output from a Kalman filter and corresponding $\rho_{x\dot{y}}$ and $\sigma_{\Delta a}$ is provided in the following section.

3.2 Semimajor Axis Accuracy from a Cartesian Filter Output

Theoretically, meeting the correlation and balance requirements described above will result in perfect semimajor axis knowledge, so a design goal for a navigation system might be to achieve these requirements. In fact, Ref. [18] categorizes filters that meet the correlation requirement as "good." However, practical experience has shown that output from CDGPS filters typically does not meet these requirements, raising the questions of why the two requirements are not met and whether meeting these requirements should be a design goal.

In Ref. [36], Busse presents a Kalman filter using CDGPS measurements that achieves very good position and velocity accuracies of $\sim$ 1 cm and $\sim$ 0.5 mm/sec velocity, respectively. These results meet neither the balance nor the correlation requirements, with $\sigma_{\dot{y}}$ approximately twenty times larger than prescribed by the balance requirement, and the correlation coefficient is roughly -0.1. This suggests a discrepancy between the goal of having high correlation and and the Kalman filter output reported in Ref. [36], which is by some definition the "best" filter for this problem. Note that extensive tuning of Q and R was done by Busse and in initial stages of this current work to produce the best possible estimate.

First, it must be determined whether a Kalman filter, by definition giving the best estimates of relative position and velocity [23], produces the best estimate of semimajor axis difference. The relative semimajor axis is a linear combination of radial position and in-track velocity, as shown in Eq. 3.10. The nominal state in Eq. 3.11 can be transformed into a state that explicitly includes the semimajor axis,

$$\mathbf{x}_t = \begin{bmatrix} \Delta a \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 2/n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \equiv T\mathbf{x}_n$$

where T is the transformation matrix between the nominal state and the transformed

state. Because T is full rank, its inverse exists. The state estimation error is defined as $\tilde{\mathbf{x}} = \hat{\mathbf{x}} - \mathbf{x}$, where $\hat{\mathbf{x}}$ is the state estimate. The estimation objective is related to the magnitude of the estimation error, which can be transformed as follows

$$\begin{aligned} \tilde{\mathbf{x}}_t^T \tilde{\mathbf{x}}_t &= (\hat{\mathbf{x}}_t - \mathbf{x}_t)^T (\hat{\mathbf{x}}_t - \mathbf{x}_t) \\ &= (\hat{\mathbf{x}}_n - \mathbf{x}_n)^T T^T T (\hat{\mathbf{x}}_n - \mathbf{x}_n) = \tilde{\mathbf{x}}_n^T S \tilde{\mathbf{x}}_n \end{aligned} \tag{3.17}$$

where $S = T^T T \geq 0$. From [23], the optimal estimate for the state $\mathbf{x}$ is found by minimizing the cost function

$$J = \mathrm{E}\left[\tilde{\mathbf{x}}^T M\, \tilde{\mathbf{x}}\right] \tag{3.18}$$

where M is any positive semidefinite matrix. The key point is that Ref. [23] shows that the optimal estimate is *independent* of the choice of $M \geq 0$. Since we can choose $M = I$, or $M = S$, as in Eq. 3.17, then the optimal estimate for $\mathbf{x}_t$ will be related to the optimal estimate for $\mathbf{x}_n$ by the linear transformation T. Thus, a Kalman filter estimating relative position and velocity *necessarily* will also yield the best possible estimate of the semimajor axis, to within the error associated with the linearization.

Proceeding with this confidence in the Kalman filter output, the issue remaining is that the output does not fulfill the balance and correlation requirements. Note that if σ_x, $\sigma_{\dot{y}}$ and $\rho_{x\dot{y}}$ could be adjusted independently, the conditions for perfect semimajor axis knowledge could be met. However, there is no mechanism in the Kalman filter for adjusting these elements of the covariance matrix independently. In this case, it is questionable whether the best strategy for achieving the lowest semimajor axis error is to require high correlation and good balance. In fact, the examples presented in the following sections, as well as experience with the CDGPS filter, suggest otherwise.

3.3 A Linear Planar Model

If a Kalman filter is producing the best estimate of the semimajor axis, but does not meet the balance and correlation requirements, the next question is to address whether the filter can be forced to meet the requirements by adjusting the input parameters

Table 3.1: Kalman Filter Elements for LPM Simulations

Description	Notation	Simulation Treatment
Dynamics Model	A	Planar Hill's Equations, constant in simulations
Measurements Model	H	Direct measures of position; GDOP varied by changing number and direction of measurements
Discrete Time Step	Δt	Varied
Process Noise	Q	Varied, to highlight effects of mis-modeled dynamics
Measurement Noise	R	Varied, to change quality of measurements

and whether this will improve semimajor axis knowledge. This investigation into the relationship between design parameters and navigation accuracy also leads directly to an understanding of what would be required to improve navigation.

Many parameters specified for a Kalman filter will affect the accuracy of the estimates. Forces not captured in the dynamics model appear as process noise. Likewise, errors and nonlinearities in the sensors will affect the measurement noise. The choices of dynamics and measurement models, the operating environment, and the extent to which nonlinearities are accentuated may all affect the Q and R defined for the filter. The set of sensor data made available to the filter will determine the measurement matrix, H. Also, the time step can affect the performance of the discrete filter.

A CDGPS navigation filter has nonlinearities in both the system dynamics and the measurement equations. Because the set of visible GPS satellites changes, the measurement matrix H will change, and the state vector length will grow or shrink as the set of estimated carrier biases change. These factors make it difficult to understand direct relationships between the filter parameters and the navigational accuracies. To begin to develop these relationships, a simplified linear example is investigated first.

This Linear Planar Model (LPM) was developed to provide insight into the behavior of a relative navigation filter using CDGPS. Table 3.1 summarizes the Kalman filter parameters used in the LPM simulations. The system dynamics in this example

are taken from the solutions of Hill's equations for radial and in-track position and velocity. Out-of-plane motion is ignored because it does not affect semimajor axis error. The system dynamics model was not varied in the simulations although its effective accuracy was varied by evaluating filter performance with different values of Q. The LPM state is defined by the relative LVLH positions and velocities in the x and y directions

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T \tag{3.19}$$

and is governed by the dynamics model,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \tag{3.20}$$

Although GPS can have as many as twelve position-related measurements, the LPM includes two or more direct measures of position that span the orbital plane. Variations of the measurement model included changing the angle between the two position measurements and increasing the number of measurements included. Also the level of noise associated with one of the two measurements was changed. The H matrix for a simple example with position measurements in both the x and y directions is

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.21}$$

Different sizes of the discrete time step, Δt, were considered in these simulations. As the time step changes, the relative importance of the dynamics model and the measurements changes. For example, if highly accurate measurements were provided at a very fast rate, the dynamics model might be of little importance. Conversely, a perfect propagator would eventually obfuscate the need for measurements at all. Changing the time step illuminates how the filter might be tilted towards one of these extremes.

The simplified LPM problem is intended to show how each parameter in the problem affects the radial position error and in-track velocity error, the correlation of the two, and ultimately, the semimajor axis error. For each design variation, the position and velocity variances are found by numerically solving a discrete algebraic Riccati equation. The discrete Riccati equation is discussed in detail in Ref. [37], but it is significantly more complicated to analyze than the continuous form shown here

$$\mathbf{0} = AP + PA^T + Q - PH^T R^{-1} HP \tag{3.22}$$

The position and velocity error variances from Eq. 3.22 are used in Eq. 3.10 to compute the corresponding semimajor axis error.

Before examples are done to discuss the changes necessary to force the correlation to -1, a general understanding of which parameters would influence the correlation is required. Preliminary simulations revealed changing Q and R would drive $\rho_{x\dot{y}}$ towards -1 only when the changes in Q and R were extreme and unrealistic. Thus three other types of changes were chosen to explore the relationship between $\rho_{x\dot{y}}$ and several key filter parameters:

1. Degrade one of two measurements
2. Bring two measurements into alignment
3. Observe effects of reducing number of measurements

Section 3.3.1 discusses LPM simulations that develop these three examples. These cases show that forcing $\rho_{x\dot{y}}$ to -1 results in position and velocity variances that are degraded and do not fulfill the balance requirement. Since the balance requirement is not met and the variances are higher, the resulting semimajor axis error is actually worse. This validates the previous analysis that the output of the Kalman filter results in the best possible knowledge of semimajor axis, and indicates that $\rho_{x\dot{y}} = -1$ not an important goal in the filter design.

3.3.1 Three LPM Examples for Correlation Demonstrations

This section presents three LPM measurement models that illustrate what is required to force $\rho_{x\dot{y}} = -1$, and the results show that the performance of the navigation filter always degrades. The baseline behavior of the LPM model was investigated to gain insight into how to influence the correlation between the radial position and in-track velocity. When two measurements spanning the plane are provided to the Kalman filter, correlations between position and velocity in the same direction, $\rho_{x\dot{x}}$ and $\rho_{y\dot{y}}$, are high. This shows that the velocity estimate in a given direction is derived from the corresponding position estimate. Conversely, the correlation between position and velocity in the different directions, $\rho_{x\dot{y}}$ and $\rho_{y\dot{x}}$ are very low. The low correlation between radial position and in-track velocity, two quantities that are coupled in the dynamics model, shows the estimate is determined by measurements and is less dependent on the dynamics model. Despite the low correlation, the position and velocity errors are very small, so the corresponding semimajor axis error is still low.

It is reasonable to expect the correlation between radial position and in-track velocity will increase only when the estimate depends more on the dynamics model embedded in the filter. This behavior is induced in examples that:

1. **Degrade one of two measurements:** two measurement directions, shown in Fig. 3–2, were chosen to provide good coverage. One of the two measurements is degraded by increasing the noise level to the point where it is unusable by the filter. This essentially causes the term $H^T R^{-1} H$ in the Kalman filter to drop rank. This means that the measurements will no longer span the full orbital plane, and the dynamics model must be employed to create the estimates of both states.

2. **Bring two measurements into alignment:** the angle between the two measurements is decreased from 90° to 0°, as in Fig. 3–3. This has an effect similar to the first example, in that the direct observability of the orbital plane decreases as the two measurements become aligned.

3. **Vary the number of measurements:** this example begins with 6 measure-

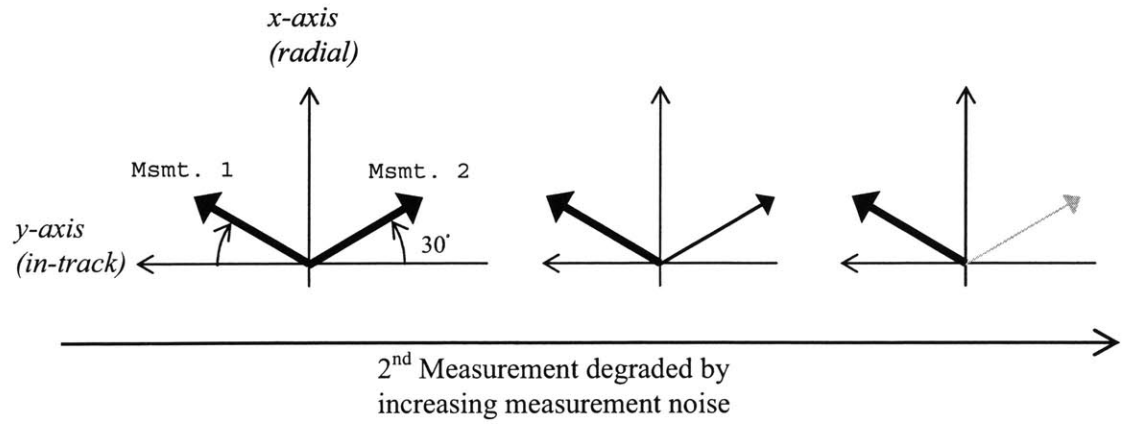

Fig. 3–2: LPM Example #1 – Filter performance as second measurement is degraded.

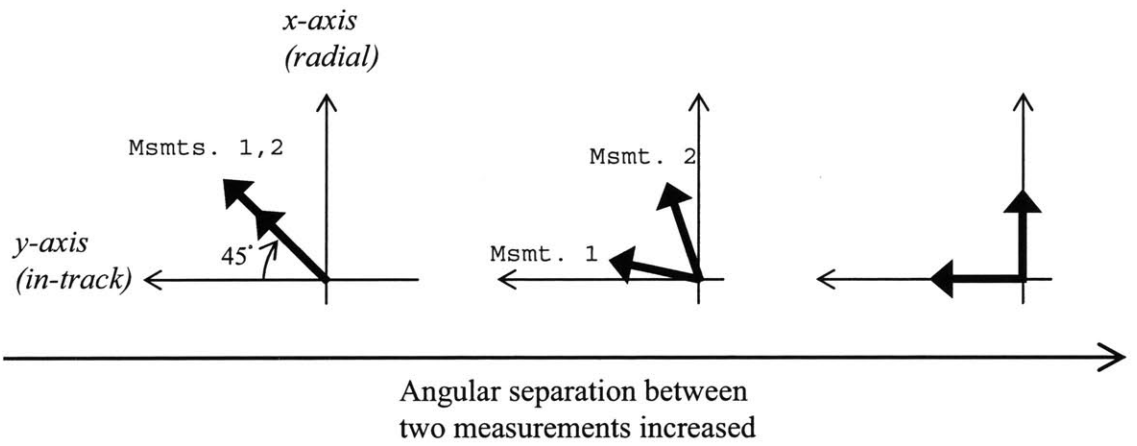

Fig. 3–3: LPM Example #2 – Filter performance as two measurements are brought into alignment.

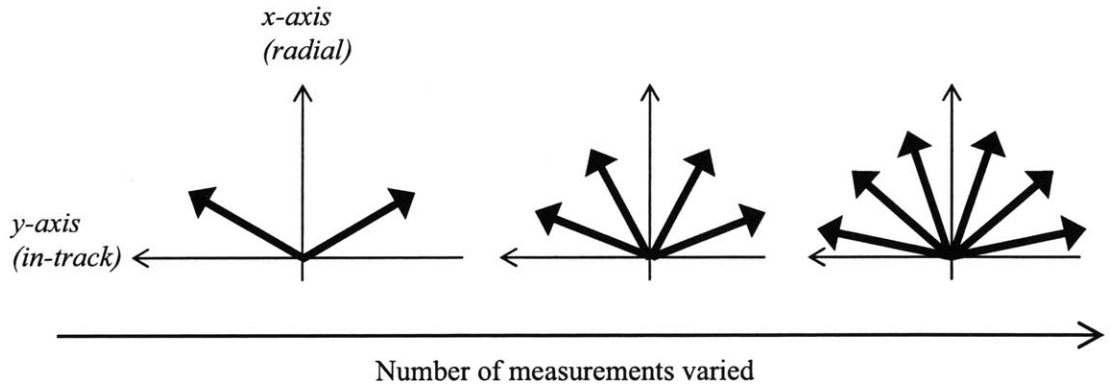

Fig. 3–4: LPM Example #3 – Filter performance as the number of measurements is varied.

ments, and then decreases the number of measurements available. This will show that the correlation becomes weaker, yet semimajor axis knowledge improves when more information, in the form of measurements, is given to the filter.

The three examples show how the measurements directly affect the error covariance matrix of the Kalman filter, and determine whether the balance and covariance requirements can be met.

EXAMPLE #1: In the first example, one of the two measurements was degraded by increasing its measurement noise well above realistic ranges, forcing the filter to depend on the other measurement. The remaining measurement does not span the orbital plane, so the filter is forced to use the dynamics model to produce the full state estimates.

Fig. 3–5 shows the values of the three parameters in Eq. 3.10, σ_x, $\sigma_{\dot{y}}$, $\rho_{x\dot{y}}$, balance index, and the resulting $\sigma_{\Delta a}$, as the second measurement degrades. The results show that as the measurement quality is reduced, the correlation $\rho_{x\dot{y}}$ tends toward -1, but σ_x and $\sigma_{\dot{y}}$ do not satisfy the balance requirement. Because the balance and correlation requirements are not met simultaneously, the increases in the position and velocity errors significantly outweigh any benefit gained from increasing the correlation. The resulting semimajor axis error is actually worse when the system is driven to have stronger correlation.

EXAMPLE #2: A variant of Example #1 provides another view of how the span of the measurement matrix affects the correlation and semimajor axis error. This example has the two position measurements of equal accuracy, that are initially along the x and y axes. The angle between the two measurements is gradually decreased until the measurement directions are nearly collinear at a 45° angle in the $x - y$ plane. Fig. 3–6 shows that the correlation between the two remains low until the angular separation is less than 20°. When the angular separation becomes smaller than a degree, the correlation drops sharply from -0.7 towards -1. As the measurements approach alignment, the uniqueness of the information that each contributes

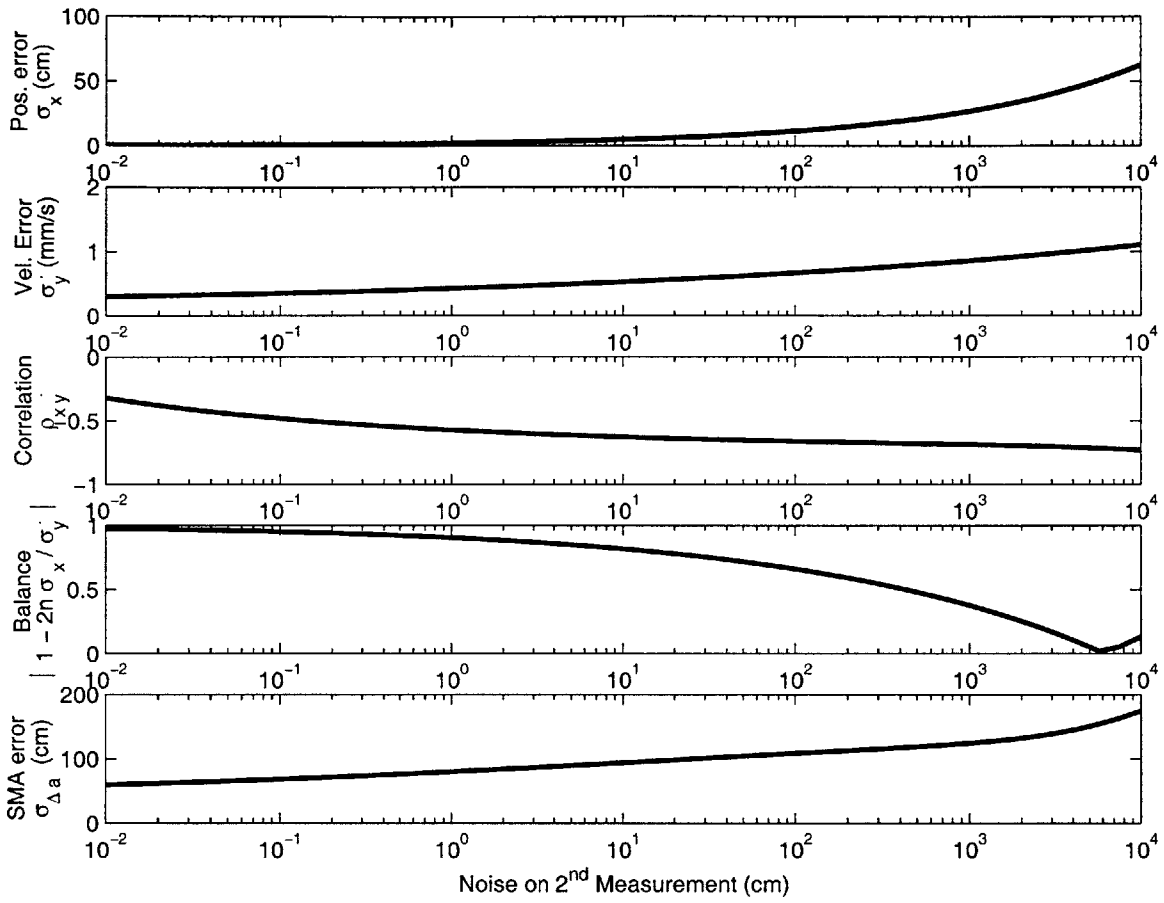

Fig. 3–5: LPM Example #1 – Filter performance as second measurement is degraded.

decreases. As the two become nearly collinear, the information provided becomes redundant. The resulting correlations quickly approach -1 as the dynamics become more important. Similar to the previous case, when the measurements no longer span the orbital plane, the position and velocity errors increase dramatically. Again, since the balance requirement is not met exactly, the increased errors in position and velocity outweigh the advantage of the increased correlation, and the final semimajor axis error is large. In the example shown, the two measurement directions converged on 45°, but similar results were seen for other terminal angles.

EXAMPLE #3: A final iteration of the simplified example looks at the effect of increasing the number of measurements from one to six. The measurements all have

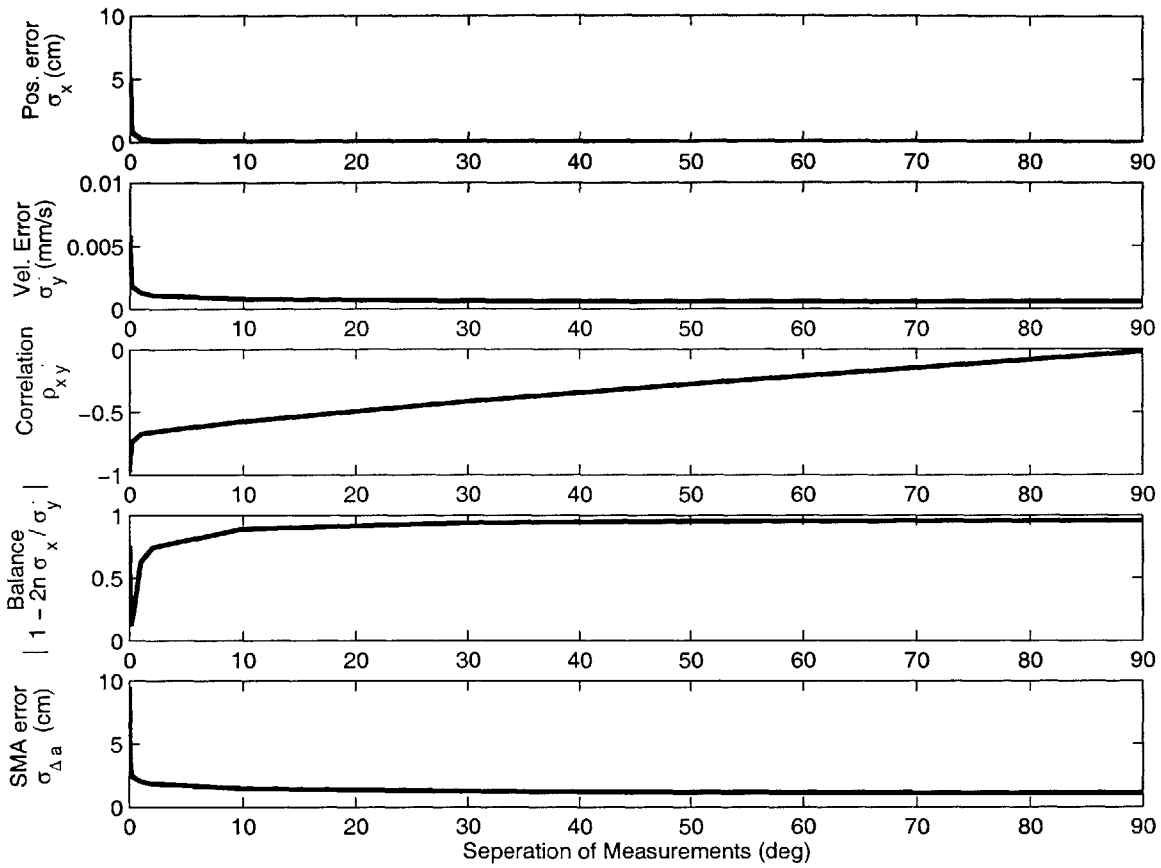

Fig. 3–6: LPM Example #2 – Filter performance versus angular separation

reasonable noise levels; none are degraded to the point they are not usable. They are equally spaced between 0° and 90° in the radial/in-track plane. The trends in this example, shown in Fig. 3–7, are consistent with the previous results. The values of σ_x and $\sigma_{\dot{y}}$ drop sharply, $\rho_{x\dot{y}}$ changes from close to -1 to around -0.5, and the balance index increases when the number of measurements is increased from one to two. When the second measurement is added, the correlation and balance index are further from meeting requirements, but the semimajor axis knowledge improves. The results show that as additional measurements are added, the position and velocity errors continue to improve. Though the degree of correlation decreases, the semimajor axis error improves when more information is used.

