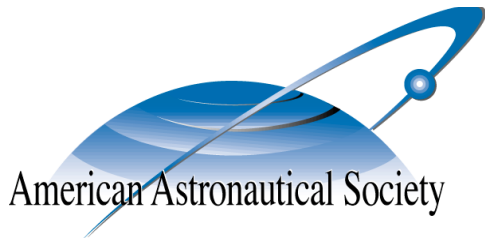


AAS 08-273



GENERALIZED LINEAR COVARIANCE ANALYSIS

J. Russell Carpenter and F. Landis Markley

The F. Landis Markley Astrodynamics Symposium

Cambridge, MD

Jun. 29–Jul. 2, 2008

AAS Publications Office, P.O. Box 28130, San Diego, CA 92198

GENERALIZED LINEAR COVARIANCE ANALYSIS

J. Russell Carpenter* and F. Landis Markley†

We review and extend in two directions the results of prior work on generalized covariance analysis methods. This prior work allowed for partitioning of the state space into “solve-for” and “consider” parameters, allowed for differences between the formal values and the true values of the measurement noise, process noise, and *a priori* solve-for and consider covariances, and explicitly partitioned the errors into subspaces containing only the influence of the measurement noise, process noise, and *a priori* solve-for and consider covariances. In this work, we explicitly add sensitivity analysis to this prior work, and relax an implicit assumption that the batch estimator’s anchor time occurs prior to the definitive span. We also apply the method to an integrated orbit and attitude problem, in which gyro and accelerometer errors, though not estimated, influence the orbit determination performance. We illustrate our results using two graphical presentations, which we call the “variance sandpile” and the “sensitivity mosaic,” and we compare the linear covariance results to confidence intervals associated with ensemble statistics from a Monte Carlo analysis.

NOTATION CONVENTION

Style	Example	Connotation
Plain	a, A	Scalar real number
Bold	\mathbf{x}, \mathbf{X}	Physical vector, i.e. an “arrow” in 3-dimensional, physical space
Calligraphic	\mathcal{T}, \mathcal{C}	Coordinate frame in which physical vectors may be expressed
Sans Serif	P, Φ	Column, row, or matrix of scalars

INTRODUCTION

In Refs. 1 and 2, Markley and others proposed a novel approach to linear covariance analysis, for both batch and sequential estimators. Unlike the methods described in Maybeck³ and Gelb,⁴ Markley, et al.’s method allowed for the partitioning of the state space into “solve-for” and “consider” parameters. Unlike the method described in Tapley, et al.,⁵ Markley, et al.’s method allowed for differences between the formal values and the true values of the measurement noise, process noise, and *a priori* solve-for and consider covariances. Unlike the method described in Jazwinski,⁶ the method of Markley et al. explicitly partitions the errors into subspaces containing only the influence of the measurement noise, process noise, and *a priori* solve-for and consider covariances.

In this paper, we review and extend in two directions the results of Refs. 1 and 2, by explicitly adding sensitivity analysis, and by generalizing the earlier results, which implicitly assumed that the batch estimator’s anchor time occurs prior to the definitive span. We also apply the method to an integrated orbit and attitude problem, in which gyro and accelerometer errors, though not estimated, influence the orbit determination performance. We illustrate our results using two graphical presentations, which we call the “variance sandpile” and the “sensitivity mosaic,” and we compare the linear covariance results to confidence intervals associated with ensemble statistics from a Monte Carlo analysis.

*Flight Dynamics Analysis Branch, NASA Goddard Space Flight Center, Code 595, Greenbelt, MD 20771.

†GN&C Systems Branch, NASA Goddard Space Flight Center, Code 591, Greenbelt, MD 20771

REVIEW AND UPDATE OF METHOD

We summarize the generalized linear covariance analysis algorithm of Refs. 1 and 2 here. The algorithm performs a general covariance analysis linearized about a given reference trajectory, $\bar{X}(t)$. It assumes that true values and formal values may differ for the *a priori* covariance, measurement noise covariance, and process noise power spectral density. We denote these parameters as P_{o-} , \hat{P}_{o-} , R , \hat{R} , Q , and \hat{Q} , respectively. The (error-free) dynamics and measurement partials, $A(t)$ and $H(t)$, and the process and measurement noise covariances, may be time-varying*. The algorithm assumes that linear, possibly time-varying, transformations partition the state space into a “solve-for” state, $s(t)$, which is to be estimated from the measurements, and a “consider” state, $c(t)$, which will not be estimated. The mapping of the state-space into solve-for and consider subspaces is defined according to

$$s(t) = S(t)x(t), \quad c(t) = C(t)x(t) \quad (1)$$

where $x(t) = X(t) - \bar{X}(t)$ is the linear deviation of the estimate from the true state. Mapping back into the original state space may be accomplished as follows:

$$M(t) = \begin{bmatrix} S(t) \\ C(t) \end{bmatrix}, \quad M^{-1}(t) = [\tilde{S}(t), \tilde{C}(t)] \quad (2)$$

$$x(t) = \tilde{S}(t)s(t) + \tilde{C}(t)c(t) \quad (3)$$

The main computational result of the original algorithm is a partitioning of the contributions to the total error from *a priori* uncertainty, measurement noise, and process noise. We denote the true and formal covariances of these error partitions as P_a , \hat{P}_a , P_v , \hat{P}_v , P_w , and \hat{P}_w , respectively. The total true variance and total formal variance are, respectively, the sums of these partitions:

$$P = P_a + P_v + P_w, \quad \hat{P} = \hat{P}_a + \hat{P}_v + \hat{P}_w \quad (4)$$

Another way of looking at the results is via difference between true and formal values of the partitions, defined as

$$\Delta P_a = SP_a S^T - \hat{P}_a, \quad \Delta P_v = SP_v S^T - \hat{P}_v, \quad \Delta P_w = SP_w S^T - P_w \quad (5)$$

It is perhaps easiest to understand the algorithm as it is applied to the sequential Kalman filter, so we begin with this. We then summarize the application of the method to the batch estimator, and generalize the batch application so as to accommodate an arbitrary location of the anchor time. We also generalize the batch form of the algorithm so that it will propagate the true and estimated covariances, and the sensitivity matrix, over a specified time span that does not necessarily correspond to the definitive span. Next, we add to the previous work a computation of the sensitivity to *a priori* errors, and show how the previous work can be used to study sensitivity to measurement and process noise.

Sequential Kalman Filter Form Let

$$\Phi_{ss}(t_i, t_o) = S(t_i)\Phi(t_i, t_o)\tilde{S}(t_o), \quad H_s(t_i) = H(t_i)\tilde{S}(t_i) \quad (6)$$

and

$$\hat{Q}_d(t_i, t_{i-1}) = \int_{t_{i-1}}^{t_i} \Phi_{ss}(t_i, \tau) \hat{Q}(\tau) \Phi_{ss}^T(t_i, \tau) d\tau \quad (7)$$

$$Q_d(t_i, t_{i-1}) = \int_{t_{i-1}}^{t_i} \Phi(t_i, \tau) Q(\tau) \Phi^T(t_i, \tau) d\tau \quad (8)$$

*The assumption that $A(t)$ and $H(t)$ are error-free is not as limiting as may at first seem, since use of incorrect parameters or model truncations by the estimator will be accommodated in the consider state analysis.

where the state transition matrix $\Phi(t_i, t_o)$ is derived from $A(t)$ in the usual way, e.g. as specified in Ref. 7. Then, propagate the various covariance partitions between measurement updates as follows:

$$\hat{P}_a(t_i^-) = \Phi_{ss}(t_i, t_{i-1})\hat{P}_a(t_{i-1}^+)\Phi_{ss}^T(t_i, t_{i-1}), \quad \hat{P}_a(t_1^-) = \hat{P}_{o-} \quad (9)$$

$$P_a(t_i^-) = \Phi(t_i, t_{i-1})P_a(t_{i-1}^+)\Phi^T(t_i, t_{i-1}), \quad P_a(t_1^-) = P_{o-} \quad (10)$$

$$\hat{P}_v(t_i^-) = \Phi_{ss}(t_i, t_{i-1})\hat{P}_v(t_{i-1}^+)\Phi_{ss}^T(t_i, t_{i-1}), \quad \hat{P}_v(t_1^-) = 0 \quad (11)$$

$$P_v(t_i^-) = \Phi(t_i, t_{i-1})P_v(t_{i-1}^+)\Phi^T(t_i, t_{i-1}), \quad P_v(t_1^-) = 0 \quad (12)$$

$$\hat{P}_w(t_i^-) = \Phi_{ss}(t_i, t_{i-1})\hat{P}_w(t_{i-1}^+)\Phi_{ss}^T(t_i, t_{i-1}) + \hat{Q}_d(t_i, t_{i-1}), \quad \hat{P}_w(t_1^-) = 0 \quad (13)$$

$$P_w(t_i^-) = \Phi(t_i, t_{i-1})P_w(t_{i-1}^+)\Phi^T(t_i, t_{i-1}) + Q_d(t_i, t_{i-1}), \quad P_w(t_1^-) = 0 \quad (14)$$

Note that the *a priori* covariance is applied only to the *a priori* partitions, and that process noise is applied only to the process noise partitions.

