10-22-2012

Linear Covariance Techniques for Closed-Loop Guidance Navigation and Control System Design and Analysis

Randall S. Christensen

David Geller

Original Article

Institution of MECHANICAL ENGINEERS

Journal of AEROSPACE ENGINEERING

Proc IMechE Part G:
J Aerospace Engineering
228(1) 44–65
© IMechE 2012
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0954410012467717
uk.sagepub.com/jaero
SAGE

Linear covariance techniques for closed-loop guidance navigation and control system design and analysis

Randall S Christensen[1] and David Geller[2]

Abstract

While linear covariance analysis is widely used for navigation system design and analysis, it is often overlooked as a tool for closed-loop guidance navigation and control (GN&C) system design and analysis. This article presents an overview of the techniques and methods required to develop a linear covariance analysis tool for a close-loop GN&C system. Then, using a simple nonlinear closed-loop GN&C problem as a guide, the capabilities of linear covariance analysis for the design and analysis of closed-loop systems are demonstrated. It is shown that linear covariance can be accurately applied to a closed-loop system with time-to-go guidance, dead-reckoning navigation, and a Kalman filter for state estimation. The accuracy and efficiency of linear covariance analysis is shown by direct comparison to Monte Carlo analysis results, and the value of linear covariance analysis is highlighted by presenting several analysis capabilities that are often required in the design and analysis of closed-loop GN&C systems. It is also shown how the efficiency of linear covariance enables new design methodologies, one of which is presented in this article, that would otherwise be prohibitive with Monte Carlo analysis.

Keywords

Closed-loop linear covariance, GN&C, Monte Carlo, Kalman filter, navigation, control, tutorial

Date received: 23 May 2012; accepted: 22 October 2012

Introduction

Monte Carlo analysis is commonly the tool of choice for closed-loop GN&C system design and analysis. However, Monte Carlo analysis may require several hundreds to thousands of simulation runs, depending on the desired probability levels and confidence intervals of the results. Another option for closed-loop GN&C system design and analysis is linear covariance analysis. Linear covariance can provide the same statistical information as a Monte Carlo analysis, but avoids the computational burden associated with the Monte Carlo approach.

Historically, linear covariance theory has been applied to general estimation theory problems,[1–3] as well as in the design and analysis of orbit determination algorithms,[4–7] inertial navigation systems,[8–11] and attitude determination systems.[12–14] These analysis approaches are more commonly known as *consider analysis*,[4,15,16] *true covariance analysis*,[12] or *generalized covariance analysis*.[13,17] In all of these examples, the effects of closed-loop guidance and control on the overall performance of the system is not considered.

Recent developments in linear covariance theory have combined the developments of Battin[18] and Maybeck[3] with continuous feedback control and model replacement[19] (i.e. state propagation using gyro or accelerometer measurements) to produce linear covariance tools that can be applied to many different types of closed-loop GN&C problems.[20] Specific applications include autonomous rendezvous,[20,21] powered lunar descent,[22] and launch vehicle ascent trajectory analysis.[23]

While the linear covariance theory for closed-loop GN&C systems has been successfully applied to these applications, the complexity of the applications has often masked the underlying theory and overshadowed the real value of the linear covariance analysis. Thus, a clear demonstration of its value and usefulness in the context of a simpler closed-loop GN&C problem is warranted.

[1] Mechanical Engineering Group, Space Dynamics Laboratory, North Logan, UT, USA
[2] Mechanical and Aerospace Engineering Department, Utah State University, North Logan, UT, USA

Corresponding author:
Randall S Christensen, Mechanical Engineering Group, Space Dynamics Laboratory, 1695 Research Parkway, North Logan, UT 84341, USA.
Email: randy.christensen@sdl.usu.edu

The objectives of this article are three-fold: first is to demonstrate the usefulness of linear covariance analysis in closed-loop system design and analysis; second is to demonstrate the accuracy and efficiency of linear covariance analysis by direct comparison to Monte Carlo analysis using a simple nonlinear closed-loop GN&C problem as a guide; third is to serve as a tutorial on linear covariance analysis for the uninitiated. The value of linear covariance analysis is highlighted by presenting several capabilities that are often required in the design and analysis of closed-loop GN&C systems. It is also shown that the efficiency of linear covariance enables large trade studies that are often prohibitive with Monte Carlo-based approaches.

The remainder of this article is organized into five sections. The 'Methods' section discusses the generic architecture of a GN&C simulation and the key metrics used in GN&C analysis. It also provides an overview of the mathematical models required for both Monte Carlo and linear covariance analysis. The 'Illustration' section defines a particular nonlinear closed-loop GN&C problem to be studied in this article. The problem contains a time-to-go guidance law, a simple dead-reckoning navigation system, and a Kalman filter state estimator. Linear covariance analysis is then applied to this problem, and a demonstration of several linear covariance analysis capabilities are presented in the 'Capabilities' section. Finally, the contents of the study are summarized in the 'Conclusions' section.

Methods

Figure 1 describes the overall setup of a Monte Carlo analysis for a generic GN&C system.[20] The variables $\mathbf{w}$, $\boldsymbol{\eta}$, and $\boldsymbol{\upsilon}_k$ (to be discussed below) are white random noise processes that, along with actuator commands, $\hat{\mathbf{u}}$, drive the dynamics of the truth model. The truth models as developed in Geller[20] account for the environmental models, sensors error models, and actuator error models. The output from the truth model is the true state $\mathbf{x}$, and the simulated sensor data $\tilde{\mathbf{z}}_k$, $\tilde{\mathbf{y}}$ (also to be discussed below). The navigation algorithm processes the measurements and produces an estimate of the state $\hat{\mathbf{x}}$, upon which the control algorithm operates and produces the actuator command, thus closing the loop. Quantities important to the GN&C analysis, $\delta\mathbf{x}$, $\delta\hat{\mathbf{x}}$, and $\delta\mathbf{e}$, are also illustrated in Figure 1. The true state dispersion, $\delta\mathbf{x}$, is the deviation of the true state from the reference trajectory. This state is absent in traditional linear covariance applications such as consider analysis, true covariance analysis, and generalized covariance analysis. The navigation dispersion $\delta\hat{\mathbf{x}}$ is the deviation of the navigation state from the same reference trajectory, and the true navigation error $\delta\mathbf{e}$ is the difference between the estimated navigation state and the corresponding true navigation state.

A Monte Carlo analysis computes the covariance of the true dispersion and error states by simply generating N samples of each state as a function of time, and then estimating the covariances as

$$D_{true}(t) \approx \frac{1}{N-1}\sum_{i=1}^{N}\delta\mathbf{x}(t)\delta\mathbf{x}^T(t) \tag{1}$$

$$D_{nav}(t) \approx \frac{1}{N-1}\sum_{i=1}^{N}\delta\hat{\mathbf{x}}(t)\delta\hat{\mathbf{x}}^T(t) \tag{2}$$

$$P_{true}(t) \approx \frac{1}{N-1}\sum_{i=1}^{N}\delta\mathbf{e}(t)\delta\mathbf{e}^T(t) \tag{3}$$

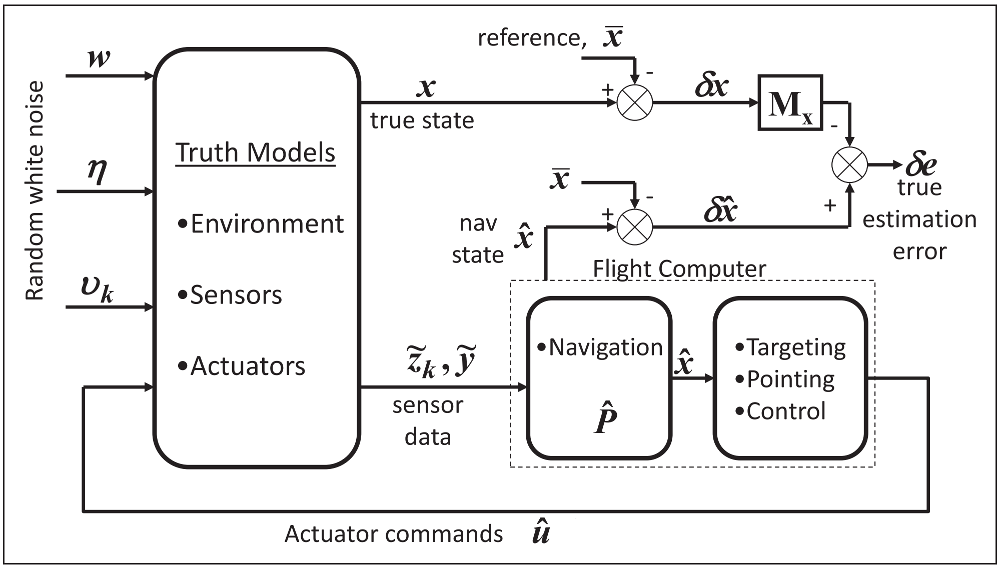

Figure 1. Generic Monte Carlo simulation for GN&C analysis.

When estimating the covariances using equations 1–3, one must consider the uncertainty in the estimates. This is done using confidence intervals given a chosen probability. Figure 2 shows that given a desired confidence interval of ±10% of the computed standard deviation, the number of Monte Carlo runs required for 90%, 95%, and 99% confidence, is approximately 600, 850, and 1500, respectively. In large simulations, even with the computational power of modern computers, the time required to perform the Monte Carlo analysis often becomes prohibitory, making large trade studies costly and impractical.

Conceptually, linear covariance analysis is a linearized version of the nonlinear Monte Carlo analysis. The nonlinear functions describing the truth models, navigation algorithms, and control blocks of Figure 1 must be linearized about a mean reference trajectory. This is often a laborious and time-consuming process, but the return on investment is large when there is a need to perform a large number of trade studies or error budget analysis. Given the validity of the linearization, linear covariance analysis produces the same statistical information as Monte Carlo analysis, but in a single run.

For a linear system, the linear covariance approach is analytical and exact. Care must be taken, however, when applying linear system analysis to a nonlinear system. Many nonlinearities are sufficiently smooth, such that in a localized region about the reference trajectory, the state dynamics can be approximated as linear. This is often true for the case of orbital dynamics, attitude dynamics, structural dynamics, as well as navigation, guidance, and control algorithms. When this condition is not true, however, good engineering judgment, or tools such as statistical linearization[24] need to be considered.

To develop a linear covariance simulation, the following steps must be taken. First, the truth models for the dynamics, sensors, and actuators are defined. The true state vector, $\mathbf{x}$, must be clearly identified, as well as the actuator commands, $\hat{\mathbf{u}}$, process noise $\mathbf{w}$, measurements, $\tilde{\mathbf{z}}_k$ and $\tilde{\mathbf{y}}$, and the associated sensor noise, $\boldsymbol{\eta}$ and $\boldsymbol{\upsilon}_k$. Biases, scale-factors, misalignment, etc., and other colored noise processes are incorporated into the truth models as additional true states. Next, the navigation state vector, $\hat{\mathbf{x}}$, and the associated navigation algorithms are defined. This step first requires the definition of a navigation *design model*, which is later discarded after the navigation algorithms are developed. Note that the navigation design model and the truth models are generally similar, but not identical. Next, the guidance and control laws are defined. Here again, the guidance and control laws are derived from a design model that is discarded once the algorithms are developed. The output of the guidance and control laws, along with compensation for actuator modeling errors, represents the actuator command defined in the truth model. Once the truth models, navigation algorithms, and guidance and control algorithms are fully developed, all the models and algorithms are linearized about the mean or nominal state time-history. Since the dynamics of the true states and the dynamics of the navigation states are coupled, an augmented state vector is created by appending the navigation state vector to the true state vector. This results in a large linear time-varying model of the entire closed-loop system. The mathematical details of this process[20] are summarized in the following section.

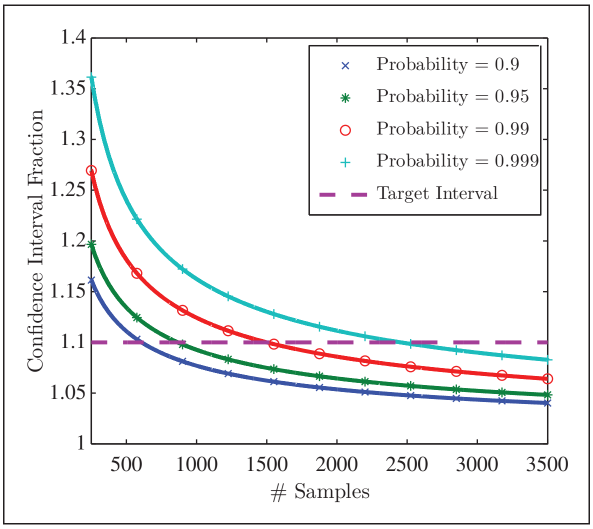

Figure 2. Confidence intervals.