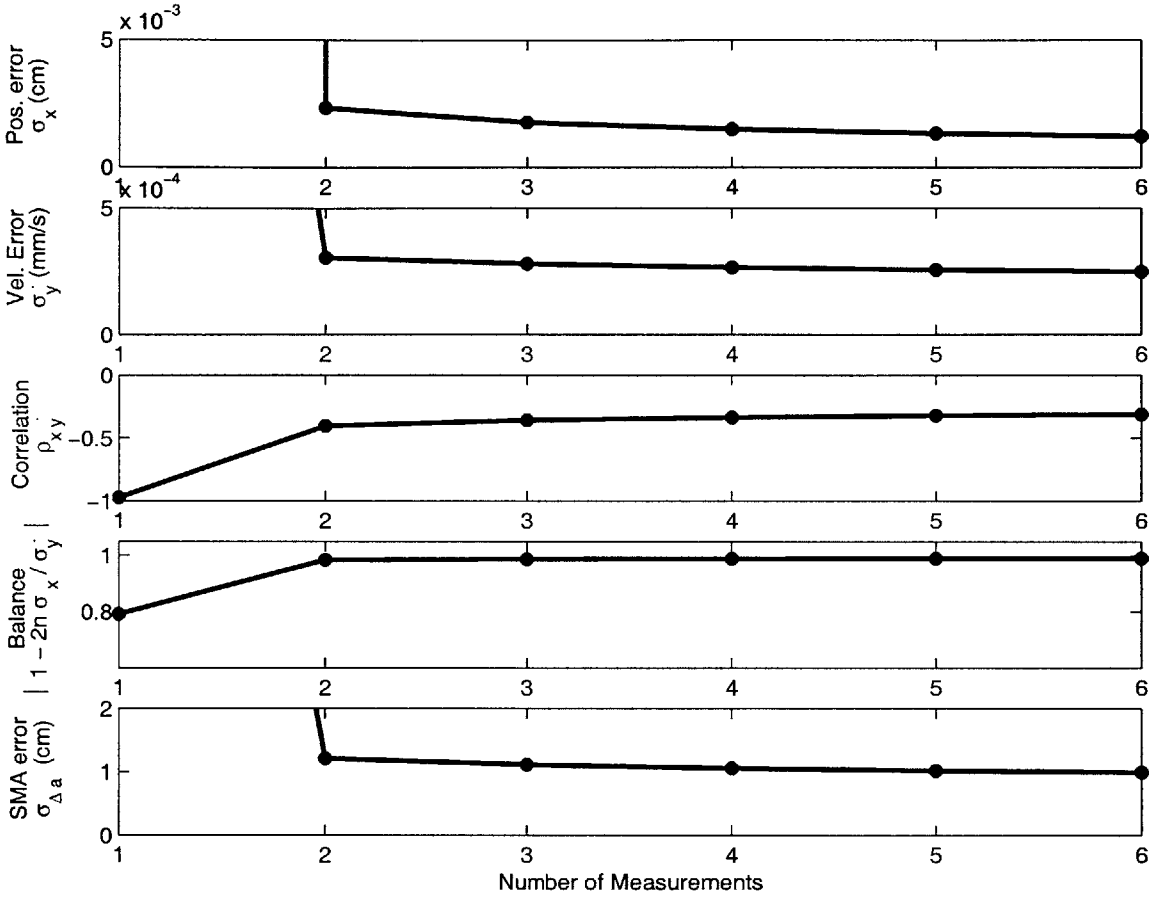

Fig. 3–7: LPM Example #3 – Filter performance versus number measurement

3.3.2 Discussion of Three LPM Examples

In the three examples, the filter can be made to rely on its dynamics model to extract good state information in the radial and in-track directions. This is accomplished by degrading the available sensing information. The increases in σ_x and $\sigma_{\dot{y}}$ that accompany the higher correlation consistently result in a larger semimajor axis error. This is a trait of a measurement driven filter, not a symptom of a deficient one. When the correlation between radial position and in-track velocity is low, the best strategy for determining relative semimajor axis is to estimate both quantities with the highest possible accuracy. The Kalman filter does this.

The behavior seen in the three variations of this simplified example provide insight

into the relative navigation system using CDGPS measurements. A GPS-based system will have many position related measurements available (around ten to twelve) that will be used by the filter. The GPS filter can also be forced to have better correlations by degrading the measurements and boosting the importance of the dynamics model. However, this results in increased semimajor axis error and is a sub-optimal strategy. Now that the LPM examples have shown that efforts to improve the navigation filter should not focus on $\rho_{x\dot{y}}$, the next section will transition to a full Nonlinear GPS Model (NGM) to explore what aspects of the filter contribute to the navigation accuracy.

3.3.3 LPM Simulations with Q and R

This section addresses the relationship between the Kalman filter parameters and the resulting estimate accuracy. These results are based on the premise that the Kalman filter produces the best answer from the given the models and measurements, and that the balance or correlation between elements of the state vector estimate are not important. Initial investigations showed that, predictably, the levels of measurement and process noise have the most influence of the estimate accuracy. These levels are indicators of how well the sensors and the dynamics are modeled. The relative levels of these noises determines how the filter will weigh new measurement information against the state propagated from the dynamics information.

Meeting the balance and correlation requirements discussed in Section 3.2 corresponds to being on the "bump" in Fig. 3–1, but the baseline results of the LMP simulation were not on this bump. This leads to the question of how changing filter inputs will move the output closer to or further from the bump. To answer this question, the LPM simulation was run for a range of measurement and process noise levels.

For each unique assignment of Q and R, the resulting error variances for radial position and in-track velocity, σ_x and $\sigma_{\dot{y}}$, were recorded. The corresponding semimajor axis error, $\sigma_{\Delta a}$, was calculated using Eq. 3.10. Fig. 3–8 shows lines of constant Q and R on axes of σ_x and $\sigma_{\dot{y}}$. The diagonal line on this graph indicates the location of

the "bump", where the balance and correlation requirements are met (which means $\sigma_{\dot{y}} = 2n\sigma_x$). By moving from one line of constant Q or R to another, one can see how decreasing the process or measurement noise would change the resulting position and velocity error. However, semimajor axis knowledge, not just position and velocity knowledge alone, is important for control performance.

Several graphs demonstrate the relationship between Q, R and $\sigma_{\Delta a}$. First, Fig. 3–9 is reproduced and lines of constant semimajor axis error added. The lines for constant Q and R are dimmed for clarity. Note the lines of constant semimajor axis are horizontal, which corresponds with the horizontal sections of the semimajor axis contours on Fig. 3–1. The effect of changes in Q and R on $\sigma_{\Delta a}$ can be assessed by looking at the constant lines for all three values. Because the lines of constant semimajor axis are horizontal, improvement in $\sigma_{\Delta a}$ can only be accomplished by moving in the vertical direction on the graph, which is equivalent to decreasing $\sigma_{\dot{y}}$. Whether this requires decreasing Q or R depends on the angles between the horizontal lines of constant $\sigma_{\Delta a}$ and the contours of constant Q and R.

Two regions are also indicated in Fig. 3–10. **Region 1** (upper left hand portion of the graph) contains lines of constant Q and R that are essentially horizontal and vertical, respectively. Improving the measurement noise in this region, or essentially moving horizontally in the graph, would have no effect on semimajor axis knowledge. Decreasing $\sigma_{\Delta a}$ would require improving the process noise to enable vertical movement on the graph.

Region 2 is closer to the line of $\sigma_{\dot{y}} = 2n\sigma_x$. Here, the lines of constant Q and R are no longer parallel and perpendicular to the horizontal lines of constant semimajor axis. This means that reducing either Q or R could improve $\sigma_{\Delta a}$. This shift from Region 1, where sensing improvement has no effect on $\sigma_{\Delta a}$, to Region 2, where it does, is also shown in Fig. 3–11 by plotting contours of constant semimajor axis on axis of Q and R. As expected, Region 1 has horizontal lines of constant $\sigma_{\Delta a}$, which directly shows that decreasing the measurement noise has little effect on semimajor axis knowledge. The portion of the graph where the contours of constant $\sigma_{\Delta a}$ are at an angle corresponds to Region 2. In this region, improvements in the measurements

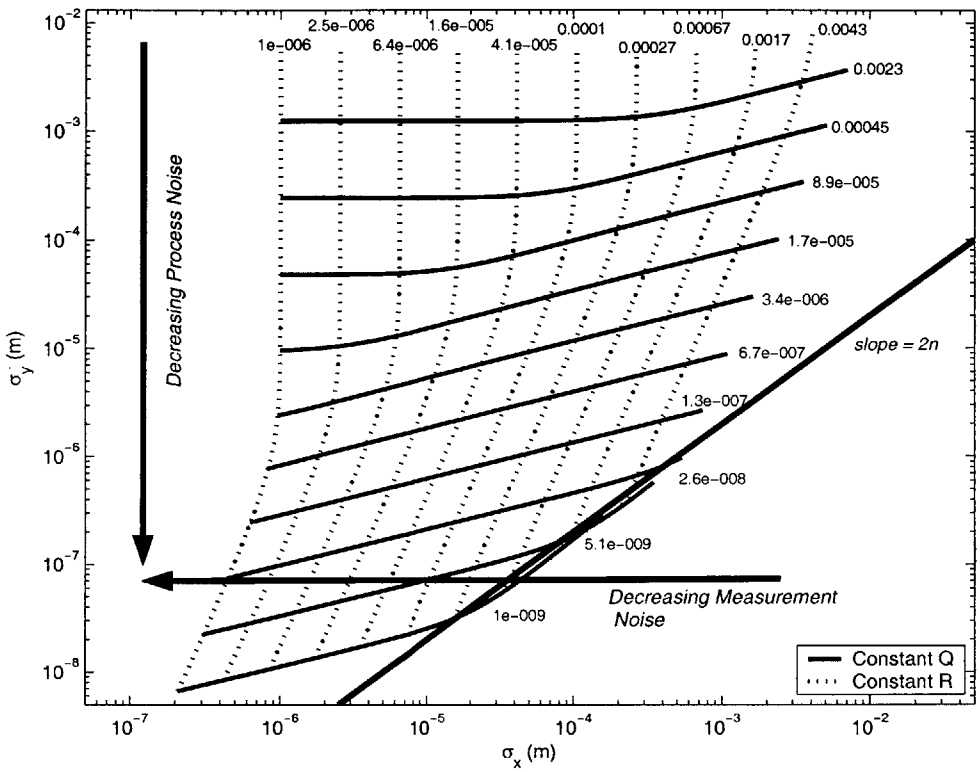

Fig. 3–8: Contours of constant Q and R are shown on axes of position and velocity accuracy.

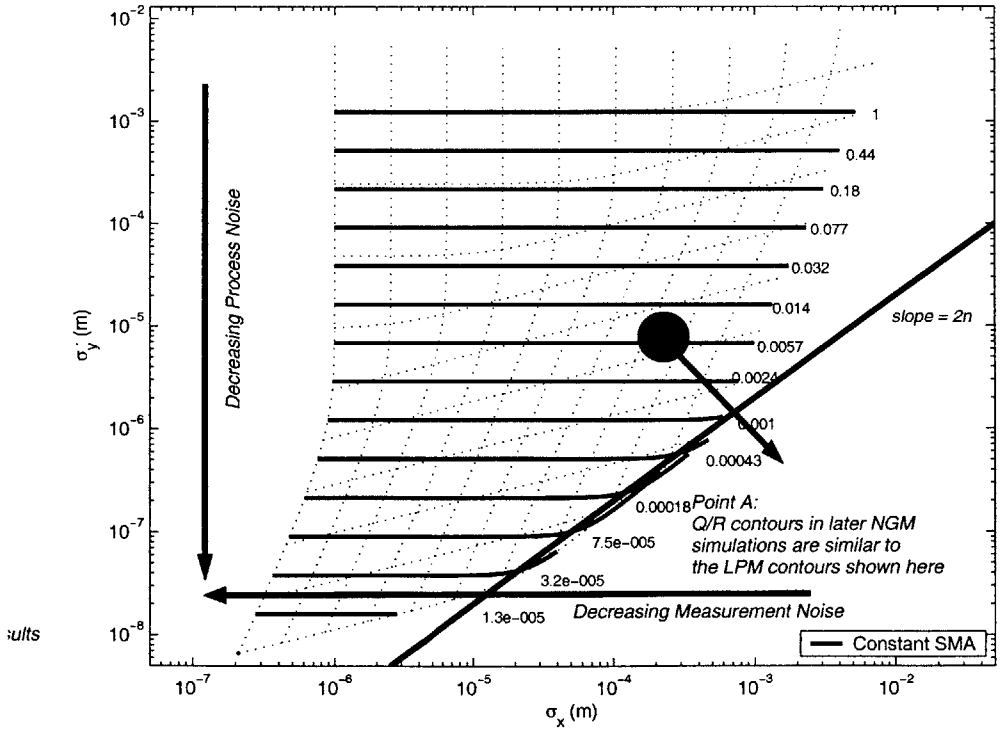

Fig. 3–9: Contours of constant semimajor axis are shown on axes of position and velocity accuracy. Contours of constant Q and R are shown as dotted lines in the background. Point A is a reference for a NGM discussion.

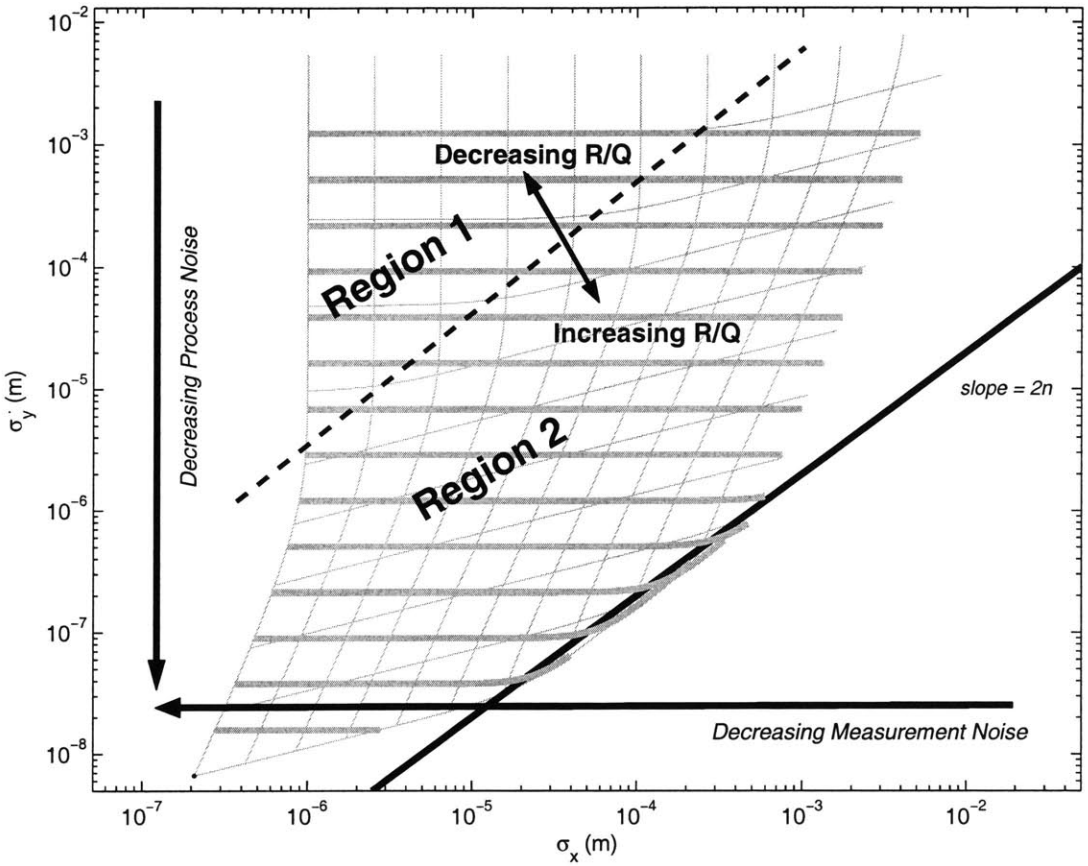

Fig. 3–10: In Region 1, decreasing the semimajor axis error requires decreasing the process noise level. In Region 2, decreasing either process or measurement noise can reduce semimajor axis error. Contours of constant Q, R, and semimajor axis, shown more clearly in Figs. 3–8 and 3–9, are in the background.

or the dynamics models (*i.e.* reduced process noise) would contribute to improved semimajor axis knowledge.

The LPM results shown thus far occupy the region above the $\sigma_{\dot{y}} = 2n\sigma_x$ line. Three questions must be addressed:

1. What features of the system cause this behavior?
2. Does this agree with expected behavior?
3. What is required to cause the LPM results to move closer to this line, and below the line?

In Region 1, $\sigma_{\Delta a}$ improvement requires decreasing $\sigma_{\dot{y}}$. This means that improvements in σ_x in this region are have no effect on semimajor axis knowledge. In Region 1,

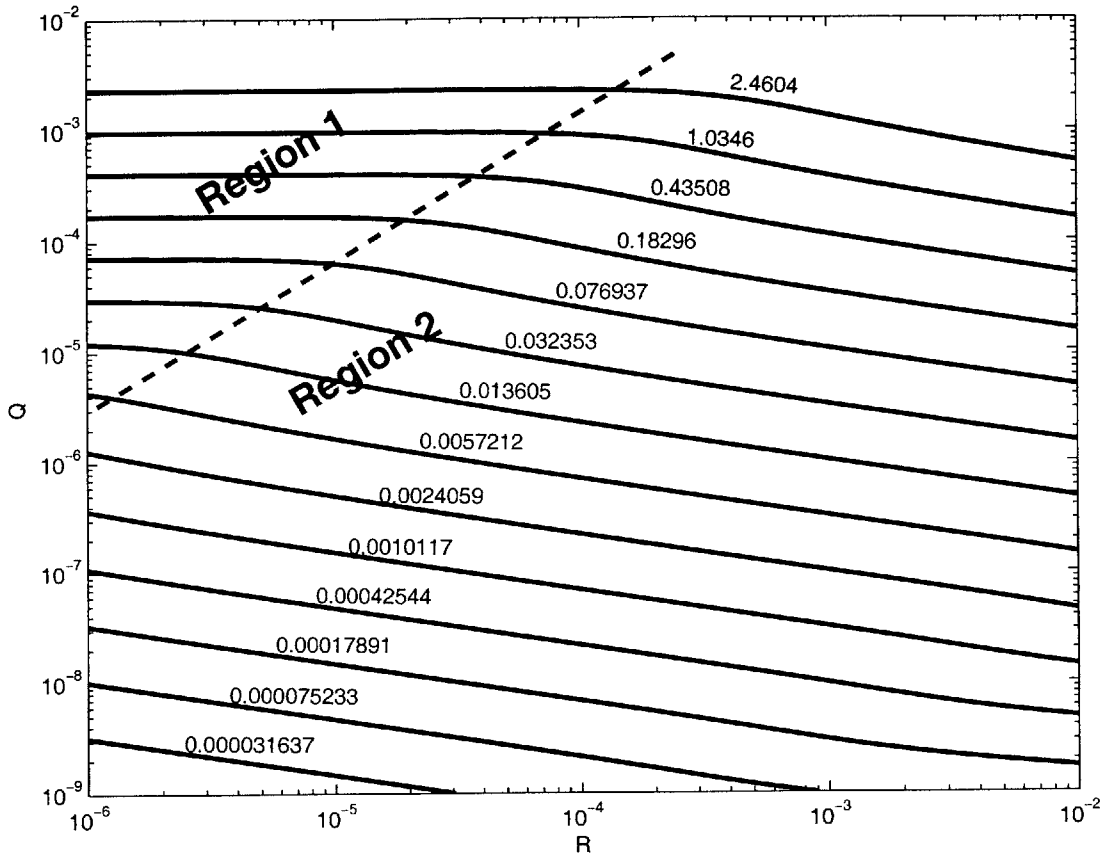

Fig. 3–11: Contours of constant semimajor axis are shown against axis of Q and R. In Region 1, decreasing the semimajor axis error requires decreasing the process noise level. In Region 2, decreasing either process or measurement noise can reduce semimajor axis error.

the position error is smaller than the balance requirement dictates. This might be viewed as "excess" position accuracy with respect to determining semimajor axis. It is useful to consider what is driving the position and velocity accuracy in the GPS problem, and the representative LPM problem. Both models proposed a navigation system based on position measurements. In the LPM, the accuracy of the position estimate is then driven by the measurement noise. With no velocity measurement, the filter must obtain a velocity estimate from the derivative of the position estimate. Since this is tied to the dynamics model, it is reasonable to see velocity accuracy more closely associated with the process noise.

These LPM trends are representative of the behavior expected from a CDGPS-

based navigation system. In the GPS problem, velocity sensing is based on Doppler measurements created in the receiver. The receiver creates the Doppler measurements by differencing the carrier phase measurements. The receiver does not model orbital dynamics in the receiver. Thus, the velocity estimate that uses the Kalman filter's orbital dynamics model should be more accurate. Adding the receiver based Doppler measurements adds virtually no information, as reported in Ref. [1].

Though a navigation filter with only GPS measurements is always expected to have "excess" position accuracy and occupy the region above the $\sigma_{\dot{y}} = 2n\sigma_x$ line, LPM simulations can be forced to produce estimates that fall below this line. As expected, this requires adding velocity measurements to the filter. As the noise on the velocity measurement is decreased and the process noise is increased, the contours of constant semimajor axis begin to turn a corner, as in Fig. 3–1. If the velocity measurement and process noises are exaggerated, by decreasing velocity noise and increasing process noise drastically, the contours transition to a vertical orientation. This change, illustrated in Fig. 3–12, is addressed only to highlight why this will never be seen in the GPS-only navigation system.

3.4 Agreement with Analytical Work

The LPM simulations discussed in the previous sections provided insight into the relationship between errors in position, velocity and semimajor axis, the correlation between position and velocity, and the levels of process and measurement noise seen by the system. These models were used to confirm observations that high correlation is not necessary to produce very accurate estimates in a CDGPS-based Kalman filter. The LPM was forced to have high correlation in several examples, but in each case, the resulting semimajor axis error *increased*, not decreased. The LPM simulations showed:

- Increasing correlation did not result in improved semimajor axis knowledge
- Semimajor axis error was strongly tied to in-track velocity error. This corresponds to being above the balance line in Fig. 3–1, which is expected since the

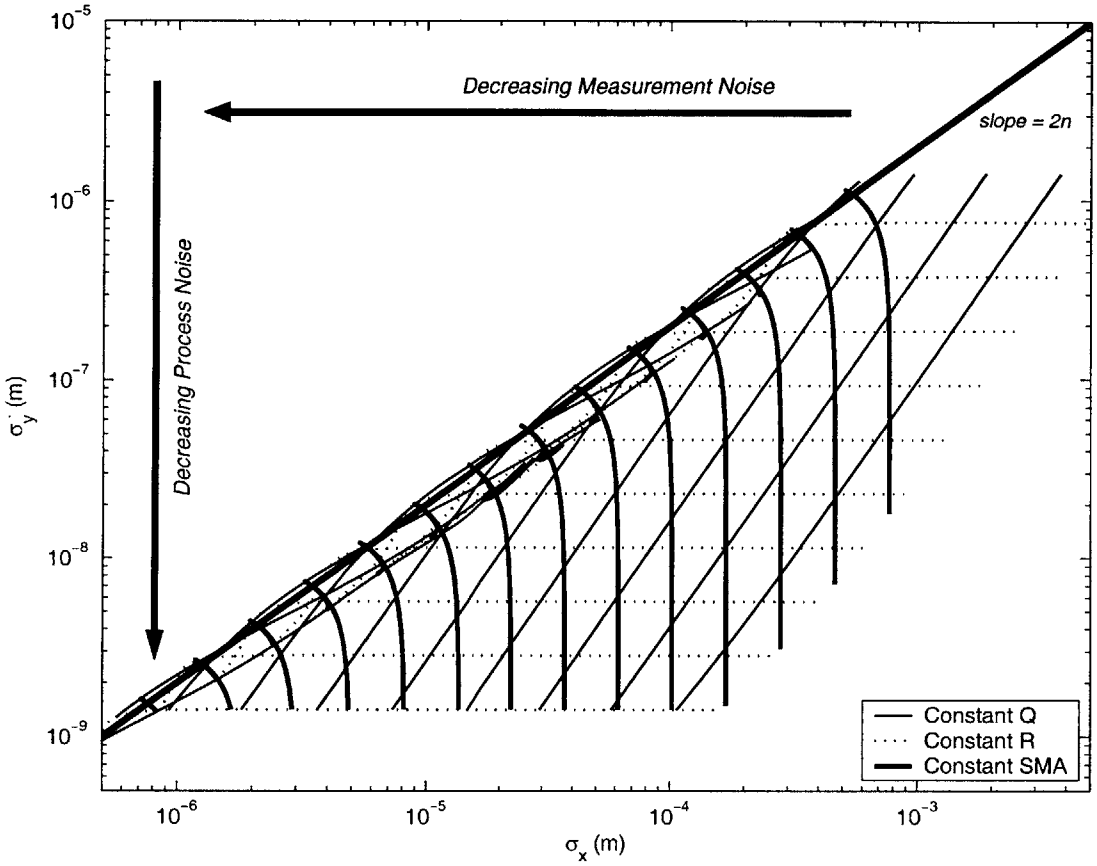

Fig. 3–12: Contours of constant Q, R, and semimajor axis are shown on axes of position and velocity error. Here, the LPM has been given extremely good velocity measurements, and the position measurements have been severely degraded. This was done to show what is required to occupy the space under the diagonal balance line

LPM and NGM rely on position measurements.

