Demonstration of Adaptive Extended Kalman Filter for Low Earth Orbit Formation Estimation Using CDGPS

Franz D. Busse, *Stanford University*
Jonathan P. How, *MIT*
James Simpson, *NASA Goddard Space Flight Center*

BIOGRAPHY

Franz Busse is completing his Ph.D. in aeronautics and astronautics at Stanford University. He received his M.S. from Stanford in 1998, and B.S. from MIT in 1995. He now works for the MIT Lincoln Laboratory.

Jonathan How is an Associate Professor in the Dept. of Aeronautics and Astronautics at MIT. He received his B.A.Sc (1987) from the University of Toronto, and SM (1990) and Ph.D. (1993) from MIT, both in the Dept. of Aeronautics and Astronautics.

James Simpson is a GNC system engineer at NASA GSFC. He received his B.S. from West Virginia University and M.S. from University of Maryland.

ABSTRACT

Carrier-phase Differential GPS is an ideal sensor for formation flying missions in low Earth orbit since it provides a direct measure of the relative positions and velocities of the vehicles in the fleet. This paper presents results for a relative navigation filter that achieves centimeter-level precision using measurements from a customized GPS receiver. This work uses a precise, robust extended Kalman filter that is based on relatively simple measurement models and a linear Keplerian propagation model. To increase the robustness of the filter in the face of environment uncertainty, the filter is adapted using MMLE algorithms. The adaptive filter is used to accurately identify the process noise and sensor noise covariance for the system. Despite the simplicity of the filter, hardware-in-the-loop simulations performed on the Formation Flying Testbed at the Goddard Space Flight Center demonstrate that this filter can achieve ≈2 cm relative position accuracy and <0.5 mm/s relative velocity accuracy for a range of Low Earth Orbit formations.

Presented at the Institute of Navigation GPS Meeting, Portland OR, September 2002

INTRODUCTION

Orbital relative navigation is a critical technology for a range of orbital formation flying missions. NASA, the Department of Defense (DoD), and the European Space Agency (ESA), are all currently planning low Earth orbit (LEO) formation flying missions [1, 2]. Some of these missions, such as TechSat-21, require relative navigation accuracy on the order of 2 cm position for separations of several kilometers. The goal of this research has been to develop a relative navigation sensor that can meet these requirements in real-time.

Carrier-phase Differential GPS provides an ideal sensor to deliver the required performance. It can be relatively low cost, and it is reliable and robust. It also has a four-in-one sensor capability, with the ability to deliver absolute state, relative state, time, and attitude sensing. GPS has already been demonstrated many times for absolute navigation on orbit [3]. (The absolute state refers to the state with respect to the center of the Earth. The relative state refers to the state of an individual vehicle with respect to some point local to the formation.) GPS has also been used on-orbit for relative navigation [3, 4, 5]. However, the actual relative navigation performance achieved on these missions is not sufficient. The best results from actual flight data has been 8-10 meters position accuracy, and about 1 cm/s velocity accuracy. To improve this, there have already been 2-vehicle hardware-in-the-loop relative navigation demonstrations using carrier differential GPS with significantly better performance. Binning used a dual frequency TurboRogue receiver and the Naval Research Laboratory's OCEAN orbital model to achieve 15cm relative position and 0.3 mm/s relative velocity accuracy [6]. Ebinuma used a simpler single-frequency receiver in closed-loop rendezvous simulations, and demonstrated 5 cm position accuracy and 1 mm/s velocity accuracy [7]. More recently, we have reported results from decentralized 4-vehicle formation hardware-in-the-loop simulations also using a single-frequency receiver [9]. For 1 km separations, relative position accuracy of 1cm and velocity accuracy of 0.3 mm/s was demonstrated.

A decentralized estimation algorithm was selected for this work on formation flying because it offers the advantages of flexibility and robustness over centralized approaches [8, 9]. Furthermore, one of the goals of formation flying is to use distributed groups of smaller, simpler, and lower cost vehicles to replace expensive, monolithic, and complex space vehicles. Therefore, the goal is to design the simplest decentralized real-time filter that can still meet the mission requirements.

An important problem facing any orbital relative navigation designer, is how to handle *uncertainty*. There are significant uncertainties in the measurements, dynamics, and disturbances that will be experienced by the formation. One of the drawbacks of complex models is that they may be based on assumptions that are not accurate, or on conditions that are not well understood. Adaptive filtering provides the ability to compensate for some of these uncertainties. As such, this paper presents a simple, decentralized, real-time adaptive extended Kalman filter for low Earth orbit relative navigation. The focus is on the adaptive techniques for dealing with uncertainty in the process noise and sensor noise. These adaptive algorithms provide additional robustness to the filter, helping achieve the desired estimator precision. This is especially important for orbital applications, where it is difficult to manually tune the filter or correct for any unexpected behavior. The effectiveness of the filter, and of the adaptive capability, are demonstrated using hardware-in-the-loop simulations performed at Goddard Space Flight Center's Formation Flying Testbed (FFTB).

EKF DESIGN

An extended Kalman filter is implemented to process the single difference GPS measurements. The measurements are provided at 1 Hz. Ref. [11] provides a detailed discussion of the types of models (measurement and dynamic) that are available for this application, and in particular, why the following filter uses a nonlinear measurement update but a linear dynamic model (sufficient for such a short propagation period). The state vector to be estimated is defined as

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

where

t_k = Time of discrete step k.

$\Delta\mathbf{r}_{ij}, \Delta\dot{\mathbf{r}}_{ij}$ = Relative position and velocity of vehicle j with respect to vehicle i, expressed in a Cartesian ECEF coordinate frame. These are the primary states of interest.

$\Delta b_{ij}, \Delta\dot{b}_{ij}$ = Differential clock offset and drift between vehicle j and vehicle i

$\Delta\beta_{ij}^m$ = Differential carrier phase bias for the signal received from GPS satellite m. This bias includes receiver line bias, and is not assumed to be an integer.

Equation 1 defines the state at discrete time-step k; the $(\cdot)_i$ and $(\cdot)_j$ refer to user vehicles, and $(\cdot)^m$ refer to the GPS NAVSTAR satellites. All the elements of the state vector are differential quantities between user vehicle j and reference user vehicle i, as indicated with the Δ symbol. The full state vector has $(8+N)$ states, where N is the number of commonly visible GPS satellites. This state has significantly fewer variables than similar filters developed by others (see [7]); the smaller state translates into smaller matrix inversions and fewer computations per time-step.