At each measurement, update the various partitions with a Kalman gain based on the formal covariances:

$$K_i = \hat{P}(t_i^-)H_s^T(t_i)[H_s(t_i)\hat{P}(t_i^-)H_s^T(t_i) + \hat{R}(t_i)]^{-1} \quad (15)$$

$$\hat{P}_a(t_i^+) = [I - K_i H_s(t_i)]\hat{P}_a(t_i^-)[I - K_i H_s(t_i)]^T \quad (16)$$

$$P_a(t_i^+) = [I - \tilde{S}(t_i)K_i H(t_i)]P_a(t_i^-)[I - \tilde{S}(t_i)K_i H(t_i)]^T \quad (17)$$

$$\hat{P}_v(t_i^+) = [I - K_i H_s(t_i)]\hat{P}_v(t_i^-)[I - K_i H_s(t_i)]^T + K_i \hat{R}(t_i) K_i^T \quad (18)$$

$$P_v(t_i^+) = [I - \tilde{S}(t_i)K_i H(t_i)]P_v(t_i^-)[I - \tilde{S}(t_i)K_i H(t_i)]^T + \tilde{S}(t_i)K_i R(t_i)K_i^T \tilde{S}^T(t_i) \quad (19)$$

$$\hat{P}_w(t_i^+) = [I - K_i H_s(t_i)]\hat{P}_w(t_i^-)[I - K_i H_s(t_i)]^T \quad (20)$$

$$P_w(t_i^+) = [I - \tilde{S}(t_i)K_i H(t_i)]P_w(t_i^-)[I - \tilde{S}(t_i)K_i H(t_i)]^T \quad (21)$$

Note that the measurement noise is applied only to the measurement noise partitions.

Batch Estimator Form The batch update maps all the innovations to an anchor time, which is associated with any *a priori* information, but which is not necessarily prior to the definitive data span; to emphasize this, we denote the anchor time with the subscript ‘*’ rather than ‘o.’ The *a priori* and measurement noise partitions can be expressed as follows:

$$K_i = [\hat{P}_{*-}^{-1} + \sum_j \Phi_{ss}^T(t_j, t_*)H_s^T(t_j)\hat{R}_j^{-1}H_s(t_j)\Phi_{ss}(t_j, t_*)]^{-1}\Phi_{ss}^T(t_i, t_*)H_s^T(t_i)\hat{R}_i^{-1} \quad (22)$$

$$\hat{P}_a(t_*^+) = [I - \sum_i K_i H_s(t_i)\Phi_{ss}(t_i, t_*)]\hat{P}_{*-}[I - \sum_i K_i H_s(t_i)\Phi_{ss}(t_i, t_*)]^T \quad (23)$$

$$P_a(t_*^+) = [I - \sum_i \tilde{S}(t_i)K_i H(t_i)\Phi(t_i, t_*)]P_{*-}[I - \sum_i \tilde{S}(t_i)K_i H(t_i)\Phi(t_i, t_*)]^T \quad (24)$$

$$\hat{P}_v(t_*^+) = \sum_i K_i \hat{R}(t_i) K_i^T, \quad P_v(t_*^+) = \sum_i \tilde{S}(t_i)K_i R(t_i)K_i^T \tilde{S}^T(t_i) \quad (25)$$

The contribution to the batch estimator’s formal variance from process noise is zero, since the batch filter does not model process noise; however, if there is really process noise present, then the true variance due to process noise is

$$P_w(t_*^+) = \Upsilon \begin{bmatrix} Q_d(t_*; t_1, t_1) & Q_d(t_*; t_1, t_2) & \cdots \\ Q_d(t_*; t_2, t_1) & Q_d(t_*; t_2, t_2) & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \Upsilon^T \quad (26)$$

where

$$\Upsilon = \begin{bmatrix} \tilde{S}(t_1)K_1 H(t_1)\Phi(t_1, t_*) & \tilde{S}(t_2)K_2 H(t_2)\Phi(t_2, t_*) & \cdots \end{bmatrix} \quad (27)$$

and

$$Q_d(t_*; t_i, t_j) = \begin{cases} \int_{t_*}^{\min(t_i, t_j)} \Phi(t_i, \tau)Q(\tau)\Phi^T(t_j, \tau)d\tau & t_* < t_i, t_* < t_j, \\ \int_{\max(t_i, t_j)}^{t_*} \Phi(t_i, \tau)Q(\tau)\Phi^T(t_j, \tau)d\tau & t_i < t_*, t_j < t_*, \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

which reduces to

$$Q_d(t_*, t_i, t_j) = \begin{cases} Q_d(t_i, t_*)\Phi^\top(t_j, t_i) & t_* < t_i \leq t_j, \\ \Phi(t_i, t_j)Q_d(t_j, t_*) & t_* \leq t_j \leq t_i, \\ \Phi(t_i, t_j)Q_d(t_*, t_j) & t_i \leq t_j < t_*, \\ Q_d(t_*, t_i)\Phi^\top(t_j, t_i) & t_j \leq t_i < t_*, \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

where we note that if $t_i = t_j > t_*$, then the definition above reduces to the usual definition of the process noise covariance, i.e. $Q_d(t_*, t_i, t_i) = Q_d(t_i, t_*)$. If $t_i = t_j < t_*$ then $Q_d(t_*, t_i, t_i) = Q_d(t_*, t_i)$, i.e. we have the usual process noise covariance, but it increases as time flows backwards. In the same fashion, the first and fourth cases in Eq. 29 produce the same numerical result, with the former corresponding to forward and the latter to reverse time flow. The same holds for the second and third cases in Eq. 29.

To propagate the estimated and true covariances, respectively, over a specified time interval (forward or backward in time), use the following:

$$\hat{P}_a(t) = \Phi_{ss}(t, t_*)\hat{P}_a(t_*^+)\Phi_{ss}^\top(t, t_*), \quad P_a(t) = \Phi(t, t_*)P_a(t_*^+)\Phi^\top(t, t_*) \quad (30)$$

$$\hat{P}_v(t) = \Phi_{ss}(t, t_*)\hat{P}_v(t_*^+)\Phi_{ss}^\top(t, t_*), \quad P_v(t) = \Phi(t, t_*)P_v(t_*^+)\Phi^\top(t, t_*) \quad (31)$$

$$P_w(t) = \Phi(t, t_*)P_w(t_*^+)\Phi^\top(t, t_*) + \Phi(t, t_*)N_d(t) + N_d^\top(t)\Phi^\top(t, t_*) + Q_d(t_*, t, t) \quad (32)$$

where

$$N_d(t) = -\sum_i \tilde{S}(t_i)K_i H(t_i)Q_d(t_*, t, t_i) \quad (33)$$

We note that when we allow for time reversal here, we are considering the *postdiction* problem; the fact that the process noise terms increase the uncertainty as time runs backwards reflects our increasing uncertainty as we propagate further into the past. One should not infer from this work that we take any position as to whether or not a physical Brownian process is diffusing backward in time.