Nonlinear modeling

The general expression for the dynamics of the true state vector can be expressed by the following nonlinear vector differential equation.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{u}}) + B\mathbf{w} \tag{4}$$

where, $\mathbf{x} \in \mathbb{R}^{n}$ is the true state vector, $\hat{\mathbf{u}} \in \mathbb{R}^{n_{\hat{u}}}$ is the actuator command vector as calculated by the flight computer, and $\mathbf{w} \in \mathbb{R}^{n_w}$ is the process noise in the truth state dynamics to account for small modeling uncertainties.

$$E[\mathbf{w}(t)] = \mathbf{0}, \quad E\left[\mathbf{w}(t)\mathbf{w}^T(t')\right] = S_w\delta(t - t') \tag{5}$$

where $E[\cdot]$ corresponds to the expected value, or mean of the quantity. Note that explicit dependencies on time can be included, but have been omitted here to simplify the notation. Continuing the development of the truth models, the inertial measurements, $\tilde{\mathbf{y}} \in \mathbb{R}^{n_{\tilde{y}}}$, (e.g. accelerometer or gyro data) are expressed as a function of the true state vector and the actuator commands

$$\tilde{\mathbf{y}} = \mathbf{c}(\mathbf{x}, \hat{\mathbf{u}}) + \boldsymbol{\eta} \tag{6}$$

where $\boldsymbol{\eta} \in \mathbb{R}^{n_{\tilde{y}}}$ is the measurement noise vector.

$$E[\boldsymbol{\eta}(t)] = \mathbf{0}, \quad E\left[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t')\right] = S_\eta \delta(t - t') \tag{7}$$

The noninertial measurements, $\tilde{\mathbf{z}}_k \in \mathbb{R}^{n_z}$, (e.g. discrete Doppler data or altimeter data) are expressed as a function of the true state vector

$$\tilde{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{v}_k \tag{8}$$

where $\boldsymbol{v}_k \in \mathbb{R}^{n_z}$is the measurement noise.

$$E[\boldsymbol{v}_k] = \mathbf{0}, \quad E\left[\boldsymbol{v}_k \boldsymbol{v}_{k'}^T\right] = R_\nu \delta_{kk'} \tag{9}$$

This completes the truth models for the dynamics, sensors, and actuators.

Next, the navigation algorithms and guidance and control algorithms are defined. Note that all elements of the flight algorithms are decorated with a '$\hat{\ }$'.

The navigation state vector, $\hat{\mathbf{x}} \in \mathbb{R}^{\hat{n}}$, is propagated and updated according to

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}}, \tilde{\mathbf{y}}) \tag{10}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \hat{K}_k\left[\tilde{\mathbf{z}}_k - \hat{\tilde{\mathbf{z}}}_k\right] \tag{11}$$

where $\hat{\tilde{\mathbf{z}}}_k \in \mathbb{R}^{n_z}$ is the estimated value of the measurement

$$\hat{\tilde{\mathbf{z}}}_k = \hat{\mathbf{h}}(\hat{\mathbf{x}}_k) \tag{12}$$

The Kalman gain is determined by

$$\hat{K}_k = \hat{P}_k^- \hat{H}_k^T \left(\hat{H}_k \hat{P}_k^- \hat{H}_k^T + \hat{R}_\nu\right)^{-1} \tag{13}$$

where the filter state covariance matrix, $\hat{P}_k$, is propagated and updated according to

$$\dot{\hat{P}} = \hat{F}\hat{P} + \hat{P}\hat{F}^T + \hat{Q} \tag{14}$$

$$\hat{P}_k^+ = \left(I - \hat{K}_k \hat{H}_k\right)\hat{P}_k^- \left(I - \hat{K}_k \hat{H}_k\right)^T + \hat{K}_k \hat{R}_\nu \hat{K}_k^T \tag{15}$$

Note that $\hat{\mathbf{f}}$ and $\hat{\mathbf{h}}$, $\hat{F}$, $\hat{H}$, $\hat{Q}$, and $\hat{R}$ are all determined by the navigation design model. The true values of the navigation state vector can be derived from the true state vector via the mapping

$$\mathbf{x}_n = \mathbf{m}(\mathbf{x}) \tag{16}$$

where $\mathbf{x}_n \in \mathbb{R}^{\hat{n}}$ are the true value of the navigation states.

Lastly, the actuator commands, $\hat{\mathbf{u}} \in \mathbb{R}^{n_{\hat{u}}}$, are represented as a function of the navigation state vector

$$\hat{\mathbf{u}} = \hat{\mathbf{g}}(\hat{\mathbf{x}}) \tag{17}$$

Linear modeling

For linear covariance analysis, the equations in the preceding section are linearized about the mean reference trajectory defined by $\overline{\mathbf{x}}$ to produce a set of equations that describe the time evolution of the true state dispersions, $\delta\mathbf{x}$, and the navigation state dispersions, $\delta\hat{\mathbf{x}}$. The state propagation equations (4) and (10) are linearized along with equations (6) and (17) to produce

$$\delta\dot{\mathbf{x}} = F_{\mathbf{x}}\delta\mathbf{x} + F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + B\mathbf{w} \tag{18}$$

$$\delta\dot{\hat{\mathbf{x}}} = \left(\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\right)\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}}\delta\mathbf{x} + \hat{F}_{\tilde{\mathbf{y}}}\boldsymbol{\eta} \tag{19}$$

where uppercase characters denote partial derivatives taken with respect to the variable indicated by the subscript and evaluated along the reference trajectory (e.g. $F_{\mathbf{x}} = \partial\mathbf{f}/\partial\mathbf{x}|_{\overline{\mathbf{x}}}$ or $\hat{G}_{\hat{\mathbf{x}}} = \partial\hat{\mathbf{g}}/\partial\hat{\mathbf{x}}|_{\mathbf{m}(\overline{\mathbf{x}})}$).

The navigation state update described by equation (11) is also linearized along with the measurement equations (8) and (12) to yield

$$\delta\mathbf{x}_k^+ = \delta\mathbf{x}_k^- = \delta\mathbf{x}_k \tag{20}$$

$$\delta\hat{\mathbf{x}}_k^+ = \left[I - \hat{K}_k\hat{H}_k\right]\delta\hat{\mathbf{x}}_k^- + \hat{K}_k H_k \delta\mathbf{x}_k^- + \hat{K}_k\boldsymbol{v}_k \tag{21}$$

where the Kalman gain, $\hat{K}_k$, is determined from equations (13) to (15).

Next the navigation state vector is appended to the true state vector to form the augmented state vector $\mathbf{X} \in \mathbb{R}^{\mathbf{n}+\hat{\mathbf{n}}}$.

$$\mathbf{X} = \begin{bmatrix} \delta\mathbf{x} \\ \delta\hat{\mathbf{x}} \end{bmatrix} \tag{22}$$

The dynamics and update equation for the augmented state vector are defined as

$$\dot{\mathbf{X}} = \mathcal{F}\mathbf{X} + \mathcal{G}\boldsymbol{\eta} + \mathcal{W}\mathbf{w} \tag{23}$$

$$\mathbf{X}_k^+ = \mathcal{A}_k\mathbf{X}_k^- + \mathcal{D}_k\boldsymbol{v}_k \tag{24}$$

where

$$\mathcal{F} = \begin{bmatrix} F_{\mathbf{x}} & F_{\hat{u}}\hat{G}_{\hat{x}} \\ \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}} & \hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}} \end{bmatrix}, \qquad \mathcal{G} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ \hat{F}_{\tilde{\mathbf{y}}} \end{bmatrix},$$
$$\mathcal{W} = \begin{bmatrix} B \\ 0_{\hat{n}\times n_w} \end{bmatrix} \tag{25}$$

$$\mathcal{A}_k = \begin{bmatrix} I_{n\times n} & 0_{n\times\hat{n}} \\ \hat{K}_k H_k & I_{\hat{n}\times\hat{n}} - \hat{K}_k\hat{H}_k \end{bmatrix}, \quad \mathcal{D}_k = \begin{bmatrix} 0_{n\times n_z} \\ \hat{K}_k \end{bmatrix} \tag{26}$$

Thus, the entire closed-loop system is represented by the linear time-varying differential equation in equation (23), and the linear difference equation in equation (24). A variety of linear system analyses can now be applied, including covariance analysis.

Note that because $E[\mathbf{x}(t)] = E[\bar{\mathbf{x}}(t)]$ and $E[\hat{\mathbf{x}}(t)] = E[\bar{\mathbf{x}}_n(t)]$ the following relationships hold

$$E[\delta\mathbf{x}(t)] = E[\mathbf{x} - \bar{\mathbf{x}}] = \mathbf{0} \tag{27}$$

$$E[\delta\hat{\mathbf{x}}(t)] = E[\hat{\mathbf{x}} - \bar{\mathbf{x}}_n] = \mathbf{0} \tag{28}$$

This implies that the mean of the augmented state vector is $E[\mathbf{X}] = \mathbf{0}$ and the covariance of the augmented system can be calculated as

$$C_A = E\left[\mathbf{X}(t)\mathbf{X}^T(t)\right] \tag{29}$$

with the following propagation and update equations

$$\dot{C}_A = \mathcal{F}C_A + C_A\mathcal{F}^T + \mathcal{G}S_\eta\mathcal{G}^T + \mathcal{W}S_w\mathcal{W}^T \tag{30}$$

$$C_A\left(t_k^+\right) = \mathcal{A}_k C_A\left(t_k^-\right)\mathcal{A}_k^T + \mathcal{D}_k R_v \mathcal{D}_k^T \tag{31}$$

where it is assumed that $\boldsymbol{\eta}$ and $\mathbf{w}$ are mutually uncorrelated. Given appropriate initial conditions for C_A, equations (30) and (31) are used to propagate and update the augmented state covariance, where the Kalman gain $\hat{K}_k$ in $\mathcal{D}_k$ is determined by equation (13) and the filter state covariance $\hat{P}_k$ is determined by equations (14) and (15).

Performance evaluation

The overall closed-loop performance of the GN&C system is evaluated by examining the covariance matrix of the true state dispersions. This matrix is easily extracted from the augmented state covariance matrix.

$$D_{true} = E\left[\delta x(t)\delta x^T(t)\right] = \begin{pmatrix} I_{n\times n} & 0_{n\times\hat{n}} \end{pmatrix} \times C_A \begin{pmatrix} I_{n\times n} & 0_{n\times\hat{n}} \end{pmatrix}^T \tag{32}$$

The covariance of the true state dispersion accounts for navigation errors, guidance/control execution errors, and all external disturbances/noise sources.

In a similar manner, the covariance matrix of the navigation state dispersion can also be extracted from the augmented covariance matrix.

$$D_{nav} = E\left[\delta\hat{x}(t)\delta\hat{x}^T(t)\right] = \begin{pmatrix} 0_{\hat{n}\times n} & I_{\hat{n}\times\hat{n}} \end{pmatrix} \times C_A \begin{pmatrix} 0_{\hat{n}\times n} & I_{\hat{n}\times\hat{n}} \end{pmatrix}^T \tag{33}$$

Finally, since the true filter error is the difference between the navigation state vector and the corresponding true state vector, the covariance matrix of the true filter error can also be extracted from the augmented covariance matrix

$$P_{true} = \begin{bmatrix} -M_{\mathbf{x}} & I_{\hat{n}\times\hat{n}} \end{bmatrix} C_A \begin{bmatrix} -M_{\mathbf{x}} & I_{\hat{n}\times\hat{n}} \end{bmatrix}^T \tag{34}$$

where $M_{\mathbf{x}}$ is the partial derivative of the mapping $\mathbf{m}(\mathbf{x})$ evaluated along the reference trajectory ($M_{\mathbf{x}} = \partial\mathbf{m}(\mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}=\bar{\mathbf{x}}}$).

Illustration

To demonstrate the capabilities of linear covariance analysis, a simple one-dimensional rocket-sled problem is examined. In this problem, the objective of the closed-loop control system is to achieve a desired final velocity at a specified final time. System performance is measured by the final velocity error and propellant usage.

The rocket-sled is modeled as a single rigid body constrained to travel perpendicular to gravity. The rocket-sled dynamics include an acceleration actuator, a nonlinear drag term, and a random acceleration disturbance. Accelerometer measurements are used for dead-reckoning, and Doppler velocimeter measurements are processed by a suboptimal Kalman filter for state estimation. A time-to-go guidance and control law is used to compute acceleration commands that are issued and sent to the acceleration actuator. A Δv state associated with propellent usage is included in the truth models to track total Δv.

The design space for this problem includes three distinct system design options. The nominal design, Option 1, utilizes the nominal time-to-go guidance law, a Kalman filter to process the Doppler velocimeter measurements, and accelerometer measurements to propagate the navigation states between velocity measurements. Option 2, is the same as Option 1, except the Kalman filter and the velocimeter measurements are removed, and only accelerometer measurements are used to maintain the navigation states, thus becoming a dead-reckoning navigation system. The final design option, Option 3, is also the same as Option 1, except that the nominal time-to-go guidance law is modified to include drag compensation. The navigation and guidance algorithms for these design options are carefully described in the sections below.

The design space also includes the level of acceleration disturbances, and all sensor and actuator performance specifications. The nominal values of these parameters are presented in Table 1. A list of other parameters that are assumed to be fixed for this problem are shown in Table 2.