The state is estimated by recursively updating the estimate with new measurements and then propagating the state estimate between the update times. The single difference measurements at each time-step are

$$y_k = \begin{bmatrix} \Delta\phi_{ij}^1(t_k) \\ \vdots \\ \Delta\phi_{ij}^N(t_k) \end{bmatrix} \tag{2}$$

The measurements are combined with the previous state estimate using the standard form:

$$\hat{x}_k^+ = \hat{x}_k^- - K_k\left(y_k - \hat{h}_k\left(\hat{x}_k^-\right)\right) \tag{3}$$

$$P_k^+ = \left(I - K_k H_k\right) P_k^- \left(I - K_k H_k\right)^T + K_k \hat{R}_k K_k^T \tag{4}$$

where $h_k(\cdot)$ is the nonlinear function relating the state to the measurement (see Eq. 10). The observation matrix in equation 4 is the Jacobian,

$$H_k = \left.\frac{\partial \hat{h}_k}{\partial x}\right|_{x=\hat{x}_k^-} \tag{5}$$

The Kalman gain, K_k is determined from

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + \hat{R}_k\right) \tag{6}$$

where $\hat{R}_k$ is the expected sensor noise covariance. The state and covariance are propagated between updates using

$$\hat{x}_{k+1}^- = \Phi_k\left(\hat{x}_k^+\right) \tag{7}$$

$$P_{k+1}^- = \Phi_k P_k^+ \Phi_k^T + \hat{Q}_k \tag{8}$$

Φ_k is the state propagation matrix, which is a linearization of the dynamic functions defined in Eqs. 13 and 15. In Eq. 8, $\hat{Q}_k$ is the expected process noise covariance.

Measurement Model

Each vehicle has a GPS receiver which collects its own independent set of measurements at each time-step. The GPS receiver provides three measurements: code phase, carrier phase, and the Doppler shift on the carrier phase. Absolute navigation uses the code and Doppler measurements. However, for the relative navigation, only the carrier phase measurement needs to be used [11].

For relative navigation, two receivers located on different vehicles collect measurements from the same NAVSTAR satellite. These are subtracted to create a single difference measurement

$$\Delta\phi_{ij}^m = \phi_j^m - \phi_i^m \tag{9}$$

This provides a direct measure of the relative state between the two vehicles. The advantage of using the single difference is that the errors in these differential terms are much smaller than in the corresponding absolute terms. Specifically, the single difference is

$$\begin{aligned}\Delta\phi_{ij}^m &= \|\mathbf{r}^{m_i} - \mathbf{r}_i\| - \|\mathbf{r}^{m_j} - (\mathbf{r}_i + \Delta\mathbf{r}_{ij})\| + \\ &\quad \Delta\beta_{ij}^m + \Delta b_{ij} + \Delta B_{ij}^m + \Delta I_{ij}^m + \nu_{\Delta\phi}\end{aligned} \tag{10}$$

where

- $\mathbf{r}^{m_i}$ = position of NAVSTAR satellite m at time of transmission of the signal measured by user i
- $\mathbf{r}_i$ = position of user i at the time the signal is measured by user i
- Δb_{ij} = differential clock offset between users i and j
- ΔB_{ij}^m = differential clock offset for GPS satellite m, between the time of transmission of signals measured by users i and j
- ΔI_{ij}^m = differential ionospheric delay
- $\Delta\beta_{ij}^m$ = differential carrier phase bias between users i and j, on signal from m
- $\nu_{\Delta\phi}$ = remaining differential noises

The measurements contribute the largest errors, and equation 10 deserves special attention. The relative states $\Delta\mathbf{r}$ and Δb are to be estimated. To determine $\hat{h}_k(\cdot)$ in Eq. 3 requires that the expected values of the other terms in Eq. 10 be used. Each of these terms will have errors that could impact the filter:

1. The signal delay caused by ionosphere, ΔI_{ij}^m, is modeled within the filter. The ionosphere model used in the filter is

$$\Delta I_{ij}^m = I_j^m - I_i^m \tag{11}$$

$$I_i^m = \frac{82.1 \times TEC}{F_c^2 \times \sqrt{\sin^2 E_i^m + 0.076} + \sin E_i^m} \tag{12}$$

   where TEC is the Total Electron Count within the ionosphere, F_c is the carrier frequency, and E_i^m is the elevation angle between the user vehicle i and the NAVSTAR satellite m. The differential ionospheric correction term is calculated using this model, and subtracted from the single difference measurements.
2. The absolute state of the master or reference vehicle, $\mathbf{r}_1$, is determined independently of the relative navigation filter. The focus of this work was not to improve the absolute state estimation process, and generally the absolute state error was on the order of 20 m. The relative state solution is not very sensitive to absolute errors of this order, and any error in the relative solution due to this error is lost in other noises.
3. The position of the NAVSTAR satellites, $\mathbf{r}^m$, and the NAVSTAR clock offset, B^m, are determined from the ephemerides that are broadcast by the NAVSTAR satellites themselves. The NAVSTAR states predicted by the ephemeris will be erroneous. As the separation between the vehicles grows, the differential ephemeris error will also grow. In this work, this differential ephemeris error is neglected, and is a leading contributor to the overall sensor noise, especially as the separation between vehicles grows.

   Special care must be taken in determining the GPS satellite states. The range measured is the range between the position of the user at the time the signal is received and the position of the GPS satellite at the time the signal is transmitted. This transmission time is determined iteratively.

   As the separation between the vehicles grows, the difference in time of transmission between the user vehicles and the same GPS satellite will also become more significant. Therefore, this iterative process must be done for both users, and both of these states are used in equation 10. Even though a simple filter is sought, if the user vehicles are far apart from each other, this process must be used or else significant errors are introduced in the relative state estimates.
4. The remaining noise, $\nu_{\Delta\phi}$, contains all other factors affecting the signal. These include multipath and noise within the receiver itself. Effectively, any errors in the other terms will also end up in the sensor noise term.

Propagation Model

A very simple set of dynamics, which are based on Kepler's differential central gravitation model, are used in the filter

$$\Delta\ddot{\mathbf{r}}_{ij} = \frac{\mu}{r_i^3}\left[\mathbf{r}_i - \frac{r_i^3\left(\mathbf{r}_i + \Delta\mathbf{r}_{ij}\right)}{\sqrt{\left(r_i^2 + 2\mathbf{r}_i\Delta\mathbf{r}_{ij} + \Delta r_{ij}^2\right)^3}}\right] + C_{ECEF} + w_{\Delta r} \quad (13)$$

where μ is the earth's gravitational constant, and

$$C_{ECEF} = 2\omega_e \times \Delta\dot{\mathbf{r}}_{ij} + \omega_e \times (\omega_e \times \Delta\mathbf{r}_{ij}) \quad (14)$$

is the correction for the rotating reference frame and ω_e as the earth's rotation rate. All other forces, such as differential drag and higher order gravity terms, are lumped together into the process noise, $w_{\Delta r}$.