Sensitivity Matrix Time and Measurement Update The sensitivity matrix shows the linear sensitivity of the solution at a specified time to mismodeling of the distributions of the *a priori* parameters (both solve-fors and considers). For the sequential filter, it is given by

$$\Sigma_a(t_i) = [I - \tilde{S}(t_i)K_i H(t_i)]\Phi(t_i, t_{i-1})\Sigma_a(t_{i-1}), \quad \Sigma_a(t_o) = [\tilde{S}(t_o), \tilde{C}(t_o)] \quad (34)$$

and for the batch estimator, it is

$$\Sigma_a(t) = S(t)\Phi(t, t_*)[I - \sum_{i=1}^k \tilde{S}(t_i)K_i H(t_i)\Phi(t_i, t_*)][\tilde{S}(t_*), \tilde{C}(t_*)] \quad (35)$$

Although it is possible to compute the sensitivities to each particular measurement and process noise sample[†], this does not appear to be particularly useful. Instead, suppose R and \hat{R} have the same structure, but differ only by a scalar multiplier, i.e., $R = r \cdot \hat{R}$ and $\hat{R} = \hat{r} \cdot \bar{R}$ (for example, let $\bar{R} = I$). Then, for the sequential filter,

$$\Delta P_v = SP_v S^\top - \hat{P}_v \quad (36)$$

will contain terms of the form $KK^\top(r - \hat{r})$. In this case, if one chooses $(r - \hat{r}) = 1$, then ΔP_v will represent the sensitivity to measurement noise mismodeling. Similarly, when $Q = q \cdot I$, and $\hat{Q} = \hat{q} \cdot I$, then $(q - \hat{q})$ will factor out of the process noise partition, so that if one chooses $q - \hat{q} = 1$, then ΔP_w will represent the sensitivity to process noise mismodeling. For the batch estimator,

$$\Delta P_v = K\bar{R}K^\top(r - \hat{r}). \quad (37)$$

In this case, if one chooses $(r - \hat{r}) = 1$, ΔP_v will represent the sensitivity to measurement noise mismodeling across the entire definitive data span. Similarly, when $Q = q \cdot \bar{Q}$, q will factor out of the (true) process noise partition, so that if one chooses $q = 1$, ΔP_w (which equals P_w since the batch has no process noise) will represent the sensitivity to process noise mismodeling across the entire interval of interest.

[†] It is particularly easy to show that for the batch estimator, the sensitivity to measurement and process noise inputs is

$$\Sigma_{vk}(t_*) = -K_k, \quad \Sigma_{wk}(t_*) = -K_k H(t_k)$$

INERTIAL NAVIGATION SYSTEM MODEL

Next, we apply the generalized method to a study of the effect of inertial navigation system (INS) parameters on an entry trajectory. This section describes the model of the INS. As part of this description, we address some issues related to attitude error modeling which appear to us to have never been adequately described in other works.

Our model includes additive bias and noise terms, and multiplicative errors. One of the multiplicative errors is associated with uncertainties in the scale factors used in the devices to convert their internal units into output units. The other error considered is that caused by small misalignments in the mounting of the gyros and accelerometers, so that they are not quite orthogonal to one another. This model shares many of the characteristics of the model used in Ref. 8, where Space Shuttle navigation performance with strapdown systems was evaluated, as well as with the model described in Ref. 9, a popular short course on inertial navigation systems. Similar models for platform-type inertial systems are described in Refs. 10 and 11. A more complicated model that shares some similarities is that of Ref. 12, in which a Kalman filter for calibration and alignment of inertial navigation systems is described.

Gyro Model

INS are generally of two types, “platform” and “strapdown,” which use the gyros in a somewhat different fashion. The platform type consists of a set of gimbals that interconnect the vehicle body to a rigid platform on which the gyros are mounted. This arrangement allows the axes along which the gyros are sensitive to rotate freely with respect to the vehicle body. A feedback system attempts to null the gyro outputs, which thus maintains the gyro platform approximately fixed in inertial space. The vehicle attitude with respect to inertial space is then given by reading out the gimbal angles. The strapdown type, which has largely replaced the platform type in current usage, dispenses with the gimbal system and rigidly fixes the gyros to the vehicle body. The strapdown INS computationally integrates the gyro outputs so as to keep track of the orientation of the body with respect to the inertial frame.

In each type of INS, errors in the gyro output affect the determination of the vehicle attitude, but in a somewhat different manner. In a platform INS, gyro output errors cause the platform to physically drift with respect to the true inertial frame. The strapdown INS measures at each instant an incremental change in the angular position of the body with respect to some inertial frame. Thus, errors in the gyro outputs create error in the computational accumulation of these angular velocity increments.

To illustrate the implications of this difference, we assume (without loss of generality) that the INS represents the attitude as a direction cosine matrix that maps from the INS case frame[‡] to the inertial frame, and we denote its estimate of the attitude as \widehat{M}_C^I . One may view this estimate as composed of either of the following successive rotations:

$$\widehat{M}_C^I = M_C^I M_C^c, \quad \text{or} \quad \widehat{M}_C^I = M_I^{\hat{x}} M_C^I \quad (38)$$

We assume that we can represent M_C^c and $M_I^{\hat{x}}$ as small angle rotations:

$$M_C^c = I - \theta^\times \quad \text{and} \quad M_I^{\hat{x}} = I - \psi^\times \quad (39)$$

where the superscript ‘ \times ’ indicates that the elements of the vector are arranged in a skew-symmetric matrix such that $\mathbf{a}^\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$. Note that θ and ψ are in general two different sets of small angle rotations – it is not necessarily the case that they are the same vector expressed in two different coordinate systems[§]. The (coordinate-independent) derivatives of these vectors with respect to the case and inertial frames are related by

$$\frac{\mathcal{I}_d}{dt} \theta = \frac{c_d}{dt} \theta + \mathcal{I} \omega^c \times \theta \quad \text{and} \quad \frac{\mathcal{I}_d}{dt} \psi = \frac{c_d}{dt} \psi + \mathcal{I} \omega^c \times \psi \quad (40)$$

[‡]For either type of INS, the case frame is rigidly fixed to the vehicle body, but may not be aligned with the body axes.

[§]However, if these rotations arise from two optimal attitude estimators processing the same data from the same initial conditions, the constraint that both estimators produce the optimal estimate implies that θ and ψ do become the same vector expressed in two different coordinate systems.

where we denote the angular velocity of frame \mathcal{C} in frame \mathcal{I} as ${}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}$. Similarly, the angular velocity estimate of the INS, $\widehat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}}$, will be corrupted by the gyro output error, and we may view this estimate as either

$$\widehat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}} = {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} + {}^{\mathcal{C}}\boldsymbol{\omega}^{\hat{\mathcal{C}}} \quad \text{or} \quad \widehat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}} = \hat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{I}}} + {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} \quad (41)$$

Thus, the coordinate-independent angular velocity error vector, $\delta {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} = \widehat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}} - {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}$, may be viewed as either

$$\delta {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} = {}^{\mathcal{C}}\boldsymbol{\omega}^{\hat{\mathcal{C}}} \quad \text{or} \quad \delta {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} = \hat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{I}}} \quad (42)$$

Recalling the description of how the platform INS uses the gyro output, it is clear that for the platform system, the gyro error corresponds to a drift of the platform with respect to inertial, and since the platform represents the INS's knowledge of the inertial frame, it is clear that the platform's drift corresponds to $\hat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{I}}}$. Thus, $\boldsymbol{\psi}$ must be related to $\hat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{I}}}$, according to

$$\frac{{}^{\mathcal{I}}d}{dt}\boldsymbol{\psi} = \hat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{I}}} \quad (43)$$

Recalling that the strapdown INS integrates the case-fixed gyro outputs, it is clear that a constant gyro drift from a strapdown gyro will appear to be constant to an observer who is also fixed in the case frame, and hence gyro drift from a strapdown INS corresponds to ${}^{\mathcal{C}}\boldsymbol{\omega}^{\hat{\mathcal{C}}}$, or

$$\frac{{}^{\mathcal{C}}d}{dt}\boldsymbol{\theta} = {}^{\mathcal{C}}\boldsymbol{\omega}^{\hat{\mathcal{C}}} \quad (44)$$

Thus, depending on whether the INS is of the platform type or the strapdown type, a constant gyro output bias will appear to be constant relative to the inertial frame, or constant relative to the case frame, respectively. These gyro biases will integrate into a misalignment $\boldsymbol{\psi}$ with respect to the inertial frame (platform INS), or into a *possibly different* misalignment $\boldsymbol{\theta}$ with respect to the case frame (strapdown INS). Note that one is free to write the equations of motion with respect to any frame one chooses, so that for example the following are completely permissible expressions:

$$\frac{{}^{\mathcal{I}}d}{dt}\boldsymbol{\theta} = {}^{\mathcal{C}}\boldsymbol{\omega}^{\hat{\mathcal{C}}} + {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} \times \boldsymbol{\theta} \quad \text{and} \quad \frac{{}^{\mathcal{C}}d}{dt}\boldsymbol{\psi} = \hat{{}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{I}}} - {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} \times \boldsymbol{\psi} \quad (45)$$