Next, the truth models, navigation algorithms, and control algorithms for the rocket-sled problem are defined. For conciseness, the final partial derivatives associated with the propagation and update of the augmented system in equations (25) and (26) are presented in Appendix 2.

Table 1. Design parameters: acceleration disturbance level and sensor/actuator specifications.

Symbol	Nominal value	Units	Description
$\sigma_{d,ss}$	0.01	m/s^2	Disturbance accelerations 3-σ
τ_d	20	s	Disturbance acceleration time constant
$\sigma_{b,ss}$	900	μg	Accelerometer bias 3-σ
τ_b	60	s	Accelerometer bias time constant
$\sigma_{s,ss}$	900	ppm	Accelerometer scale factor 3-σ
τ_s	60	s	Accelerometer scale factor time constant
$\sigma_{e,ss}$	0.03	m/s^2	Actuator bias 3-σ
τ_e	60	s	Actuator bias time constant
σ_a	150	$\mu g\sqrt{s}$	Accelerometer measurement noise 3-σ
R_v	0.01	m/s	Velocity measurement noise 3-σ

Table 2. Fixed design parameters.

Symbol	Value	Units	Description
k	1	s	Controller gain
$v_{t,12}$	1039.65	m/s	Pseudo target for Options 1 and 2
$v_{t,3}$	1010.00	m/s	Pseudo target for Option 3
Δt_{accel}	0.01	s	Accelerometer sampling time
t_f	100	s	Time for the target velocity
Δt_{vel}	10	s	Velocimeter sampling time
α	0.00001	1/m	Drag coefficient

Truth models

The truth model for the rocket-sled problem consists of 6 states

$$\mathbf{x} = \begin{bmatrix} v & d & b & s & e & z \end{bmatrix}^T \tag{35}$$

where v is the vehicle velocity, d is the disturbance acceleration on the vehicle, b is the accelerometer bias, s is the accelerometer scale factor, e is actuator bias, and z is Δv used by the propulsion system.

The dynamics of each element of the truth vector are as follows

$$\dot{v} = d + a_{act} - \alpha v^2 \tag{36}$$

$$\dot{d} = -\frac{d}{\tau_d} + w_d \tag{37}$$

$$\dot{b} = -\frac{b}{\tau_b} + w_b \tag{38}$$

$$\dot{s} = -\frac{s}{\tau_s} + w_s \tag{39}$$

$$\dot{e} = -\frac{e}{\tau_e} + w_e \tag{40}$$

$$\dot{z} = | a_{act} | \tag{41}$$

where, a_{act} is the true or actual continuous thrust acceleration from the actuator, and α is the drag coefficient.

It is important to note that the four parameters states, d, b, s, and e are defined as first-order Markov processes or exponentially correlated random variables (ECRV) where the driving noise is assumed to be Gaussian

$$E[w_d(t)] = 0, \qquad E[w_d(t)w_d(t')] = \frac{2\sigma^2_{d,ss}}{\tau_d}\delta(t - t') \tag{42}$$

$$E[w_b(t)] = 0, \quad E[w_b(t)w_b(t')] = \frac{2\sigma^2_{b,ss}}{\tau_b}\delta(t - t') \tag{43}$$

$$E[w_s(t)] = 0, \quad E[w_s(t)w_s(t')] = \frac{2\sigma^2_{s,ss}}{\tau_s}\delta(t - t') \tag{44}$$

$$E[w_e(t)] = 0, \quad E[w_e(t)w_e(t')] = \frac{2\sigma^2_{e,ss}}{\tau_e}\delta(t - t') \tag{45}$$

and where $\sigma^2_{d,ss}$ $\sigma^2_{b,ss}$, $\sigma^2_{s,ss}$, and $\sigma^2_{e,ss}$ are the steady-state variance of the disturbance accelerations, accelerometer bias, accelerometer scale factor, and actuator bias, respectively. The parameters τ_d, τ_b, τ_s, and τ_e correspond to the associated ECRV time constants.

The thruster acceleration model is define as a commanded thrust plus a random bias.

$$a_{act} = \hat{a}_{com} + e \tag{46}$$

where, $\hat{a}_{com}$ is the actuator thrust command as calculated by the flight computer.

There are two measurements available in this problem. The first is a measurement of the acceleration of the vehicle corrupted by scale factor, bias, and noise. The accelerometer model is described as

$$\tilde{a} = (1 + s)(a + b + \eta_a) \tag{47}$$

where, a is the true acceleration of the vehicle and η_a is the accelerometer noise.

The accelerometer noise η_a is assumed to be a Gaussian random process with

$$E[\eta_a(t)] = 0, \quad E[\eta_a(t)\eta_a(t')] = \sigma_a^2\delta(t - t') \tag{48}$$

where σ_a^2 is the power spectral density of the accelerometer measurement noise.

Note that the true acceleration can be expressed as

$$a = d + a_{act} - \alpha v^2 \tag{49}$$

Substituting equation (46) into equation (49) yields

$$a = d + \hat{a}_{com} + e - \alpha v^2 \tag{50}$$

Finally, substituting equation (50) into equation (47) yields the final form for the output of the accelerometers as a function of the true states and noise sources.

$$\tilde{a} = (1 + s)\left(d + \hat{a}_{com} + e - \alpha v^2 + b + \eta_a\right) \tag{51}$$

The second measurement is a discrete Doppler velocity measurement corrupted by noise

$$\tilde{v}_k = v_k + \eta_k \tag{52}$$

where the Doppler noise is assumed to be Gaussian with

$$E[\eta_k] = 0, \quad E[\eta_k \eta_{k'}] = \sigma_v^2 \delta_{kk'} \tag{53}$$

where σ_v^2 is the variance of the velocity measurements.

Navigation and control algorithms

The navigation state vector consists of velocity, accelerometer bias, and accelerometer scale-factor.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{v} & \hat{b} & \hat{s} \end{bmatrix}^T \tag{54}$$

For the development of the navigation filter, a design model (dm) must be defined and subsequently linearized to define the elements of a Kalman filter. The state vector of the filter design model can be defined as

$$\mathbf{x}^{dm} = \begin{bmatrix} v^{dm} & b^{dm} & s^{dm} \end{bmatrix}^T \tag{55}$$

where

$$\dot{v}^{dm} = a^{dm} \tag{56}$$

$$\dot{b}^{dm} = -\frac{b^{dm}}{\tau_b^{dm}} + w_b^{dm} \tag{57}$$

$$\dot{s}^{dm} = -\frac{s^{dm}}{\tau_s^{dm}} + w_s^{dm} \tag{58}$$

The discrete Doppler velocity measurements can be expressed as a function of the state vector plus noise.

$$z_k^{dm} = \hat{h}\left(\mathbf{x}_k^{dm}\right) + \eta_k^{dm} = v_k^{dm} + \eta_k^{dm} \tag{59}$$

where the Doppler noise is assumed to be a Gaussian random sequence with

$$E\left[\eta_k^{dm}\right] = 0, \quad E\left[\eta_k^{dm} \eta_{k'}^{dm}\right] = R_v^{dm} \delta_{kk'} \tag{60}$$

The true acceleration a^{dm} in equation (56) is modeled as a function of measured acceleration, bias, scale factor, and accelerometer noise, and can be determined by rearranging equation (47) to yield

$$a^{dm} = \frac{\tilde{a}}{(1 + s^{dm})} - b^{dm} - \eta_a^{dm} \tag{61}$$

Substituting equation (61) into equation (56) yields the final form of the design model.

$$\dot{v}^{dm} = \frac{\tilde{a}}{(1 + s^{dm})} - b^{dm} - \eta_a^{dm} \tag{62}$$

$$\dot{b}^{dm} = -\frac{b^{dm}}{\tau_b^{dm}} + w_b^{dm} \tag{63}$$

$$\dot{s}^{dm} = -\frac{s^{dm}}{\tau_s^{dm}} + w_s^{dm} \tag{64}$$

where the noise in the two ECRV's are defined similar to equations (43) and (44).

Equations (62) to (64) can be put in vector format and expressed as

$$\dot{\mathbf{x}}^{dm} = \hat{\mathbf{f}}\left(\mathbf{x}^{dm}, \tilde{a}\right) + \hat{B}\mathbf{w}^{dm} \tag{65}$$

where

$$\hat{\mathbf{f}} = \begin{bmatrix} \dfrac{\tilde{a}}{(1 + s^{dm})} - b^{dm} \\ -\dfrac{b^{dm}}{\tau_b^{dm}} \\ -\dfrac{s^{dm}}{\tau_s^{dm}} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{w}^{dm} = \begin{bmatrix} \eta_a^{dm} & w_b^{dm} & w_s^{dm} \end{bmatrix}^T$$

The system described in equation (65) is then linearized about the reference trajectory to yield the following propagation and update equations

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}}, \tilde{a}), \ \ \hat{\mathbf{x}}_0 = \text{given} \tag{66}$$

$$\dot{\hat{P}} = \hat{F}\hat{P} + \hat{P}\hat{F}^T + \hat{B}\hat{Q}\hat{B}^T, \ \ \hat{P}_0 = \text{given} \tag{67}$$

where,

$$\hat{Q} = E\left[\mathbf{w}^{dm}\left(\mathbf{w}^{dm}\right)^T\right] = \begin{bmatrix} (\sigma_a^{dm})^2 & 0 & 0 \\ 0 & \dfrac{2(\sigma_{b,ss}^{dm})^2}{\tau_b} & 0 \\ 0 & 0 & \dfrac{2(\sigma_{s,ss}^{dm})^2}{\tau_s} \end{bmatrix} \tag{68}$$

$$\hat{F} = \left.\frac{\partial \hat{\mathbf{f}}\left(\mathbf{x}^{dm}, \tilde{a}\right)}{\partial \mathbf{x}^{dm}}\right|_{\hat{\mathbf{x}}} = \begin{bmatrix} 0 & -1 & \dfrac{-\tilde{a}}{(1+\hat{s})^2} \\ 0 & -1/\tau_b^{dm} & 0 \\ 0 & 0 & -1/\tau_s^{dm} \end{bmatrix}_{\hat{\mathbf{x}}} \tag{69}$$

Note that in this extended Kalman filter implementation $\hat{F}$ is evaluated using the current estimate of the state. For the linear covariance analysis, it will be evaluated along the reference trajectory as in a standard Kalman filter.

To complete the navigation algorithm, it remains to define the state and state covariance update equations given the Doppler velocity measurement. The update equations are detailed below.

$$\hat{\mathbf{x}}_{t_k}^{+} = \hat{\mathbf{x}}_{t_k}^{-} + \hat{K}_k[\tilde{z}_k - \hat{z}_k] \tag{70}$$

$$\hat{P}_k^{+} = \left(I - \hat{K}_k\hat{H}_k\right)\hat{P}_k^{-}\left(I - \hat{K}_k\hat{H}_k\right)^T + \hat{K}_k R_v^{dm}\hat{K}_k^T \tag{71}$$

where

$$\hat{H}_k = \left.\frac{\partial \hat{\mathbf{h}}\left(\mathbf{x}^{dm}\right)}{\partial \mathbf{x}^{dm}}\right|_{\hat{\mathbf{x}}}$$

$$\hat{K}_k = \hat{P}_k^{-}\hat{H}^T\left(\hat{H}\hat{P}_k^{-}\hat{H}_k^T + \hat{R}_v\right)^{-1}$$

The superscripts − and + denote the state or state covariance before and after the update, respectively.

For guidance and control, two control laws will be considered. The nominal control law is a time-to-go guidance law given by

$$\hat{a}_{com} = \frac{v_t - \hat{v}}{k + \left(t_f - t\right)} \tag{72}$$

where, v_t is a pseudo target target selected to achieve the desired velocity, 1000 m/s, at specified final time t_f, and k is a positive controller gain to avoid the singularity when $t = t_f$.

The second control law considered is identical to the first, but with the addition of a drag compensation term

$$\hat{a}_{com} = \frac{v_t - \hat{v}}{k + \left(t_f - t\right)} + \hat{\alpha}\hat{v}^2 \tag{73}$$

where $\hat{\alpha}$ is the flight computer's value for the drag coefficient. Although it is not required, the accelerometer errors and the velocimeter noise specifications in equations (60), (68), and (69) as well as the drag coefficient used in equation (73) are assumed to be equal to the true values.

Initial conditions

Two sets of similar initial conditions are required for this problem. The first set specifies the initial condition of the Monte Carlo simulation, while the second set specifies the initial condition of the linear covariance simulation.