Eq. 13 handles the position and velocity states. The clock offset and drift dynamics are modeled only as,

$$\Delta\ddot{b}_{ij} = w_{\Delta b} \quad (15)$$

The noise in the clock model is on the order of 0.05 m/s^2, while the noise in the motion dynamics is on the order of 10^{-6} m/s^2, for vehicles 100m apart.

Finally, the carrier phase bias states are considered as constants, so they have no dynamic propagation over time. Note that the unknown or unmodeled relative dynamics errors are much smaller than the corresponding absolute dynamics errors. Just as in the case of the measurement model, this also makes it desirable to estimate the relative state rather than the absolute states.

Filter Initialization

Since this is a nonlinear system, it is especially sensitive to the filter initialization. The initial state estimate is determined by taking the least-square solution of the differential code phase measurements ($\Delta\rho$) and differential Doppler measurement ($\Delta\dot{\phi}$). For close formations (vehicles less than 1 km apart), this method gives an initial estimate with an accuracy of 2–4 m. However, for larger separations the error in the least squares solution increases because the "common line-of-sight assumption" breaks down. To compensate, Park's algorithm [10] for correcting measurements is implemented. Using this technique, separations of 100's of kilometers can still be initially determined to within a few meters for position and several centimeters per second for velocity. The full initialization routine is discussed in Ref. [9].

IDEAL PERFORMANCE

To demonstrate the filter performance, actual prototype GPS receivers are used in hardware-in-the-loop simulations. Four receivers are used, simulating formations of four vehicles, which allows us to compare the results of multiple simultaneous independent relative solutions. The GPS receivers used for this work are modified receivers based on the GP2015 and GP2021 chipset by Zarlink (formerly Mitel/Plessey) [11, 9]. The receiver exhibits a carrier phase measurement noise of $\approx$ 5mm in ground tests, which should provide a conservative bound on orbital noise levels.

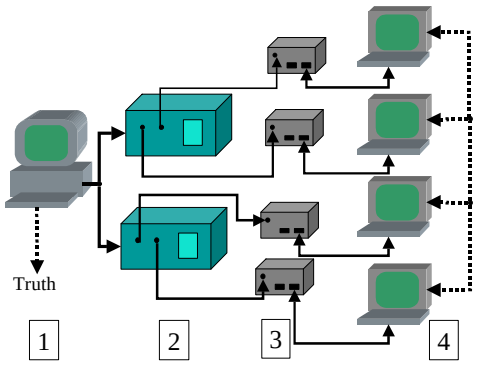

Fig. 1: Shows the GSFC FFTB setup, including: 1) DS10 Workstation, which sends orbital states to 2) STR4760 GPS signal generators, which sends actual RF signals, received by 3) GPS receivers, which perform real-time absolute navigation and also collect measurements that are saved on 4) data storage PC's.

The NASA Goddard Space Flight Center has a state-of-the-art Formation Flying Test-Bed. At its heart is the Spirent STR Series Multichannel Satellite Navigation Simulator [16]. Figure 1 shows a simple block diagram of the experimental set-up.

A variety of scenarios have been run using this orbital simulator. These different formations are all based on the same master orbit. The master orbit has an altitude of ~450km, eccentricity of 0.005, and 28.5° inclination. The simulation used a tenth-order gravity model and an atmospheric drag model (based on Lear's atmospheric model). For the measurements, a diverging ephemeris and clock model was used and a standard orbital ionospheric model. The ionospheric model is the same as appears in equation 12. There is no multipath noise in the simulation, but this is not expected to be a significant error source for microsatellites. All vehicle models are identical, with a surface area of 1m^2, drag coefficient of 2, and 0.1 metric tonne (the smallest mass the simulator would model). The antenna is at the center of gravity, and the vehicle maintained a nadir earth-pointing attitude. A standard hemispherical antenna gain pattern was used.

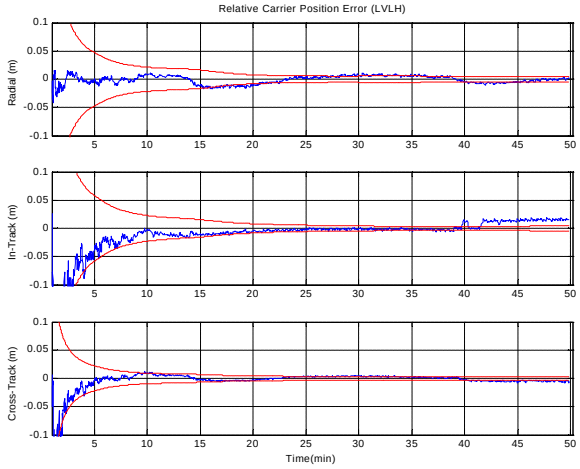

Fig. 2: Relative position performance (1 km elliptical formation).

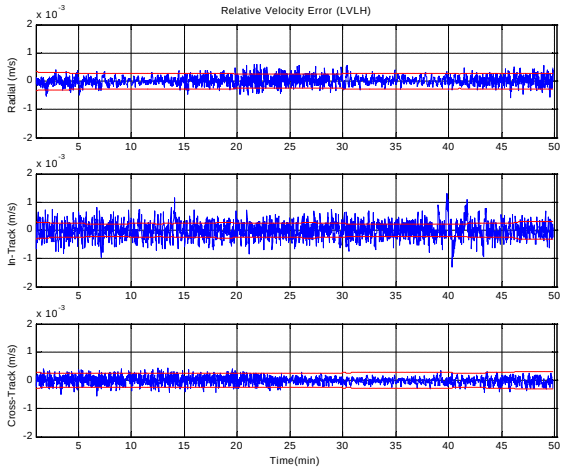

Fig. 3: Relative velocity performance (1 km elliptical formation).

For all experiments, data storage of the raw measurements began after all four of the receivers achieved navigation fixes. The receivers performed numerous "warm" starts, where they were provided almanac information for the NAVSTAR constellation, a rough current position guess (often wrong by several hundred kilometers) and a rough current time guess (off by up to fifteen seconds). A navigation fix was usually attained, and active tracking begun, in less than two minutes after the simulation began.