Henceforth, we only consider strapdown INS. For many such systems, the gyro error may be modeled as

$$\delta {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} = {}^{\mathcal{C}}\boldsymbol{\omega}^{\hat{\mathcal{C}}} = \mathbf{b}_{g\mathcal{C}} + \mathbf{S}_g {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} + \boldsymbol{\Gamma}_g {}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}} + \boldsymbol{\varepsilon}_\omega, \quad (46)$$

where $\mathbf{b}_{g\mathcal{C}}$ is an angular velocity in the case frame that biases the gyro, \mathbf{S}_g is the gyro scale factor matrix, $\boldsymbol{\Gamma}_g$ is the gyro nonorthogonality matrix, ${}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}$ is the representation of ${}^{\mathcal{I}}\boldsymbol{\omega}^{\mathcal{C}}$ in coordinates fixed to the INS case, $\boldsymbol{\varepsilon}_\omega$ is a Gaussian white noise that produces angular random walk, and where

$$\mathbf{S}_g = \begin{bmatrix} S_{gx} & 0 & 0 \\ 0 & S_{gy} & 0 \\ 0 & 0 & S_{gz} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Gamma}_g = \begin{bmatrix} 0 & 0 & 0 \\ -\gamma_{gyz} & 0 & 0 \\ \gamma_{gzy} & -\gamma_{gzx} & 0 \end{bmatrix}. \quad (47)$$

Here, the elements of the matrix \mathbf{S}_g represent scale factor errors, and the elements of the matrix $\boldsymbol{\Gamma}_g$ represent nonorthogonality errors. The latter arise from the fact that (for example) the y gyro will measure not only the angular velocity about \mathcal{C}_y as it is intended to, but also a small projection of the angular velocity about \mathcal{C}_x [¶].

[¶]Here we assume that as part of the INS calibration procedure, the x gyro is defined to be co-linear with the case x axis, the x - y case frame is defined to include the y gyro, and misalignments of the y and z gyros from the two axes orthogonal to this reference are measured. In practice, the case frame is defined by an alignment device such as an optical corner cube attached to the outside of the case, and the relationship between the frame defined externally by the optical cube and the internally-defined case frame is loaded into the INS's firmware.

Accelerometer Model

Since we have assumed a strapdown INS, the accelerometers measure the case coordinates of the “sensed acceleration”, \mathbf{a}_C , i.e. that part of the second time derivative of \mathbf{r} , the INS position vector, with respect to an inertial frame, which is due to non-gravitational, or “contact,” forces. If the INS is located at the spacecraft center of gravity (c.g.), \mathbf{r} represents the spacecraft position vector as well. If not, the accelerometer will measure not only the external acceleration on the spacecraft, but also the linear acceleration at the c.g. to INS moment arm due to any spacecraft rotational motion. This must be subtracted from the INS measured acceleration, typically by the navigation software. This model does not explicitly include the effect INS errors have on this c.g. offset calculation.

For many strapdown accelerometers, we may assume that the error may be modeled as

$$\delta \mathbf{a}_C = \mathbf{b}_{aC} + \mathbf{S}_a \mathbf{a}_C + \Gamma_a \mathbf{a}_C, \quad (48)$$

where \mathbf{b}_{aC} is the accelerometer bias, \mathbf{S}_a is the accelerometer scale factor matrix, Γ_a is the accelerometer nonorthogonality matrix, and where

$$\mathbf{S}_a = \begin{bmatrix} S_{ax} & 0 & 0 \\ 0 & S_{ay} & 0 \\ 0 & 0 & S_{az} \end{bmatrix}, \quad \Gamma_a = \begin{bmatrix} 0 & 0 & 0 \\ -\gamma_{ayz} & 0 & 0 \\ \gamma_{azy} & -\gamma_{azx} & 0 \end{bmatrix}. \quad (49)$$

Combined Model

It is convenient to recast Eqs. 46 and 48 as follows. Define $\mathbf{D}(\mathcal{I}\omega_C^C) = \text{diag}(\mathcal{I}\omega_C^C)$, and $\mathbf{s}_g = [S_{gx}, S_{gy}, S_{gz}]^T$. Then $\mathbf{S}_g \mathcal{I}\omega_C^C = \mathbf{D}(\mathcal{I}\omega_C^C) \mathbf{s}_g$. Similarly, define $\mathbf{D}(\mathbf{a}_C) = \text{diag}(\mathbf{a}_C)$, and $\mathbf{s}_a = [S_{ax}, S_{ay}, S_{az}]^T$. Then $\mathbf{S}_a \mathbf{a}_C = \mathbf{D}(\mathbf{a}_C) \mathbf{s}_a$. Define γ_g and $\mathbf{F}(\mathcal{I}\omega_C^C)$ such that $\Gamma_g \mathcal{I}\omega_C^C = \mathbf{F}(\mathcal{I}\omega_C^C) \gamma_g$, i.e. $\gamma_g = [\gamma_{gyz}, \gamma_{gzy}, \gamma_{gzx}]^T$, and

$$\mathbf{F}(\mathcal{I}\omega_C^C) = \begin{bmatrix} 0 & 0 & 0 \\ -\mathcal{I}\omega_{Cx}^C & 0 & 0 \\ 0 & \mathcal{I}\omega_{Cx}^C & -\mathcal{I}\omega_{Cy}^C \end{bmatrix}. \quad (50)$$

Similarly, define γ_a and $\mathbf{F}(\mathbf{a}_C)$ such that $\Gamma_a \mathbf{a}_C = \mathbf{F}(\mathbf{a}_C) \gamma_a$. Then, Eqs. 46 and 48 may be written as

$$\delta \mathcal{I}\omega_C^C = \mathbf{b}_{gC} + \mathbf{D}(\mathcal{I}\omega_C^C) \mathbf{s}_g + \mathbf{F}(\mathcal{I}\omega_C^C) \gamma_g + \epsilon_\omega \quad (51)$$

$$\delta \mathbf{a}_C = \mathbf{b}_{aC} + \mathbf{D}(\mathbf{a}_C) \mathbf{s}_a + \mathbf{F}(\mathbf{a}_C) \gamma_a \quad (52)$$

Now, let $\mathbf{x}_g = [\theta_C^T, \mathbf{b}_{gC}^T, \mathbf{s}_g^T, \gamma_g^T]^T$. Then, one can rewrite Eq. 46 as

$$\begin{pmatrix} \frac{c_d}{dt} \theta_C \\ \frac{c_d}{dt} \mathbf{b}_{gC} \\ \frac{d}{dt} \mathbf{s}_g \\ \frac{d}{dt} \gamma_g \end{pmatrix} = \begin{bmatrix} \mathbf{O}_{3 \times 3} & \mathbf{I}_3 & \mathbf{D}(\mathcal{I}\omega_C^C) & \mathbf{F}(\mathcal{I}\omega_C^C) \\ \mathbf{O}_{9 \times 12} & & & \end{bmatrix} \mathbf{x}_g + \begin{bmatrix} \mathbf{I}_3 \\ \mathbf{O}_{9 \times 3} \end{bmatrix} \epsilon_\omega \quad (53)$$

$$\dot{\mathbf{x}}_g = \mathbf{A}_g \mathbf{x}_g + \mathbf{B}_g \epsilon_\omega, \quad (54)$$

where we have chosen to write the equations of motion for the gyro misalignment and drift with respect to the case frame, and we recall that since the scale factor and nonorthogonality coefficients are only shown as column arrays for convenience, they are not properly viewed as physical vectors that take derivatives in any particular coordinate system. In similar fashion, let $\mathbf{x}_a = [\mathbf{b}_{aC}^T, \mathbf{s}_a^T, \gamma_a^T]^T$, with $\dot{\mathbf{x}}_a = \frac{c_d}{dt} \mathbf{x}_a = 0$, and cast the acceleration error $\delta \mathbf{a}_C$ as

$$\delta \mathbf{a}_C = \begin{bmatrix} \mathbf{I}_3 & \mathbf{D}(\mathbf{a}_C) & \mathbf{F}(\mathbf{a}_C) \end{bmatrix} \mathbf{x}_a \quad (55)$$

$$= \mathbf{H}_a \mathbf{x}_a. \quad (56)$$

We refer to the combination of the two models above as our Inertial Measurement Unit (IMU) model. When we integrate these results to produce position and velocity errors, we have a full INS model. When an INS incorporates sensed acceleration into its computation of the total acceleration on the vehicle, its gyro misalignment will corrupt the calculation:

$$\widehat{\frac{\mathcal{I}_d}{dt} \mathbf{v}_I} = -\frac{\mu}{\hat{r}^3} \hat{\mathbf{r}}_I + \widehat{\mathbf{M}_C^T \hat{\mathbf{a}}_C} \quad (57)$$

$$\approx -\frac{\mu}{r^3} \mathbf{r}_I + \mathbf{G}(\mathbf{r}_I) \delta \mathbf{r}_I + \mathbf{M}_C^T [\mathbf{I} - \boldsymbol{\theta}_C^\times] (\mathbf{a}_C + \delta \mathbf{a}_C) \quad (58)$$

where the gravity gradient matrix, $\mathbf{G}(\mathbf{r}_I)$, is the Jacobian of the two-body gravity term, $-\mu \mathbf{r}_I / r^3$. Thus, to first order, we may approximate the total acceleration error equations of motion as

$$\frac{\mathcal{I}_d}{dt} \delta \mathbf{v}_I = \mathbf{G}(\mathbf{r}_I) \delta \mathbf{r}_I + \mathbf{M}_C^T \mathbf{a}_C^\times \boldsymbol{\theta}_C + \mathbf{M}_C^T \delta \mathbf{a}_C \quad (59)$$