For the Monte Carlo analysis, the initial conditions for the true states, filter navigation states, and filter covariance are required. The initial true states are determined from the initial reference state plus a random component based upon the covariance of the initial true state.

$$\bar{\mathbf{x}}(t_0) = 0_{n\times 1} \tag{74}$$

$$P_{\mathbf{xx}}(t_0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{d,ss}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{b,ss}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{s,ss}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{e,ss}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{75}$$

The initial filter state is always set equal to the initial value of the reference navigation state

$$\hat{\mathbf{x}}(t_0) = \mathbf{m}(\bar{\mathbf{x}}, t_0) = 0_{\hat{n}\times 1} \tag{76}$$

and the initial value of the navigation filter covariance is set to

$$\hat{P}(t_0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_{b,ss}^2 & 0 \\ 0 & 0 & \sigma_{s,ss}^2 \end{bmatrix} \tag{77}$$

For the linear covariance analysis, the initial augmented covariance matrix must be consistent with the initial conditions of the Monte Carlo simulation. For this problem, the initial covariance of the augmented system can be partitioned as

$$C_A(t_0) = \begin{bmatrix} E\left[\delta\mathbf{x}_0\delta\mathbf{x}_0^T\right] & 0_{n\times\hat{n}} \\ 0_{\hat{n}\times n} & E\left[\delta\hat{\mathbf{x}}_0\delta\hat{\mathbf{x}}_0^T\right] \end{bmatrix} \tag{78}$$

here

$$E\left[\delta\mathbf{x}_0\delta\mathbf{x}_0^T\right] = P_{\mathbf{xx}}(t_0) \tag{79}$$

$$E\left[\delta\hat{\mathbf{x}}_0\delta\hat{\mathbf{x}}_0^T\right] = 0_{\hat{n}\times\hat{n}} \tag{80}$$

where equation (80) is consistent with the fact that the initial filter state never varies.

Mean reference trajectory

The mean reference velocity and acceleration profiles for Design Options 1, 2, and 3 are shown in Figure 3.

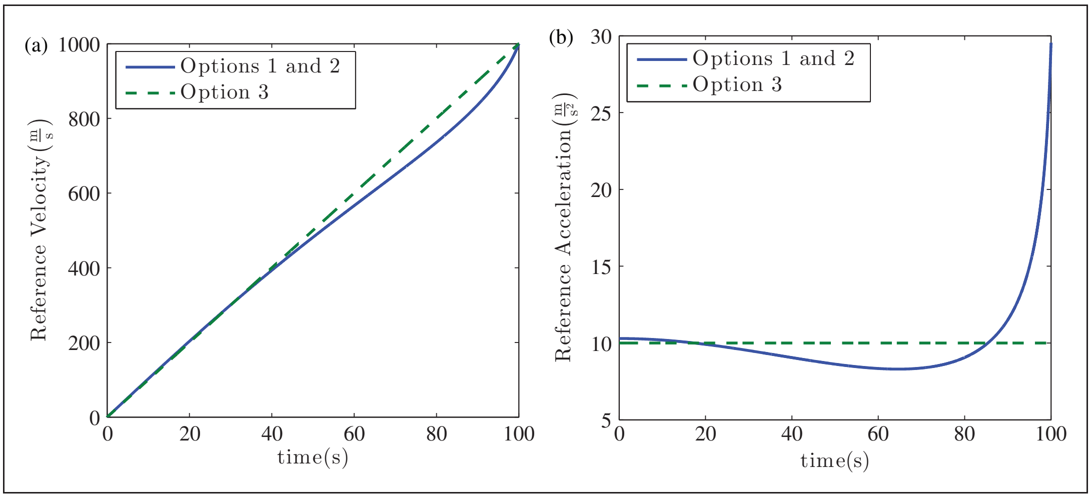

Figure 3. Mean reference velocity (left) and mean reference acceleration (right).

The reference trajectories for Options 1 and 2 are identical, but the inclusion of drag compensation in the guidance law for Option 3 causes significant differences. Since the modified control law can anticipate accelerations due to drag, Option 3 exhibits a very constant acceleration profile and a corresponding linear velocity profile. Options 1 and 2, however, deviate from the linear velocity profile due to the unanticipated influence of drag throughout the trajectory and exhibit a large increase in actuator thrust toward the end of the trajectory. Although the final reference velocity for all options is the same (1000 m/s), the total Δv for Option 3 is 1333 m/s and exceeds the 1297 m/s of Options 1 and 2.

Capabilities

Monte carlo-like results

The first objective of a linear covariance analysis is to be able to quickly duplicate the results of traditional Monte Carlo analysis for a nonlinear closed-loop system with or without a Kalman filter for navigation state estimation. To illustrate this, Option 1 was analyzed with both Monte Carlo analysis and linear covariance analysis. Figure 4 shows the 3-σ true velocity dispersion as a function of time. The light gray lines represent 500 individual Monte Carlo runs. The blue solid line shows the 3-σ dispersion of all Monte Carlo runs with the 95% confidence intervals shown in dashed red. The green dotted line shows the results of a single linear covariance run, which lies inside the confidence intervals, thus replicating the information contained in the Monte Carlo analysis. Since the linear covariance analysis results replicated all the Monte Carlo analysis in this article, only the linear covariance results will be presented from this point forward.

The difference in efficiency of the two techniques was also determined by comparing the execution time. The Monte Carlo analysis took 1914 s to complete, while the linear covariance analysis took 2.6 s to complete, resulting in a speed-up of over 700. As will be illustrated later in this section, the efficiency of linear covariance enables large trade studies that are often prohibitive in the Monte Carlo world.

Navigation and dispersion error budgets

An important feature of linear covariance analysis is that it can be used to quickly compute error budgets for both the true navigation errors and the state dispersions. In this type of analysis, the components of the total navigation error or total state dispersion due to particular error sources are computed. This is accomplished by first zeroing out the variances of all true sources of error, and then rerunning the linear covariance analysis with the variance of only one true error source (or variances of one group of true error sources) turned on at a time. Using this approach, the total state dispersion or the total navigation error is always equal to the square-root of the sum of the individual errors squared (RSS), provided that the individual sources of error are uncorrelated. The navigation algorithm/filter parameters are not varied in this analysis, only the variances of true error sources are turned off and on.

As an example, consider Option 1 in which a Kalman filter is used to process velocimeter measurements, and accelerometer measurements are used to propagate the state between velocity measurement updates. Figure 5 shows the components of velocity estimation error due to each error source as a function of time. The sources of error in this problem are disturbance accelerations, accelerometer errors – bias, scale-factor, and noise – actuator noise, and

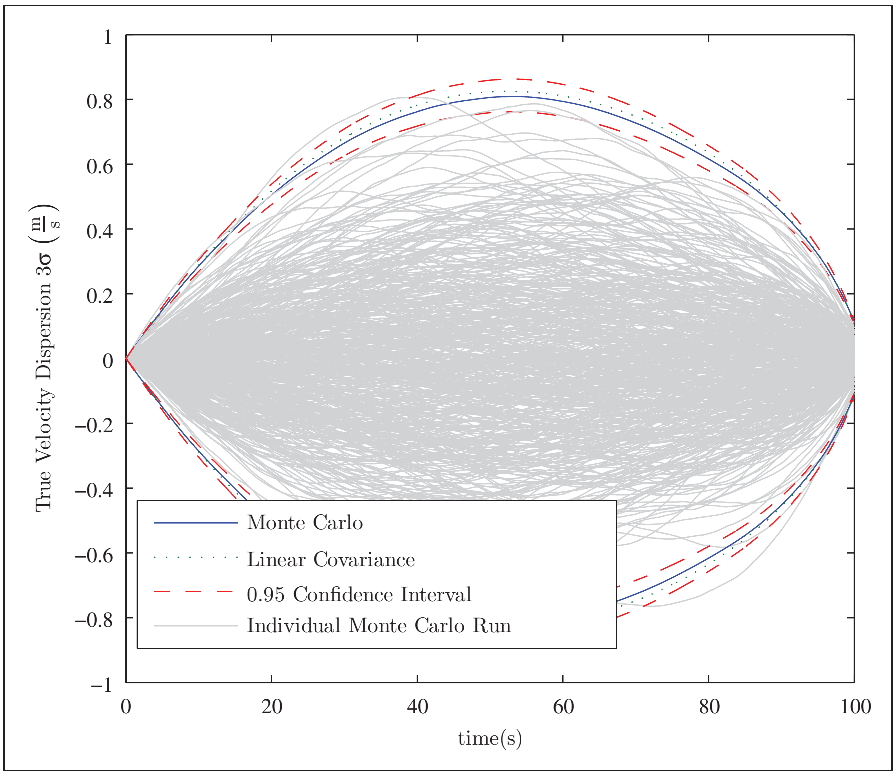

Figure 4. Comparison of Monte Carlo and linear covariance analysis results.

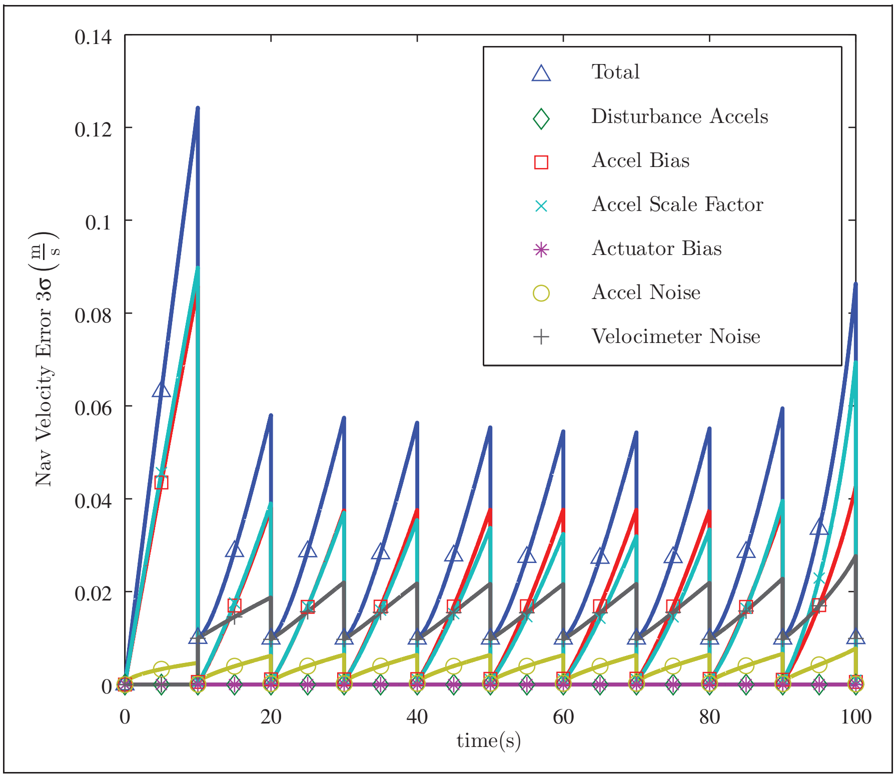

Figure 5. Components of the total navigation velocity error for Option 1.

velocimeter/Doppler noise. Several items are noteworthy. First, the Kalman filter is clearly in operation as is evident in the sharp decrease in error following the measurement update every 10 s. In addition, the change in the error growth rate after the first measurement at $t = 10$ s shows that the Kalman filter has successfully estimated the two dominant sources of error growth, the accelerometer bias and scale

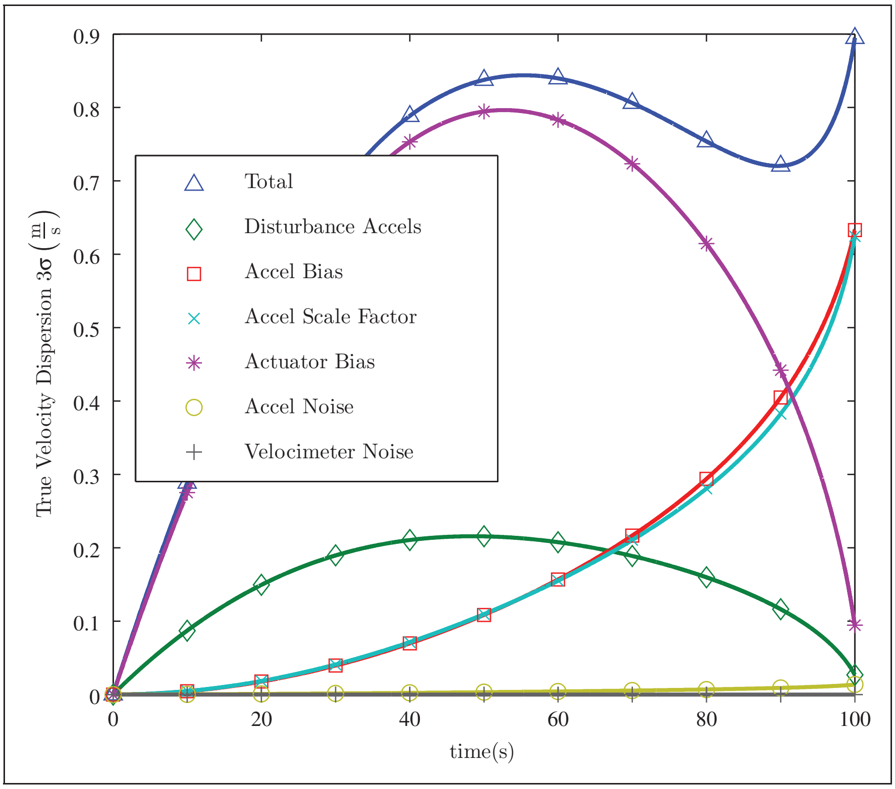

Figure 6. Components of the true velocity dispersion for Option 2.

factor. The rate of error growth increases, however, toward the end of the trajectory since the large accelerations shown in Figure 3 amplify the effect of scale factor estimation error. The effect of the Doppler noise does not appear until after the first Doppler measurement at $t = 10$ s, and the effect of the accelerometer noise is minimal throughout. Finally, as expected neither the disturbance accelerations nor actuator bias errors sources contribute to the velocity estimation error.