Figures 2 and 3 show typical errors in relative position and relative velocity, respectively. These are errors for simulation of a vehicle in a 1×2 km elliptical formation. The three plots show the errors in Radial (R), In-Track (I), and Cross-Track (C) directions. The position scale is ±10 cm, and the velocity scale is ±2 mm/s. From the position plot, it can be seen that the carrier-phase biases are determined within 5 minutes of the start of the filtering. After that, as new measurements come on-line, the bias re-initialization is seamless, and does not disturb the position estimate. The velocity plot shows fairly "white" error, with a very low mean. This is an encouraging sign that the filter is performing very well, even with the unmodeled drag and other biased noise sources. Table 1 summarizes the performance for this formation case. The actual numbers are Root-Mean-Square combinations of 3 simultaneous relative navigation solutions (the three differential solutions from the four-vehicle simulated formation).

Table 1: Relative State Estimation Results: 1km In-Plane Elliptic Formation

Rel. State Error		Mean (μ)	St. Dev.(σ)
Position (cm)	R	0.251	0.454
	I	1.061	0.660
	C	0.159	0.285
Velocity (mm/s)	R	0.032	0.156
	I	0.001	0.275
	C	0.017	0.107

EFFECT OF UNCERTAINTY

The previous results demonstrate the filter performance assuming all the conditions are well known and correctly modeled. However, it is quite possible that these conditions or models may not reflect the actual performance on-orbit. For example, the receiver may have more or less noise on the signal reception. Or the differential process noise may be much larger or smaller than expected. Any of the terms contributing to the phase measurement (as in Eq. 10), or any of the noise sources lumped into the process noise $w_{\Delta r}$ could contribute to the overall sensor or process noise.

Many of these noise sources can be treated as Exponentially Correlated Random Variables (ECRV). Some of these ECRV noises, such as the ionosphere, will have relatively long time constants, even to the extent that they appear to be constant biases. The noise sources also include ECRVs with short time-correlation, to the extent that they can be modeled as white noise. These short-correlation sources are also important, and can also influence the performance of the filter. The covariance of the sensor noise is represented by $\hat{R}$ in the filter, and the covariance of the process noise is represented by $\hat{Q}$.

A sensitivity analysis illustrates the importance of $\hat{Q}$ and $\hat{R}$. For a linear, time-invariant system, it is easy to perform a parameter sensitivity analysis (see Gelb [20]). A simple linear system was simulated. Then, keeping all other conditions constant, the filter was implemented using a range of design $\hat{R}$ values (and $\hat{Q}$

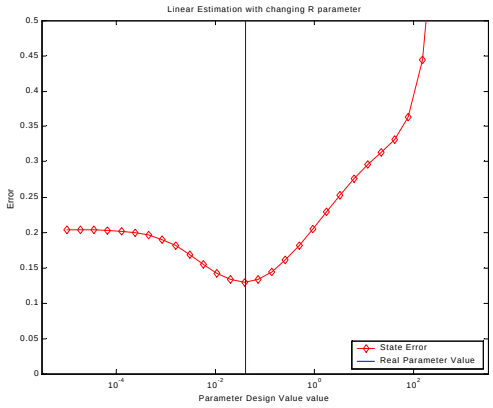

Fig. 4: Design R sensitivity, linear example

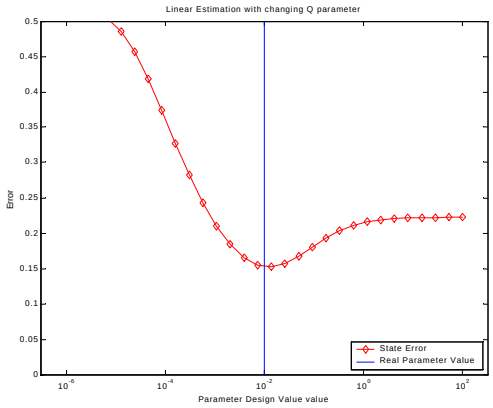

Fig. 5: Design Q sensitivity, linear example

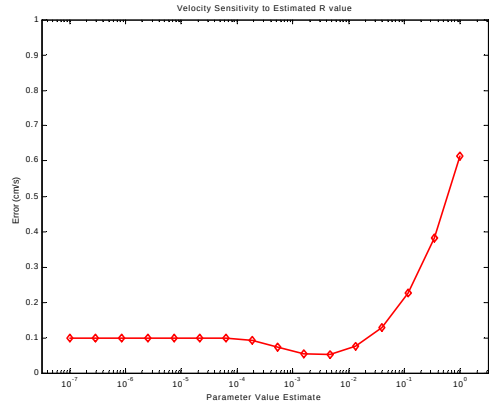

Fig. 6: Design R sensitivity, real data

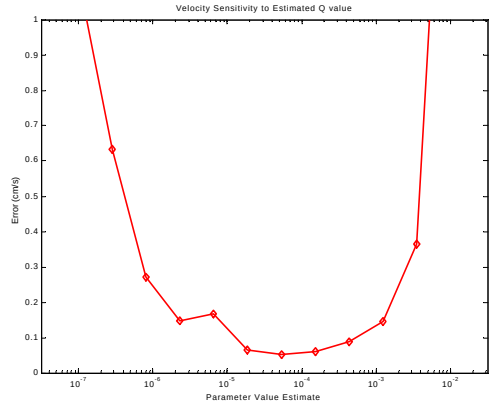

Fig. 7: Design Q sensitivity, real data

constant), and then implemented using a range of design $\hat{Q}$ values (holding $\hat{R}$ constant). Figures 4 show the results for $\hat{R}$, and 5 shows the results for $\hat{Q}$. The vertical axes in both figures is the state error, the horizontal axes are the design parameter values. The dashed vertical line marks the real parameter value. As expected, when the design value matches the real value, the error is minimized. These simple linear results match closely with the theoretical expectation illustrated by Gelb.

The same study can be performed for our actual application as well. Actual data is stored from hardware-in-the-loop tests. Using this same set of data and conditions, only the $\hat{R}$ and $\hat{Q}$ design values are changed in the filter. The resulting relative velocity state estimate error is then measured for each filter. This provides a measure of the sensitivity of the filter to the design parameter value.

For these sensitivity curves, the actual parameter value can only be estimated by looking for the minimum error points. These sensitivity curves are also sensitive to time-step size (Δt) and formation separation ($\Delta \mathbf{r}$). The $\hat{R}$ sensitivity, for the conditions shown, is very similar to the linear case. The $\hat{Q}$ has much more pronounced error on either side of the minimum point. These sensitivity curves clearly illustrate the importance of using the correct value of $\hat{R}$ and $\hat{Q}$ in the filter design; using the wrong value can be potentially devastating to the filter accuracy. Furthermore, the minimum point on these curves will be different for different conditions, some of which are unknown.