These models may be combined as follows:

$$\begin{pmatrix} \frac{\mathcal{I}_d}{dt} \delta \mathbf{r}_I \\ \frac{\mathcal{I}_d}{dt} \delta \mathbf{v}_I \\ \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_g \end{pmatrix} = \begin{bmatrix} \mathbf{O}_{3 \times 3} & \mathbf{I}_3 & \mathbf{O}_{3 \times 9} & \mathbf{O}_{3 \times 12} \\ \mathbf{G}(\mathbf{r}_I) & \mathbf{O}_{3 \times 3} & \mathbf{M}_C^T \mathbf{H}_a & \mathbf{M}_C^T \mathbf{X}_{ag} \\ \mathbf{O}_{12 \times 3} & \mathbf{O}_{12 \times 3} & \mathbf{O}_{9 \times 9} & \mathbf{O}_{12 \times 12} \\ \mathbf{O}_{12 \times 3} & \mathbf{O}_{12 \times 3} & \mathbf{O}_{9 \times 9} & \mathbf{A}_g \end{bmatrix} \begin{pmatrix} \delta \mathbf{r}_I \\ \delta \mathbf{v}_I \\ \mathbf{x}_a \\ \mathbf{x}_g \end{pmatrix} + \begin{bmatrix} \mathbf{O}_{15 \times 3} \\ \mathbf{B}_g \end{bmatrix} \boldsymbol{\varepsilon}_\omega \quad (60)$$

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \boldsymbol{\varepsilon}_\omega \quad (61)$$

where $\mathbf{X}_{ag} = [\mathbf{a}_C^\times, \mathbf{O}_{3 \times 9}]$ couples the gyro and accelerometer errors.

EXAMPLE

Next, we describe our example problem, then use the generalized covariance method to analyze the performance and sensitivities of the inertial navigation system.

Description of Example Problem

In our example, we assume that a global navigation satellite system type of receiver provides vector measurements of position throughout an entry trajectory, with measurement noise standard deviation three times less than the estimator assumes. Figs. 1 and 2 show some salient characteristics of the example. The simulation begins after the de-orbit burn, and one sees in the top plot of Fig. 1 that the vehicle reaches a peak deceleration of about $12 \times$ Earth's surface gravity about half-way through the entry. The vehicle executes 14 bank maneuvers, which are a combination of rolls and yaws, as the top and bottom plots of Fig. 2 show, so as to control its landing point. The middle and bottom plots of Fig. 1 show accelerations from side forces created by the vehicle's lift during these bank maneuvers.

A Monte Carlo simulation produced the gyro and accelerometer error data that Figs. 1 and 2 also show. This simulation assumed that the elements of $\boldsymbol{\theta}_C(t_o)$, \mathbf{b}_{gC} , \mathbf{s}_g , $\boldsymbol{\gamma}_g$, \mathbf{b}_{aC} , \mathbf{s}_a , $\boldsymbol{\gamma}_a$ and $\boldsymbol{\varepsilon}_\omega$ are zero mean, timewise uncorrelated, Gaussian-distributed random processes, and are uncorrelated with one another. Table 1 lists the relevant simulation parameters that we used.

Navigation Analysis of Sample Problem

In using any linear covariance analysis method, we always recommend at least one actual simulation as a check against linearization problems. In the present work, we generate random deviations from the reference as initial conditions for each of 25 Monte Carlo cases. We integrate each deviated case, and use this as truth for measurement simulation and estimation error generation. From these, we generate the time series of estimation errors and residuals for each case.

Table 1 Simulation Parameters

Parameter	Standard Deviation	Units
Gravitational Constant	4.305×10^4	km^3/sec^2
Measurement Time Interval	2	sec
True Position Measurement Noise	0.305	m
Formal Position Measurement Noise	0.914	m
Initial Position Error	30.5	meters
Initial Velocity Error	3.05	cm/sec
Accelerometer Bias	60	μg
Accelerometer Scale Factor	500	ppm
Accelerometer Nonorthogonality	10	ppm
Initial Gyro Angular Error	42	arcsec
Gyro Bias Drift	0.01	deg/hr
Gyro Scale Factor	33	ppm
Gyro Nonorthogonality	20	ppm
Gyro Random Walk Intensity	0.025	$\text{deg/hr}^{1/2}$

Description of Plots We will be discussing a number of plots below, and it will be helpful to describe our plot conventions in some detail before delving into the results themselves.

A typical output of Monte Carlo simulations is the ensemble of the time histories of the true estimation errors, along with their formal errors, which are the standard deviations from the diagonals of the estimator's covariance matrix. Fig. 4 is an example of such a result. In this and similar figures to follow, we plot the true error time series as a sheaf of yellow lines, and the ensemble mean of the true errors as a single blue line. We plot the empirical ensemble deviation of the true errors as a thick cyan band that covers ± 1 to $\pm 3\sigma$'s. We show twice the true standard deviation from the linear covariance analysis as a series of black + marks. We show twice the ensemble mean of the formal standard deviations from the Monte Carlo analysis as a series of green + marks. We compute a 95% confidence interval based on a hypothesis test of the variance of the Monte Carlo results against the true variance from the linear covariance analysis, which we show as a translucent red band bounded by red + marks. These sorts of results are available from existing linear covariance and Monte Carlo simulation procedures.

We have chosen to illustrate the capability of the generalized covariance method to partition the covariance using what we call "variance sandpiles," which are stacked area charts showing the time series of each solve-for variance's contribution from *a priori* error variance, measurement noise variance, and process noise variance. Fig. 7 is an example of such a result. Here, we plot the components of the true variance as a positive sandpile, and the components of the formal variance as an inverted sandpile on the negative ordinate, and relabel the negative ordinate to indicate this^{||}.

A useful result of many types of covariance analysis is an error budget showing the sensitivity of each solve-for parameter to errors in the consider parameters. We show such results as "sensitivity mosaics," which are checkerboard plots of the log magnitude of the elements of the sensitivity matrices. Fig. 8 is an example of such a plot.

Results We performed a number of studies of our example problem using the generalized method, which we briefly summarize before discussing our final results. First, we solved for all of the states in the INS model to study the observability of the various states. From this work, we concluded that accelerometer bias was weakly observable and accelerometer scale factor was strongly observable along the two directions generally

^{||} Depending on the relative magnitudes of these results, we could also illustrate the sandpiles in other ways, e.g. if all the delta variances were positive, the sandpile could show the formal variance, and the deltas due to each component. If all the deltas were negative, the sandpile could show the true variance, and negatives of the deltas.

normal to the drag acceleration. Accelerometer nonorthogonality and all of the gyro rate errors were not observable. Gyro misalignment was strongly observable, along with position and velocity. Based on these results, we next examined various combinations of observable states as solve-for parameters. Among these combinations, we studied (1) solving for position, velocity, accelerometer bias and scale factor, and gyro misalignment; (2) solving for only position and velocity; and (3) solving for linear combinations of the observable accelerometer and gyro states. The latter was particularly easy with the generalized formulation, since we needed only to modify the S and C matrices. From this work, we concluded that the best performance was available from case (1). Figs. 3 – 8 show the results from this case we obtained with the sequential form of the method. Finally, we studied this case using the batch estimator. Although this produced comparable results, we found that the large size of the Υ matrix, by which the batch form captures the effect of process noise, caused the batch analysis to take considerably longer to run.

Discussion There are several notable features of Fig. 3, which shows the Monte Carlo and linear covariance results for the position states**. The agreement between the true standard deviation from the linear covariance analysis (black + marks) and the ensemble mean of the formal standard deviations from the Monte Carlo analysis (green + marks) indicates that nonlinearities are not significantly affecting the results. These are bounded by the Monte Carlo confidence interval (red band), indicating that the null hypothesis, which is that the Monte Carlo cases actually are governed by the moments of the linear covariance analysis, cannot be rejected at the 95% confidence level. The large sawtooth increases in the errors between measurement updates lead us to suspect that process noise, which only comes from gyro random walk in this example, is a strong factor influencing performance.