The error budget for the dispersion states is determined in a similar manner. This error budget is generally much more valuable than the navigation error budget because the dispersions represent the overall performance of the closed-loop system accounting for navigation errors, guidance and control errors, disturbances, and actuators errors.

As an example, consider Option 2 in which the Kalman filter and velocity measurements are absent, and only accelerometer measurements are used to propagate the state. Figures 6 and 7 show how each source of error contributes to two key metrics, the true velocity dispersion and the true Δv dispersion. Of note in both plots is the increasing contribution of accelerometer error parameters which, because of the absence of a Kalman filter, become the main sources of error at the end of the trajectory. The actuator bias and the disturbances make significant contributions to the dispersion during the burn, but make only a small contribution at the end of the burn. The effect of accelerometer noise is very small throughout, and the effect of velocimeter noise is zero as expected.

Navigation, guidance, and control law evaluation

Linear covariance analysis is also very useful in evaluating the effect of different navigation, guidance, and control laws on overall system performance. Two examples will be illustrated. First, the effect of the navigation system in Option 1 (Kalman filter, accelerometer measurements, and velocimeter measurements) will be compared to the effects of the dead-reckoning navigation system in Option 2. Then, the effects of the nominal time-to-go guidance law in Option 1 will be compared to the time-to-go guidance law with drag compensation in Option 3.

Figure 8 shows the velocity estimation error as a function of time for the two different navigation schemes. In the case of dead-reckoning, the error continues to grow with time as expected. The Kalman filter, however, exhibits a sawtooth shape due to the processing of velocity measurements every 10 s.

The effect of these two different navigation schemes on the final velocity dispersion (i.e. the miss velocity) and total Δv is shown Figures 9 and 10. It is clear that the addition of the velocimeter measurements and the Kalman filter results in an overall dramatic reduction in the final velocity and Δv dispersions. It is also clear that this is due to the Kalman filter's ability to estimate the accelerometer bias and scale factor.

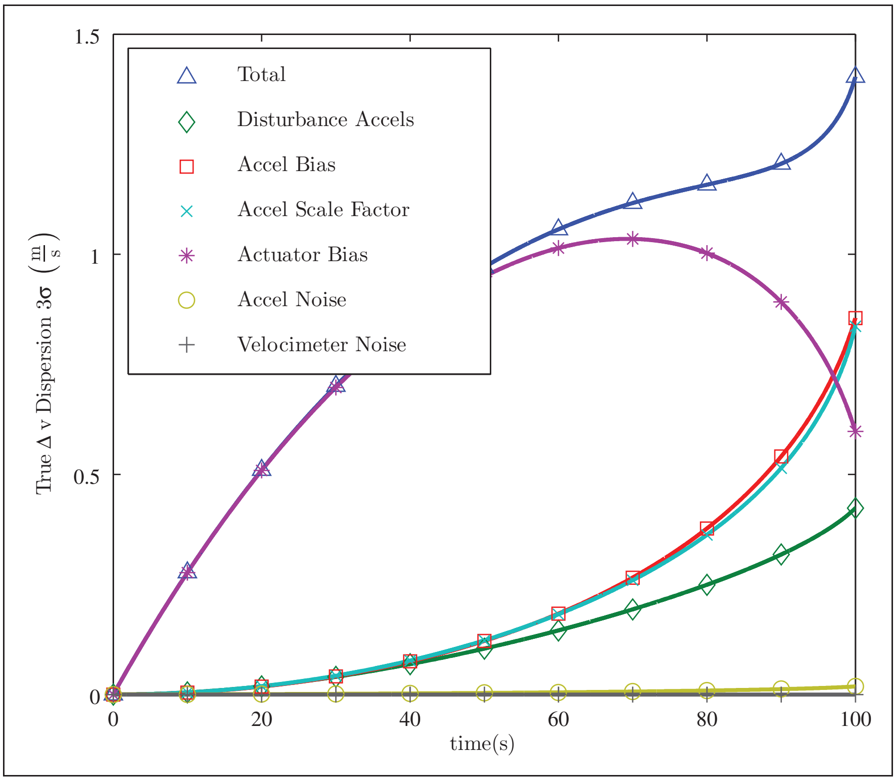

Figure 7. Component of the true Δv dispersion for Option 2.

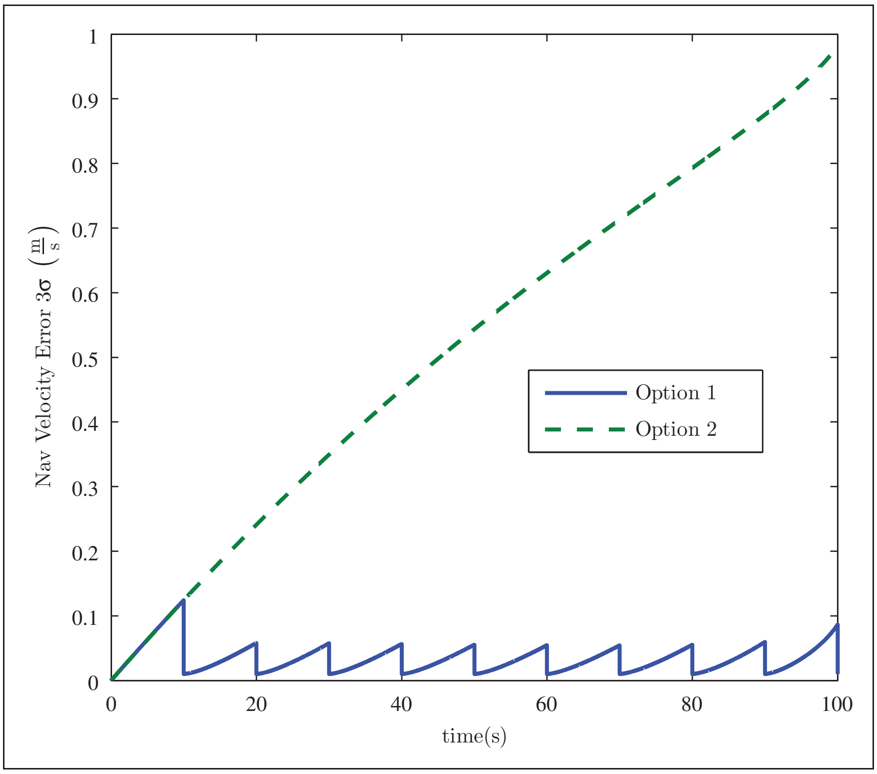

Figure 8. Navigation velocity error for Options 1 and 2.

A similar analysis can be conducted for the two guidance laws in Options 1 and 3. Figure 11 shows that the drag compensation has very little impact on the Kalman filter navigation error. It is only at the very end of the trajectory that the modified control law has a slight advantage over the nominal control law.

Figures 12 and 13 illustrate how the error budgets change due to the addition of the drag model in

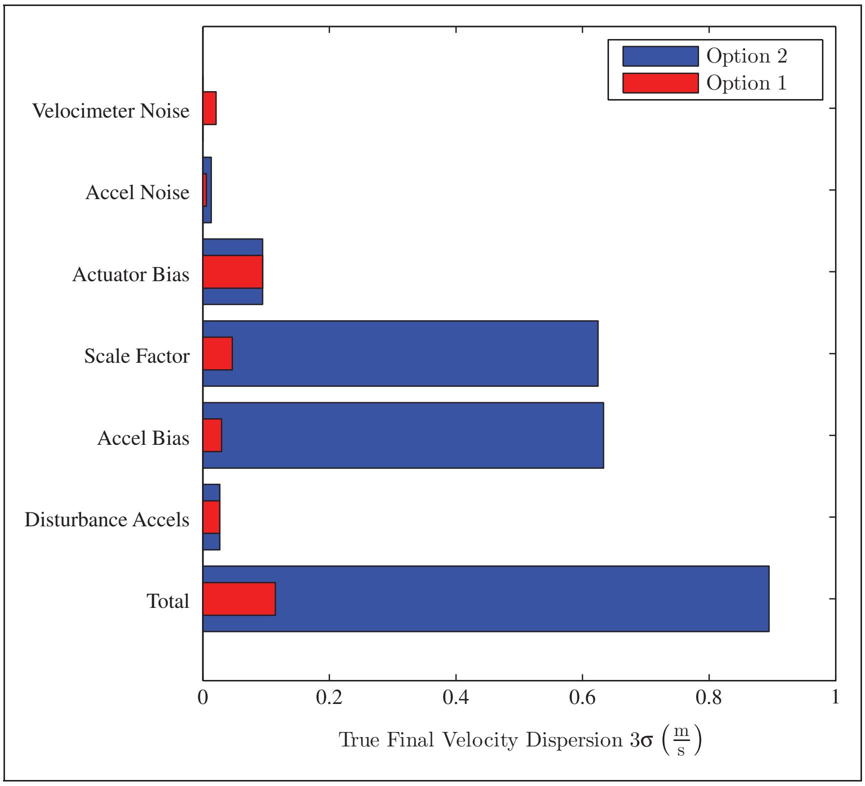

Figure 9. Components of the final velocity dispersions for Options 1 and 2.

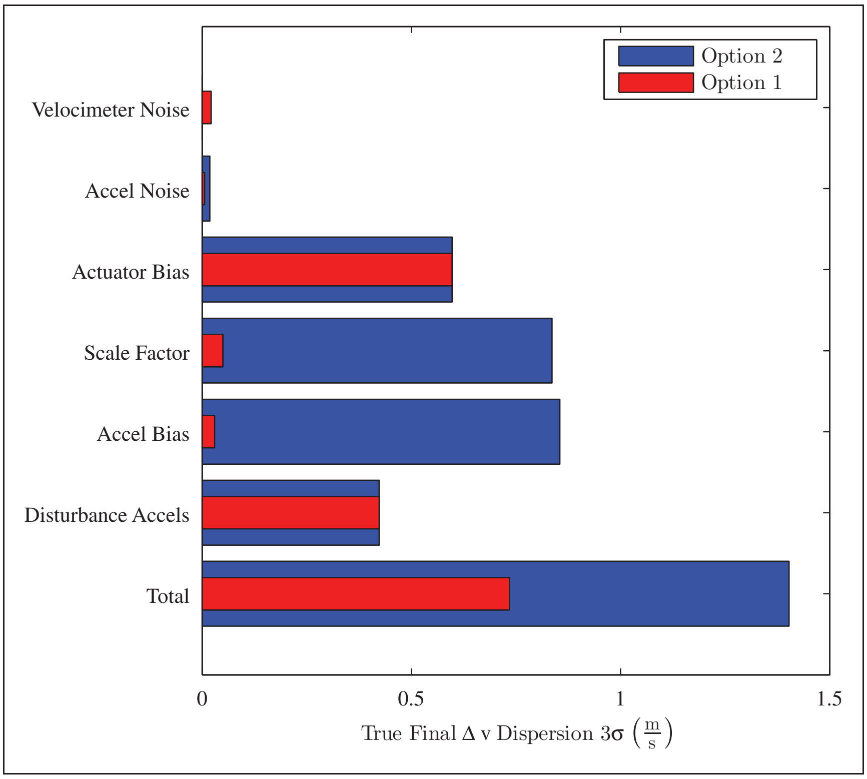

Figure 10. Components of the total final Δv dispersion for Options 1 and 2.

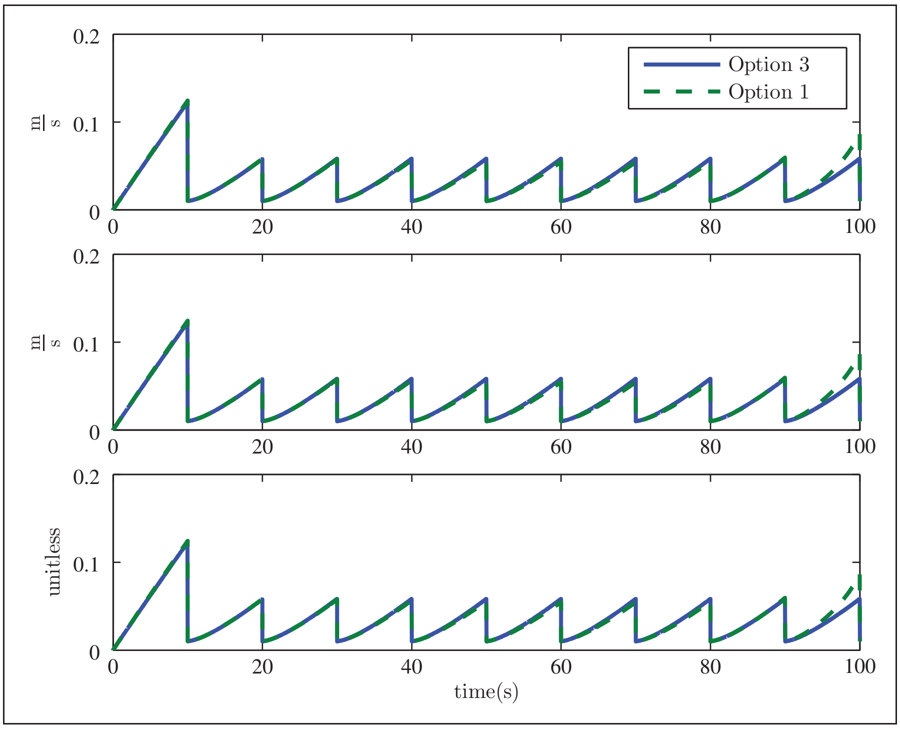

Figure 11. Comparison of velocity estimation errors for Options 1 and 3. Top: velocity estimation error. Middle: accelerometer bias estimation error. Bottom: accelerometer scale factor estimation error.