- Process and measurement noise strongly influenced correlation. If the ratio R/Q is increased, this effectively allows the filter to trust the dynamics model and ignore the measurements, which naturally leads to higher correlation.
- Process and measurement noise levels also strongly influenced semimajor axis error.

Recent analytical work by Professor Alfriend at Texas A&M University confirms these findings [38]. The analysis, summarized below, develops analytical relationships between the process and measurement noise levels and the errors and correlations.

The link between the two is enabled through the algebraic Riccati equation. The matrix Riccati equation has no analytic solution for anything more complicated than a double integrator. However, for a Kalman filter with a time step, $\Delta t \approx 1$ second for a 90 minute orbit, the coupling between motion in the x and y directions is very weak and the dynamics can be well approximated as two weakly coupled double integrators. More formally, by defining $\epsilon = n\Delta t$, Hill's equations from Eq. 2.35 becomes

$$x'' = 2\epsilon y' + 3\epsilon^2 x + (\Delta t)^2 f_x \tag{3.23}$$

$$y'' = -2\epsilon x' + (\Delta t)^2 f_y \tag{3.24}$$

where $(*)' = (*)\Delta t$. When $\epsilon \ll 1$, which is true for this application, the x and y dynamics can be written as double integrators for which the solution of the Riccati equation is easily found.

The combined dynamics then consist of two double integrators with coupling terms of order ϵ and ϵ^2. The covariance for the full state is represented as Taylor expansion in ϵ. By substituting the expressions for P and A in the Riccati equation (Eq. 3.22) and grouping terms to the same power in ϵ, it is possible to solve for the coupling terms in the covariance matrix [38]. This gives the P_{xx}, $P_{\dot{y}\dot{y}}$, and $P_{x\dot{y}}$ entries of the covariance matrix, which leads to expressions for the correlation coefficient between the x and $\dot{y}$ estimates and the variances for the semimajor axis. The full analysis in Ref. [38] then shows that the resulting correlation coefficient can be written as

$$\rho_{x\dot{y}} = -n\sqrt{\frac{\sigma_R}{\sigma_Q}} \tag{3.25}$$

which, upon substitution in Eq. 3.10, gives a semimajor axis error of

$$\sigma_{\Delta a} = \frac{2}{n}\,\sigma_{\dot{y}} \tag{3.26}$$

$$= \frac{2^{\frac{5}{4}}}{n}\,\sigma_Q^{\frac{3}{4}}\sigma_R^{\frac{1}{4}} \tag{3.27}$$

The variances σ_Q and σ_R refer to the standard deviation of the process and measurement noises. The process noise is assumed to be the same for all states and the measurement noise is assumed to be the same for all measurements. Eqs. 3.25 – 3.27 predict the correlation and semimajor axis error variance that can be expected from a Kalman filter. Since these predictions were developed from model based on two double integrators, the results may not immediately map to the full CDGPS problem. However, these equations should agree with LPM simulations.

Prior to applying these equations, it is prudent to verify the ranges of σ_Q and σ_R for which the equation is valid. The correlation coefficient, ρ_{xy}, has a limit of -1. Now

$$-n\sqrt{\frac{\sigma_R}{\sigma_Q}} \geq -1 \tag{3.28}$$

implies

$$\frac{\sigma_Q}{\sigma_R} \geq n^2 \tag{3.29}$$

must be true. In Low Earth Orbit, where $n \approx 0.001$, Eq. 3.29 implies that σ_R must be no more than six orders of magnitude larger than σ_Q. With this insight, the LPM model was evaluated for a wide range of σ_Q and σ_R, and the "true" semimajor axis variance was calculated using Eq. 3.10. This true value was compared to both forms of the prediction error given in Eqs. 3.26 and 3.27. The true correlation was calculated using the covariance matrix from the Riccati solver, as defined in Eq. 3.12. The performance of the two forms of the semimajor axis prediction equations as well as the correlation prediction equation is shown in Fig. 3–13. The three subplots show the percentage errors for the two forms of the semimajor axis and for the correlation,

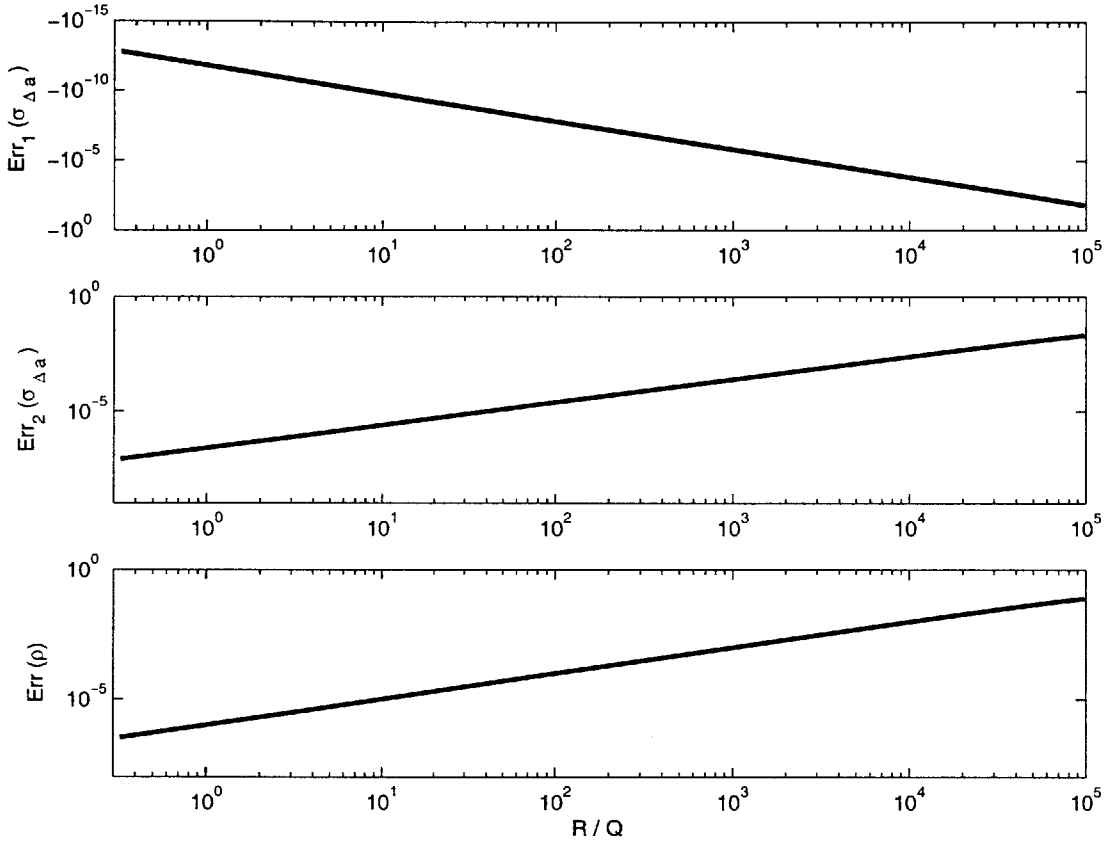

Fig. 3–13: Predictions for Semimajor Axis and Correlation (shows percentage errors.)

defined as

$$\begin{aligned} \mathrm{Err}_1(\sigma_{\Delta a}) &= \frac{\sigma_{\Delta a\mathrm{true}} - \sigma_{\Delta a\mathrm{pred1}}}{\sigma_{\Delta a\mathrm{true}}} \\ \mathrm{Err}_2(\sigma_{\Delta a}) &= \frac{\sigma_{\Delta a\mathrm{true}} - \sigma_{\Delta a\mathrm{pred2}}}{\sigma_{\Delta a\mathrm{true}}} \\ \mathrm{Err}(\rho) &= \frac{\rho_{\mathrm{true}} - \rho_{\mathrm{pred}}}{\sigma_{\Delta a\mathrm{true}}} \end{aligned}$$

$$\sigma_{\Delta a\mathrm{pred1}} = 2\frac{\sigma_{\dot{y}}}{n}, \qquad \sigma_{\Delta a\mathrm{pred2}} = 2^{\frac{5}{4}}\frac{\sigma_Q^{\frac{3}{4}}\sigma_R^{\frac{1}{4}}}{n}, \qquad \rho_{\mathrm{pred}} = -n\sqrt{\frac{\sigma_R}{\sigma_Q}}$$

The percentage errors for all three comparisons are very small, indicating very good agreement between the prediction equations and LPM simulation results. Also, each of the errors grows linearly with σ_R/σ_Q, and therefore also with $\rho_{x\dot{y}}$. The linearity seen in the percentage errors is especially encouraging, because the plots represent

simulations run with wide ranges of σ_Q and σ_R.

Note that Eq. 3.25 requires that the process noise be a million times smaller than the sensor noise. Many space-rated GPS receivers produce differential carrier phases measurements with millimeters of error. To achieve $\rho_{x\dot{y}} = -1$, the dynamics environment of the vehicle would have to modeled to nanometer-level accuracy. This is not currently possible, and if it were, designers probably would not bother with CDGPS. This interpretation of the relationship between correlation and process and measurement noises concurs with conclusions drawn from the three LPM examples.

3.5 A Nonlinear GPS Model

The LPM simulations provided a baseline of the expected behaviors from the NGM simulations and an indication of what parameters to vary. As in the LPM experiments, the NGM was simulated for an array of values of Q and R. For each set of Q and R, the full GPS simulation was run for 3000 seconds, to allow sufficient time for the filter to converge to steady state. The absolute truth states for the vehicles were propagated to create the simulated GPS measurements and differenced to create the relative state truth. The position and velocity errors were determined from the standard deviation of the estimation error, not from a Riccati equation as in the LPM cases. Note that in the nonlinear case Eq. 3.10 was found to give inaccurate estimates of the semimajor axis error, so the orbital elements for the two vehicles were calculated. The absolute semimajor axes were differenced at each time step and the differential semimajor axis errors were stored. The square root of the variance of these estimation errors (after the filter had converged) was recorded.

The simulation software for the NGM model propagates the orbit and provides simulated measurements to the estimator. Because a large number of simulations was required to observe relationships between filter parameters and the navigation accuracy, it was not practical, or perhaps even possible, to employ hardware-in-the-loop simulations. Generally, the software simulations produce results that are more accurate than the hardware simulations. However, previous and current work has

observed that trends seen in the software simulations are also seen in the hardware simulations.

The NGM simulation tools were originally developed by Busse [1], and were modified as needed for this work. The user has a great deal of control over how nonlinear effects and what perturbations are modeled in the truth propagator and are accounted for in the filter. This is very useful in trying to relate the NGM to the LPM. The following discussion begins with an environment that simulates all of the perturbations and nonlinear effects. Though unrealistic, the initial nonlinear effects are reduced by placing the vehicles 1 meter apart. This minimizes truncation error in the relative orbital dynamics equations. It also causes the line-of-sight vectors to the GPS satellites for the two vehicles to be nearly identical, which is another assumption made by the filter.

Three NGM simulations illustrate the relationship between noises and estimator performance. The goal of the first simulation is to create a model that has the full NGM state, with clock and carrier bias terms included, and that uses simulated GPS measurements, but that also looks as much like the LPM as possible. To accomplish this, many real-world effects that are usually included in software simulations, such as ephemeris error, clock error, communication outages, and measurement cycle slips, are not included. The first simulation is run at a 0.1 second time step. The second simulation, run at a 1 second time step, is intended to be an intermediate step between an unrealistic, nearly linear simulation, and the fully nonlinear simulation that models real world error sources. It also introduced very low levels ephemeris errors and clock errors. The third NGM simulation, also at a 1 second time step, included all available error sources in the models for the environment and sensors.

3.5.1 NGM Simulations with Q and R

The NGM analysis begins with a very simply model that focuses on the relative orbital dynamics and GPS measurements, and takes steps to reduce nonlinearities associated with their equations. Starting with such a simple model has two main advantages. First, the simple NGM model, like the LPM, uncovers the basic relationships between

process and measurement noises and semimajor axis error. Second, starting with the very simple model allows the trends caused by clock, ephemeris, and other extraneous errors to be identified.

The first NGM simulation removes most of the simulated errors to begin with a problem that is most like the LPM. To further reduce the importance of the nonlinearities that the NGM dynamics model introduces, the filter is run at a 0.1 second time step. Fig. 3–14 shows contours of constant Q, R and $\sigma_{\Delta a}$ that are very similar to the LPM. As in the LPM, Region 1 contains lines of constant Q and R that are essentially horizontal and vertical. Improving the measurement noise in this region would have no effect on semimajor axis knowledge. Reductions in process noise are required to reduce $\sigma_{\Delta a}$. In Region 2, improvements in the measurements or the dynamics models would contribute to improved semimajor axis knowledge.

An interesting change was observed in the second simulation, when the time between measurement updates was increased to 1 second. In the first NGM simulation and the LPM simulation, the lines of constant R bent to the left at the bottom. In the second NGM simulation, in Figure 3–15, the lines of constant R bend to the right before turning back left. This would mean that improving process noise and keeping the measurement noise constant results in improved velocity accuracy, but causes the position accuracy to actually degrade. This change reflects the increased effect of the dynamics nonlinearities.

However, the second simulation retains the major trends of the first NGM simulation and the LPM. In Region 1, constant measurement noise contours are vertical, and constant contours for process noise and semimajor axis are horizontal. This means that process noise reductions are required to improve semimajor axis knowledge, and measurement noise reductions have no impact. In Region 2, the lines of constant Q are similar to the LPM model, starting horizontal and sloping upwards slightly near the line of $\sigma_{\dot{y}} = 2n\sigma_x$. The tops of the contours of R share the vertical orientation seen in the LPM, though, as discussed above, nonlinear effects begin to reshape the bottom sections of the contours.

The final NGM simulation includes errors introduced by clock, ephemeris, and

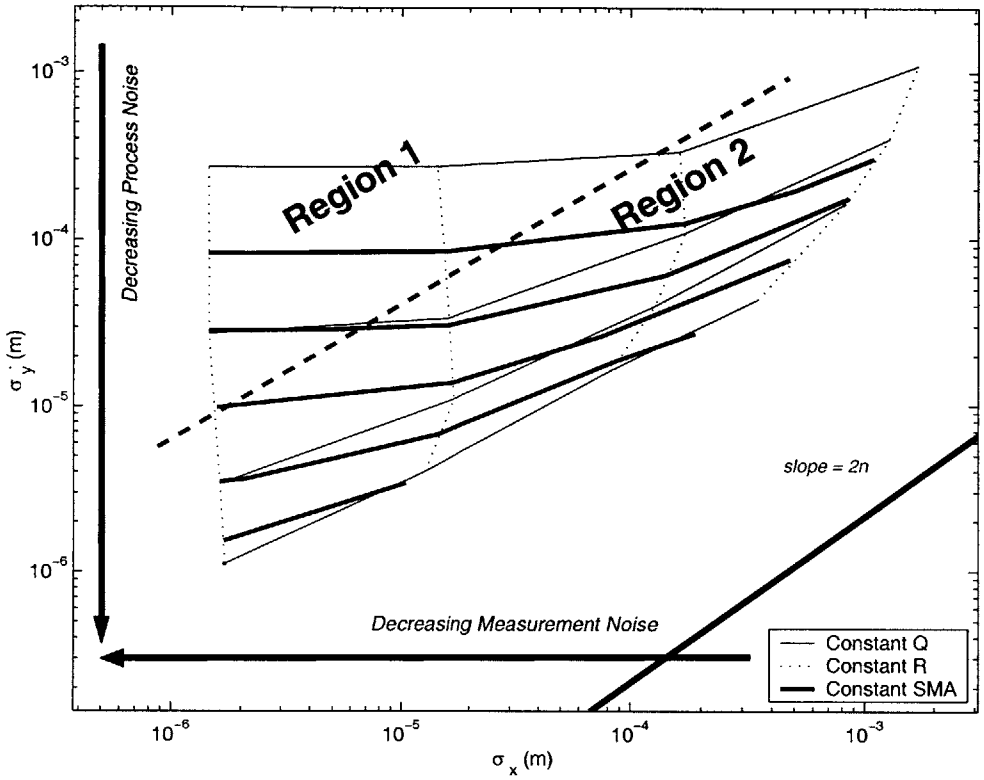

Fig. 3–14: Contours of constant Q, R and semimajor axis are shown for NPM results. Nonlinear effects were minimized for this simulation, which was run with a 0.1 second time step.

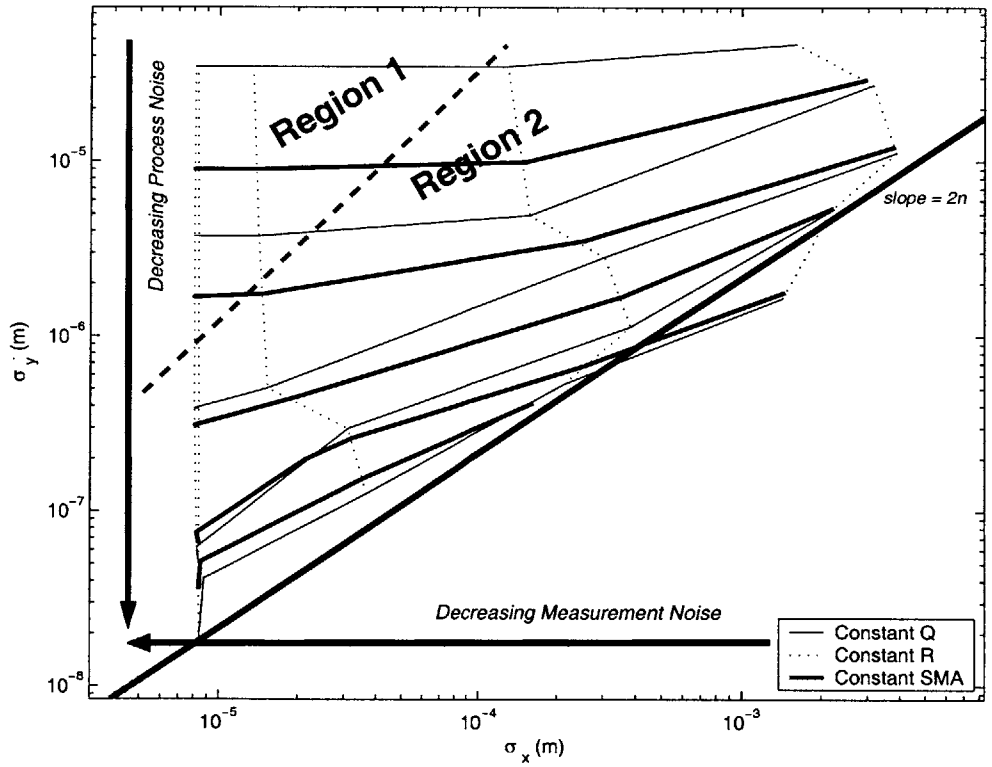

Fig. 3–15: Contours of constant Q, R and semimajor axis are shown for NPM results. Clock, ionospheric, and absolute state errors were removed in this simulation, which had a 1 second time step.

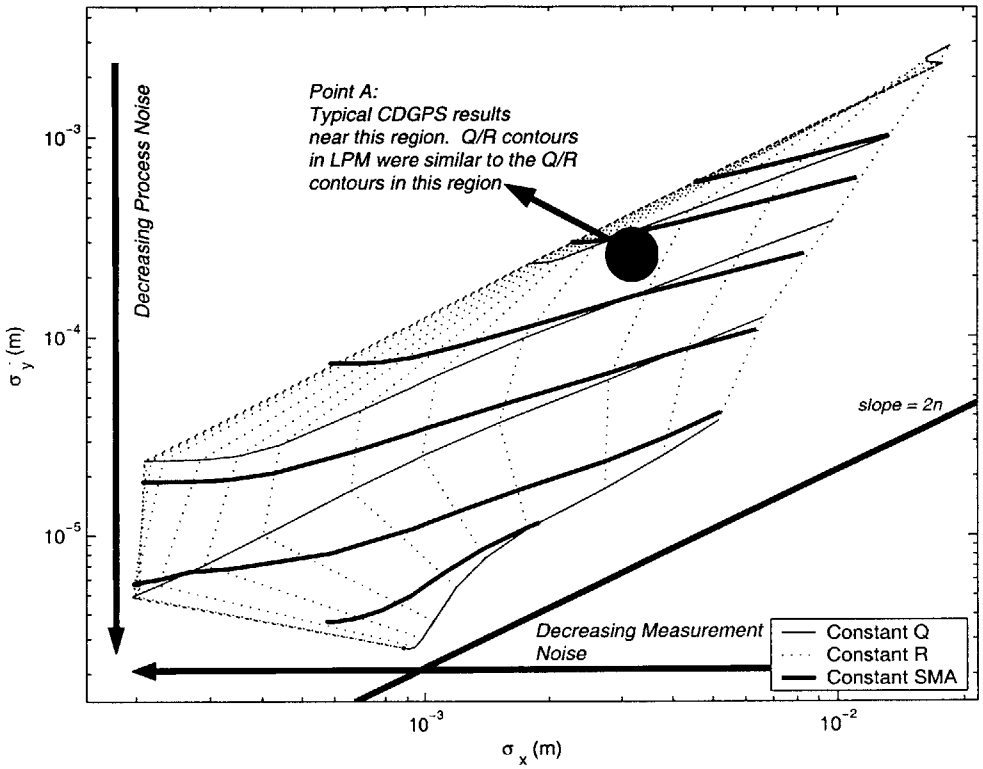

Fig. 3–16: Contours of constant Q, R and semimajor axis are shown for NPM results. This NPM included clock, ionospheric, and absolute state errors.

absolute state uncertainty, which were excluded from the first two simulations. Fig. 3–16 shows constant lines of Q, R and $\sigma_{\Delta a}$ on axis of σ_x and $\sigma_{\dot{y}}$. The line of $\sigma_{\dot{y}} = 2n\sigma_x$ is included as a reference, but will not necessarily represent the same transition point that it did in the LPM. The region that coincides with typical CDGPS accuracy is indicated on the plot. Contours of constant process noise tend to the horizontal direction and contours of constant measurement noise tend towards vertical. The contours of constant semimajor axis have a mildly positive slope, but are still fairly similar to the contours seen on the LPM plots. The trends for constant Q, R lines are in general agreement with the LPM plots, in the sense that both show upward titled parallelograms (see Point A in Figs. 3–9 and 3–16). In going from the LPM to the NGM, the horizontal lines of constant semimajor axis become slightly sloped. This indicates that position accuracy will have some effect on $\sigma_{\Delta a}$ in the NGM, where it had no effect in the LPM. The slope of the NGM contours is still small, which

indicates the velocity error will still dominate semimajor axis knowledge.

The major departure from LPM behavior is seen the lines of constant measurement noise. In the LPM case, reducing measurement noise always resulted in reductions of σ_x. In Region 1 of the LPM, Fig. 3–10, the lines of constant R were vertical and orthogonal to the lines of constant semimajor axis. In this region, decreasing R would not improve $\sigma_{\Delta a}$, but, improvements to σ_x were always possible. In the NGM cases, the contours of constant R begin to bunch when they become very small. This implies that further reductions will impact neither $\sigma_{\Delta a}$ nor σ_x. The fact that σ_x does not continue to improve as the measurement noise is decreased suggests that some aspect of the NGM that models realistic phenomenon, such as clock error, is limiting the overall accuracy.

Fig. 3–16 suggests several things about the performance of our CDGPS-based relative navigation filter. Reducing either process or measurement noise can have a positive effect on semimajor axis. The stance taken in the development of prior work was that a coarse dynamics model is sufficient because measurement updates are performed very frequently [1]. To a large degree, that assumption was correct. The performance of that filter was very good and met the stated requirements. Still, this analysis suggests that future work to incorporate a higher fidelity dynamics model, thereby reducing the process noise, could improve the semimajor axis knowledge. Finally, this analysis shows that noise from sources besides the CDGPS measurements and the relative orbital dynamics model, such as from the clock and absolute state error, also impacts the performance. When these errors are introduced, there is a limit to how much the position, velocity, and semimajor axis performance can be improved. It is expected that there should be a limit to how much velocity and semimajor axis can be improved by increasing the sensor accuracy. However, in the LPM, sensor improvement always resulted in improved position accuracy, which is not true for the NGM.