In simulation, the filter can be tuned, and the best value of $\hat{Q}$ and $\hat{R}$ can be determined. However, the experienced engineer recognizes that reality often exhibits different behavior than simulation. The ideal values of these parameters in simulation may not represent ideal values in real operation. This example has shown that the performance is sensitive to these parameters, in real operation it may be more or less sensitive over the range of parameter values. The robust design must be prepared to handle this uncertainty.

ADAPTIVE ESTIMATION

To robustly handle uncertainty in the standard deviation of the sensor and process noises, an adaptive filter can be applied that identifies the value of R or Q. There are many different adaptive filters in existence. The Method of Maximum Likelihood Estimation (MMLE) is a technique applied to Kalman Filters. Originally proposed by Mehra [12, 13], variations of the technique have been used in many filter applications. The routines presented in this work are modified from algorithms summarized by Maybeck [14]. The basic premise is to use the measurement and state residuals to modify the parameter values for sensor and process noise. A similar variant has also been employed by Campana for absolute GPS navigation [15].

Process Noise

The process noise covariance Q is a measure of the uncertainty in the state dynamics during the time interval between measurement updates. It is generally defined in terms of the process noise applied to the system, *i.e.,* $w_k \sim \mathcal{N}(0, Q_k)$.

Adaptive filtering requires that the parameter of interest be measured. Our observation of Q is obtained from the difference between the state estimate before and after the measurement update:

$$\begin{aligned} Q^\star &= \Delta x_k \Delta x_k^T + P_k^- - P_k^+ - \hat{Q}_k^- \quad (16) \\ \Delta x_k &= \hat{x}_k^+ - \hat{x}_k^- \quad (17) \end{aligned}$$

where:

$\hat{x}_k^+$ = *à posteriori* state estimate
$\hat{x}_k^-$ = *à priori* state estimate
P_k^+ = *à posteriori* state covariance estimate
P_k^- = *à priori* state covariance estimate
$\hat{Q}_k^-$ = Current expected process noise covariance

Eq. 16 is the core of the adaptive routine, and is worth some consideration. The term Δx_k is the state residual; it represents the difference between our state estimate before the measurement update and after the measurement update. If this residual has a large value, then it indicates that we are not predicting the future state very well, because when the measurements are applied, there is a large jump in the state estimate. As the filter converges, this residual should decrease, as our ability to predict the next state should be improving.

The first term of Eq. 16 is a measure of the state residual. The next part of the equation is a measure of what one might expect this residual to be. These terms could be considered the expected change in covariance. It may be conceptually clearer by rewriting Eq. 16 as

$$\begin{aligned} Q^\star &= \Delta x_k \Delta x_k^T - \left[P_k^+ - \left(P_k^+ - \hat{Q}_k^- \right) \right] \\ &= \Delta x_k \Delta x_k^T - \left[P_k^+ - \left(\Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T \right) \right] \quad (18) \end{aligned}$$

where Φ_{k-1} is the state propagation matrix for the time interval from $k-1$ to k. Eq. 18 shows that $Q^\star$ is the residual minus the change in the *à posteriori* covariances between two contiguous time-steps.

This measure of the process noise, $Q^\star$, is then combined with the current estimate $\hat{Q}$ in a moving average (or low pass filter),

$$\hat{Q}_k^+ = \hat{Q}_k^- + \frac{1}{L_Q} \left(Q^\star - \hat{Q}_k^- \right) \quad (19)$$

where L_Q is the window size that sets the number of updates being averaged. If L_Q is small, then each update is weighted heavily, but if L_Q is large, then each update has a small effect. The performance of the adaptive routine is very sensitive to the selection of L_Q, and should be selected for each application.

For the orbital relative navigation application using GPS, this Q-adaptive scheme did not perform well without additional measures. To improve performance, the discrete formulation was then placed into continuous form. Then this continuous $\hat{Q}$ was integrated anew through the state dynamics to create a new discrete Q. In other words, if

$$\hat{Q}_k = \begin{bmatrix} Q_{pos} & Q_{pv} \\ Q_{vp} & Q_{vel} \end{bmatrix} \quad (20)$$

then

$$\begin{aligned} q_c &= \mathrm{diag}\left(Q_{vel}\right) \frac{1}{\Delta t} \quad (21) \\ Q_c &= \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{diag}\left(q_c\right) \end{bmatrix} \quad (22) \end{aligned}$$

and this is re-discretized by

$$\hat{Q}_k^+ = \int \Phi_k Q_c \Phi_k^T d\tau \quad (23)$$

This updated estimate of $\hat{Q}$ is then used for the state propagation between time-step k and $k+1$. This process helps by isolating the error due to process noise, which shows up in velocity, from the bias measurement errors, which show up in position. Without this step, the estimate of $\hat{Q}$ remains much too high because of the position errors, and cannot resolve the small disturbances.

Sensor Noise

The routine for adaptively identifying the sensor noise is similar to that of process noise, but it is only a

function of the measurement update and is based on the measurement residual. R is also defined in terms of the sensor noise,*i.e.*, $\nu_k \sim \mathcal{N}(0, R_k)$.

To identify R, form a measure of the sensor noise

$$R^\star = \Delta y \Delta y^T - H_k P_k^+ H_k^T \quad (24)$$
$$\Delta y = y_k - \hat{y}_k^+ \quad (25)$$

where H_k is the observation matrix at time k. For nonlinear systems, H is the Jabobian of the nonlinear measurement function. Δy is a measure of the covariance of the measurement residual. y_k is the actual set of measurements taken this time-step. $\hat{y}_k^+$ is the *à posteriori* measurement expectation,

$$\hat{y}_k^+ = h_k\left(\hat{x}_k^+\right) \quad (26)$$

Note that $\hat{y}^+$ uses the *à posteriori* state estimate, so this routine is run *after* the measurement update. This requires $\hat{y}$ to be computed a second time in the update process. The second term in Eq. 24, HP^+H^T, removes the amount of expected measurement error due to the state uncertainty. What is left is a measure of the "unexpected" measurement error, which should be attributable to sensor noise.

The R estimate is also updated with this sensor noise measure through a moving average

$$\hat{R}_k^+ = \hat{R}_k^- + \frac{1}{L_R}\left(R^\star - \hat{R}_k^-\right) \quad (27)$$

where, as before, the sensor noise window size, L_R, is very application specific.