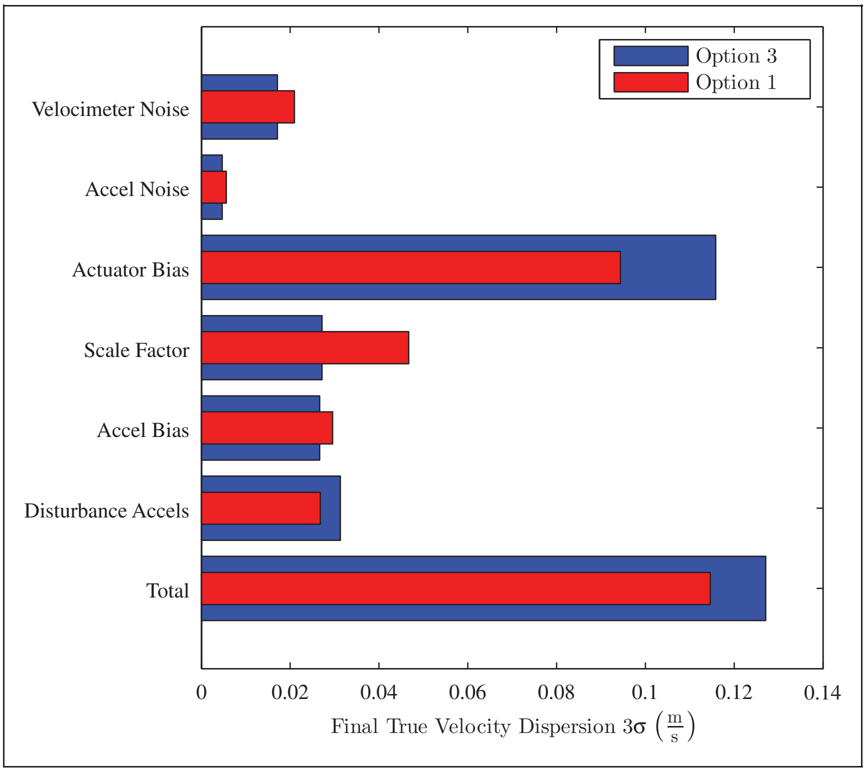

Figure 12. Final velocity dispersions for Options 1 and 3.

the modified control law. The velocity dispersion contribution from the velocimeter and accelerometer errors decrease, but the contribution from the actuator bias error and environment disturbances increase, with a slight increase in the overall final velocity dispersion.

From a Δv perspective, Figure 13 shows that the modified control law again decreases the contribution

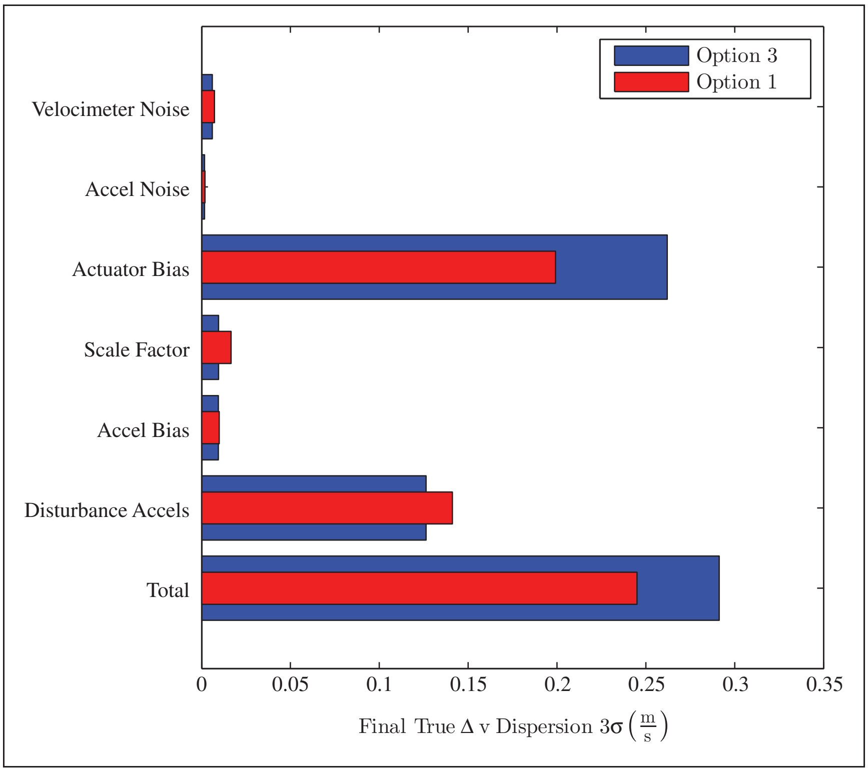

Figure 13. Final Δv dispersions for design Options 1 and 3.

Table 3. Sensor and actuator specifications: low accuracy, nominal, and high accuracy.

Symbol	Low accuracy	Nominal value	High accuracy	Units	Description
$\sigma_{b,ss}$	9000	900	90	µg	Accelerometer bias 3-σ
$\sigma_{s,ss}$	9000	900	90	ppm	Accelerometer scale factor 3-σ
$\sigma_{e,ss}$	0.3	0.03	0.003	m/s^2	Actuator bias 3-σ
σ_a	1500	150	15	µg$\sqrt{s}$	Accelerometer measurement noise 3-σ
R_ν	0.1	0.01	0.001	m/s	Velocity measurement noise 3-σ

of the velocimeter and accelerometer error terms, but also decreases the contribution of the disturbance environment. Thus the only contribution that increased upon changing adding drag compensation is that of the actuator bias. Considering the nominal trajectory differences discussed in the Mean reference trajectory subsection, this results in a total fuel usage of 1333±0.29 m/s for Option 3 and 1297±0.25 m/s for Option 1. Thus, there appears to be no benefit in using the ad hoc modified control law except for a reduction in the level of applied accelerations.

Sensor/actuator specifications and disturbances

Disturbances and sensor/actuator specifications play a large role in overall closed-loop system performance. While an accurate and expensive suite of instruments may improve the performance of the system, a relatively inaccurate low cost suite of instrument may be suitable in meeting mission requirements. Linear covariance analysis is especially suited to conduct this type of analysis.

Consider the performance of the closed-loop system in Option 3 for three different instrument suites: a low-cost/low-accuracy suite of instruments, a nominal suite of instruments, and a high-cost/high accuracy suite of instruments. The specifications of these three instrument suites are shown in Table 3.

Figures 14 and 15 show the associated closed-loop system performance as measured by the final velocity and Δv dispersions. The results show a strong coupling between sensor/actuator specifications and the final velocity and Δv dispersions, as expected, with the coupling being especially strong for actuator errors. Notice that while the sensor and actuator specifications are varying by an order of magnitude, the final miss distance and Δv dispersion is not varying by an order of magnitude. This illustrates a nonlinear relationship between sensor/actuator specifications and system performance, and can be very useful to the

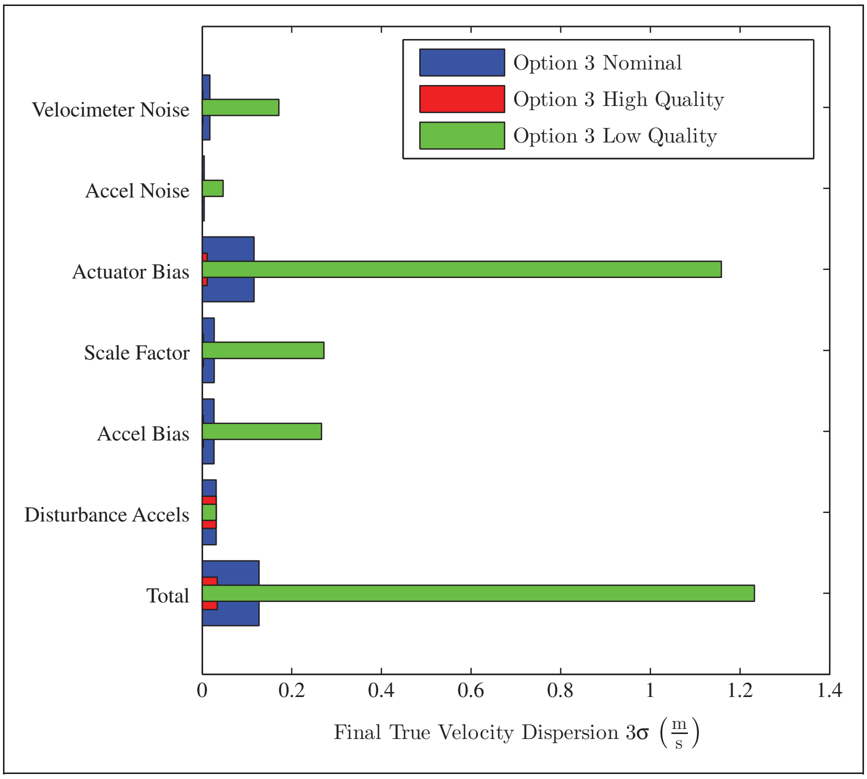

Figure 14. Final velocity dispersions for 3 different sensor/actuator suites.

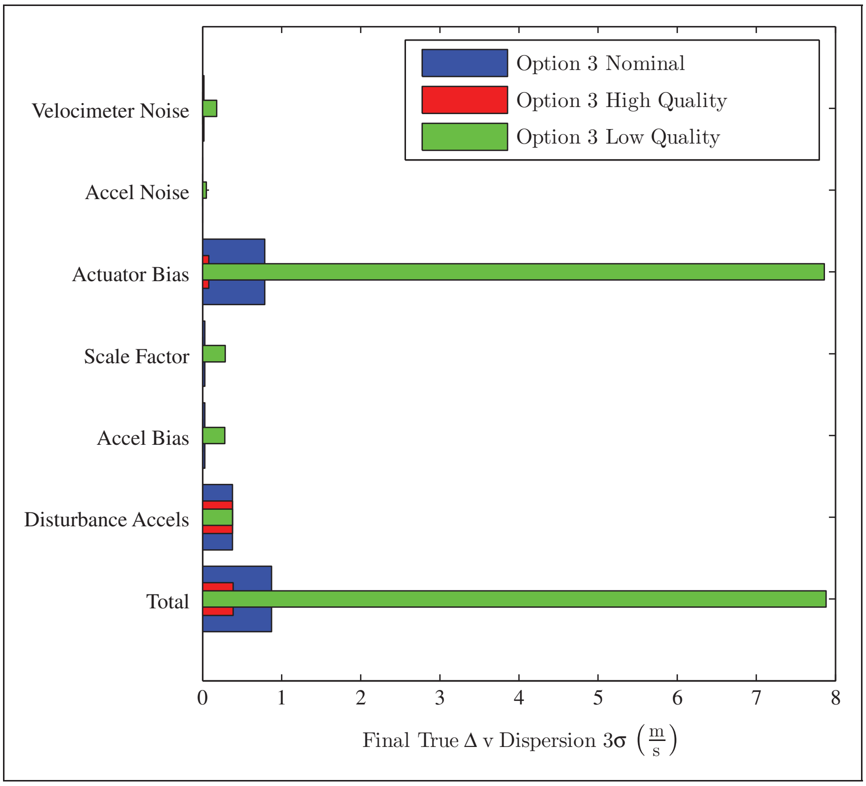

Figure 15. Final Δv dispersions for 3 different sensor/actuator suites.

system designer in determining an optimal design based on overall performance and cost.