The progression of the three sets of simulations describe above shows the Nonlinear GPS Model gradually being changed into a model whose behavior resembles the Linear Planar Model analyzed in the previous section. Nonlinearities in measurement

and dynamics, perturbations in the environment, and uncertainty introduced by an inaccurate vehicle clock bend the contours of constant Q and R to some degree. However, the general trends, especially in the area where typical CDGPS filters are found, are retained. The similarities between the linear and nonlinear results suggest that the insights gained from the LPM examples about the relationships between correlation and errors, and between process and measurement noises and errors are relevant to the more complex CDGPS filters.

3.6 Summary of Noise and Navigation Accuracy

This chapter was motivated by the desire to 1) determine what metrics should be used to characterize the filter performance as "good", and 2) explore what parameters in the Kalman filter have the most impact on the performance of the navigation system. Closed loop control performance, which the navigation system exists to support, is dependent on accurate knowledge of the semimajor axis. This chapter started by showing that a Kalman filter estimating relative position and velocity will also produce the best estimate of the differential semimajor axis, because these quantities can be related by a linear transformation.

Previous research has suggested that if the position and velocity errors have high correlation and the proper balance, semimajor axis error can effectively be canceled out, but this idea did not concur with previous and current experience with CDGPS-based navigation systems. Three examples based on a linear planar model were presented to show that the standard GPS problem could be modified to yield highly correlated estimates, but this was always accompanied by an increase in the semimajor axis error. This is because the Kalman filter does not allow independent control over σ_x, $\sigma_{\dot{y}}$, and $\rho_{x\dot{y}}$, which is required to meet both the correlation and balance requirements.

The simple linear model was then used to explore how the filter design parameters affect the navigation accuracy, and the results showed the relationship between the filter inputs, measurement and process noise, and the filter output (navigation errors

and their correlation). The simulation results were shown to match a recently derived analytic expression for the correlation coefficient. Similar tests were done using a full nonlinear GPS simulation. The general trends seen in the linear model were retained in the nonlinear results, suggesting that the insights gained from the linear examples are valuable in understanding complex CDGPS filters. The differences in the linear and nonlinear simulations were also investigated, and these highlighted the effects that realistic phenomenon, such as clock and ephemeris error, have on the performance of the filter.

Chapter 4

The Unscented Kalman Filter for Relative Orbital Navigation

As discussed in Chapter 3, the Kalman filter provides the best state estimate for a linear system. The Extended Kalman Filter (EKF) used in much of this research was developed to handle nonlinear systems. The assumptions made in the EKF require ignoring terms in the dynamics and measurement equations, some of which are not necessarily small [23]. The Unscented Kalman Filter (UKF) was proposed by Julier and Uhlmann as a method to better handle the propagation of the state estimates and their uncertainty through the measurement and/or dynamic nonlinearities. Ref. [39] shows that the UKF will result in accuracy equal or greater than the EKF, with a similar computational burden.

The purpose of this chapter is to investigate whether the UKF may improve the relative navigation for formation flying spacecraft, particularly when vehicle separations exceed 1 km. This chapter begins with a discussion of the nonlinearity seen in the relative navigation problem. Next, additive and square-root forms of the UKF are presented. Tests are performed with simulated measurements to show that there is little performance difference between the two forms of the UKF for this application. These simulation also show that the additive form requires much less computation, so it is used in subsequent comparisons

A set of simulations was run to compare to performance of the UKF to the perfor-

mance of the EKF. Simulations were run using data generated with orbits calculated in FreeFlyer^{TM} and using stored GPS receiver measurements compared to truth data from experiments at GSFC. The results of these simulations confirm that the UKF out-performs EKF when the separation between vehicles is greater than 1 km or the discrete time step of the filter is greater than $10 - 15$ seconds.

4.1 Nonlinearity in Relative Orbital Dynamics

In a relative orbital navigation system using GPS measurements, nonlinear equations govern both the system dynamics and the measurement process. The system dynamics for this CDGPS-based relative navigation filter are based on the nonlinear Hill's equations, as discussed in Section 2.2.1. Hill's equations were originally developed in the 1960's for rendezvous problems, and therefore accuracy over long time periods was not addressed [40]. The work performed in Ref. [1] employs an EKF with measurements every second, and the results suggest that the nonlinearity in this case is negligible. However, as the time step between measurements is increased, or when measurements become unavailable for a period of time, the error accumulated during the time propagation step will become increasingly important.

Previous work in absolute orbital navigation has shown that the degradation due to linearization errors is more pronounced in the estimate of the covariance than in the estimate of the mean [40]. The dynamics used in a relative state estimator are also expected to have errors in the mean and covariance. Hill's equations assume the vehicles are close and that the reference orbit is circular [34]. Alfriend presents a modeling error index to quantify both linearization and modeling error [41]. He shows that Hill's equations have a high modeling error index for separations of even a few kilometers and eccentricities as small as 0.001.

When the filter time step is increased, it is very important that the dynamics be modeled accurately to obtain good estimates. This problem is particularly apparent when the basic Hill's equations are used in the relative navigation for widely separated spacecraft, because the error in these dynamics grows rapidly with the separation dis-

tance. These propagation errors can be decreased by using an improved dynamics model that includes additional nonlinear effects. However, if the filter equations do not properly handle these added nonlinearities, then these more sophisticated models of the dynamics may not lead to improved estimator performance. Because EKF uses a linearized form of the nonlinearities in the system dynamics to propagate the covariances, including additional gravity terms in the model may not improve the EKF's performance. Since UKF does not linearize the dynamics, the nonlinear refinements to the model should improve performance, but, as discussed in the following sections, the UKF still has to make an approximation when the propagated covariance is calculated. This chapter compares the performance of the EKF and UKF and determines the extent to which these improvements are possible when J_2 perturbation terms are added to the dynamics model, as in Eq. 2.19.

4.2 The Unscented Kalman Filter

The development of the EKF propagation and update equations requires ignoring terms which may not be small for many nonlinear systems [23, 19]. The inherent linearization in the process typically introduces significant biases in the estimation results. Although the EKF has been widely used for many years, experience has shown that it is only reliable for systems that are almost linear on the time scale of the update intervals [42]. In contrast, the UKF does not require the linearization of any nonlinear functions[1]. Instead, the UKF uses a set of points, called *sigma points*, that are distributed around the current estimate. These sigma points are chosen so that their mean matches the current estimate and the covariance of the distribution of sigma points matches the current covariance of the estimate [39]. The often stated premise of this approach is "it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation" [39].

During the propagation step of the filter, each sigma point is propagated forward through the actual nonlinear dynamics equation. After the propagation, the set of

[1]Although Ref. [43] presents an opposing view, countered by Ref. [44].

sigma points is condensed back to a single state estimate using a weighted sum of the propagated sigma points. The propagated state covariance is also set to be the calculated covariance of the propagated sigma points. Similarly, the measurement update step requires updating the set of sigma points using the nonlinear measurement equation. The updated sigma points are then condensed back into a single state and their covariances is used for the updated state estimate covariance.

While the EKF handles the nonlinearities by approximating them (*i.e.,* by linearization) in the measurement update and time propagation steps, the UKF approximates the *distribution* of $\hat{\mathbf{x}}$, with the sigma points. The mean and covariance of the original state estimate are represented precisely in the distribution of the sigma points. The mean and covariance of the propagated states (*i.e.,* either time update or measurement update) are correct to second order as well [39], which means that the UKF calculates the mean with more accuracy than the EKF, and the covariance with accuracy of at least the EKF [39]. In fact, Theorem 2 of Ref. [42] strengthens this statement to *the prediction algorithm introduces errors in estimating the mean and covariance at the fourth and higher orders in the Taylor series.*

4.2.1 The Standard Form of the UKF

The original form of the UKF requires augmenting the state with process and measurement noise variables. However, as shown in the equations included in the next section, the number of sigma points required is determined by the length of the state (or augmented state). Since the nonlinear state propagation and measurement update is performed for each sigma point, a larger state can require many more calculations. Fortunately, if the measurement and process noises are purely additive, the standard form can be reduced to what is called the additive form of the UKF (UKF-A), which does not require state augmentation [45]. The UKF-A has a smaller state and a reduced computational burden. The original augmented form of the UKF was implemented, but since the process and measurement noises in the relative orbital navigation filter are additive, as seen in Chapter 2, the reduced additive form can be used for this application. The two UKF forms were compared, and the results showed

that the performances are virtually identical and that the computation time for the standard form is much higher than for the additive form. Thus, the original form was abandoned in favor of the additive form of the UKF, which presented in the following section.

4.2.2 The Additive Form of the UKF

The additive form of the UKF (UKF-A) is used for systems whose process and measurement noises are purely additive, as in the case of a CDGPS relative navigation filter. This form is preferred over the standard form of the UKF because it has a smaller state vector, resulting in fewer sigma points and less computation. Similar to the standard form, the UKF-A uses the nonlinear dynamics and measurement equations and employs a set of sigma points in each time propagation and measurement update step.

At each step, $2n + 1$ sigma points are required, where n indicates the length of the state vector. The sigma points that are used for the time propagation and measurement update steps are chosen to have a cumulative mean and standard deviation identical to the prior estimate. The sigma points are formed by adding and subtracting scaled columns of the matrix square root of the covariance matrix to the original state estimate. Adding these $2n$ columns to the original state vector produces the required $2n+1$ sigma points [45, 19]. This procedure is shown below in the presentation of the UKF-A algorithm.

Several weights and constants are used in the UKF algorithm. The set of scalar weights used to recombine the sigma points into the a posteriori state mean and

covariance estimates, $\{W_i\}$, are defined by

$$W_0^{(m)} = \frac{\lambda}{n+\lambda} \tag{4.1}$$

$$W_0^{(c)} = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \tag{4.2}$$

$$W_i^{(m)} = \frac{1}{2(n+\lambda)} \quad , \; i = 1, \ldots, 2n \tag{4.3}$$

$$W_i^{(c)} = \frac{1}{2(n+\lambda)} \quad , \; i = 1, \ldots, 2n \tag{4.4}$$

The scaling parameter λ is $\lambda = \alpha^2(n+\kappa) - n$. The related parameter, $\gamma = \sqrt{n+\lambda}$, is used in the sigma point calculation. The parameter α, set between 10^{-4} and 1, determines the spread of points around the original estimate. The parameter β influences how much the prior estimate of the covariance is weighted and is typically set to 2 for state estimation problems.

The algorithm for the UKF-A is presented in Table 4.1. In several places, the nonlinear dynamics equation $\mathbf{f}$ or measurement equations $\mathbf{h}$ is applied to a matrix of vectors, producing matrix output. First the sigma points are formed around the $\hat{\mathbf{x}}_{k-1}$ state estimate (Eq. 4.7). The sigma points are propagated (Eq. 4.8) and weighted sums of the sigma points are used to calculate propagated state estimate and covariance, $\hat{\mathbf{x}}_k^-$ and P_k^- (Eqs. 4.9 and 4.10). A new set of sigma points is calculated for the measurement update step (Eq. 4.11). The measurement equation is used to create an expected measurement for each of the sigma points (Eq. 4.12), which is then condensed with a weighted sum to create an overall, composite expected measurement, $\hat{\mathbf{y}}_k^-$ (Eq. 4.13). Covariance matrices are calculated with the expected measurements (Eqs. 4.14 and 4.15), and a Kalman gain matrix is formed (Eq. 4.16). Finally, the updated state and covariance, $\hat{\mathbf{x}}_k$ and P_k, are calculated (Eqs. 4.17 and 4.18) [45].

As a practical note, it was observed that numerical errors would sometimes introduce very small imaginary components when the matrix square root of the covariance was computed. This tends to disrupt the algorithm, so preventative measures should be included.

Another form of the UKF, called the square root form (UKF-S), has been pro-

Table 4.1: UKF-A Algorithm

1. The UKF-A is initialized with

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \tag{4.5}$$
$$P_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \tag{4.6}$$

For $k \in \{1, \ldots, \infty\}$,

2. Calculate sigma points:

$$\mathcal{X}_{k-1} = [\ \hat{\mathbf{x}}_{k-1},\ \ \hat{\mathbf{x}}_{k-1} + \gamma\sqrt{P_{k-1}},\ \ \hat{\mathbf{x}}_{k-1} - \gamma\sqrt{P_{k-1}}\ ] \tag{4.7}$$

3. Propagate the sigma points with the nonlinear dynamics equation $\hat{\mathbf{x}}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k)$ and find the covariance:

$$\mathcal{X}^*_{k|k-1} = \mathbf{f}(\mathcal{X}_{k-1},\ \mathbf{u}_{k-1}) \tag{4.8}$$
$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{X}^*_{i,k|k-1} \tag{4.9}$$
$$P_k^- = \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{X}^*_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathcal{X}^*_{i,k|k-1} - \hat{\mathbf{x}}_k^-)^T + Q \tag{4.10}$$

4. Create a new set of sigma points and, with the nonlinear function measurement equation $\hat{\mathbf{y}}_\mathbf{k} = \mathbf{h}(\hat{\mathbf{x}}_\mathbf{k})$, create expected measurements:

$$\mathcal{X}_{k|k-1} = [\ \hat{\mathbf{x}}_k^-,\ \ \hat{\mathbf{x}}_k^- + \gamma\sqrt{P_k^-},\ \ \hat{\mathbf{x}}_k^- - \gamma\sqrt{P_k^-}\ ] \tag{4.11}$$
$$\mathcal{Y}_{k|k-1} = \mathbf{h}(\mathcal{X}_{k|k-1}) \tag{4.12}$$
$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \tag{4.13}$$

5. Create the Kalman gain and perform a measurement update:

$$P_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} = \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-)(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-)^T + R \tag{4.14}$$
$$P_{x_k y_k} = \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-)(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-)^T \tag{4.15}$$
$$K_k = P_{x_k y_k} P^{-1}_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} \tag{4.16}$$
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \tag{4.17}$$
$$P_k = P_k^- - K_k P_{\tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k} K_k^T \tag{4.18}$$

posed as an alternative to the additive form. The square root form makes use of Cholesky factorization, and in some applications may increase numerical stability and performance. Both the UKF-A and the UKF-S are evaluated. The algorithm for the UKF-S is presented in Section 4.2.3 and a comparison of the UKF-A and the UKF-S follows in Section 4.2.4.

4.2.3 The Square Root Form of the UKF

Van der Merwe and Wan introduced the Square Root form of the UKF (UKF-S) [46] to handle some of the numerical issues referred to in the previous section. The computational burden of the UKF-S is expected to be similar to that of the UKF-A. However, the UKF-S may provide improved numerical stability and it also guarantees that the covariance matrix will be positive semi-definite, which is required for filter stability.

The main innovation introduced in the UKF-S is the use of the Cholesky factor, S, of the covariance matrix. The Cholesky factor, S, is initialized taking the matrix square root of the initial P with a Cholesky factorization. Three linear algebra techniques are recommended by Merwe for use in the UKF-S [46]:

- **QR decomposition** The QR decomposition factors a matrix A into the product of an orthogonal matrix, Q, and an upper triangular matrix, R, so $A^T = QR$. The upper triangular part of R is the transpose of the Cholesky factor of a matrix P, defined such that $P = AA^T$. The MATLAB function `qr` returns the *lower* Cholesky factor. The transpose of the output of `qr` should be used to get the *upper* Cholesky factor required in this algorithm.

- **Cholesky factor updating** Consider a matrix $P = AA^T$ that has a Cholesky factor of S. If the matrix P is updated so that $P^+ = P \pm \sqrt{\nu}\mathbf{u}\mathbf{u}^T$, the Cholesky factor of the updated P^+ is found using the `cholupdate` MATLAB function, $\texttt{chol}(P^+) = \texttt{cholupdate}(S, \mathbf{u}, \pm\nu)$. When $\mathbf{u}$ is a matrix, the function performs a Cholesky update for each of the columns of $\mathbf{u}$.

- **Efficient Least Squares** The MATLAB function "/" finds the solution to $(AA^T)\mathbf{v} = A^T\mathbf{b}$ using an efficient QR decomposition method.

These three techniques were incorporated into the Square Root Unscented Kalman Filter used in this research. The UKF-S algorithm is summarized in Table 4.2.

4.2.4 Comparison of Additive and Square Root UKF's

The UKF-S has the potential for improved performance and numerical stability, but these improvements are not guaranteed, so both the Additive and Square Root form of the UKF (UKF-A and UKF-S) were implemented and compared. Since UKF was designed to better handle the nonlinearities, a 10 km baseline example was used for this comparison. The estimator was given simulated measurements that were created from stored absolute trajectories. White noise was applied to the dynamics and the measurements. The performance was seen to vary when the filter was run for different randomly generated measurement noise sequences. Twenty different random measurement noise profiles were generated and stored, and the simulation was run for each profile. The average of the results from 20 profiles provided a better basis for evaluating the filter than the results from any single noise profile. Because nonlinearities in the dynamics may have more of an effect when the filter has a longer time step, the UKF-A and UKF-S were compared for time steps between 5 and 60 seconds.

Fig. 4–1 shows the differences in the means and standard deviations for the position and velocity estimates from the UKF-A and UKF-S. The results for the individual runs are shown in the background in gray, and the mean over all the noise profiles is shown in a strong black line. The differences for any single run is very small, and, when averaged over the 20 noise profiles, the difference is negligible. Also, the mean differences do not grow or shrink significantly as the time step is increased. The results, summarized in Table 4.3, indicate that there is no performance advantage to using the UKF-S in relative navigation filters. Because there is no significant performance difference, and the UKF-A is easier to implement, it was the one used for

Table 4.2: UKF-S Algorithm

1. The UKF-S is initialized with

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \tag{4.19}$$
$$S_0 = \texttt{chol}\left(E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]\right) \tag{4.20}$$

For $k \in \{1, \ldots, \infty\}$,

2. Calculate sigma points

$$\mathcal{X}_{k-1} = [\ \hat{\mathbf{x}}_{k-1},\ \ \hat{\mathbf{x}}_{k-1} + \gamma S_k,\ \ \hat{\mathbf{x}}_{k-1} - \gamma S_k\ ] \tag{4.21}$$

3. Propagate sigma points using nonlinear dynamics equation and find covariance

$$\mathcal{X}^*_{k|k-1} = \mathbf{f}(\mathcal{X}_{k-1},\ \mathbf{u}_{k-1}) \tag{4.22}$$
$$\hat{\mathbf{x}}^-_k = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{X}^*_{i,k|k-1} \tag{4.23}$$
$$S^-_k = \texttt{qr}\left(\left[\ \sqrt{W_1^{(c)}}(\mathcal{X}^*_{1:2n,k|k-1} - \hat{\mathbf{x}}^-_k),\ \ \sqrt{Q}\ \right]\right) \tag{4.24}$$
$$S^-_k = \texttt{cholupdate}\left(\ S^-_k,\ \ \mathcal{X}^*_{0,k} - \hat{\mathbf{x}}^-_k,\ \ W_0^{(c)}\ \right) \tag{4.25}$$

4. Create new set of sigma points and, using the nonlinear measurement equation, and perform the Cholesky update

$$\mathcal{X}_{k|k-1} = [\ \hat{\mathbf{x}}^-_k,\ \ \hat{\mathbf{x}}^-_k + \gamma S^-_k,\ \ \hat{\mathbf{x}}^-_k - \gamma S^-_k\ ] \tag{4.26}$$
$$\mathcal{Y}_{k|k-1} = \mathbf{h}(\mathcal{X}_{k|k-1}) \tag{4.27}$$
$$\hat{\mathbf{y}}^-_k = \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \tag{4.28}$$
$$S_{\tilde{y}_k} = \texttt{qr}\left(\left[\ \sqrt{W_1^{(c)}}(\mathcal{Y}_{1:2n,k} - \hat{\mathbf{y}}_k),\ \ \sqrt{R}\ \right]\right) \tag{4.29}$$
$$S_{\tilde{y}_k} = \texttt{cholupdate}\left(\ S_{\tilde{y}_k},\ \ \mathcal{Y}_{0,k} - \hat{\mathbf{y}}_k,\ \ W_0^{(c)}\ \right) \tag{4.30}$$

5. Create the Kalman gain and perform the measurement update

$$P_{x_k y_k} = \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}^-_k)(\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}^-_k)^T \tag{4.31}$$
$$K_k = (P_{x_k y_k} / S^T_{\tilde{y}_k}) / S_{\tilde{y}_k} \tag{4.32}$$
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^-_k + K_k(\mathbf{y}_k - \hat{\mathbf{y}}^-_k) \tag{4.33}$$
$$U = K_k\ S_{\tilde{y}_k} \tag{4.34}$$
$$S_k = \texttt{cholupdate}(S^-_k, U, -1) \tag{4.35}$$

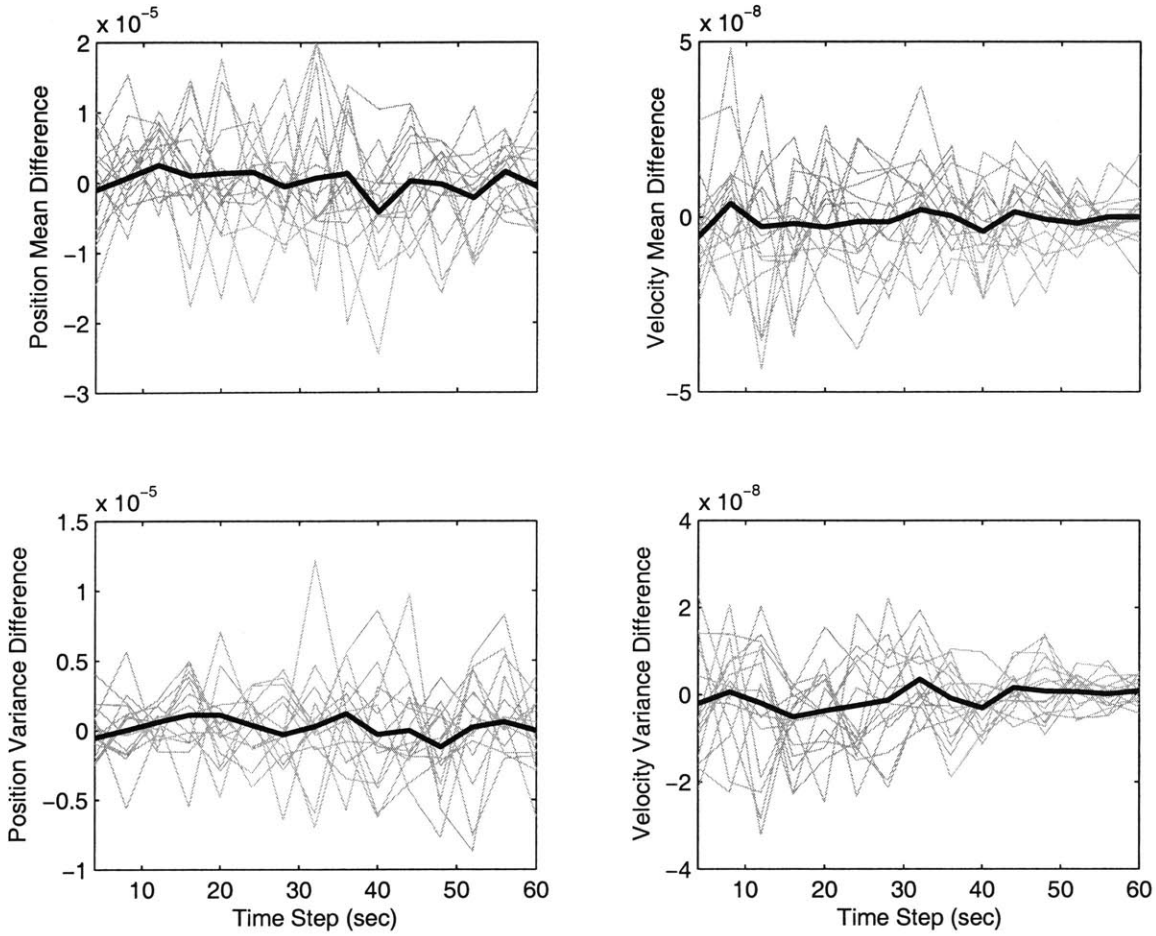

Fig. 4–1: Performance Difference Between UKF-A and UKF-S: Negligible

comparisons of the UKF and EKF.

4.3 EKF versus UKF for Relative Navigation

Since the UKF was developed to better handle nonlinear dynamics and measurements, it is expected that the UKF will perform better than the EKF only when the nonlinearities become significant. Section 4.1 discussed nonlinearities seen in the relative navigation problem. The errors associated with nonlinearities are expected to increase as the baseline distance between the two vehicles grow and as the filter time step increases.