One of the characteristic issues of GPS estimation, is that the measurements come and go continuously. This means that the measurement vector changes dimension, and the placement of a given channel within the vector will change. As such, care must be taken to ensure that the R matrix matches the same channels throughout the process. When new measurements come on-line, it is assumed that this channel will have a noise level that is similar to the other channels. Therefore, the newly introduced elements of the $\hat{R}$ matrix, r_{ij} are initialized as

$$r_{ij,k} = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} r_{nn,k-1} & : \quad i = j \\ 0 & : \quad i \neq j \end{cases} \quad (28)$$

Identify Noise

MATLAB simulations with known noise levels added to the dynamics and measurements were performed to demonstrate the effectiveness of the adaptive schemes to identify the environment noises, even with an erroneous initial estimate of those noise. To demonstrate the sensor noise adaptive capability, measurements were created in MATLAB with white noise of

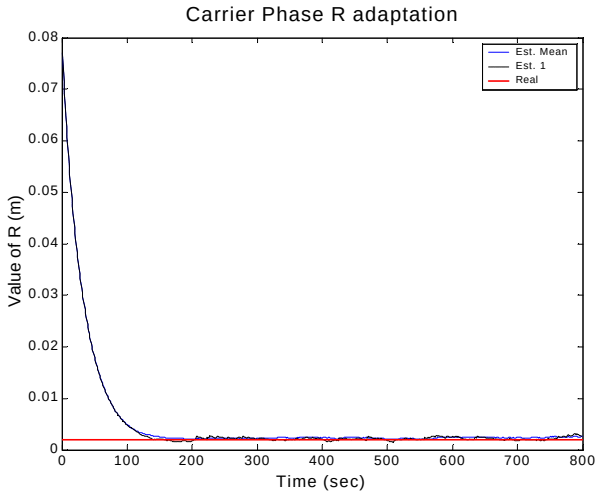

Fig. 8: Ability to identify sensor noise. All channels are able to identify the correct sensor noise level, even with an initial guess that is 20 times too high.

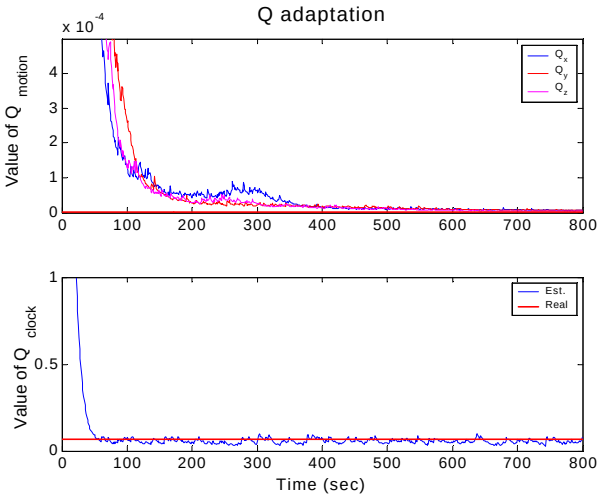

Fig. 9: Ability to identify process noise. The process noise level in the three directions of motion, as well as the clock noise which is orders of magnitude larger, can be identified.

2 mm added to the carrier signal from each GPS satellite. There was no ionospheric, ephemeris, or other bias in the simulated signal. The absolute and relative filters were the same other than the R-adaptation algorithm. Figure 8 shows the ability of the adaptive filter to identify the noise level on the carrier signal. The solid line marks the mean estimate over all valid channels, while the dashed line marks the noise estimate on channel one. Even with an initial guess of $\hat{R} = (10\text{cm})^2$, the filter is able to correctly identify the noise level.

The process noise can also be identified successfully. Figure 9 shows in the upper plot the ability to identify the level of process noise in the motion dynamics, and the lower plot shows the ability to identify the clock

drift noise. Notice that the levels of the noises are orders of magnitude apart, and yet they can still be accurately identified. For this demonstration, only white process noise was added to the simulated Keplerian motion, so there is no differential drag or other biased noise included. This was done so that the process noise level was a known value. The algorithm also works in the presence of more realistic disturbances.

Application Notes
Some important notes about the use of the algorithms:

1. These two routines, the Q-adaptation and R-adaptation, do not run simultaneously. If both R and Q are unknown, a serial approach of running the routines one after another could be implemented. Further research is being done in this area.
2. The routines are functions of steps, not time. So if there is a longer time interval between updates, they will take more time to identify the parameters.
3. It may be desirable to change window sizes during the filtering process. Currently, we begin with a short window (approximately 10 steps), and then after a few hundred steps, switch to a longer window size.
4. The routines were always able to identify the parameter if the initial guess was larger than the parameter value, but they did not always converge for initial guesses below the actual values.
5. These routines are designed to identify changes in the standard deviations of the white noise rather than biases or time constants in a ECRV noise model. Further work will investigate other adaptive algorithms that can identify these parameters.

Active Thrusting
Formation flying involves active control of the vehicles within the formation. A special case of adapting the Q parameter occurs during active control because the uncertainty associated with the propulsion devices on spacecraft will be far greater than the natural differential disturbances. The filter must account for this new acceleration noise when thrusters are turned on.

It is assumed that any intended control input (a thruster firing) can be input to the filter. The filter handles the thrusting as follows:

$$\ddot{\hat{x}}_k = f\left(\hat{x}_k\right) + \Delta\hat{u}_k \tag{29}$$

where $\Delta\hat{u}_k$ is the expected differential control input (acceleration due to thrusters) at time-step k. The nonlinear propagation function $f(\cdot)$ is defined in equation 13.

To account for the greater uncertainty associated with the thrusters, the covariance is also updated, so that Eq. 21 can be augmented in the following way:

$$q_c \rightarrow q_c + q_u \tag{30}$$

$$\text{and } q_u = \left(K_u \cdot \text{abs}\left(\Delta\hat{u}_k\right)\right) \tag{31}$$

where q_u is the thruster uncertainty adjustment. For this work, the constant K_u was set at 10%, which represents the uncertainty in the thruster noise.

ADAPTIVE PERFORMANCE

This section presents three simulations that clearly demonstrate the advantages of using an adaptive filter for this application. The first demonstration compares the adaptive filter performance with the performance of the filter using a reasonable, but wrong, parameter setting. The second compares the covariance behavior using the adaptive filter and using a good, but fixed, parameter setting. Finally, the third is an example of the active thrusting adaptation working successfully.