Another important sensor specification is the stability (or time constant) of the navigation sensor errors such as the scale factor and bias. Linear covariance analysis can also be used to analyze the effects of these time constants. To illustrate this capability, error budgets for Option 2 are created with short, nominal, and long time constants of $\tau_b = 6$, 60, 600 s for the accelerometer bias and $\tau_s = 6$, 60, 600 s for the accelerometer scale factor. Figures 16 and 17 show that for the dead-reckoning approach to navigation, it is advantageous, both from a velocity dispersion and Δv standpoint, to use sensors whose time constants are short, approaching white noise. Without an external measurement to help estimate these

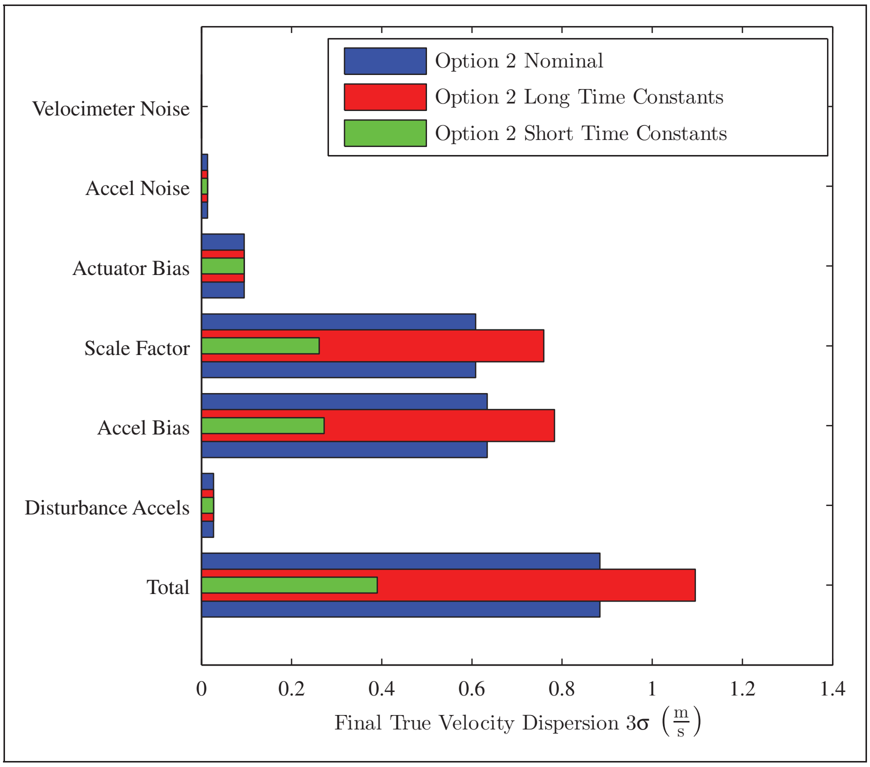

Figure 16. Final velocity dispersions for varying sensor time constants.

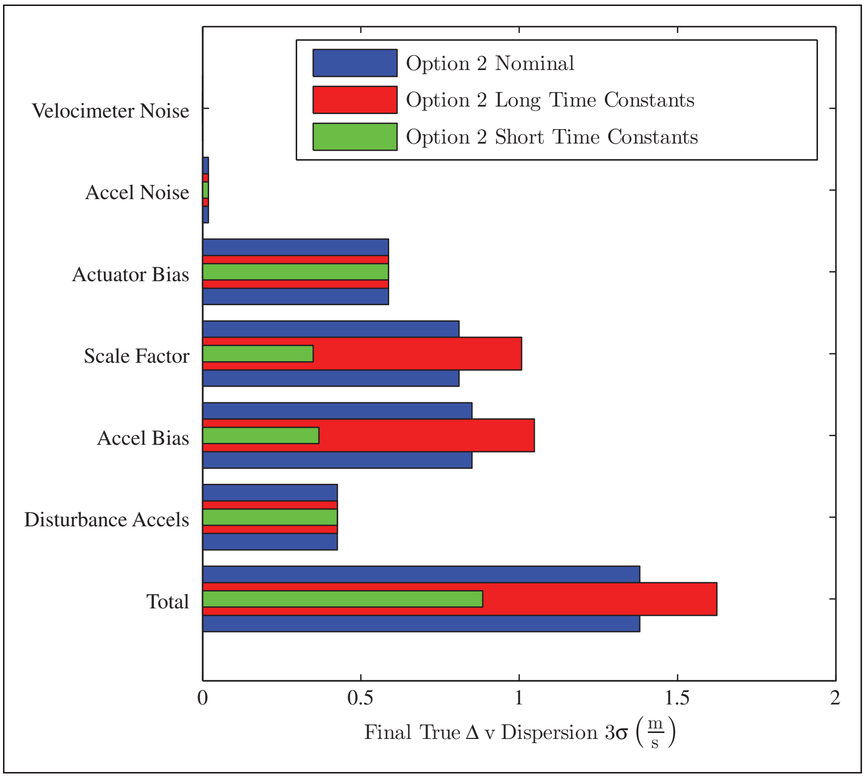

Figure 17. Final Δv dispersions for varying sensor time constants.

errors, the best choice is to select errors sources with short time constants whose effects can be averaged out. An error budget of this type clarifies the coupling of time constants to the overall dispersion, and enables the designer to make critical sensor choices supported by data.

For a fixed set of sensors and actuators, it is also important to understand how the closed-loop system will perform in the presence of varying disturbances. In this analysis, Option 3 is investigated with a high, nominal, and low level of acceleration disturbances, $\sigma_{d,ss} = 0.1\,\mathrm{ms}^2$, $\sigma_{d,ss} = 0.01\,\mathrm{ms}^2$, and $\sigma_{d,ss} = 0.001\,\mathrm{ms}^2$, respectively. Figures 18 and 19 again show the associated closed-loop system performance as measured by the final velocity and Δv dispersions. The results show a strong coupling between the disturbance acceleration and the final velocity and Δv dispersions, as expected. Notice again that the final velocity and Δv dispersions do not vary linearly with the disturbance level.

Closed-loop system design

Since linear covariance is very efficient, it can be used to span all of the different combinations of sensor specs, actuator specs, control laws, and navigation schemes described in this article. In this case, it becomes an excellent preliminary design tool. Using the linear covariance analysis described in this article, the total simulation time required to sweep all combinations of the design space was 38 h. A Monte Carlo approach to this analysis would have been extremely prohibitive, requiring more than 26,000 h or almost 1110 days of simulation time! Thus, the efficiency of linear covariance analysis becomes very relevant, enabling very large trade studies that are not feasible with traditional Monte Carlo methods.

Figures 20 and 21 illustrate the results of spanning the design space. Each point in the graphs represents a single linear covariance run with an associated final velocity dispersion and worst-case $\Delta v = \Delta v_{nominal} + 3\sigma_{\Delta v}$. The dashed lines represent arbitrarily chosen design requirements for the final velocity dispersion and worst-case Δv of 3 m/s and 1335 m/s, respectively. Thus, given design requirements, one is able to identify all possible designs that meet those requirements.

While the different designs can be explored in many ways, one particularly useful way is to color or shape the points according to a quantity of interest. Figure 20 shows the design space with the color/shape of the point indicating the navigation and control scheme (i.e. Option 1, 2, or 3). From this plot it is evident that the lowest performing designs are associated with Option 2 (a dead-reckoning approach to navigation) with a relatively small number of designs meeting the requirements. Although the majority of the designs associated with Options 1 and 3 meet the requirements, they may also be more costly requiring a Doppler velocimeter and a Kalman filter.

Figure 21 shows the same plot as 20, but with the color/shape of the point indicating the actuator bias

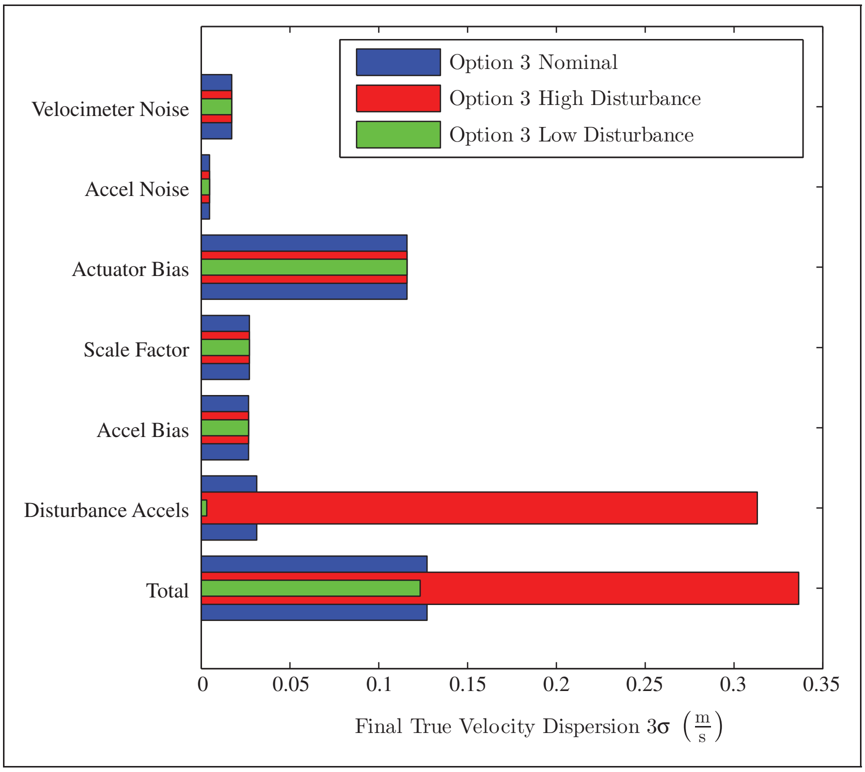

Figure 18. Final velocity dispersions for 3 different disturbance levels.

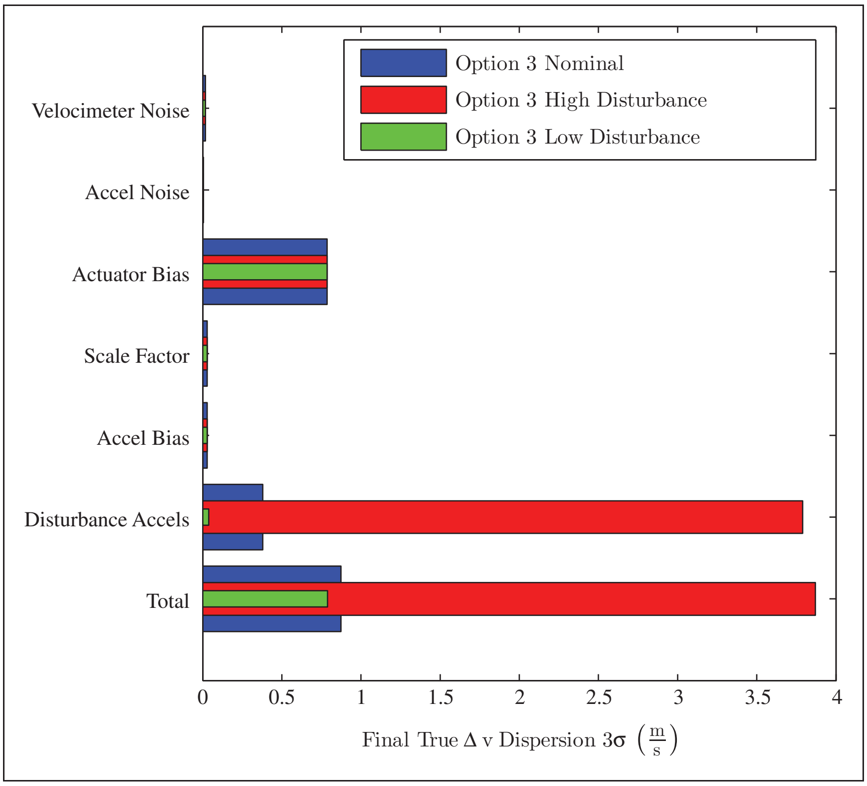

Figure 19. Final Δv dispersions for 3 different disturbance levels.

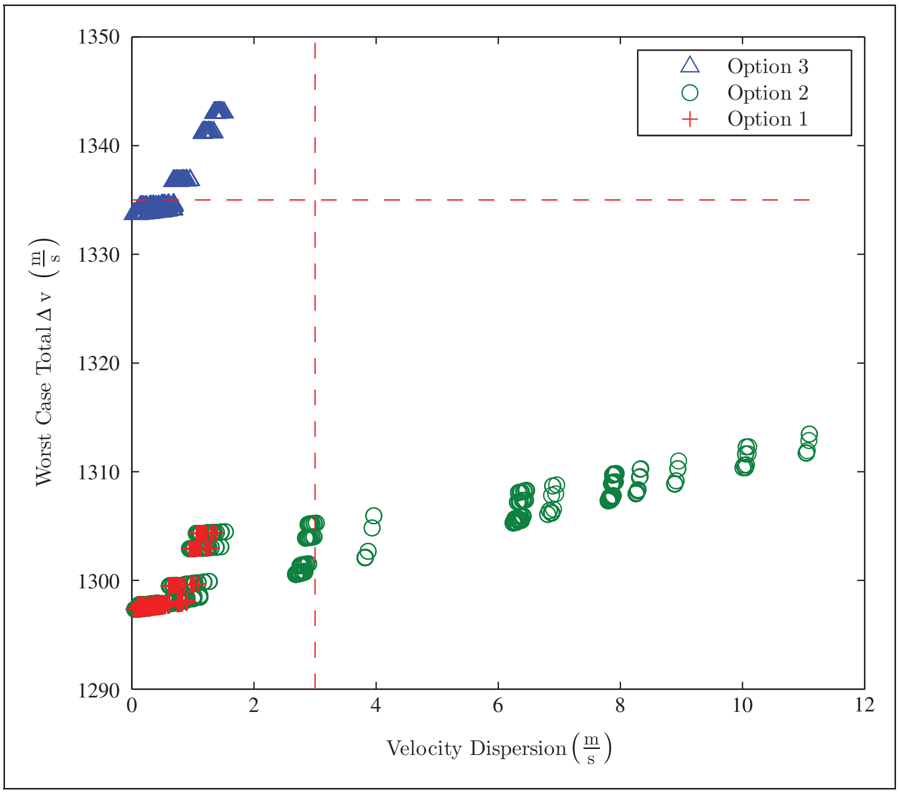

Figure 20. System performance over entire design space as a function of navigation and control law selection.