Table 4.3: Comparison of the Standard Additive and Square Root Forms of the UKF

	Position (m)	Velocity (m)
Mean, $\mu_{UKF-S} - \mu_{UKF-A}$	3.8277e-005	4.5264e-008
Standard Deviation, $\sigma_{UKF-S} - \sigma_{UKF-A}$	1.2474e-005	3.0500e-008

4.3.1 Summary of Simulations to Compare EKF and UKF

Several sets of simulations were created to explore the performance differences in the EKF and UKF. To observe the effects of both increased baseline and time step, the simulations were conducted for baselines of 100 m, 1 km, and 10 km for time steps between 5 seconds and 1 minute. Comparisons were repeated in both a simulated environment and a more realistic environment created using stored data from GSFC. The two environments have different advantages. The total simulation environment provides control over every variable, from noise levels, to perturbations, to satellite coverage. However, simulation results are usually required to be corroborated by data from real hardware. For this reason, simulations based on FreeFlyer^{TM} trajectories and simulations base on recorded hardware experiments at GSFC are used. A summary of these environments is given below:

1. FreeFlyer^{TM}-based simulations The original MATLAB simulation trajectories were created with a simple propagator. Perturbations and other real-world effects were coarsely simulated by adding white noise into the dynamics propagation. This is exactly the dynamics model used in the EKF and the UKF. The simulation is more realistic if it uses truth trajectories created with a dynamics model that has a much higher fidelity than the model used in the filter. The FreeFlyer^{TM} commercial orbital dynamics simulation software can create very high fidelity trajectories that include perturbation forces including higher order gravity terms, solar radiation pressure, 3rd body gravity effects, and aerodynamic forces [47]. For the EKF/UKF comparison, FreeFlyer^{TM} was used to create the truth trajectories from which the simulated measurements were derived. Each simulation was repeated for 20 stored noise profiles and the results

Table 4.4: Summary of the Simulations and Results.

Baseline	FreeFlyer™ data	GSFC data
100 m	Fig. 4–2	Fig. 4–5
1 km	Fig. 4–3	Fig. 4–6
10 km	Fig. 4–4	Fig. 4–7

were averaged.

2. GSFC stored-data simulations The GSFC Formation Flying Testbed has a Spirent simulator that models vehicle motion and the GPS satellite constellation and creates an RF signal that mimics the input of the vehicle antenna in space. The Spirent trajectories can be stored as truth data and the output of the GPS receiver is stored to provide future measurement inputs to the filter. The truth and measurements can be post-processed to evaluate various estimators. More information about the setup used at GSFC is included in Chapter 5 as part of a discussion about online estimation.

A summary of the simulations used to compare the EKF and UKF is shown in Table 4.4. The results of the comparisons are shown in Figures 4–2 to 4–7. Each figure has subplots for position and velocity. The mean of the estimate error and bounds for the standard deviation of the error are shown for both the EKF and UKF. The errors are shown against an axis indicating the discrete time step of the filter. These six plots contain a large amount of information, and will be used to demonstrate several trends. The set of figures will be discussed: 1) as individual entities, 2) across a set of baseline distances, evaluating FreeFlyer™ and GSFC results separately, and 3) between FreeFlyer™ and GSFC results, evaluating each baseline distance separately. Prior to initiating a detailed discussion of the results, a summary of the questions addressed is listed below:

1. **Examining a Single Figure**

   - How do the errors for the EKF and UKF results compare?

- Do the means grow as the time step is increased?
- Do the standards deviations grow as the time step is increased?
- Does the time step increase have a greater effect on position or velocity estimates?

2. **Comparing plots across 3 baselines for same environment**
   - How do the mean values change across the 3 baselines?
   - Do the standard deviations increase across the 3 baselines?
   - If the standard deviation increases with time step for one baseline, does it in the others? If not, is there a reasonable explanation?
   - Does changing the baseline affect the position and velocity differently?

3. **Comparing FreeFlyer™ GSFC plots for same baselines**

   The results from the FreeFlyer™ and GSFC simulations are not expected to agree numerically. The FreeFlyer™ based simulations provide much more control over the truth trajectories and perturbation effects. The measurements are created by adding white noise to the output of equations. This model was also used in the estimator. At GSFC, the real Orion™ hardware with the many associated uncertainties, was used to create the measurements. For example, thermal variation has been known to affect the measurements. Clock uncertainties, poor electrical connections, or unexpected interference might also contribute errors. In general, the simulated measurements produce better estimates. However, trends seen in simulation are also found in hardware tests, so both are used in the comparisons.

   - When the trends in the FreeFlyer™ and GSFC simulations are similar, how do the FreeFlyer™ results reinforce interpretation for GSFC performance?
   - When the trends are different, do simulation differences provide a reasonable explanation?

- Finally, do the FreeFlyer^{TM} and GSFC provide the same answer for the question of when does the UKF performs better?

Several expectations about the performance of the EKF and UKF and about the results of FreeFlyer^{TM} and GSFC simulations are also useful to consider before proceeding:

- The UKF should perform better when the dynamics model breaks down. The dynamics model breaks down when the time step or baseline distance increases significantly.

- Dynamics models in both the EKF and UKF are more similar to the model that created the FreeFlyer^{TM} trajectory. Since velocity estimate depends on quality of velocity model, FreeFlyer^{TM} simulations may perform better for velocity estimates.

- The filter measurement models closely match those used to create measurements from the FreeFlyer^{TM} simulated trajectory. The hardware used to record GSFC measurements may or may not reflect the measurement model in the filter. This suggests FreeFlyer^{TM} simulations may perform better.

Though results are shown for both position and velocity estimates, the velocity performance is much more important. As discussed in Chapter 3, the error in the velocity estimate has more effect on the knowledge of semimajor axis error than does the error in position. Since the semimajor axis error influences closed loop control performance, and the navigation system exists primarily to aid in the control system, the velocity performance will be used as the final discriminator between the EKF and UKF.

4.3.2 Comparison for Single Baseline, as Time Step Increases

Each of the six figures shows the mean and the 1σ bounds around the mean for position and velocity errors. The individual figures are useful in evaluating whether there is an advantage in using the UKF for a particular scenario. For example, at

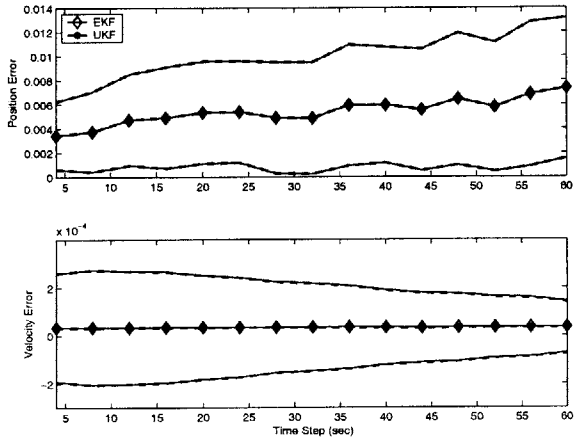

Fig. 4–2: FreeFlyer^{TM}, baseline 100 m

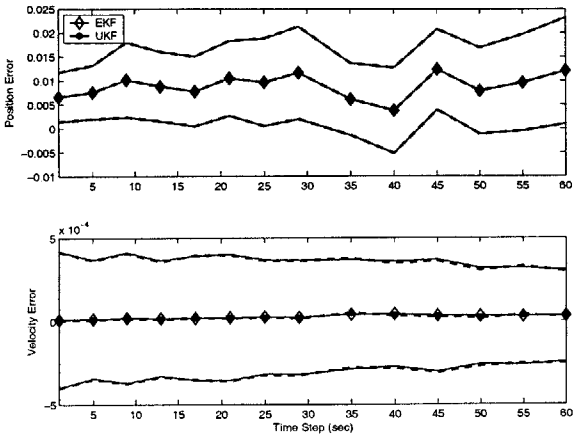

Fig. 4–5: GSFC, baseline 100 m

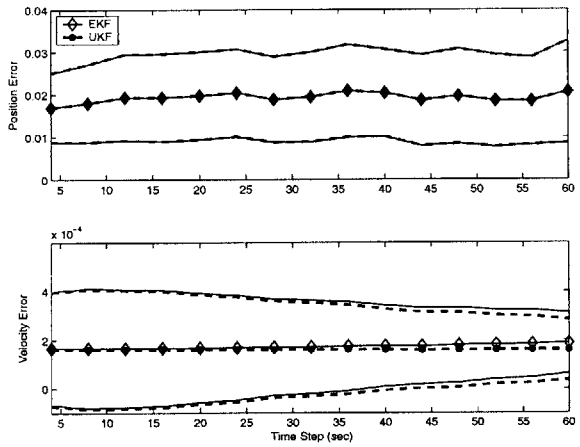

Fig. 4–3: FreeFlyer^{TM}, baseline 1 km

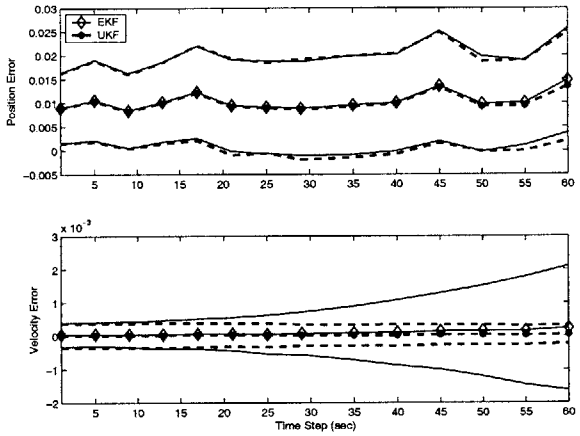

Fig. 4–6: GSFC, baseline 1 km

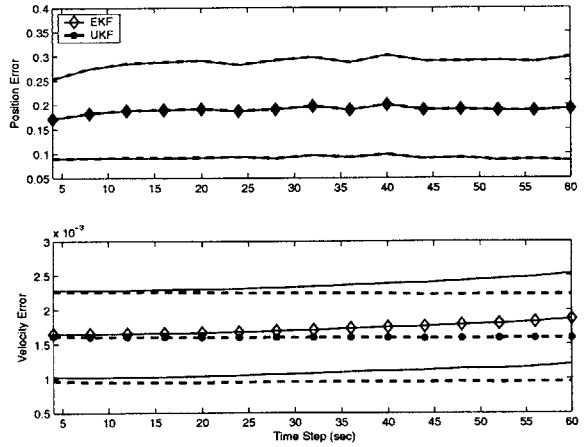

Fig. 4–4: FreeFlyer^{TM}, baseline 10 km

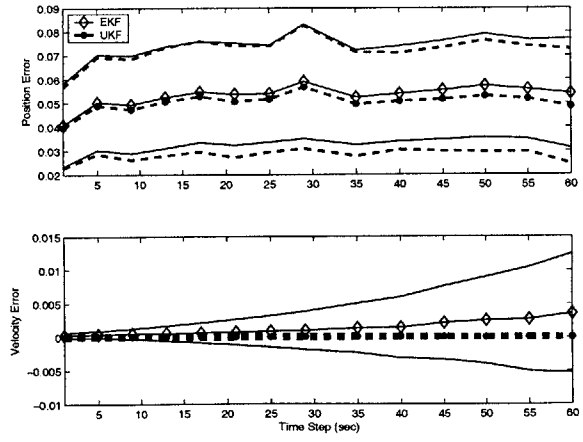

Fig. 4–7: GSFC, baseline 10 km

100 m, the position and velocity means are nearly identical for the UKF and EKF. The standard deviation increases very slightly with the time step. In general, the nonlinearities at 100 m, even at a longer time step, are not significant enough to warrant using a UKF. At 1 km and 10 km, the velocity means are higher for the EKF, and the velocity standard deviations diverge as the time step increases. In these cases, the UKF would be a better choice.

4.3.3 Comparison as Baselines Increase

Figures 4–2 to 4–7 show that as the baseline is increased, the mean values for position and velocity errors also increase. This degradation of estimation accuracy concurs with results reported in Ref. [1]. When the nonlinear equations were linearized, the higher order terms of the series expansion are truncated. The linearization error increases with the distance between vehicles. The truncation is accounted for in the filter by including it in the process noise term. The process noise must therefore be increased as the baseline (and the corresponding truncation error) is increased. The results of this increase in process noise are increases in the position and velocity errors. Similarly, as the baseline grows, the standard deviation trends change from being tapered or constant as the time step increases, to diverging with time step increases.

Though the errors grow with the baseline size, the UKF still offers advantages over the EKF. This is seen in the velocity performance of the UKF, particularly in Figures 4–6 and 4–7. Interesting differences in the velocity standard deviation of the FreeFlyer^{TM} and GSFC simulations arise and are discussed in the following section.

4.3.4 Comparison for FreeFlyer^{TM} and GSFC simulations

As stated before, the results from FreeFlyer^{TM} and GSFC simulations are not expected to agree numerically. The 100 m baseline case was unremarkable: the EKF and UKF produced nearly identical results, with position means and standard deviations both showing slight increases in the FreeFlyer^{TM} and GSFC simulations. The

velocity means were constant for both simulation environments. The velocity standard deviation bounds became smaller, though this effect was more subtle for the GSFC simulations.

With 1 km baselines, the position trends are similar in the FreeFlyer^{TM} and GSFC simulations. The mean values increase slightly with time step, and the standard deviation bounds flare slightly at the largest time steps. There is no significant difference in EKF and UKF performance in position estimation. As in the at 100 m simulations, the mean of velocity errors show little growth as the time step increases. The velocity mean of the EKF is slightly higher than the UKF mean for the largest time steps.

An interesting difference is seen in the standard deviations of velocity error produced by the FreeFlyer^{TM} and GSFC simulations, at a 1 km baseline. In the FreeFlyer^{TM} simulations, the standard deviations for both the EKF and the UKF decrease as the time step increases. In the GSFC simulations, the UKF velocity standard deviation shows a slight decrease as well. However, this is nearly obscured by the dramatic *increase* in the EKF velocity standard deviation. The question is why the EKF velocity diverges in GSFC simulations and not in the FreeFlyer^{TM} simulations. Recall that the dynamics model in the filter, described in Chapter 2, is a simple 2-body model with J_2 perturbations. This is closer to the model used to create the FreeFlyer^{TM} trajectory than it is to the model that governs the GSFC dynamics. FreeFlyer^{TM} corresponds to a case that is in between "perfectly modeled dynamics" and "fully realistic dynamics." Since the velocity estimates strongly depend on the quality of the dynamics model, it is reasonable to expect that the GSFC simulations will show poor performance for combinations of the separation & time step that are shorter than for the FreeFlyer^{TM} simulations. In addition, the "measurements" used in the FreeFlyer^{TM} simulations were created with the measurement equation, while those used in the GSFC simulations were recorded with real hardware. Accordingly, FreeFlyer^{TM} simulations also fall somewhere between "perfectly modeled measurements" and "fully realistic measurements."

This difference in velocity standard deviation behavior in FreeFlyer^{TM} and GSFC

simulations continues in the 10 km baseline examples. In the GSFC simulations, in Fig. 4–7, the EKF velocity mean becomes much larger than the UKF velocity mean at large time steps exceeding even the UKF 1σ bounds. The EKF is so much worse in this case that the UKF mean and standard deviation lines appear on top of each other – the EKF velocity standard deviation bounds diverge fairly explosively. Overall, the errors in the EKF velocity at 10 km eclipse the UKF errors. In comparison, while the FreeFlyer^{TM} simulations show the UKF errors are smaller than the EKF errors, the difference is not as dramatic. This is attributed to the differences in the FreeFlyer^{TM} and GSFC simulation setups.

Also of significance, in the 1 km FreeFlyer^{TM} simulations, the velocity standard deviations decreased for the larger time steps, in Fig. 4–3. Conversely, they appear nearly constant in the 10 km FreeFlyer^{TM} simulations, in Fig. 4–4. This suggests the FreeFlyer^{TM} simulations will follow a trend seen in the 100 m and 1 km GSFC simulations, where the velocity standard deviation *decreased* for the larger time steps in the former, but *increased* for larger time steps in the latter. The trend appears more gradually in the FreeFlyer^{TM} simulations than the it did in the GSFC simulations, where the EKF velocity standard deviations failed dramatically when the distance was increased from 100 m to 1 km. Still, it appears that the dynamics model is beginning to fail even in the FreeFlyer^{TM} simulations. This trend is affirmed for the FreeFlyer^{TM} model in a final simulation with extreme nonlinearities, in Section 4.4, where the EKF mean and standard deviations diverge with the sharpness seen in the GSFC simulations.

Overall, the UKF performs better than the EKF when the baseline distance and time step are both increased. This assessment is based on the smaller means and standard deviation bounds of the velocity errors, which is the parameter that most strongly influences closed loop control performance. The advantage of the UKF is especially apparent in GSFC simulations, whose dynamics and measurements are more realistic than the FreeFlyer^{TM} simulations.

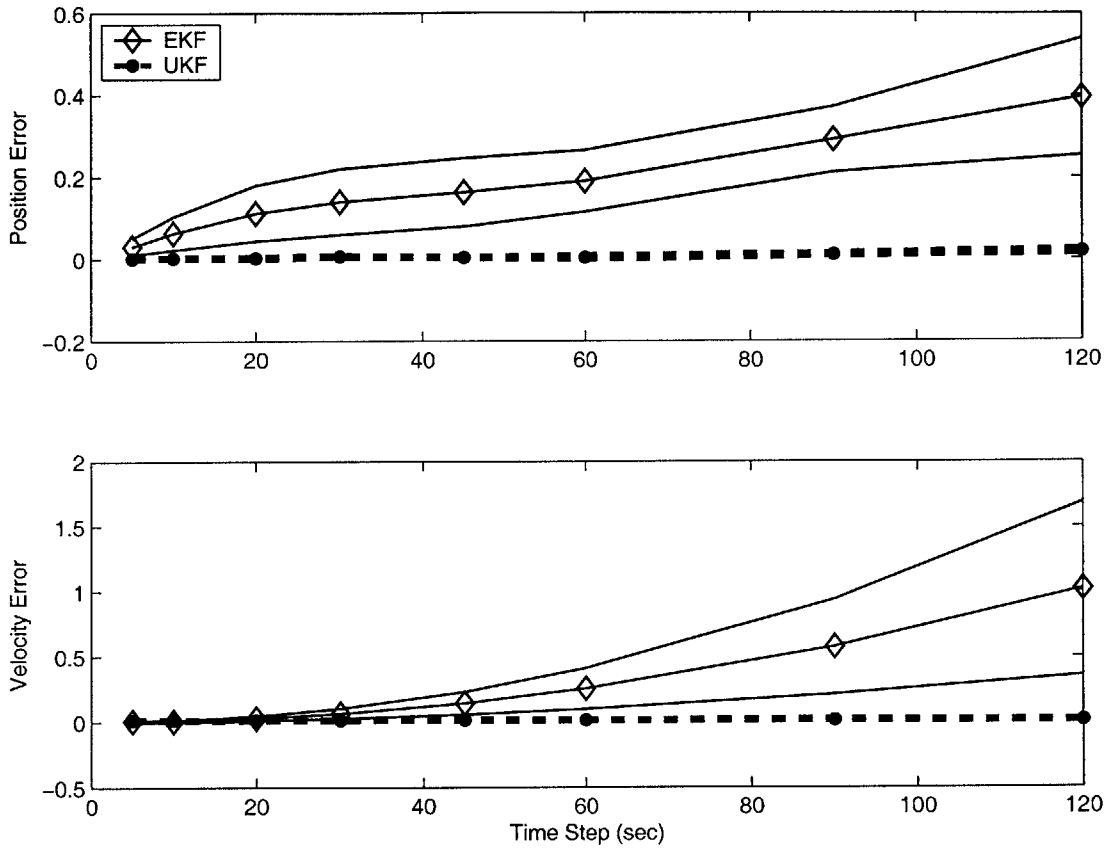

Fig. 4–8: FreeFlyer^{TM}, baseline 100 km

4.4 A Final Example

The previous discussion presented examples that showed the UKF outperforming the EKF as the baselines and time steps increased. This confirmed the hypothesis that the UKF, designed to better handle nonlinearities in the dynamics and measurement models, would become advantageous when the system nonlinearities are accentuated.

A final example with an extremely long baseline of 100 km is presented in Fig. 4–8. The nonlinearities are especially insidious here, as Hill's equations fail rapidly at this large separation. Also, at this distance, the ionospheric effects would begin to dominate [48]. At very small time steps, the EKF and UKF performance is comparable. This is because the estimate is corrected very frequently with new measurements. At higher time steps, the filter depends on the dynamics model to propagate be-

tween states, but when the dynamics model is poor, the propagated state error grows rapidly. If this error is not corrected quickly, then the filter will diverge. The UKF uses the nonlinear dynamics and measurement equations and employs a much better method of propagating the state error covariance. As a result, the performance is more consistent at longer baselines and longer time steps, and the UKF does not diverge, which is in stark contrast to the EKF results in Fig. 4–8.

This example underscores the potential for the UKF in situations when nonlinearities, including those caused by long time steps or large separations, are particularly important. Long time separations and eccentric orbits are likely to be required of some future missions and could reasonably benefit from the UKF.

4.5 Summary

Overall, the UKF performs better than the EKF for larger separations and longer time steps. This assessment is based on the smaller means and standard deviation bounds of the velocity errors, which is the parameter that most strongly influences closed loop control performance. The advantage of the UKF is especially apparent in the GSFC simulations, which represents a more realistic setup (in particular the measurements) than the FreeFlyer^{TM} simulations.

Chapter 5

Development of a Closed Loop Navigation System Architecture

The work in this thesis builds on research by Olsen and Busse [49, 1]. Olsen's demonstration of CDGPS-based relative navigation used blimps and an indoor GPS system as a surrogate for a space demonstration [9]. Busse extended this work to a simulated space environment by performing off-line estimation with data from a high fidelity GPS RF signal generator at NASA GSFC [50]. The next steps in demonstrating CDGPS technologies are online estimation and closed loop navigation. In preparation for these demonstrations, several components of the navigation system had to be updated. The navigation system includes GPS receivers, GPS Receiver Monitor interface software, Estimation code, Control code, and an orbital dynamics model that defines the truth trajectory. The GPS receivers, GPS Receiver Monitor interface software, and Estimation code are addressed in this research. The communication interfaces between these three components, and between the Estimation and Control blocks are also discussed. The Control algorithms, truth modeling, and related communication interfaces are addressed in Ref. [20].

This chapter begins with a comparison of the simulation architecture for the original and current simulation setups. Next, the changes made to the individual components of the navigation system are described. Finally, tests performed with a Spirent Simulator at NASA Goddard Space Flight Center (GSFC) are discussed.

5.1 Original and Updated Navigation Systems

Two milestones in advancing these relative navigation algorithms towards space flight are *online estimation* and *closed loop navigation.* Online estimation requires the GPS receivers track RF signals continuously, and, in real time, execute the estimation algorithm. When the loop is closed, the Estimation block provides state estimates to the Control block, and the Control block generates commands that are applied to the system and fed back into the Estimation block. Closing the loop introduces communication requirements between the Estimation and Control blocks. If the control portion of closed loop control is disabled, the system reverts to online estimation. In this sense, online estimation requires only a subset of the closed loop control functions. Both online estimation and closed loop navigation were accomplished at GSFC using the architecture described in this section.

Diagrams of the original [1] and current navigation systems are shown in Fig. 5–1. The boxed portion of each diagram signifies which steps are performed in real time at GSFC and is repeated N times, once for each vehicle in the formation. The original system begins with receivers that transmit measurements to intermediate GPS Receiver Monitor (GPS-RM) software. The GPS-RM software stores the absolute carrier measurements from the receiver in data files. At a later time, the data files for all vehicles are imported into a single MATLAB Estimation block, and the carrier differential measurements are created by differencing the absolute measurements. This enables off-line evaluation of the Kalman Filter. Thruster firings were considered by reading a pre-planned thrusting sequence from a file. This architecture clearly cannot support online estimation or closed loop navigation.