The position variance sandpiles (left column of Fig. 6) allow us to confirm that the spikes are due to process noise. They also show that measurement noise is a significant contributor to the formal variance, but not the true variance (recall that the simulation was run with three times less measurement noise on the true measurements than the filter assumed). Referring to the sensitivity mosaic in Fig. 8, we can see that the IMU parameters, especially gyro drift, are significant contributors to position and velocity errors.

Considering now the accelerometer and gyro errors in Figs. 4 – 5, we see that the z -component of the accelerometer scale factor error somewhat exceeds the 95% confidence limit during the initial convergence of the filter, possibly indicating stronger nonlinear effects not captured by the linear covariance analysis (Fig. 4). Once the z -component of the gyro angular error converges after $t = 40$ (Fig. 5), the confidence interval violation for the accelerometer scale vector stops, suggesting that the cross-product between gyro angular error and sensed acceleration is playing a role. The sandpiles for the weakly observable accelerometer parameter variances (Fig. 7) are dominated by *a priori* error, while the strongly observable gyro angular error variance sandpile (right column of Fig. 6) is dominated by process and measurement noise, as with the other strongly observable states.

SUMMARY AND CONCLUSION

We have updated and extended previous work on linear covariance analysis and used our generalized method to study the navigation performance of an inertial navigation system during an entry trajectory. As part of this work, we clarified some issues concerning the equations of motion for modeling attitude errors.

We showed that sensing drag acceleration during the entry, along with position measurements from a GPS-like device, allows the INS to update its attitude error states. We found that the linear covariance analysis results agreed well with a 25-case Monte Carlo simulation, indicating the lack of any strongly nonlinear effects in our example problem. Using the unique capability of the generalized method to partition the errors, we were able to *quantify* the extent to which process and measurement noise dominate the performance in estimating the strongly observable states, and to quantify the extent to which *a priori* error strongly affects the performance in estimating the weakly observable states. We illustrated the sensitivity of the INS to various unobservable parameters, and showed that position and velocity errors are strongly sensitive to gyro bias.

**Velocity results are generally similar to position, so we omit them for brevity.

ACKNOWLEDGMENTS

Lee Bryant of the Johnson Space Center produced the reference data for the entry simulation. Chunlei Rui of Northrop Grumman initiated the discussion of attitude error modeling which caused us to clarify our thinking on this matter.

REFERENCES

- [1] F. L. Markley, E. Seidewitz, and M. Nicholson, "A General Model for Attitude Determination Error Analysis," *NASA Conference Publication 3011: Flight Mechanics/Estimation Theory Symposium*, May 1988, pp. 3–25.
- [2] F. L. Markley, E. Seidewitz, and J. Deutschmann, "Attitude Determination Error Analysis: General Model and Specific Application," *Proceedings of the CNES Space Dynamics Conference*, Toulouse, France, November 1989, pp. 251–266.
- [3] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, Vol. 1. New York: Academic Press, 1979.
- [4] A. Gelb, ed., *Applied Optimal Estimation*. The M.I.T. Press, 1974.
- [5] B. D. Tapley, B. E. Schutz, and G. R. Born, *Statistical Orbit Determination*. Academic Press, 2004.
- [6] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [7] C.-T. Chen, *Linear System Theory and Design*. Orlando, FL: Hold, Rinehart and Winston, Inc., 1984.
- [8] D. Riegsecker, "On-Orbit Attitude Errors," Memorandum E43-96-105, C. S. Draper Laboratory, July 25, 1996.
- [9] S. C. Bose, "Integrated Navigation Systems (GPS/INS)," Short course lecture notes.
- [10] "Mathematical Models of Attitude Hardware," *Spacecraft Attitude Determination and Control* (J. R. Wertz, ed.), Boston: Kluwer Academic Publishers, 1978.
- [11] "Inertial Measurement Unit," *Onboard Navigation System Characteristics, JSC-26289*, NASA Johnson Space Center, June 1993.
- [12] M. Grewal, V. Henderson, and R. Miyasako, "Application of Kalman Filtering to the Calibration and Alignment of Inertial Navigation Systems," *IEEE Transactions on Automatic Control*, Vol. 36, Jan 1991, pp. 4–12.

IMU Model Acceleration Errors

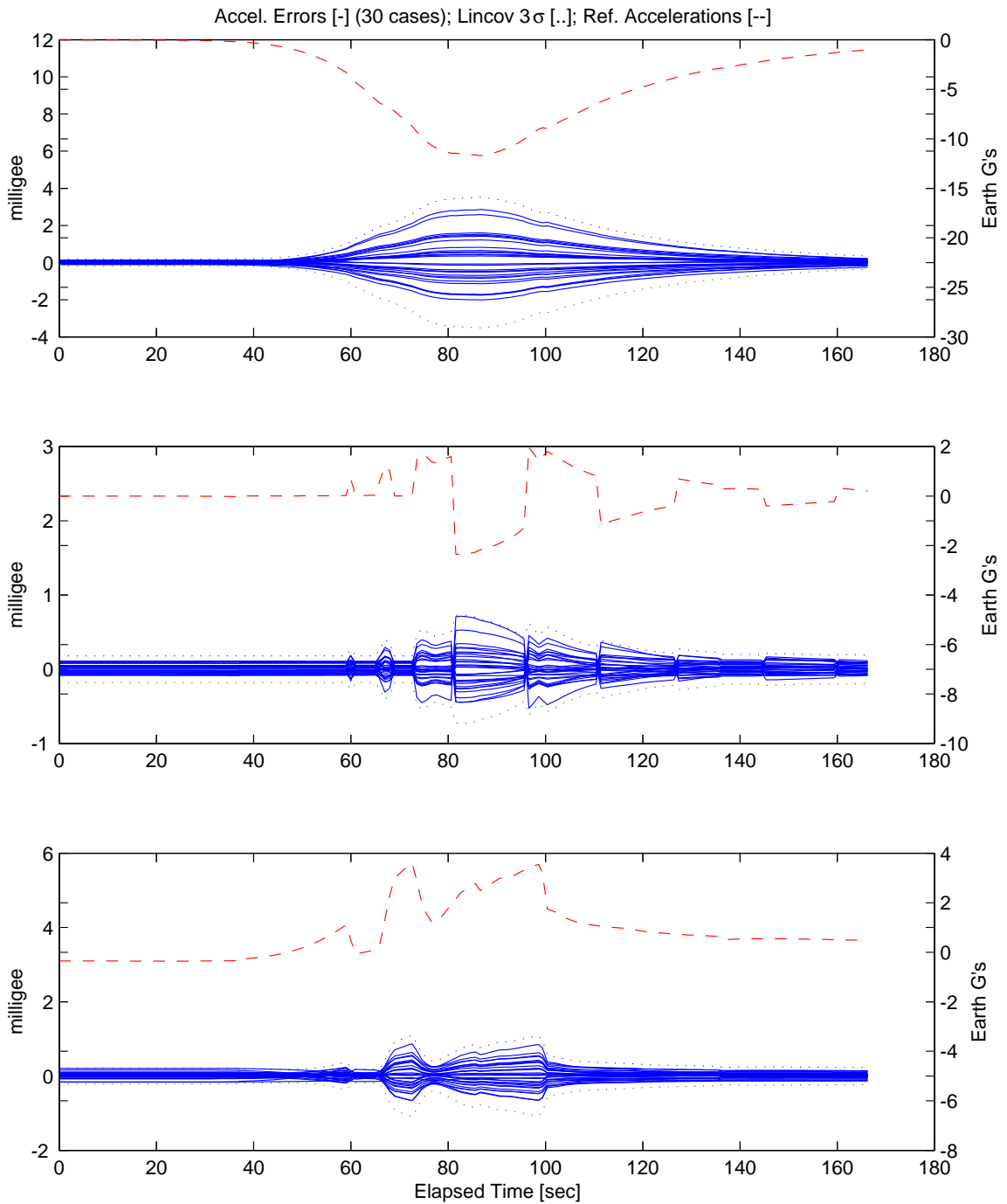


Figure 1 Monte carlo simulation of case coordinates of accelerometer errors (left scale, solid blue lines) and their formal 3σ values (left scale, dotted blue lines), shown with reference accelerations (right scale, dashed red lines); the x , y , and z coordinates are shown in the top, middle, and bottom subplots, respectively.