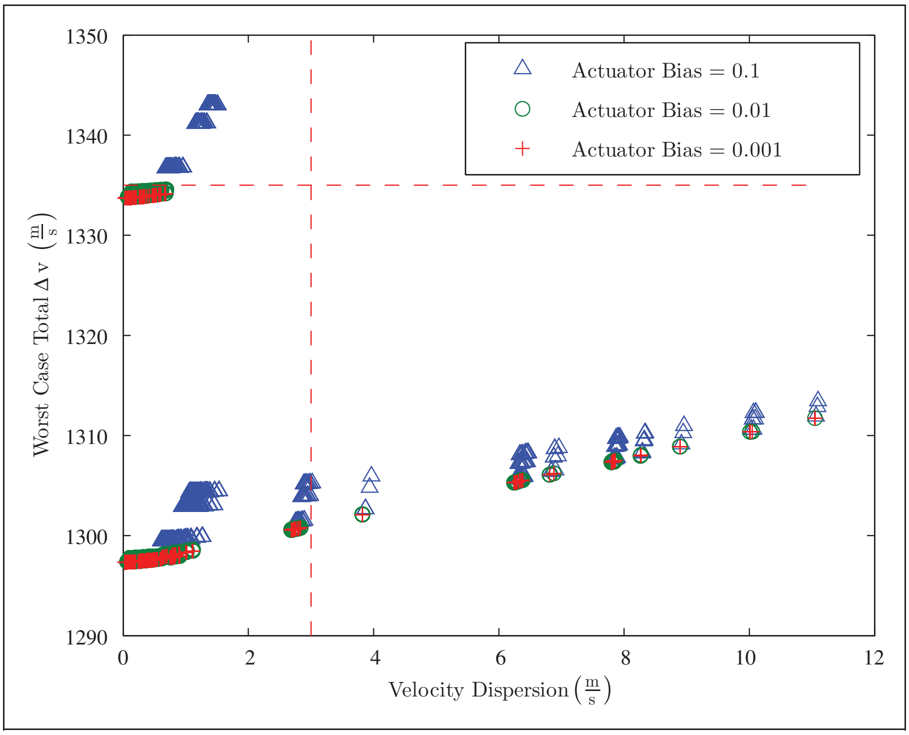

Figure 21. System performance over entire design space as a function of actuator bias.

level. In this plot it is apparent that the actuator bias strongly influences trends in the design space, as expected given the high contribution evident in the error budgets mentioned earlier. Thus the designer can use these plots in conjunction with others to make trades between sensor and actuator specifications as well as navigation and control schemes, and determine the specific design that meets mission requirements, but minimizes cost and complexity.

Conclusions

The underlying theory and value of linear covariance is illustrated in the analysis of a closed-loop rocket-sled navigation and control problem. Using the rocket-sled GN&C problem as an example, linear covariance analysis was shown to produce the same statistical results as Monte Carlo analysis with drastically higher efficiency. The value of linear covariance analysis was highlighted by presenting several analysis capabilities that are often required in the design and analysis of close-loop GN&C systems. The analysis capabilities include: (1) developing navigation and true dispersion error budgets, (2) evaluating the effect of changes to the navigation and control scheme on the error budgets, and (3) evaluating the effect of sensor/actuator specifications and disturbance levels on the error budgets. A new design methodology was also presented in which an exhaustive search of the design space – sensor and actuator specifications, disturbances, and navigation and control laws – can be considered. Using the rocket-sled example, it was shown that linear covariance analysis makes this new design methodology feasible. A design approach of this completeness, enables the designer to identify the system configuration that minimizes cost and complexity, and yet still meets mission requirements.

Funding

This research was conducted with support and funding from the Space Dynamics Laboratory and the Utah State University Research Foundation.

References

1. Gelb A. *Applied optimal estimation*. Cambridge, MA: M.I.T. Press, 1974.
2. Crassidis JL and Junkins JL. *Optimal estimation of dynamic systems*. New York: Chapman and Hall CRC, 2004.
3. Maybeck PS. *Stochastic models, estimation, and control*. vol. 1, New York: Navtech Book and Software Store, 1994.
4. Tabley BD, Schutz BE and Born GE. *Statistical orbit determination*. London: Elsevier Academic Press, 2004.
5. Ionasescu R. Orbit determination covariance analysis for the cassini solstice mission. In: *AIAA-2010-8264, AIAA/AAS astrodynamics specialist conference*, Toronto, Ontario, 2–5 August 2010.
6. Park R, Chodas P and Roundhill I. Covariance analysis of cassini titan flyby using sar and altimetry data. In: *AIAA-2006-6398, AIAA/AAS astrodynamics specialist conference and exhibit*, Keystone, Colorado, 21–24 August 2006.

7. Euler EA. Orbit determination covariance analysis for the cassini solstice mission. In: *AIAA-2010-8264, AIAA/AAS astrodynamics specialist conference*, Toronto, Ontario, 2–5 August 2010.
8. Stastny NB, Bettinger RA and Chavez FR. Lincov analysis of an automated celectial inertial navigation approach for geo satellites. In: *AIAA 2008-6756, AIAA/AAS astrodynamics specialists conference*, Honolulu Hawaii, 2008, pp.184–196.
9. Kreigsman BA and Mahar KB. Gravity model errors in mobile inertial-navigation systems. *J Guid Contr Dyn* 1986; 9(3): 312–318.
10. Edwards RM. Gravity model performance in inertial navigation. *J Guid Contr Dyn* 1986; 5(1): 73–78.
11. Jekili C. Precision free-inertial navigation with gravity compensation by an onboard gradiometer. *J Guid Contr Dyn* 2006; 29(3): 73–78.
12. Bar-Itzhack IY and Harmann RR. Covariance simulation of the euve update filter. In: *Flight mechanics/estimation theory symposium*, Greenbelt, MD, 1989, pp.225–237.
13. Markley FL, Seidewitz E and Nicholson M. A general model for the attitude determination error analysis. In: *NASA conference publication 3011: flight mechanics/estimation theory symposium*, Greenbelt, MD, 1988.
14. Van Der Ha JC. Attitude determination covariance analysis for geostationary transfer orbits. *J Guid Contr Dyn* 1986; 9(2): 156–163.
15. Woodbury D and Junkins J. On the consider kalman filter. In: *AIAA guidance, navigation, and control conference, AIAA-2010-7752*, Toronto, Ontario, 2–5 August 2010.
16. Ambrosuis BAC, Hesper ET and Wakker KF. Application of global positioning system for hermes rendezvous navigation. *J Guid Contr Dyn* 1993; 16(1): 197–205.
17. Carpenter JR and Markley FL. Generalized linear covariance analysis. In: *AAS/ AIAA F. Landis Markley astrodynamics symposium AAS 08-273*, Cambridge, MD, 2008.
18. Battin RH. *An introduction to the mathematics and methods of astrodynamics*, AIAA Education Series. New York: AIAA, 1987.
19. Pittelkau ME. Rotation vector in attitude estimation. *J Guid Contr Dyn* 2003; 26(6): 855–860.
20. Geller DK. Linear covariance techniques for orbital rendezvous analysis and autonomous onboard mission planning. *J Guid Contr Dyn* 2006; 29(6): 1404–1414.
21. Woffinden DC. *Angles-only navigation for autonomous orbital redezvous*. Dissertation. Utah Sate University, 2008.
22. Moesser T, Geller D and Robinson S. Guidance and navigation linear covariance analysis for lunar powered descent. In: *AAS/AIAA astrodynamics specialists conference*, Girdwood, Alaska, August, 2011 AAS Paper II-532.
23. Rose T and Geller D. Linear covariance techniques for powered ascent. In: *AIAA guidance, navigation, and control conference*, Toronto, Ontario, 2–5 August, AIAA-2010-8175.
24. Zarchan P. Complete statistical analysis of nonlinear missile guidance - slam. *J Guid Contr* 1979; 2(1): 71–78.

Appendix I

Notation

a	true acceleration
$\hat{a}_{com}$	actuator thrust command
b	accelerometer bias
C_A	covariance of augmented state vector
d	disturbance accelerations
D_{nav}	covariance of navigation state dispersions
D_{true}	covariance of true state dispersions
e	actuator bias
$I_{n\times n}$	identity matrix
k	controller gain
$\hat{P}$	flight computer covariance of navigation state error
P_{true}	true covariance of navigation state error
$P_{\mathbf{xx}}$	true state covariance
s	accelerometer scale factor
$\hat{\mathbf{u}}$	continuous actuator commands issued by the flight computer
v	velocity
v_t	pseudo target velocity
$\mathbf{w}$	true state process noise
$\mathbf{x}$	true states
$\hat{\mathbf{x}}$	navigation state vector
$\mathbf{x}_n$	true navigation states
$\mathbf{X}$	augmented state vector
$\tilde{\mathbf{y}}$	continuous inertial measurement
z	fuel usage state (Δv)
$\tilde{\mathbf{z}}_k$	discrete noninertial measurements at time t_k
α	drag coefficient
$\delta\mathbf{e}$	true navigation error
$\delta\mathbf{x}$	true state dispersions
$\delta\hat{\mathbf{x}}$	navigation state dispersions
$\boldsymbol{\eta}$	continuous inertial measurement noise
$\boldsymbol{\upsilon}_k$	discrete noninertial measurement noise at time t_k
$0_{m\times n}$	matrix of zeros

Superscript

dm	parameters and states used in the design model
^	parameters, variables, and functions associated with the flight algorithms
-	reference values
~	measured values

Appendix 2

Matrix partial derivatives

This section documents the matrices required to propagate and update the covariance of the augmented system defined in the ‘Linear modeling’ section. Note that an over bar denotes a quantity along the reference trajectory. For example, $\bar{v}$ and $\hat{\bar{a}}_{com}$ denote the velocity and commanded acceleration at a given time along the reference trajectory.

$$F_x = \begin{bmatrix} -2\alpha\bar{v} & 1 & 0 & 0 & 1 & 0 \\ 0 & -1/\tau_d & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/\tau_b & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/\tau_s & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/\tau_e & 0 \\ 0 & 0 & 0 & 0 & \dfrac{\hat{\bar{a}}_{com}}{\left|\hat{\bar{a}}_{com}\right|} & 0 \end{bmatrix} \tag{81}$$

$$F_{\hat{u}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dfrac{\hat{\bar{a}}_{com}}{\left|\hat{\bar{a}}_{com}\right|} \end{bmatrix} \tag{82}$$

$$\hat{G}_{\hat{x}} = \begin{bmatrix} \dfrac{-1}{k + (t_f - t)} & 0 & 0 \end{bmatrix} \text{ for the nominal controller} \tag{83}$$

$$\hat{G}_{\hat{x}} = \begin{bmatrix} \dfrac{-1}{k + (t_f - t)} + 2\alpha\bar{v} & 0 & 0 \end{bmatrix} \tag{84}$$

for the modified controller

$$\hat{F}_{\tilde{\mathbf{y}}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{85}$$

$$C_x = \begin{bmatrix} -2\alpha\bar{v} & 1 & 1 & \hat{\bar{a}}_{com} - \alpha\bar{v}^2 & 1 & 0 & 0 \end{bmatrix} \tag{86}$$

$$\hat{F}_{\hat{\mathbf{x}}} = \begin{bmatrix} 0 & -1 & -\tilde{\bar{a}} \\ 0 & -1/\tau_b & 0 \\ 0 & 0 & -1/\tau_s \end{bmatrix} \tag{87}$$

$$C_{\hat{\mathbf{u}}} = [1] \tag{88}$$

$$S_w = \begin{bmatrix} \dfrac{2\sigma_{d,ss}^2}{\tau_d} & 0 & 0 & 0 \\ 0 & \dfrac{2\sigma_{b,ss}^2}{\tau_b} & 0 & 0 \\ 0 & 0 & \dfrac{2\sigma_{s,ss}^2}{\tau_s} & 0 \\ 0 & 0 & 0 & \dfrac{2\sigma_{e,ss}^2}{\tau_e} \end{bmatrix} \tag{89}$$

$$S_\eta = \sigma_a^2 \tag{90}$$

$$R_v = \sigma_v^2 \tag{91}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{92}$$

$$\hat{H}_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{93}$$