Wrong Parameter Correction
Consider the 1km formation case presented previously in the "Ideal Performance" section. That example used the process noise setting of $\hat{Q} = 10^{-4}$, and the results were presented in Figures 2 and 3. However, now consider the performance if we had instead used a reasonable, but incorrect, process noise estimate of $\hat{Q} = 10^{-6}$ in the filter. This one parameter was changed in the filter (all other conditions were the same) and it was applied to the same saved data to obtain the result in Figure 10 (same scales). Comparing figures 2 and 10, it is clear that the solution diverges and the filter does not perform well.

The adaptive filter was used with the same data sets and conditions. The initial estimate of the process noise is $\hat{Q}_0 = 1\text{m/s}^2$. Figure 11 displays the resulting relative position error and it is clear that the performance is much better. In fact, the performance is close to that of the ideal filter performance shown in Figure 2. The adaptive filter works well in correcting for an inaccurate estimate of the noise covariances.

Covariance Comparison
Another benefit of the adaptive filter is that it keeps the covariance consistent with the real performance. Consider an example with a longer time-step and a larger separation between vehicles, both of which tend to increase the differential process noise. This increased noise impacts the relative velocity estimation performance. Figure 12 shows the velocity performance from

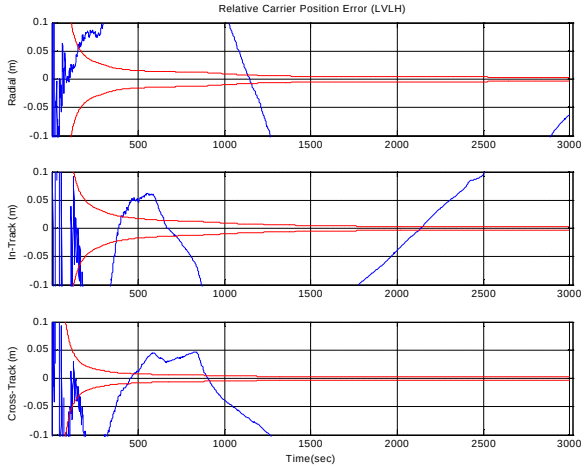

Fig. 10: Relative position error, with an incorrect $\hat{Q}$ value, with no adaptation

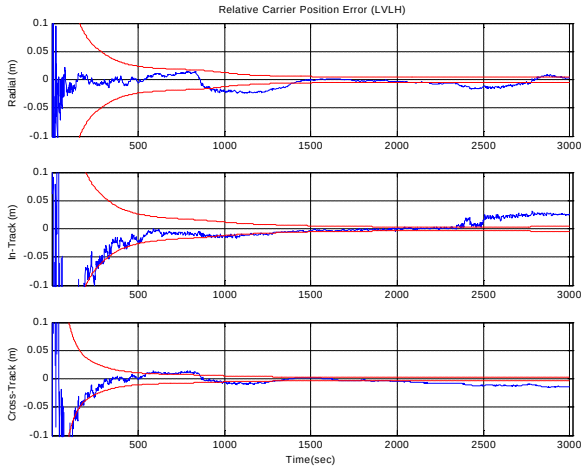

Fig. 11: Relative position error, using Q–adaptation. Now the performance is greatly improved, and is about the same as using the ideal $\hat{Q}$ value.

a poor case, with no adaptive filtering and a design $\hat{Q} = 10^{-4}\text{m/s}^2$. Note that the velocity error drifts significantly outside of the covariance bounds. Figure 13 shows the same data with the Q-adaptation filter. The results in this case clearly shows that the covariance bounds (marked by the dashed lines) are adapted (*e.g.,* in-track between 1200–1800 secs) to reflect the larger error. This has the effect of slightly increasing the standard deviation of the error, but tends to reduce the mean error. It also provides confidence that we will not end up with a "smug" filter that allows the estimate diverge. Divergence can occur when the expected $\hat{Q}$ is much lower than $\hat{R}$, and the covariance decreases to the point that the new measurement updates are ignored. The covariance behavior illustrated in figure 13 is very desirable, and suggests improved filter robustness in the face of uncertain noise conditions.

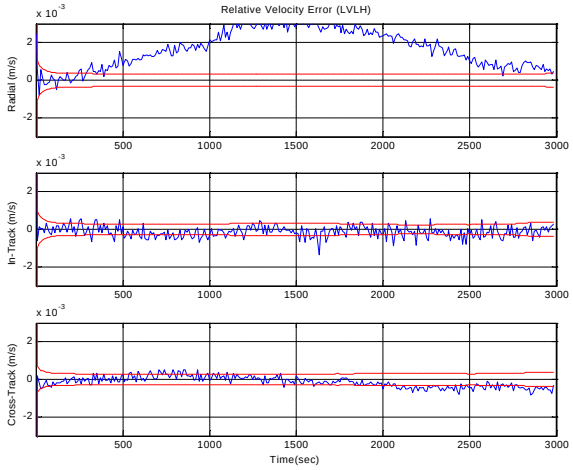

Fig. 12: Relative velocity error, without adaptation, for long separation and long time interval. Notice the error is well outside the covariance bounds (marked by the dashed lines).

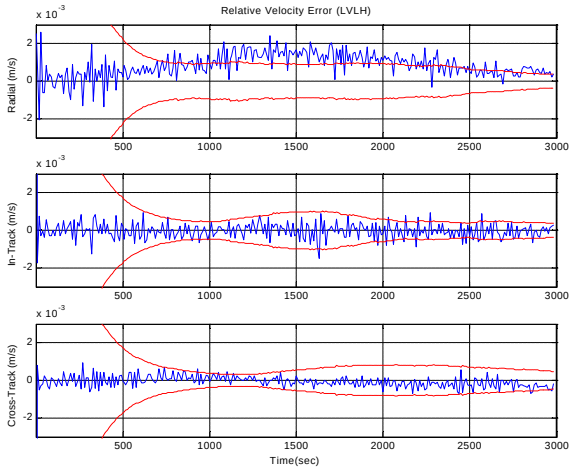

Fig. 13: Relative velocity error for the same case as 12, with Q–adaptation. Now the covariance bounds better reflect the error experienced (and also mitigates the error bias).