IMU Model Gyro Rate Errors

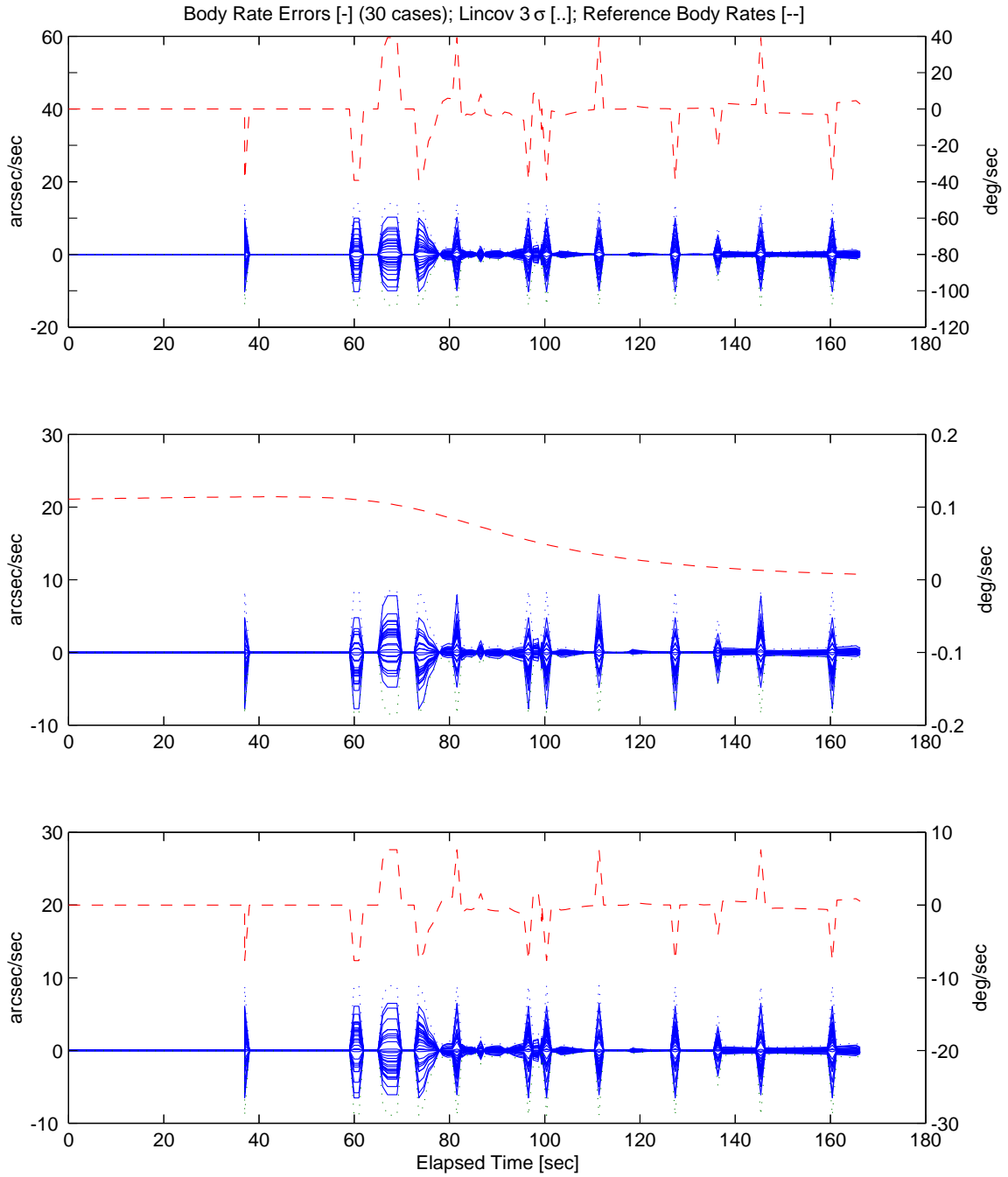


Figure 2 Monte carlo simulation of case coordinates of gyro rate errors (left scale, solid blue lines) and their formal 3σ values (left scale, dotted blue lines), shown with reference body rates (right scale, dashed red lines); the x , y , and z coordinates are shown in the top, middle, and bottom subplots, respectively.

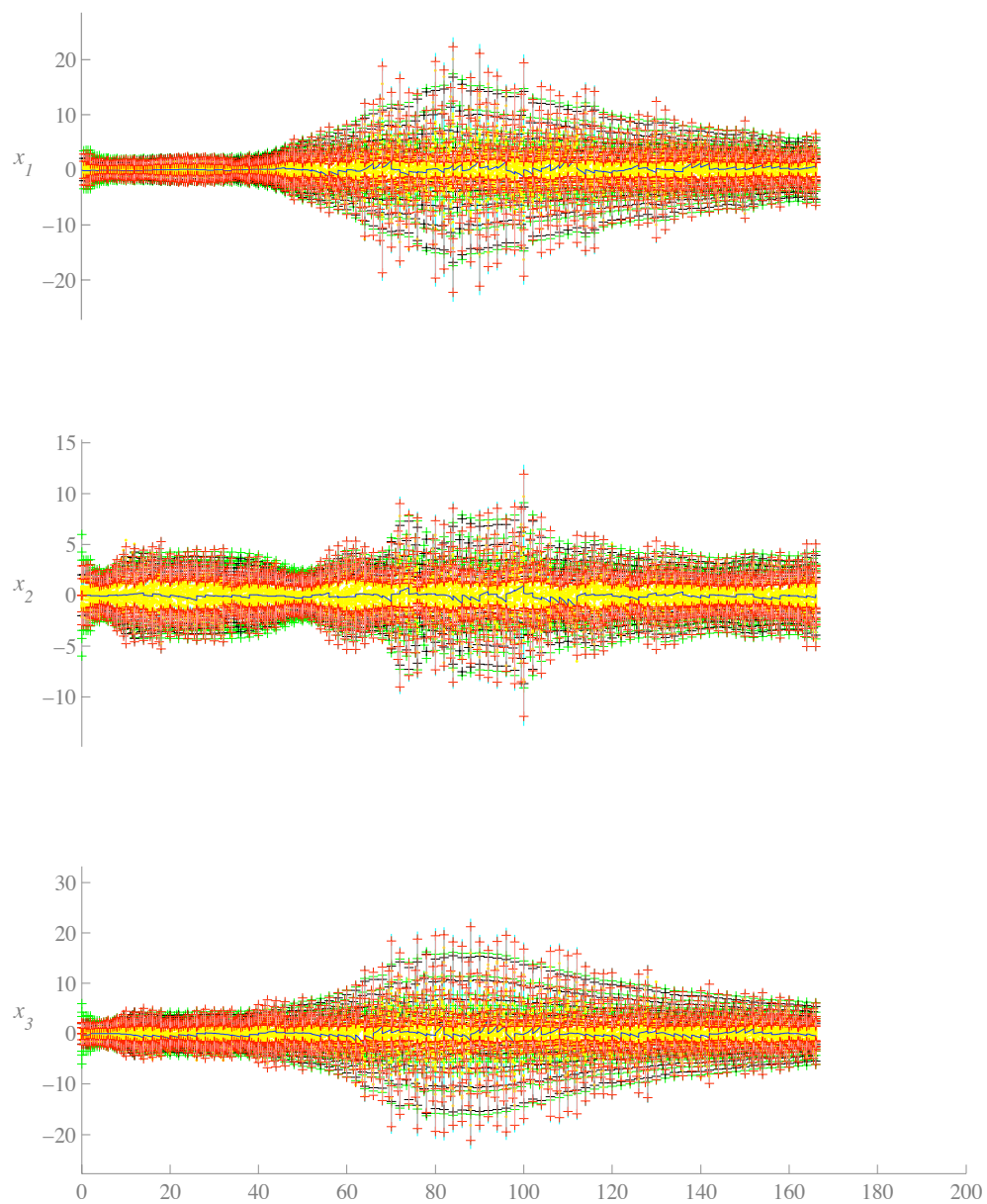


Figure 3 Position Errors from 25-case Monte-Carlo Simulation.