In the new architecture, the GPS-RM also continuously reads measurements from the GPS receiver. However, instead of storing the data, the GPS-RM on each vehicle spools the measurements to the local Estimation block over a TCP/IP interface. The Leader Estimation block calculates an estimate for the absolute state, which is used in the local Control block. It also sends its raw carrier measurements to the Follower vehicles, which use them to create the carrier differential GPS measurements. Each

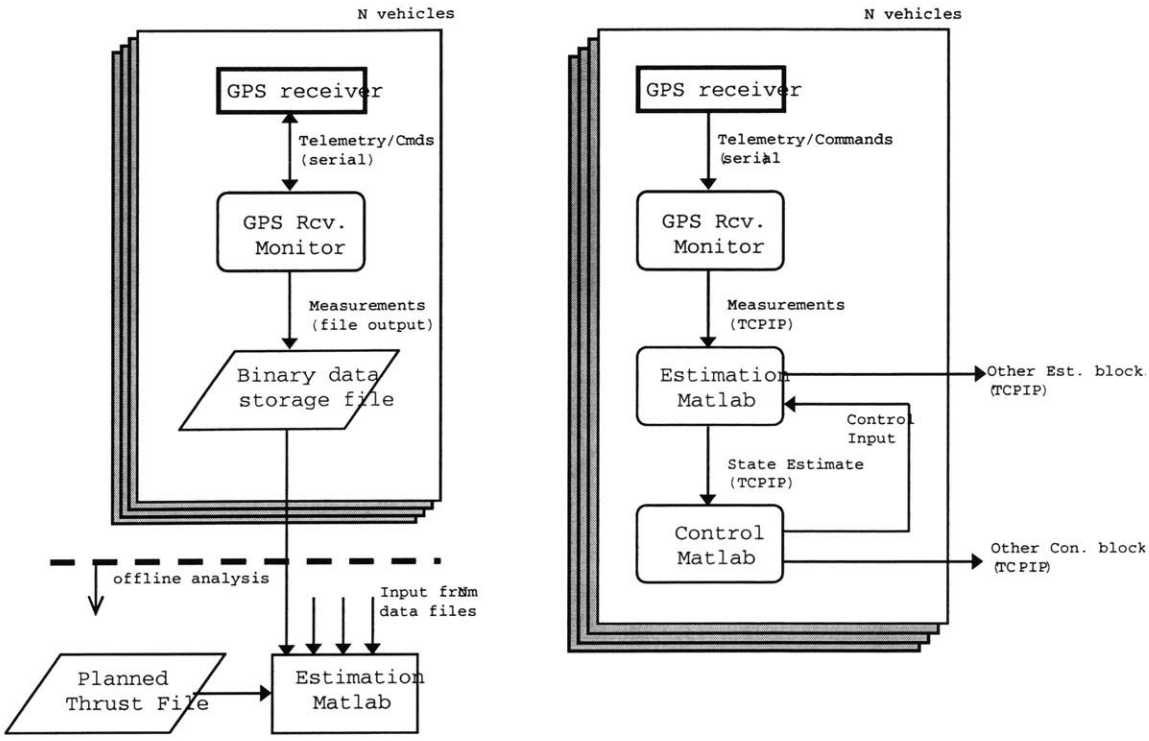

Fig. 5–1: Comparison of Original (left) and Current (right) Estimation Simulation Architectures.

Estimation block communicates with the local Control block: the Estimation block on the Leader sends an absolute estimate to the Control block on the Leader, and the Estimation blocks on the Followers send relative estimates to the Control block on the local Follower. The Control blocks use the estimates to calculate control inputs for each vehicle, which are communicated back to the local Estimation blocks.

The decentralized formulations were developed in Ref. [51] and extended in Ref. [1]. The updated navigation testbed developed in this research truly implements this decentralization. Figure 5–2 illustrates the communication between the GPS-RM, Estimation blocks, and Control blocks for the Leader and Follower vehicles, for a fleet of n vehicles.

The increase in communication required for a closed loop scenario is considerable. The original system performed all computations in a single MATLAB thread, but this clearly is not sufficient for closed loop demonstrations. The system Estimation block read measurement data and pre-planned thrust sequences from files, and then

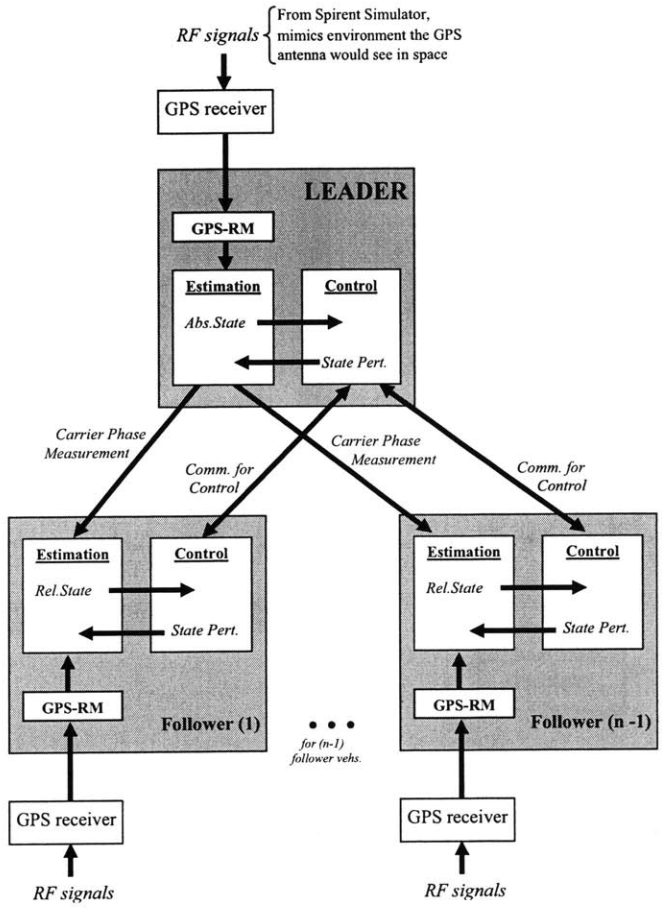

Fig. 5–2: Decentralized Communication between Navigation System Components in the Leader and Follower Vehicles, for a fleet of n vehicles.

computed estimates for all follower vehicles. The current system, operating online, requires navigation to be performed in separate MATLAB threads for each vehicle. The algorithm structure in the MATLAB code is identical to the structure that would be used in the C version. The use of MATLAB simplifies the implementation and debugging process, while preserving the functionality that would be provided with a C code implementation. To move from the architecture used by the original navigation system to the architecture desired for the updated navigation system, changes in several of the system components were required. A detailed look at the updated navigation system will highlight the nature of the required system changes.

Fig. 5–3 shows a detailed diagram of the information flow in the closed loop architecture. This represents the flow for one *follower vehicle.* The flow for the

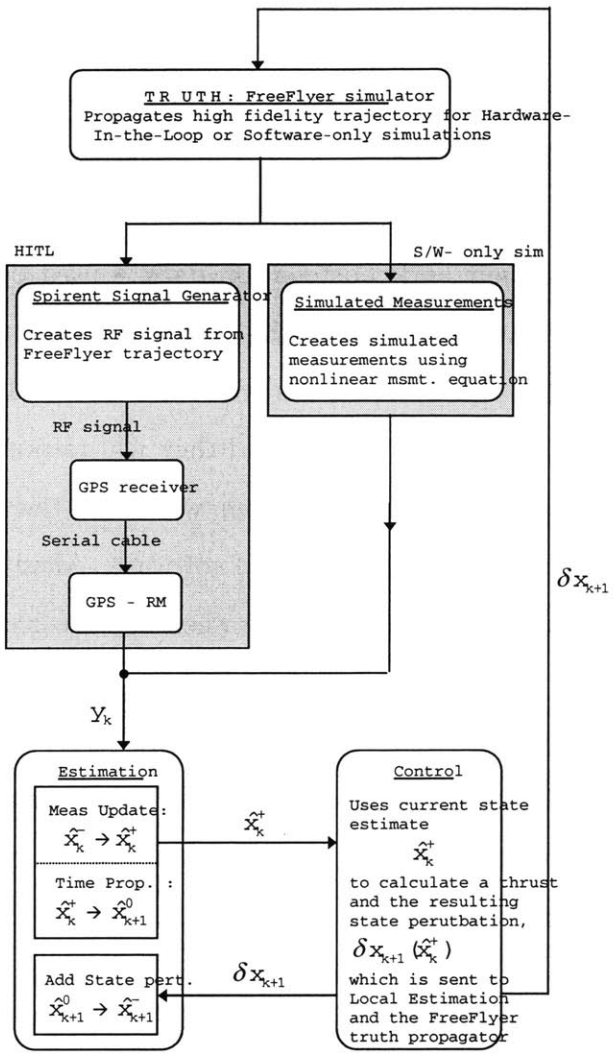

Fig. 5–3: Block Diagram of the Closed Loop Testing Setup

leader vehicle is similar, but the functions inside the Leader Estimation block are slightly different. The functions in the Leader and Follower estimation blocks are discussed in further detail in Section 5.2. Each simulation uses a truth model that is created using FreeFlyer^{TM}, which has a high fidelity orbital dynamics propagator and can model a wide range of perturbations [47]. Measurements are derived from the FreeFlyer^{TM} truth trajectory data in two ways. If a purely software simulation is desired, the measurement equations can be used to directly create a measurement, including additive white noise if desired. This option is useful for some debugging and initial demonstrations, but hardware-in-the-loop (HITL) simulations require inserting a GPS receiver between the FreeFlyer^{TM} truth model and the estimator.

For the HITL simulations, the FreeFlyer^{TM} truth trajectory is transmitted to the Spirent SimGen software [52]. SimGen reads the truth trajectory over TCP/IP, and exports this, as well as information about the GPS constellation, to the Spirent RF signal generator. The vehicle and constellation data is used by the signal generator to create an RF signal that mimics the GPS signal the vehicle antenna would receive if it were actually located in Space, on the truth trajectory. The receivers can then be connected to the simulator "antennas", and they will respond as if in orbit. If the HITL simulation branch is chosen, the receiver will track the RF signal and output carrier measurements at 1 Hz to the GPS-RM software. After startup, the GPS-RM software functions primarily to pass data between the receiver and the Estimation block.

The measurements $\mathbf{y}_k$, whether the real or simulated, are transmitted to the local Estimation block over TCP/IP. To form the CDGPS measurement, the Follower must also acquire both the local GPS measurement and the Leader vehicle's measurement. The communication link between the Leader and Follower Estimation blocks that enables this exchange, is shown explicitly in Fig. 5–1 (it is implied in Fig. 5–3). The Leader vehicle transmits its carrier phase measurement to each of the Followers, which will difference the Leader's measurement with its own phase measurement (with matching time tag) to create a differential measurement [1]. After the differential measurement is calculated, the Estimation block performs a measurement update of the state $\hat{\mathbf{x}}_k^- \rightarrow \hat{\mathbf{x}}_k^+$. The next step in the estimation process is the time propagation from $\hat{\mathbf{x}}_k^+ \rightarrow \hat{\mathbf{x}}_{k+1}^0$. The state $\hat{\mathbf{x}}_{k+1}^0$ represents the estimate which is propagated without the control. An explanation for handling the control is given in the following.

After the propagation, the estimator must incorporate the feed forward control input from the Control block. This feeding forward of the control commands is absolutely necessary, because if the control block implements a thruster firing and the estimator does not model it, the estimate will diverge. The Control block uses the updated state estimate to form a control input. The controller uses a Zero-Order-Hold control model, which is discussed further in Section 5.2 and in Ref. [20]. This type of control model sends the estimator a state perturbation, $\delta\mathbf{x}_{k+1}$, which is added

Table 5.1: Summary of Code Changes Made to Accommodate Online Estimation & Control

Code	Description
Estimation MATLAB Code	Distribute for real time decentralized execution, Add communication links between Leader & Follower Estimation blocks, GPS-RM, and Control; Change control implementation to complement ZOH controller
GPS Receiver Monitor	Convert to C++, add realtime measurement output via TCP/IP
Control MATLAB Code	Add comm. links to FreeFlyer and Estimation; Other control algorithm work [20]
Truth Simulator	Implement FreeFlyer simulation; and add comm. links from FreeFlyer to SimGen and Control [20]

to the propagated state $\hat{\mathbf{x}}^0_{k+1}$ to represent the effect the control input will have on the state [53]. The Control block also sends the state perturbation, which is equivalent to a control input in a ZOH model, to the FreeFlyer^{TM} simulator, so the control can be implemented in the truth model. The state perturbation is added to the propagated state $\hat{\mathbf{x}}^0_{k+1}$ to produce $\hat{\mathbf{x}}^-_{k+1}$, which completes the estimation loop.

The changes required to implement this architecture are summarized in Table 5.1. The changes associated with GPS and estimation are addressed in this thesis. The changes associated with the Control and Truth models are addressed in Ref. [20].

5.2 Real Time Estimation

The original Estimation code was written to read measurement and control inputs from a file. The relative estimates for all vehicles were computed in a single MATLAB thread, and the estimates were stored in a file. The updated architecture requires data interfaces between the local GPS-RM and Estimation blocks, between the Leader and Follower Estimation blocks, and between the local Estimation and Control blocks. Communication between other elements of the system, including those between the Control blocks, between FreeFlyer^{TM} and the Control, and between FreeFlyer^{TM} and

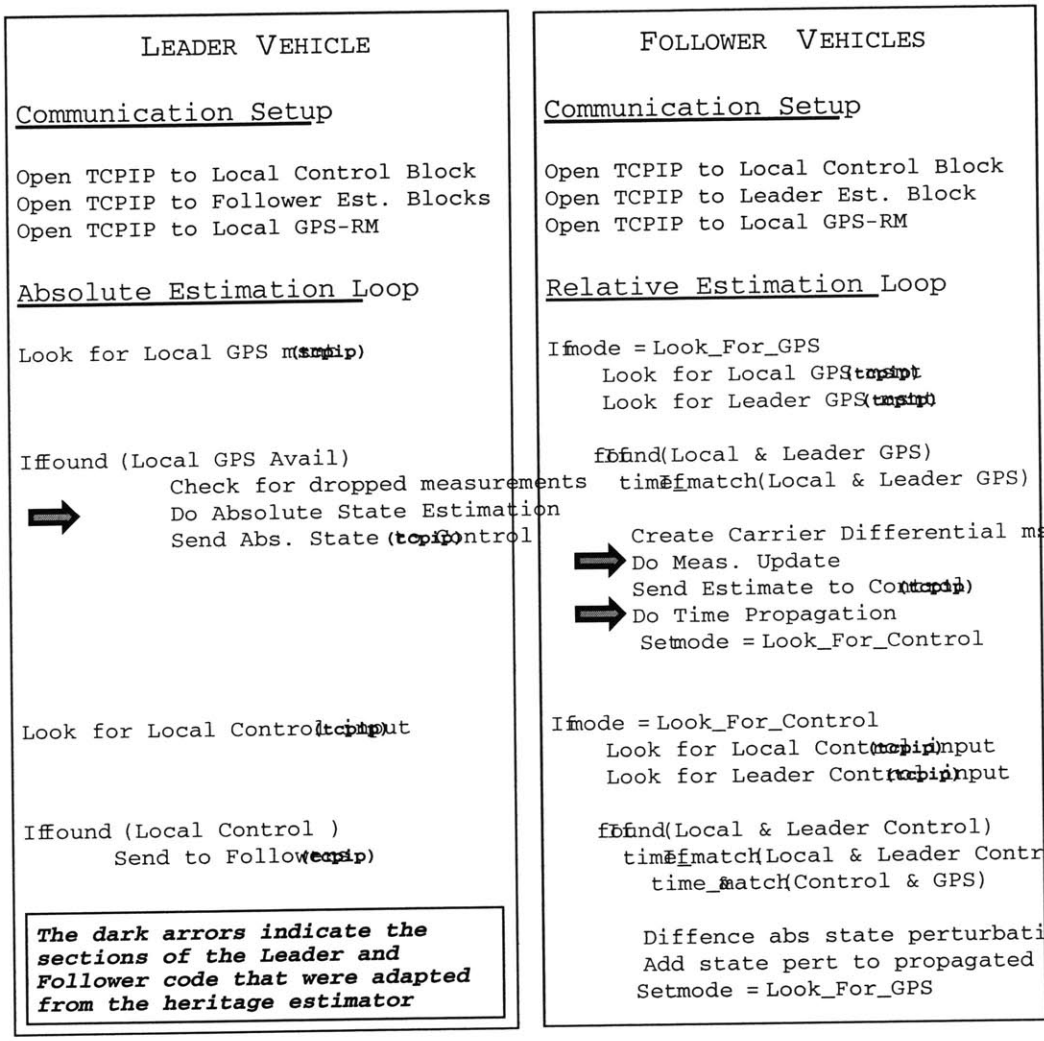

Fig. 5–4: Pseudocode for the Leader and Follower blocks of the Navigation Executive

SimGen, must also be executed in a timely manner. The second group of communication interfaces is discussed in Ref. [20].

A Navigation Executive was written in MATLAB to oversee the functions of the Estimation block and the communication required by the Estimation block. The Navigation Executive logic for the Leader and Follower vehicles is shown in two pseudocode blocks in Fig. 5–4. Both blocks begin by opening the TCP/IP communication connections to the local Control and local GPS blocks, and to the appropriate estimation blocks. The local Control and local GPS block refer to the Control MATLAB thread and the GPS-RM session on the same vehicle as the particular estimation block. These communication interfaces are shown in the block diagram in Fig. 5–3.

The Absolute and Relative Estimation Loops inside the Navigation Executive include the steps that are executed continuously. Logic that performs time checking and synchronization is a very important in ensuring the Absolute and Relative Navigation Loops work together properly. The Absolute Estimation Loop has minimal logic to verify that the incoming measurements and control inputs have the proper time stamps and are in the proper sequence. The burden of time checking is placed primarily on the Relative Estimation Loop. The simulation setup used a version of MATLAB TCP/IP that includes automatic error checking, which is slow, and was run on machines with Windows XP, which is not a real time operating system. Occasionally, the machine would "hang," delaying data from the GPS-RM or the Control block. This can cause the Navigation Executive to, for example, receive two GPS measurements before the control measurement. Because the Relative Estimation Loop requires certain packets of information before proceeding with each step, communication synchronization is critical. To allow ample time for all the communication and computation to occur, the Estimation and Control blocks were executed at a 4 second time step. This enabled the system to better tolerate the intermittent lags and alleviated some of the timing and synchronization problems.

In addition to increasing the time step, a time checking system that involved buffering and dumping measurements and control inputs helps the Relative Estimation System maintain synchronicity. The Relative Estimation Loop in the Navigation Executive stores all incoming data in buffers. When the times of two inputs (either Control or GPS) match the time of the next expected input, the Relative Estimation Loop will proceed. If the loop waits for a certain period of time for an expected data input and does not receive it, it will proceed without it. If the missed input is a GPS measurement, the current estimate is just propagated forward in time. The late measurement must then be dumped when it finally arrives[1]. If the missed input is a control input, the control input is set to zero and the filter proceeds. The current setup has similar logic in the Control block to ensure that control is not implemented

[1]It is possible to still use this late measurement, but if new measurements have already arrived, it will be of limited value.

on the system if the planned input is not sent to the Estimation block in a timely manner.

The tasks in the Absolute Estimation Loop of the Navigation Executive are fairly straightforward. The Absolute Estimation Loop looks for a local GPS measurement on the GPS-RM TCP/IP connection, and, upon finding one, verifies the measurement carries the appropriate time stamp. It then calculates an absolute state estimate and sends it to the local Control block. The Absolute Estimation Loop also sends the raw GPS measurements to the Follower Estimation blocks, which creates a Carrier Differential measurement from the raw Leader and Follower measurements. The Absolute Estimation Loop also continuously scans for incoming control inputs from the local Control block. Upon finding one, it sends the control input to the Follower vehicles.

The Relative Estimation Loop of the Navigation Executive has two modes of operation *Look_for_GPS* and *Look_for_Control*, to help it perform the required time checking and synchronization. In the first mode, the executive polls the Local GPS-RM and the Leader Estimation ports to collect the Local and Leader GPS measurements. When *both* are found, the time matching functions perform the necessary data buffering and dumping to ensure the measurements are synchronized. The appropriate pair of Local and Leader GPS measurements is then differenced by the Follower to create a CDGPS measurement, which is used in the measurement update step.

After the measurement update, the next step of the filter is the time propagation. The standard form of the Kalman filter models the control input in the dynamics equations used in the propagation. This is the model used in the original navigation system. However, the Control block used in the updated system uses a different model. Because even small errors in the way the control input is modeled in the filter can result in large estimation errors, the original estimation algorithm was changed to properly account for the different control model. The Control block uses a Zero-Order-Hold (ZOH) form of he controller and calculates a state perturbation that represents the change in state that the thruster firing would cause. To accurately reflect the ZOH model, the Estimation scheme implements a state perturbation rather than trying to

model the control input in the dynamics equation. This means that the state $\hat{\mathbf{x}}_k^+$ will be propagated (with no control input), to produce an intermediate state, $\hat{\mathbf{x}}_{k+1}^0$. When the state perturbation $\delta\mathbf{x}_{k+1}$ is available, it is added to the propagated intermediate state to produce,

$$\hat{\mathbf{x}}_{k+1}^- = \hat{\mathbf{x}}_{k+1}^0 + \delta\mathbf{x}_{k+1} \tag{5.1}$$

The time propagation can be done before the state perturbation is received, which helps simplify the time synchronization problems.

The Relative Estimation loop transitions from the *Look_for_GPS* mode to the *Look_for_Control* mode after the time propagation. In the *Look_for_Control* mode, it scans the Local Control and Leader Estimation ports until it has control inputs (*i.e.,* absolute state perturbation) from both the Leader and the Follower. A time stamp validation similar to that done for the GPS measurements is also performed. The relative state perturbation is calculated and added to the propagated state to form $\hat{\mathbf{x}}_{k+1}^-$. The loop is completed when the mode is reset to *Look_for_GPS*.

5.3 Receiver Monitor Code Changes

The GPS Receiver Monitor (GPS-RM) code serves as an interface between the GPS receiver and the MATLAB Estimation block. The original monitor code was written in *C* for a Borland compiler. The GPS-RM provides a DOS-based interface that displays real time telemetry and provides limited receiver commanding capabilites. Additional commands were added by Busse to complement upgrades to the receiver firmware [54].

Previous estimation work using this receiver has been performed off-line, using data files that were written by the GPS-RM software and contain raw GPS measurements. To perform online estimation, the real time measurements must be provided to the MATLAB Estimation block. One simple way to import data into MATLAB in real time was to use TCP/IP communication. The GPS-RM *C* code was converted to *C++* so that TCP/IP communication could be added. The new version of the GPS-RM is able to write the measurements to a file and/or output them in real time over

TCP/IP. This ability to output measurements in real time enables demonstrations of online estimation and closed loop navigation.

The Estimation code depends on acquiring GPS measurements from the GPS-RM at regular intervals. Though time checking logic was added to the Estimation block to account for late or dropped measurements, the measurement time stamps must be aligned with equally spaced intervals. For example, the time stamps should be multiples of every n seconds, like in the sequence,

$$[\ 122556.9999 \quad 122558.9999 \quad 122560.9999 \quad \ldots\ ]$$

However, it was observed on several occasions that a sequence of time stamps would more closely resemble:

$$[\ 122556.9999 \quad 122558.9999 \quad 122561.4444 \quad 122563.4444 \quad 122564.9999 \quad \ldots\ ]$$

This would cause the Estimation block to crash. The cause of this behavior was found in the GPS-RM code. The GPS-RM code originally output code when a set number of seconds measured by the system clock had passed. The machines were heavily taxed when the GPS-RM code, the Estimation block MATLAB, and the Control block MATLAB were running. Windows XP is not a real time operating system, and sometimes system lags would delay the GPS-RM when it tried to read the system clock. The GPS-RM was modified so that it would send measurements based on the time tag associated with the data from the GPS receiver. This made more sense and fixed some of the timing related problems seen during testing.

5.4 Hardware Closed Loop Navigation Tests

The architecture to support online estimation and closed loop navigation demonstrations was discussed in the previous sections. Changes in the Estimation software and the GPS-RM code that support these goals were discussed in detail. The Control system used in this navigation system is the topic of research in Ref. [20].

This section discusses hardware tests that helped develop these architecture changes, which were performed at NASA GSFC. These HITL test used FreeFlyer^{TM} to simulate a truth trajectory. The Spirent signal generator mimics RF signals for the GPS receivers. Thus, executing the actual estimation and control algorithms with measurements from real hardware is possible. Section 5.4.1 discusses the equipment setup at GSFC used for HILT testing. Section 5.4.2 presents results that demonstrate the infrastructure for closed loop navigation.

5.4.1 Closed Loop Test Setup

Fig. 5–5 shows the setup at NASA GSFC, with the major system components highlighted. GSFC provided the Spirent desktop that ran the SimGen software, and the Spirent Signal Generator. The other components of the navigation system were provided by MIT. Four Orion GPS receivers were available. Each vehicle in the formation was simulated with a separate laptop that ran the GPS-RM software, an Estimation MATLAB thread, and a Control MATLAB thread. The implementation of control commands was simulated by incorporating the state perturbations in the FreeFlyer^{TM} truth orbit propagator, which was run on a separate laptop.

Connecting these components together required several types of communication links. The RF signal was transmitted via coaxial cable to the GPS receivers. The GPS receivers communicated with the GPS-RM code over serial cables. TCP/IP communication was used for interfaces between the GPS-RM, the Estimation blocks, the Control blocks, and the FreeFlyer truth model. These interfaces were discussed in previous sections and were also diagrammed in Fig. 5–1. The TCP/IP communication between vehicles was enabled by putting all the laptops on a local network. The Spirent desktop was also on the local network.