Thrust Demonstration

It is also important to demonstrate effective estimation during active thrusting. Sample control inputs were performed during the simulation using a script of previously generated control commands. An experiment was performed using a 3-vehicle formation, where the formation was changed from an in-track formation to an in-plane elliptical formation (an In-track formation has the vehicles moving in the same orbit one after the other). The commands are fed forward to the Kalman filter. However, to demonstrate robustness, the value fed to the filter was changed by 10% to simulate uncertainty in the actual thruster behavior. There is a 10 second thruster firing of 0.1 m/s^2 in the radial direction. Figure 14 shows the velocity error for the two relative states during the experiment. Both follower vehicles fire their thrusters at the same

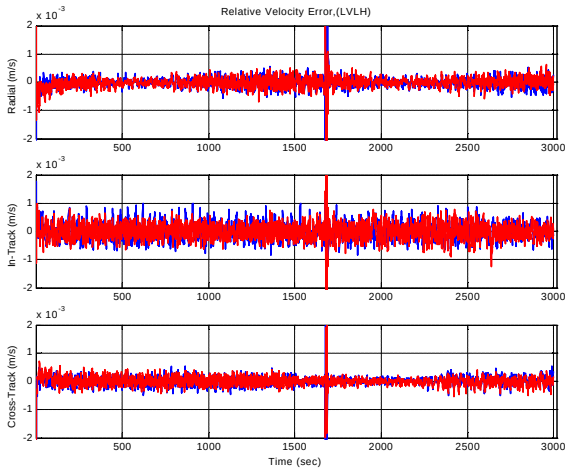

Fig. 14: Velocity Error during a commanded maneuver. Even with large error in commanded thrust level, the velocity estimate returns immediately after firing to previous error levels.

time (in opposite radial directions). During the actual thruster firing, the velocity estimate jumps, which is partly due to the 10% simulated error, and also because of the increase in the Q matrix during active control. The main point of these results is that the estimate returns almost immediately to the previous accuracy level directly after the thruster firing. The position estimate maintains accuracy throughout the thruster firing.

CONCLUSIONS

This paper has focused on demonstrating that a simple, yet robust, extended Kalman filter is sufficient to achieve centimeter-level accuracy for formation estimation. Adaptive filtering techniques, based on MMLE algorithms, have been developed to enhance this decentralized filter. The adaptation is designed to augment (rather than replace) the process of modeling the disturbance environment, but it provides an additional level of robustness in the face of uncertainty. The experimental results in this paper demonstrate that the MMLE algorithms are very effective for identifying the standard deviation of the noises present.

This simple, adaptive EKF has been demonstrated using hardware-in-the-loop simulations at Goddard Space Flight Center. These simulations demonstrate less than 2 cm relative position error and less than 0.3 mm/s relative velocity error for formation separations of 1-2 km. Measurement and process noises are a function of separation between vehicles, and for longer separations (on the order of 10-20 km), the resulting position error levels are higher (on the order 5 cm), while velocity is not affected as strongly.

ACKNOWLEDGMENTS

We gratefully acknowledge R. Burns, J. Leitner, and F. Bauer at NASA Goddard Space Flight Center for their help and support in developing and testing this receiver using the FFTB. This work was funded through NASA grant NAG5-10719.

REFERENCES

[1] F.H. Bauer, K. Hartman, J.P. How, J. Bristow, D. Weidow, F. Busse, "Enabling Spacecraft Formation Flying through Spaceborne GPS and Enhanced Automation Technologies," Proc. of *Institute of Navigation,* July 1999

[2] J. Leitner, F. Bauer, D. Folta, R. Carpenter, M. Moreau, and J.P. How, "Formation Flight in Space," *GPS World*, Feb. 2002, pp. 22–31.

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

[4] D. Highsmith, P. Axelrad, "Relative State Estimation Using GPS Flight Data from co-Orbiting Spacecraft," Proceedings of the *Institute of Navigation GPS Conference*, Nashville, TN, Sep. 1999.

[5] I. Kawano, M. Mokuno, T. Kasai, T. Suzuki, "First Autonomous Rendezvous Using Relative GPS Navigation by ETS-VII," *Journal of the Institute of Navigation,* Vol. 48, No. 1, Spring 2001.

[6] P. Binning, I. Galysh, "Satellite to Satellite Relative Navigation Using GPS Pseudoranges," Proc. of *ION National Technical Meeting,* Santa Monica, CA, Jan. 1997.

[7] T. Ebinuma, R. Bishop, E. Lightsey,"Hardware-in-the-Loop GPS Test Facility for Spacecraft Autonomous Rendezvous," Proceedings of *ION GPS,* Salt Lake City, UT, Sep. 2001.

[8] C.W. Park, P. Ferguson, N. Pohlman, J.P. How, "Decentralized Relative Navigation for Formation Flying Spacecraft using Augmented CDGPS," Proc. of *ION GPS,* Salt Lake City, UT, Sep. 2001.

[9] F.D. Busse, J.P. How, "Real-Time Experimental Demonstration of Precise Decentralized Relative Navigation for Formation Flying Spacecraft," Proc. of *AIAA Guidance, Navigation, and Control Conference,* Monterey, CA, Aug. 2002.

[10] C.W. Park, *Precise Relative Navigation Using Augmented CDGPS,* Ph.D. Dissertation, Stanford University, Dept. of Mechanical Engineering, June 2001.

[11] F.D. Busse, *Precise Formation Estimation in Low Earth Orbit Using GPS*, Ph.D. Dissertation, Stanford University, Dept. of Aeronautics and Astronautics, Oct 2002.

[12] R.K. Mehra, "On the Identification of Variances

and Adaptive Kalman Filtering," *IEEE Transactions on Automatic Control,* Vol. AC-15, No. 2, April 1970.

[13] R.K. Mehra, "Approaches to Adaptive Filtering," *IEEE Transactions on Automatic Control*, October 1972.

[14] P. Maybeck, *Stochastic Models, Estimation, and Control,* Volume 2, Academic Press, New York, 1982.

[15] R. Campana, L. Marradi, "GPS-based Space Navigation: Comparison of Kalman Filtering Schemes," Proc. of *Institute of Navigation GPS Meeting*, Salt Lake City, UT, Sep. 2000.

[16] *STR Series Multichannel Satellite Navigation Simulator Reference Manual,* May 2001.

[17] M.J. Unwin, P.L. Palmer, Y. Hashida, C.I. Underwood, "The SNAP-1 and Tsinghua-1 GPS Formation Flying Experiment," Proc. of *Institute of Navigation,* Salt Lake City, UT, Sep. 2000.

[18] E. Olsen, *GPS Sensing for Formation Flying Vehicles,* Ph.D. Thesis, Stanford University, Dept. of Aeronautical and Astronautical Engineering, Dec 1999.

[19] E.G. Lightsey, *Development and Flight Demonstration of a GPS Receiver for Space,* Ph.D. Thesis, Department of Aeronautics and Astronautics," Stanford University, Feb., 1997.

[20] A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974