5.4.2 Closed Loop Demonstrations

The testbed was developed to simulate formations of multiple vehicles. Four vehicles were simulated during recent GSFC tests, but the infrastructure can accommodate

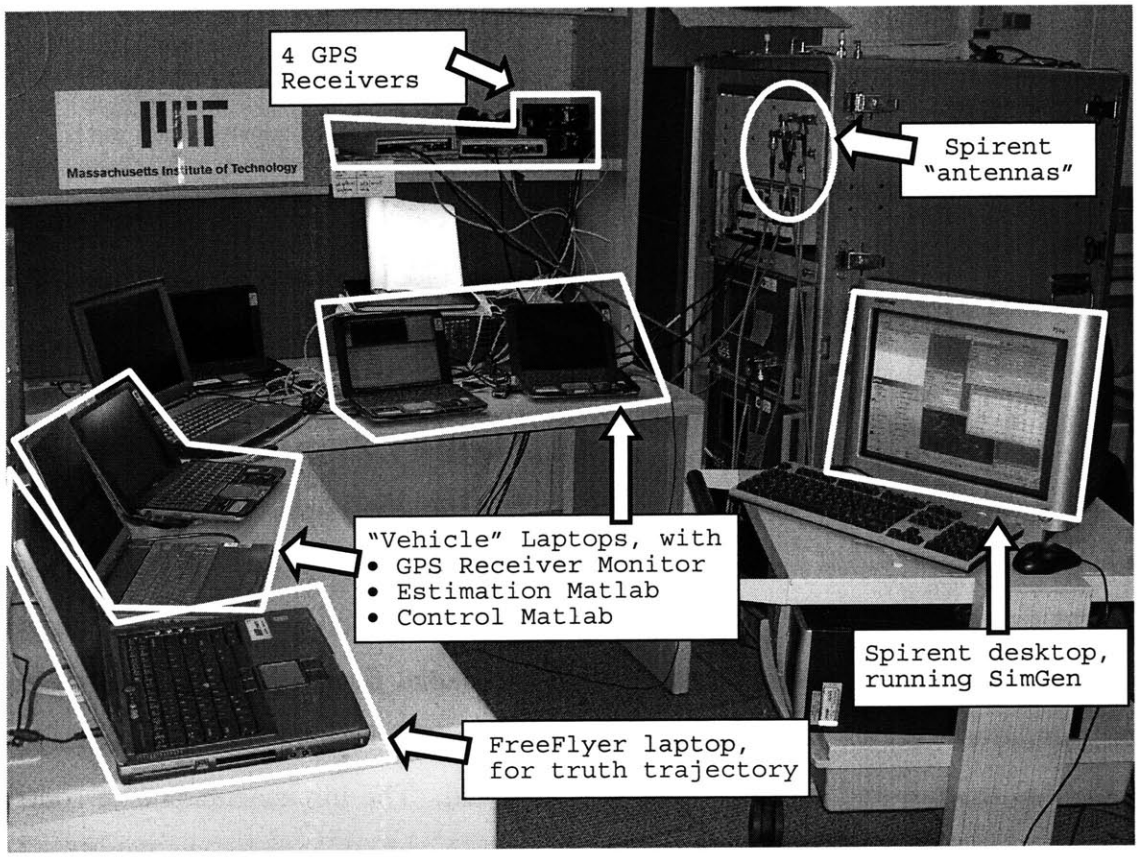

Fig. 5–5: HITL Navigation Demonstration Setup at GSFC

additional vehicles if desired. The demonstrations discussed previously report the results for a pair of vehicles in LEO, with in-track separation of 20 m. Three demonstrations were done to show the system evolving from the original architecture to an architecture that can perform closed loop navigation tests:

- Demo 1 – Online Estimation, no communication interfaces with Control established
- Demo 2 – Online Estimation, communication interfaces with Control established, but control is zeroed out
- Demo 3 – Closed Loop Navigation

The primary goal of these tests was to demonstrate the successful development of the infrastructure needed to accomplish closed loop navigation. The three progressive

tests allowed the gradual build-up of the system.

Demo 1 incorporates changes that allow real time estimation. The original system was expanded by adding communication links between the GPS-RM and Estimation blocks, decentralizing the Estimation computation to the individual vehicles, and adding communication links between the Leader and Follower Estimation blocks. The GPS-RM module on each vehicle reads the GPS measurements and transmits them to the local Estimation block in real time. The Leader Estimation block then transmits its measurements to the Follower vehicles so they can create CDGPS measurements. The Follower Estimation blocks then compute, display, and store estimates in real time. In this first test, the communication link between Estimation and Control is not established. This results in an entirely open loop scenario.

Figs. 5–6 and 5–7 show the performance of the online estimation in Demo 1. The performance is similar to levels seen in previous research [1]. The filter produced position errors of ~ 3 cm, and velocity errors of $\sim 0.2 - 0.5$ mm/sec. These results demonstrate that online estimation can provide good performance.

The next two demonstrations focused less on performance and more on demonstrating that the infrastructure for closed loop navigation was completely functional. While some of the communication interfaces can be tested without hardware, many of the timing and synchronization issues discussed in Section 5.2 are not apparent until the hardware is included. Much of the work at GSFC involved troubleshooting these unforseen problems and developing logic to make the Estimation and Control blocks robust to implementation on the real hardware. The estimation and control algorithms have each been shown to work well individually. The infrastructure developed in this work provides the apparatus to further refine the closed loop performance when the two algorithms are working together.

Demo 2 builds on the online estimation shown in Demo 1 by adding the communication links to the Control blocks. In addition to computing the estimate in real time, the Estimation block now transmits that estimate to the local Control block, and reads a control input (which is zeroed out for Demo 2) at each time step. While this may not appear to add much new functionality, Demo 2 incorporates additional

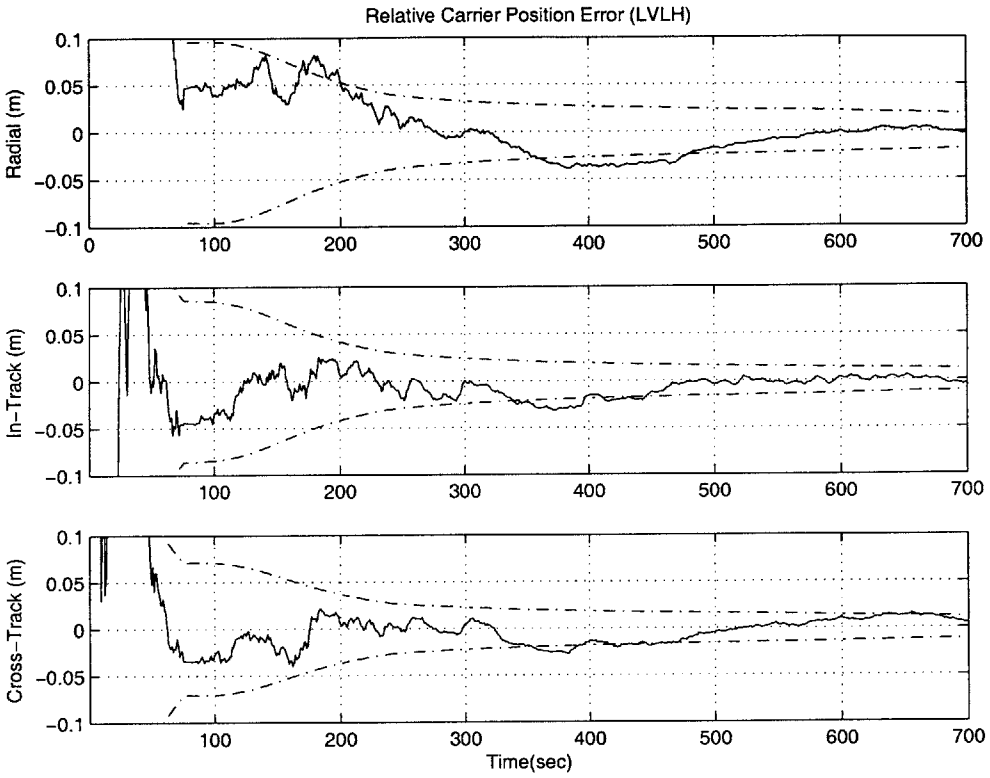

Fig. 5–6: Demo 1:Relative Position Error

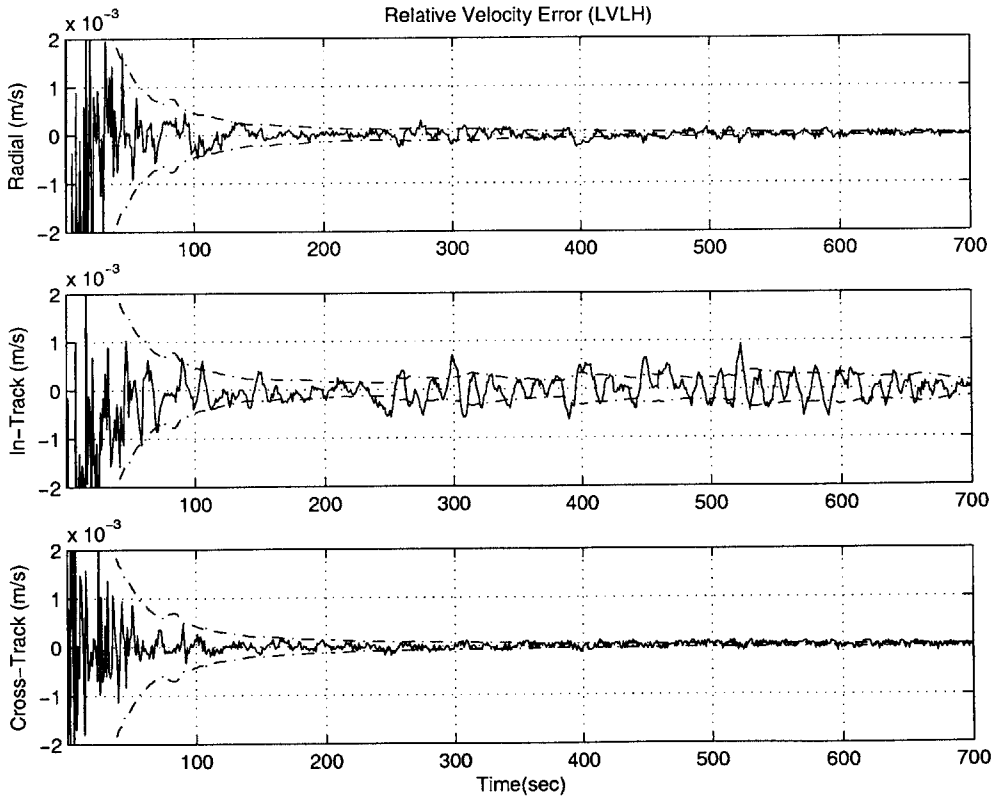

Fig. 5–7: Demo 1: Relative Velocity Error

synchronization and timing issues logic so that the control communication links do not disrupt the estimation process. The filter nominally requires a control input packet at every time step. If the control input is received late, the filter will fall behind, in the sense that the state perturbation will not be added until after the next measurement is received.

The results of Demo 2 are shown in Figs. 5–8 and 5–9. Fig. 5–10 was included to show the number of GPS Satellite Vehicles (SVs) for which both the Leader and Follower have measurements. Around 700 seconds into the demonstration, and at 4 other times, the number of SVs momentarily drops to 0. This occurs when the system hangs. If one of the vehicle laptops delays the communication of the measurement between the GPS-RM and the local Estimation block, or delays the communication of the Leader measurement from the Leader Estimation block to the Follower block, the filter will not be able to produce a CDGPS measurement in time. The filter then "gives up" on receiving the measurement for that time step, and propagates the current estimate forward in time. When the filter does not receive a measurement in time, it is usually not because the receiver drops the measurement, but because a delay occurs somewhere in the communication cycle. Thus, the measurement will eventually arrive at the Estimation block, but must be dumped because it is late and the filter has already propagated past the time of the measurement. The filter buffers and dumps until once again it has Leader and Follower measurements that are on time and synchronized. Depending on the length of the original hang time, it may take the filter several time steps to get back in synch. During this period, no measurements are available to the filter, and the quality of the estimate will degrade. This is seen in the velocity error in Fig. 5–9, at times corresponding to the measurement dropouts.

While the brute force method of buffering and dumping measurements to ensure synchronization may not be the best approach, it does allow the communication links necessary for closed loop navigation to be established. The infrastructure now has the functionality necessary to support closed loop navigation. Testing found that a time step of 4 seconds, which was used in Demo 2, normally allowed all the communications and calculations to be completed in a timely manner. However, system "hangs" still

caused measurement drop outs at a 4 second time step, which is why the estimator performance in Demo 2 (Figs. 5–8 and 5–9) is not as good as it was in Demo 1 (Figs. 5–6 and 5–7). With a longer time step, the system has more time to wait before dropping late measurements. For example, the 10 second time step used in Demo 3 was sufficient to ensure that any delayed measurements almost always arrived in time to be used.

There are several other approaches to fix these measurement dropouts and the associated performance degradation seen in Demo 2. These include using a computer with a faster processor, using a faster TCP/IP library , or eliminating or reducing the processor-intensive real time display functions of the GPS-RM. While system "hangs" can cause performance degradation, if a sufficiently long time step is used or if these infrastructure issues are addressed, then the online estimation will yield good performance.

The closed loop navigation example for Demo 3 was chosen to show two vehicles that are driven from a 20 m in-track separation to a 10 m separation. Fig. 5–11 shows plots of the control input magnitude and the vehicle separation. The estimator runs for around 1700 seconds with no control inputs, so the filter will have plenty of time to converge. If control is initiated before the estimate has converged, the filter and thus the controller, will likely diverge rapidly. When the control is turned on, the vehicle separation, reflected in the estimate, decreases steadily. Demo 3 shows two significant points about the estimator:

- It is correctly incorporating the feed forward control state perturbations
- It is providing a sufficiently accurate estimate to the Control block

Thus, the navigation system infrastructure has demonstrated sufficient ability and robustness to host closed loop tests for estimation and control algorithms.

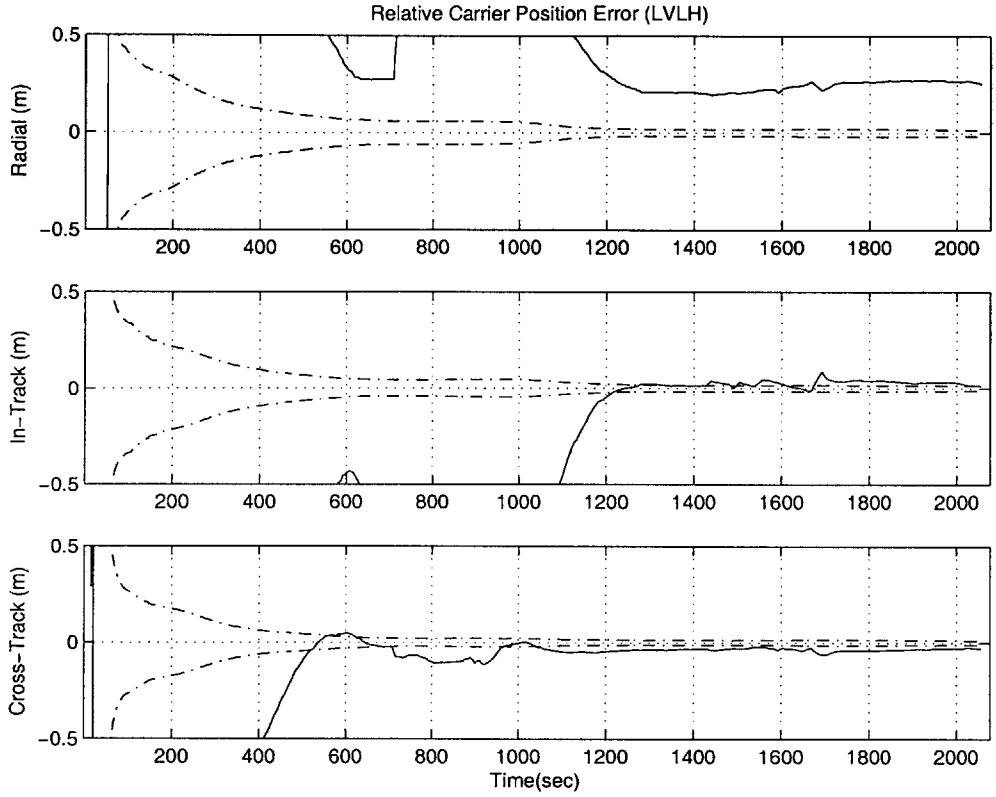

Fig. 5–8: Demo 2: Relative Position Error

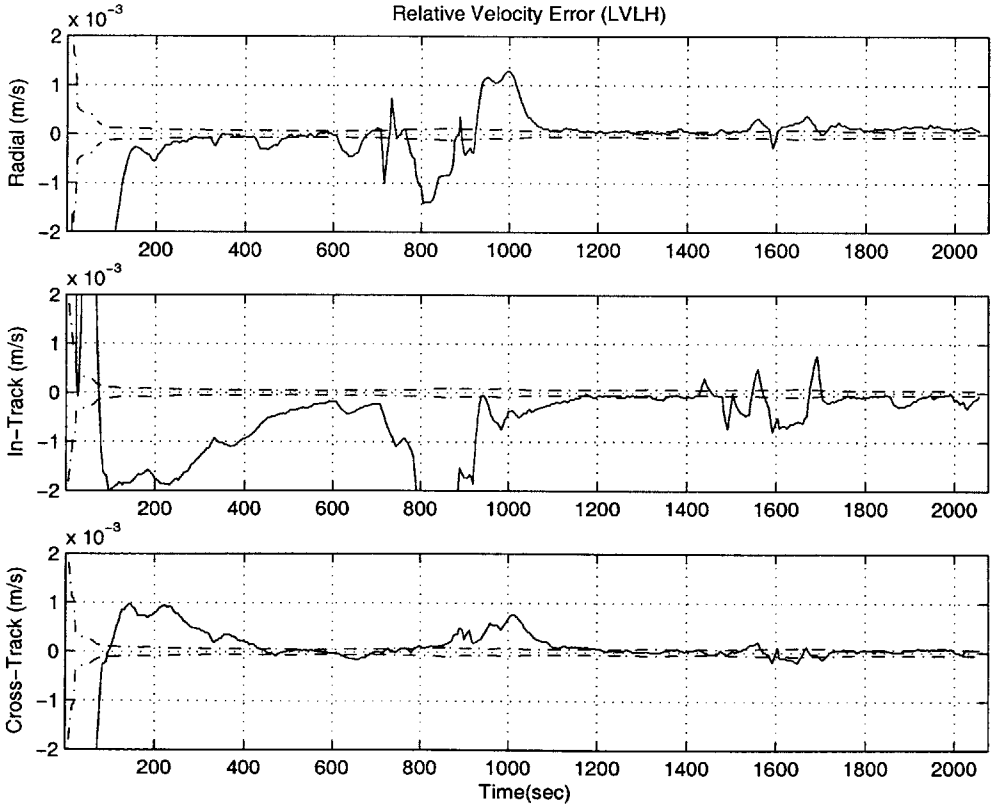

Fig. 5–9: Demo 2: Relative Velocity Error

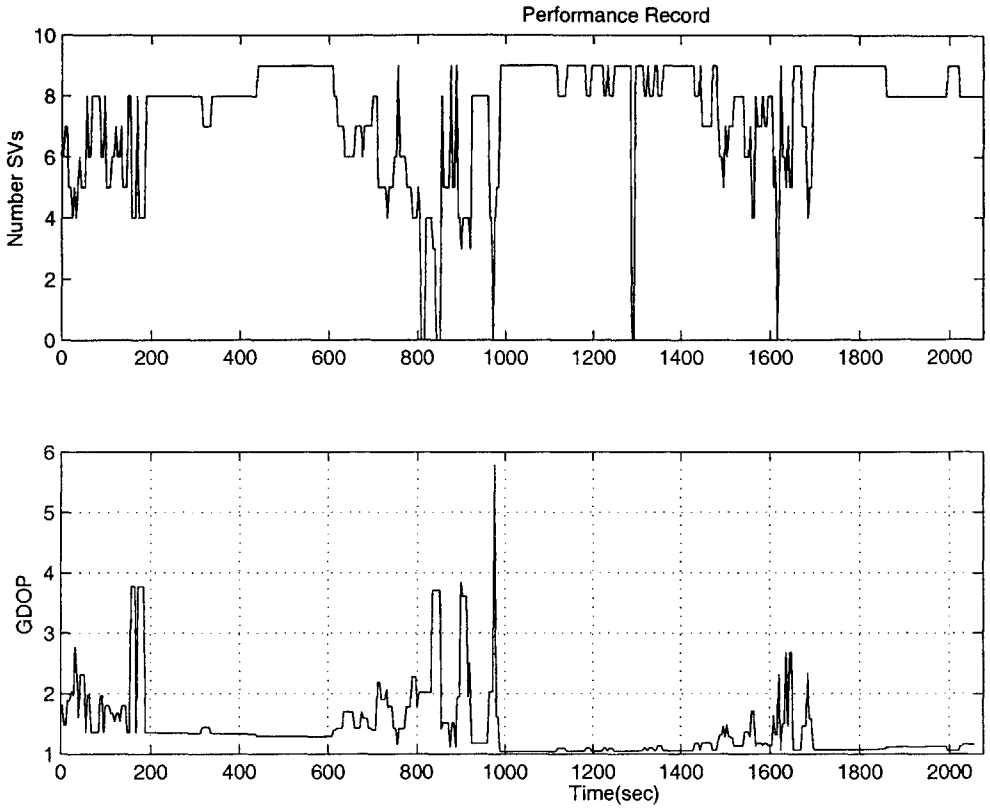

Fig. 5–10: Demo 2: Satellite Tracking

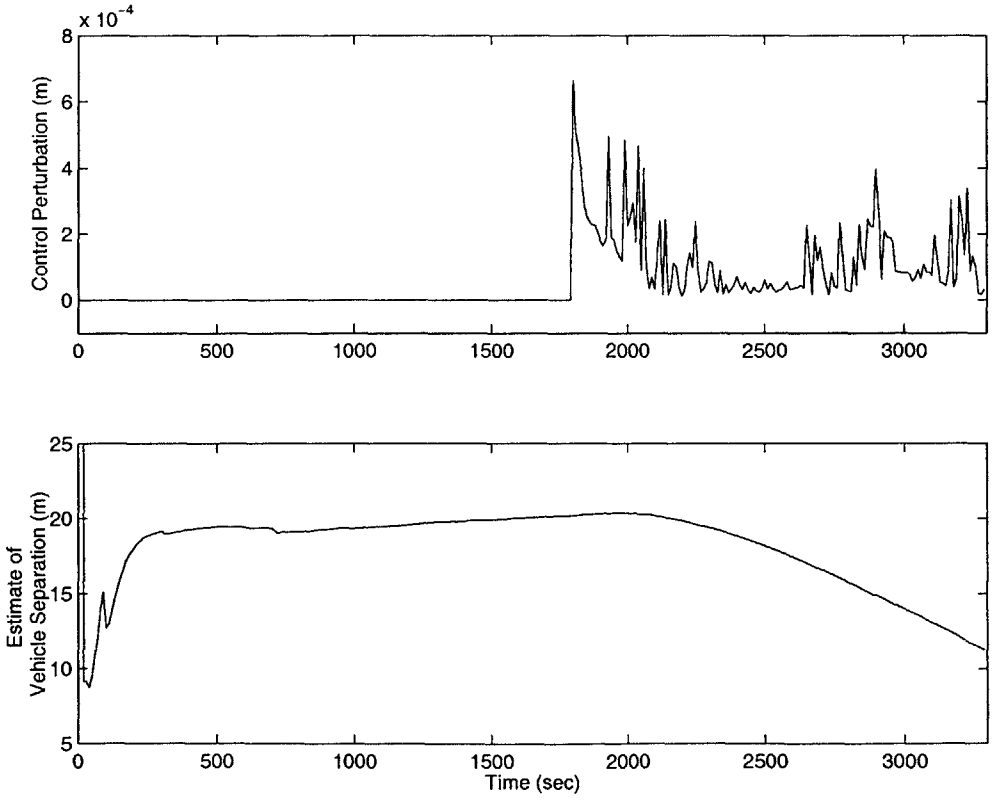

Fig. 5–11: Demo 3: Closed loop navigation; Control Inputs and Estimate of Vehicle Separation shown

5.4.3 Future Work for Closed Loop Demonstrations

The closed loop navigation demonstrations with the updated architecture will not emulate a spacecraft navigation system with complete accuracy. However, the goal of this setup was to demonstrate the Estimation algorithm, the Control algorithm, and the two algorithms working together with real hardware in a closed loop scenario. To maintain this focus on the algorithms, some realism was sacrificed by executing the algorithms in MATLAB, instead of *C*, by hosting the algorithms on Windows laptops, rather than real time processors, and by performing inter-vehicle communication with TCP/IP over a network, rather than with a wireless modem system that might be used on future microsatellites [55]. However, the flexibility and portability of our architecture is very well suited for algorithm development, which ultimately is the focus of this research. The navigation system developed for closed loop simulations, and the complementary control work done in Ref. [20], provides the framework for improving the future closed loop demonstrations on the hardware and, eventually, in Low-Earth Orbit.

Chapter 6

Conclusion

6.1 Thesis Contributions

This thesis presented simulations and discussions that help reveal fundamental relationships between the dynamic system, the measurements, and the filter. This knowledge can be used to highlight steps to improve navigation for formation flying missions. The following outlines the main results and contributions of this thesis:

Correlation as a Design Goal

Previous research suggested that, to minimize semimajor axis estimation error, the correlation coefficient $\rho_{x\dot{y}}$ between the radial position and in-track velocity should be large. This thesis investigated that conjecture for the Kalman filters designed for relative navigation of spacecraft using Carrier-phase differential GPS measurements. The standard CDGPS results typically yield correlation coefficients close to -0.1, which are quite different than the -1 that had been associated with "good" navigation filters. Several examples were presented to show that the filter could be forced to create solutions with a stronger correlation, but these resulted in larger errors in the semimajor axis (*i.e.*, the performance was consistently *worse*). The best strategy for determining relative semimajor axis is to seek the optimal estimate of relative position and velocity, which is provided by the Kalman Filter.

Noise and Estimation Accuracy

An additional goal was to explore what parameters in the Kalman filter have the

most impact on the performance of the navigation accuracy. A linear, planar model was used to reveal the basic relationships between measurement and process noises, and position, velocity, and semimajor axis accuracy. These simulations were shown to be consistent with recently developed analytic prediction of the correlation coefficient. Simulations with a nonlinear, GPS-based model showed very similar trends. When realistic errors were introduced in clock, ephemeris, and absolute state, additional limitations of complex CDGPS-based filters were observed. Generally, when the noise on the carrier differential phase measurement is decreased, a corresponding decrease in position error is seen. However, in the nonliear models, the continued improvement in position stalls after a point, suggesting that the realsistic errors from clock, ephemeris, and absolute state, will dominate CDGPS measurement decreases below a certain, low noise level.

Unscented Kalman Filter in Relative Navigation

Busse [1] explored the effect of nonlinearities in the estimation problem for relatively short vehicle separations and time steps. The goal of this work was to consider the filter design for larger separations ($\sim$ 10 km) and time steps ($\sim$ 60 sec). An Unscented Kalman filter (UKF) was developed for the relative orbital navigation because the UKF was developed to handle nonlinearities in the measurement and dynamics equations better than the Extended Kalman Filter.

Several demonstrations were run using both simulated measurements calculated from FreeFlyer^{TM} trajectories and stored measurements from receiver tests at NASA GSFC. These tests spanned filter time steps between 5 and 60 seconds and included baselines of 0.1, 1, and 10 km. These examples clearly demonstrated that the EKF gave very poor velocity estimates at the larger separations and time steps, agreeing with results in Ref. [1]. In systems with pronounced nonlinearities, the UKF is recommended because its relative navigation performance was much better than the EKF for comprable computational burden.

HITL Testbed for Closed Loop Navigation and Control

A hardware-in-the-loop testbed was developed to demonstrate CDGPS technologies in closed loop. The testbed uses the Spirent Simulator at NASA GSFC, which creates

RF signals to emulate a space environment for the GPS receivers. A Navigation Executive was written to coordinate the communication between system elements and execute the steps of the decentralized Kalman Filter. Changes were also made to the GPS Receiver Monitor software to support a closed loop architecture. The decentralized algorithms developed by Park and Busse supported real time applications, but they were not tested in real time. This work extended those algorithms for real time implementation by developing the communications infrastructure between the algorithmic and hardware components of the navigation system. Demonstrations of the new setup were performed at NASA GSFC which showed that the testbed has sufficient flexibility and robustness to host future closed loop experiments for estimation and control algorithms.

6.2 Future Work

Part of the motivation for this thesis was to understand the factors that influence filter performance. The results from Chapter 3 helped suggest avenues of work that will have the greatest impact on CDGPS-based relative navigation filters. The results suggest that improvements to the dynamics might be a first area of work. Experiments with the Unscented Kalman Filter also suggest additional work on the dynamics model would be useful. The UKF handles longer baselines and longer time steps better, so a high quality, computationally intense dynamics model could be accommodated with a UKF that has a time step greater than the current rate of 1/4 Hz.

The closed loop HILT testbed provides the infrastructure for future analysis of estimation and control algorithms. The testbed can also be used to evaluate the performance of GPS receiver algorithms. Future work might demonstrate estimation and control for a formation with long baselines and an eccentric orbit in the real time testbed. This would require the aforementioned changes to the dynamics model, the adoption of a GPS receiver that could track the GPS signal during the high dynamics of an eccentric orbit, and the patience of researchers during the much longer period of an eccentric orbit. If the orbit is extremely eccentric, the orbits will travel above

the GPS satellite constellation. In this case, the superior covariance propagation provided by the UKF may allow sufficient navigation through the long outages of GPS measurements.

Space navigation continues to present many challenges and opportunities, which are multiplied as the field is extended to formation flying satellites. CDGPS-based relative estimation presents an exciting area of research with much promise in the future.

Appendix A

Reference Frames

The references frames used in this thesis are based on definitions found in Ref. [28]. Descriptions and illustrations of the references frames appear below. The conversions between these reference frames are described in Ref. [28].

Earth Centered Inertial

The Earth Centered Inertial (ECI) reference frame, shown in Fig. A–1, is essentially fixed with respect to celestial bodies. The origin of the reference frame is the center of mass of the Earth. The x-axis extends through the equator, towards the first point of Aries, which is a fixed point in the celestial sphere. The z-axis is aligned with the axis of rotation of the Earth, pointing towards the North Pole, and the y-axis completes a right-handed coordinate system. The non-rotating ECI frame is useful for describing orbiting bodies whose motion is independent of the rotation of the Earth.

Earth Centered Earth Fixed

The Earth Centered Earth Fixed (ECEF) reference frame, shown in Fig. A–1, has coordinate axes that rotate with Earth. The origin of the ECEF frame is the same as the origin of the ECI frame, the center of mass of the Earth. The x-axis intersects the Greenwich meridian and the Earth's equator. The z-axis is aligned with the axis of rotation of the Earth, pointing towards the North Pole, and the y-axis completes a right-handed coordinate system. The ECEF and ECI frames are aligned at the

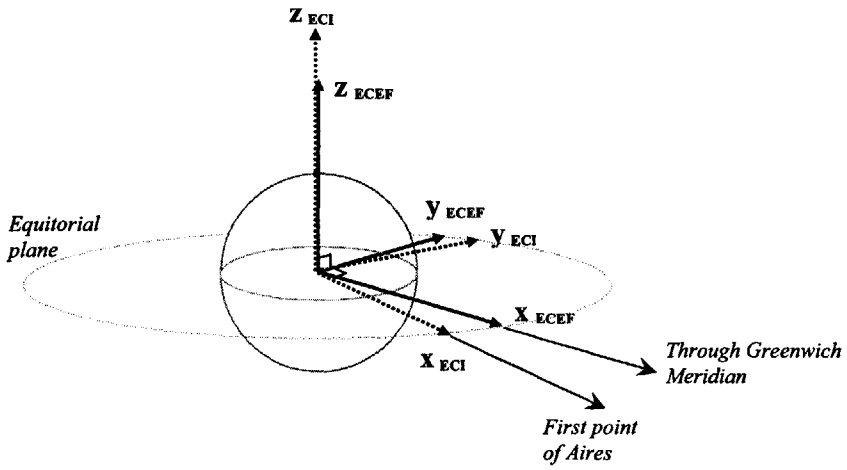

Fig. A–1: ECI and ECEF Reference Frames

moment when the Greenwich meridian is aligned with the first point of Aries. The ECEF reference frame rotates with respect to the ECI frame at the rotation rate of the Earth. The ECEF reference frame is not a natural frame for describing orbital motion, but it becomes important when relating to a location on the surface of the Earth. The Global Positioning System was created as a navigation system for objects on or near the surface of the Earth, and thus adopted the ECEF reference frame as a standard. The relative navigation in this work is done in an ECEF reference frame, and converted to ECI or the local frame described below as necessary.

Orbital Elements

Though not strictly a reference frames, a set of orbital elements provides the same information as a Cartesian state in ECI or ECEF. The Cartesian reference frames are defined completely independent of the notion of an orbit. The orbital elements are useful in visualizing the shape of an orbit and the location of a vehicle on the orbit. The orbital elements, shown in Fig. A–2, are Ω, i, ω, a, e, and θ. The first three elements, Ω, i, ω, describe the orientation of the orbital ellipse with respect to the Earth. Longitude of the ascending node, Ω, is the angle between x_{ECI} (or equivalently the first point of Aries), and the intersection of the equatorial plane and the orbit. In particular, this specifies the one of two intersecting points that corresponds to

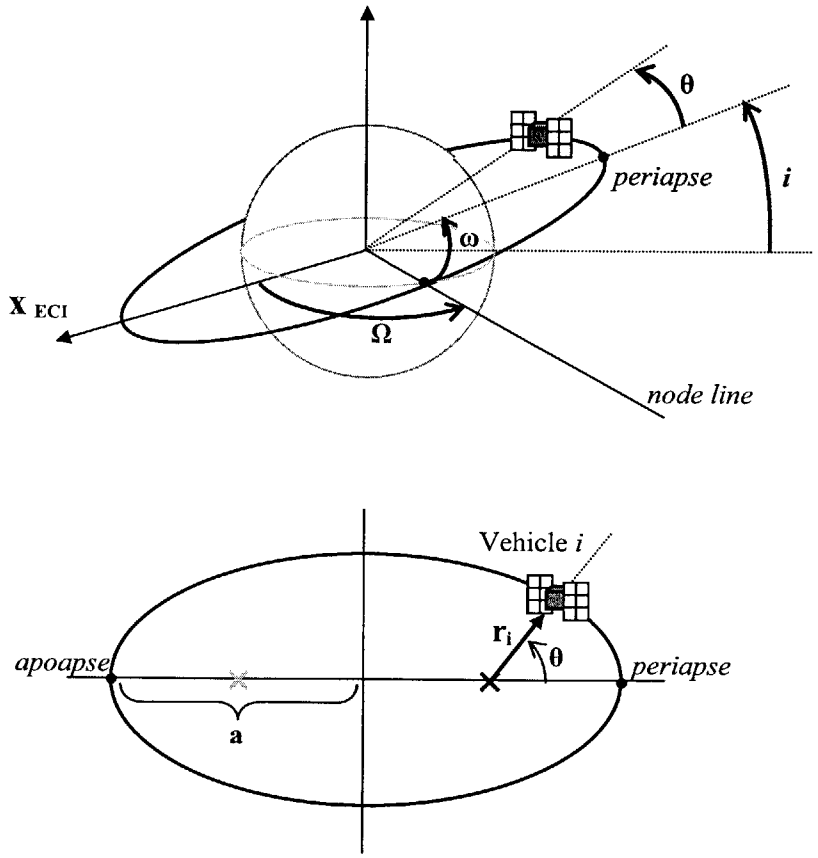

Fig. A–2: Orbital Elements

the location where the orbiting body crosses from the southern hemisphere to the northern hemisphere. The line through this point and the origin is the node line. The inclination, i, indicates the tilt of the orbit, or the angle between the equatorial plane and the orbital plane. The argument of perigee, ω, is the angle between the node line and the periapse of the orbit.

The next two elements, a and e, specify the size and shape of the orbital ellipse. The semimajor axis, a, is half the distance between apoapse and periapse, and thus determines the size of the orbit. The eccentricity, which is dimensionless, is related to the shape of the orbit, and essentially indicates how "squashed" the ellipse is. The eccentricity for elliptical orbits is between 0 and 1. A perfectly circular orbit has an eccentricity of 0, which results in a constant radius that is identical in size to the

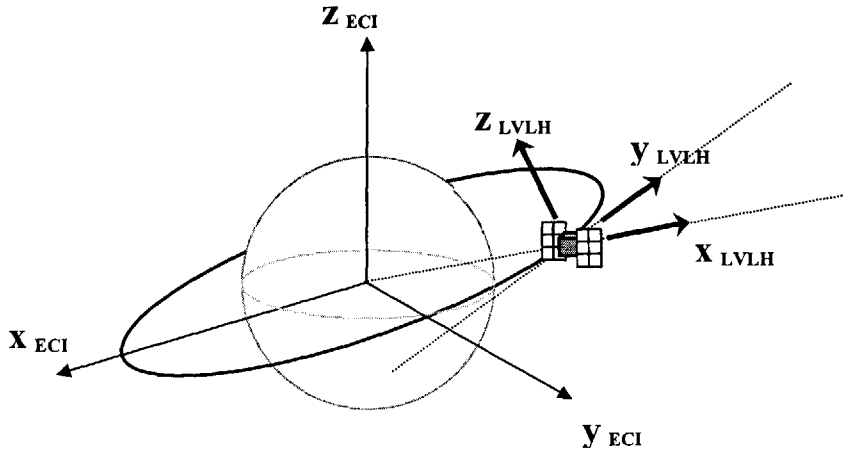

Fig. A–3: LVLH Reference Frame

semimajor axis.

The final orbital element, θ, indicates the position of the vehicle in the orbit, and is the angle between the vector through perigee and the vehicle's position vector. This set of elements is called the *Keplarian* orbital elements. Alternative representations are available in Ref. [28].

Local Vertical Local Horizontal

The reference frames presented so far have all originated at the center of the Earth. A reference frame that is locally based simplifies expressions for relative positioning. The Local Vertical Local Horizontal (LVLH) reference frame, shown in Fig. A–3, has an origin that is fixed to a convienient point on the vehicle From an inertial perspective, the axes of the reference frame rotate with the vehicle. The LVLH x-axis is always aligned with the radius vector from the center of the Earth to the vehicle. The y-axis is parallel and in the same direction as the velocity vector. The z-axis completes the right-handed system. The x, y, and z axes corresponds to the radial, in-track, and cross-track directions.

Bibliography

[1] F. Busse, "Precise Formation-State Estimation in Low Earth Orbit Using Carrier Differential GPS," Ph.D. Dissertation, Stanford University, Dept. Aeronautics and Astronautics, Feb. 2003.

[2] J. Leitner, F. Bauer, D. Folta, M. Moreau, R. Carpenter, J. How, "Distributed Spacecraft Systems Develop New GPS Capabilities," in *GPS World: Formation Flight in Space,* Feb. 2002.

[3] J. How, R. Twiggs, D. Weidow, K. Hartman, F. Bauer, "Orion - A Low-cost Demonstration of Formation Flying in Space using GPS," *Proceedings of AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Boston, MA, Aug. 10-12, 1998, Collection of Technical Papers (A98-37348 10-13), Reston, VA, American Institute of Aeronautics and Astronautics, 1998, p. 276-286.

[4] F. Bauer, K. Hartman, J. Bristow, D. Weidow, J. How, F. Busse, "Enabling Spacecraft Formation Flying Through Spaceborne GPS and Enhanced Autonomy Technologies," *ION-GPS '99*, Proceedings of the 12th International Technical Meeting of the Satellite Division of the Institute of Navigation, Nashville, TN, Sept. 14-17, 1999 (A01-27218 06-32), Alexandria, VA, Institute of Navigation, 1999, p.369-383.

[5] M. Tillerson, " Coordination and Control of Multiple Spacecraft using Convex Optimization Techniques," S.M. Thesis, Massachusetts Institute of Technology, Dept. Aeronautics and Astronautics, June 2002.

[6] M. Tillerson, G. Inalhan, and J. How, "Coordination and Control of Distributed Spacecraft Systems Using Convex Optimization Techniques," in *International Journal of Robust and Nonlinear Control,* Vol 12, Issue 2-3, Feb.-Mar. 2002, p.207-242.

[7] J. How and M. Tillerson, "Analysis of the Impact of Sensor Noise on Formation Flying Control," *Proceedings of the 2001 American Control Conference*, Arlington, VA, June 25-27, 2001, Proceedings. Vol. 5 (A01-45851 12-63), Piscataway, NJ, Institute of Electrical and Electronic Engineers, 2001, p. 3986-3991.

[8] F. Busse, J. How, J. Simpson, and J. Leitner, "PROJECT ORION-EMERALD: Carrier Differential GPS Techniques and Simulation for Low Earth Orbit Formation Flying," presented at the *IEEE Aerospace Conference,* Mar 10-17, 2001.

[9] E. Olsen, C.W. Park, and J. How, "3D Formation Flight using Differential Carrier-phase GPS Sensors," in *Proc. of the ION-GPS Conference,* Spring, 1999, Vol. 146, No. 1, pp. 35–48.

[10] F. D. Busse, J. Simpson, and J. P. How, "Demonstration of Adaptive Extended Kalman Filtering for LEO Formation Estimation Using CDGPS," *Navigation* Journal of the Institute of Navigation, Vol. 50, No. 2, Summer 2003, pp. 79–94.

[11] C.W. Park, J. How, and L. Capots, 'Sensing Technologies for Formation Flying Spacecraft in LEO Using Inter-Spacecraft Communications System," the *Navigation* J. of the Institute of Navigation, Vol. 49, No. 1, Spring 2002, pp. 45–60.

[12] S. Leung, O. Montenbruck, "Real-Time Navigation of Formation Flying Spacecraft using Single-Frequency GPS," submitted to *AIAA Journal of Guidance, Control and Dynamics*, Jan. 2004.

[13] G. Inalhan, M. Tillerson, J. How, "Relative Dynamics & Control of Spacecraft Formations in Eccentric Orbits," AIAA Journal of Guidance, Control, and Dynamics (0731-5090), Vol. 25, No. 1, Jan.-Feb. 2002, p. 48-59.

[14] J. Garrison, P. Axelrad, "Relative Navigation in Highly Elliptical Orbits Using an Iterative Nonlinear Filter," in *ION GPS Conference*, Sept. 1997.

[15] W. Bamford, T. Ebinuma, E.G. Lightsey, "Navigation of Large Autonomously Controlled Formations," in *AIAA Guidance, Navigation, and Control Conference*, Aug. 2003.

[16] E. Gill, B. Naasz, T. Ebinuma, "First Results from a Hardware-in-the-Loop Demonstration of Closed-Loop Autonomous Formation Flying," in *AAS Guidance and Control Conference*, Feb. 2003.

[17] J.R. Carpenter, E. Schiesser, "Semi major Axis Knowledge and GPS Orbit Determination," in *Journal of the Institute of Navigation* Vol. 48, No. 1, Spring 2001.

[18] J.R. Carpenter, K.T. Alfriend, "Navigation Accuracy Guidelines for Orbital Formation Flying," in *AIAA Guidance, Navigation, and Control Conference*, Aug. 2003.

[19] S. Julier, J. Uhlmann "Unscented Filtering and Nonlinear Estimation," in *Proceedings of the IEEE*, Vol. 92, No. 3, Mar. 2004.

[20] L. Breger, " Model Predictive Control for Formation Flying Spacecraft ," S.M. Thesis, Massachusetts Institute of Technology, Dept. Aeronautics and Astronautics, June 2004.

[21] R. Brown, P. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* Wiley, 1997.

[22] J. How. *"Advanced Estimation for GPS and Inertial Navigation"* Class Notes, Massachusetts Institute of Technology, Spring, 2004.

[23] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[24] R.S. Bucy, P.D. Joseph. *Filtering for Stochastic Processes with Applications to Guidance.* Interscience Publishers, New York, 1968.

[25] B.W. Parkinson, J.J. Spilker, Jr., P. Enge. *Global Positioning System: Theory and Applications, Volumes 1 and 2*, American Institute of Aeronautics and Astronautics, Progress in Astronautics and Aeronautics Series, Volumes 163 and 164, Washington, DC, 1996.

[26] P. Misra, P. Enge, *Global Positioning System – Signals, Measurements, and Performance,* Ganga-Jamuna Press, Lincoln, MA, 2001.

[27] E. Kaplan, *Understanding GPS Principles and Applications,* Artech House Publishers, Boston, MA 1996.

[28] D. Vallado. *Fundamentals of Astrodynamics and Applications, Second Edition.* Microcosm Press, El Segundo, CA, 2001.

[29] S. Schweighart, R. Sedwick, "Development and Analysis of a High Fidelity Linearized J2 Model for Satellite Formation Flying," in *AIAA Guidance, Navigation, and Control Conference*, Fall 2001.

[30] S. Schweighart, " Development and Analysis of a High Fidelity Linearized J2 Model for Satellite Formation Flying" S.M. Thesis, Massachusetts Institute of Technology, Dept. Aeronautics and Astronautics, June 2001.

[31] `http://www-ccar.colorado.edu/~ stauch/GRAVITY_HTML/gravity.htm`

[32] J. Prussing, B. Conway. *Orbital Mechanics.* Oxford University Press, 1993.

[33] M. Fleck, S. Starin, "Evaluation of a Drag-Free Control Concept for Missions in Low Earth Orbit," in *AIAA Guidance, Navigation and Control Conference,* Aug. 2003.

[34] M. Kaplan. *Modern Spacecraft Dynamics and Control.* Wiley, 1976.

[35] B. Hofmann-Wellenhof, H. Lichtenegger, J. Collins. *Global Positioning System Theory and Practice.* Springer-Verlag, 1994.

[36] F. Busse, J. How, "Real-Time Experimental Demonstration of Precise Decentralized Relative Navigation for Formation Flying Spacecraft," in *AIAA Guidance, Navigation, and Control Conference*, Aug. 2002.

[37] M. Grewal, A. Andrews. *Kalman Filtering.* Wiley, 2001.

[38] K.T. Alfriend. "Navigation Correlation Problem," *personal correspondence*, May, 2004.

[39] S. Julier, J. Uhlmann "A New Extension of the Kalman Filter to Nonlinear Systems," in *SPIE AeroSense Symposium*, April 1997.

[40] J. Junkins, M. Akella, K.T. Alfriend, "Non-Gaussian Error Propagation in Orbital Mechanics," in *The Journal of the Astronautical Sciences*, Vol. 44, No. 4, 1996.

[41] K.T. Alfriend, H.Yan. "An Evaluation and Comparison of Relative Motion Theories," in *AAS/AIAA Astrodynamics Specialist Conference,* Aug. 2003.

[42] S. Julier, J. Uhlmann. "A New Method for the Nonlinear Transformation on Means and Covariances in Filters and Estimators," in *IEEE Transactions on Automatic Control,* Vol. 45, No. 3, Mar. 2000.

[43] T. Lefebvre, H. Bruyninckx, J. De Schutter. "Comment on 'A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators', " in *IEEE Transactions on Automatic Control,* Vol. 47, No. 8, Aug. 2002.

[44] S. Julier, J. Uhlmann. "Authors' Reply to Comment on 'A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators', " in *IEEE Transactions on Automatic Control,* Vol. 47, No. 8, Aug. 2002.

[45] S. Haykin. *Kalman Filtering and Neural Networks.* Wiley, 2001.

[46] R. van der Merwe, E. Wan, "The Square-Root Unscented Kalman Filter for State and Parameter Estimation," in *International Conference on Acoustics, Speech, and Signal Processing*, May, 2001.

[47] A.I. Solutions, "FreeFlyer User's Guide",Version 4.0, March 1999.

[48] W.A. Feess, S.G. Stephens, "Evaluations of GPS Ionospheric Models, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-23, No. 3, pp.332-338, 1987.

[49] E. Olsen, "GPS Sensing for Formation Flying Vehicles," Ph.D. Dissertation, Stanford University, Dept. Aeronautics and Astronautics, Nov. 1999.

[50] F. Busse, J. How. "Real-time Experimental Demonstration of Precise Decentralized Relative Navigation for Formation Flying Spacecraft," in *AIAA Guidance, Navigation, and Control Conference,* Aug. 2002.

[51] C. Park, " Precise Relative Navigation Using Augmented CDGPS," Ph.D. Dissertation, Stanford University, Dept. Mechanical Engineering, June 2001.

[52] "STR4760 High Dynamic GPS/SBAS Multichannel Simulator" `http://www.positioningtechnology.co.uk/datasheets/str4760.html`

[53] G. Franklin, J. Powell, M. Workman, "Digital Control of Dynamic Systems," Third Edition, Addison-Wesley, 1998.

[54] F. Busse. "GPS Payload: Interface Control Document." ORION Project Documentation, Massachusetts Institute of Technology, Apr. 2001.

[55] P. Ferguson, F. Busse, and J. How, "Navigation Performance Predictions for the Orion Formation Flying Mission," in *International Symposium on Formation Flying: Missions and Technologies,* Oct. 2002.

4088-45