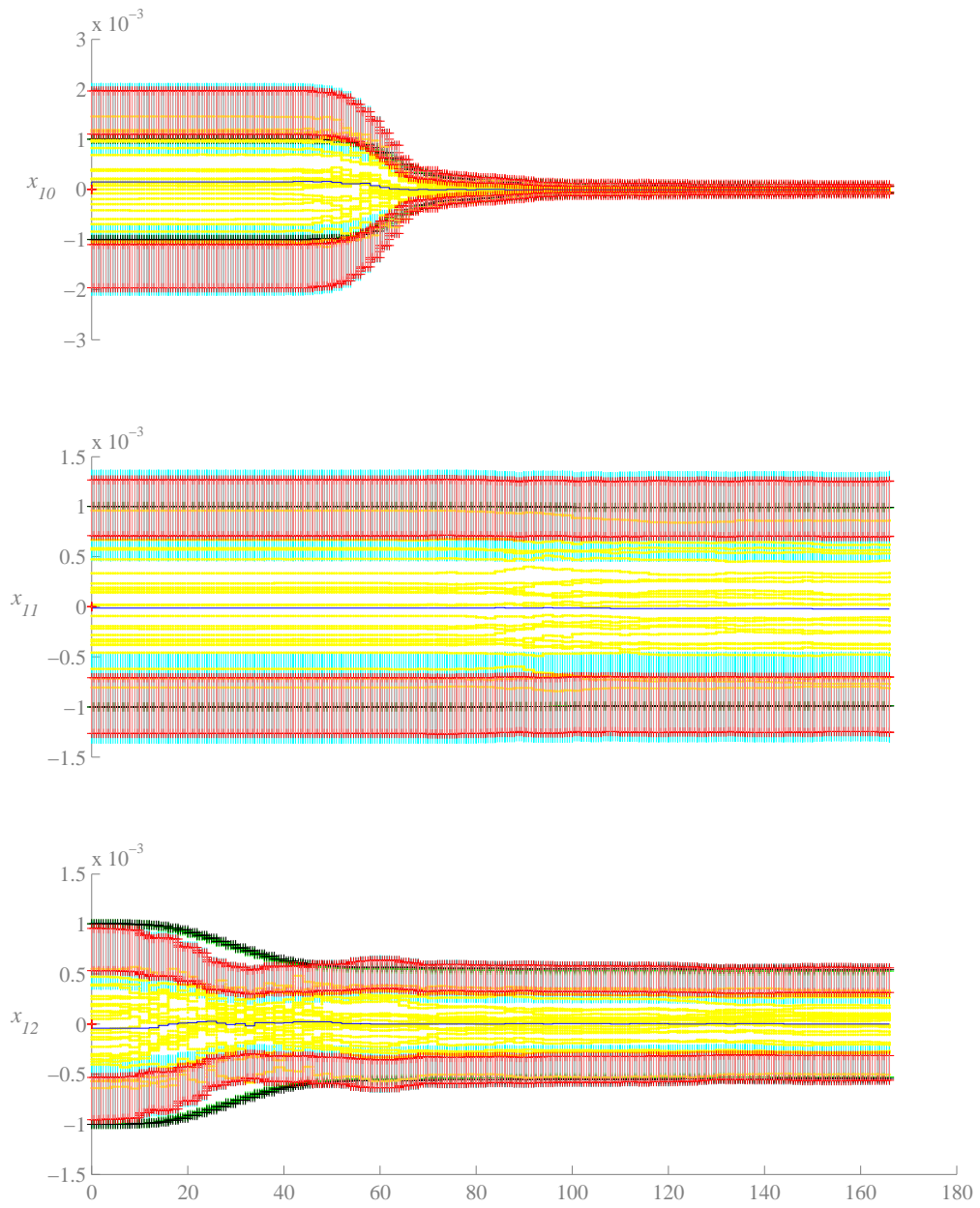


Figure 4 Accelerometer Scale Factor Errors from 25-case Monte-Carlo Simulation.

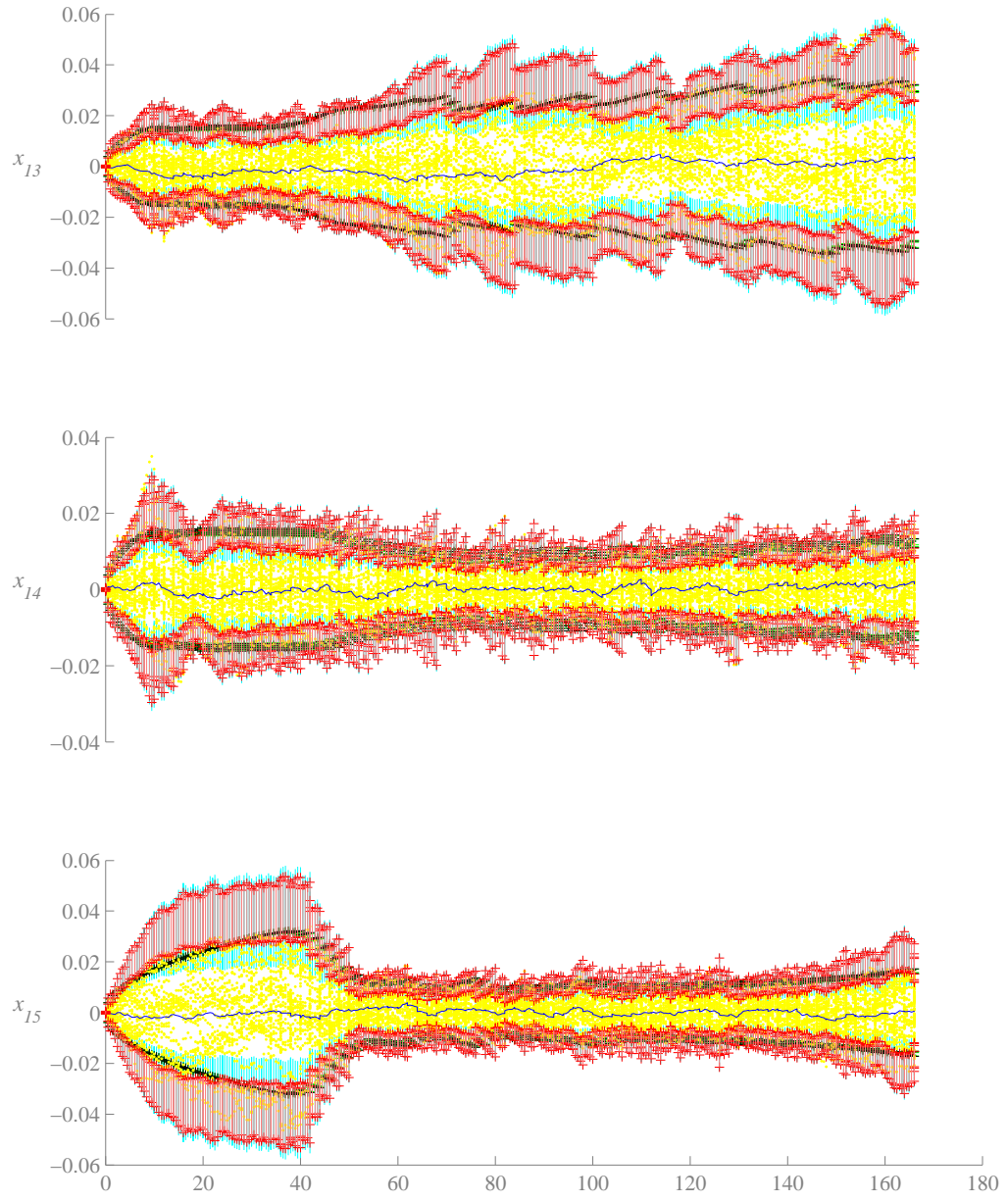
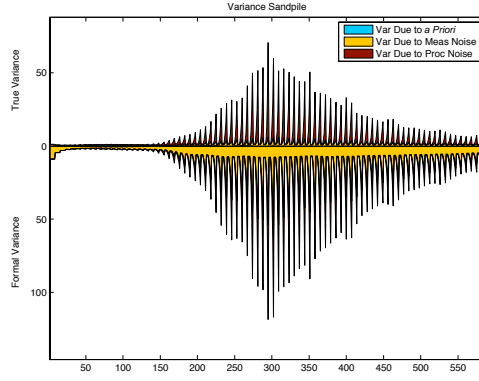
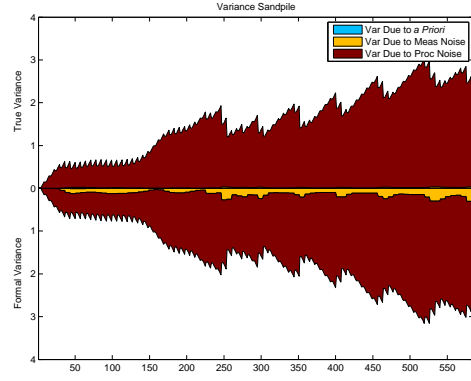


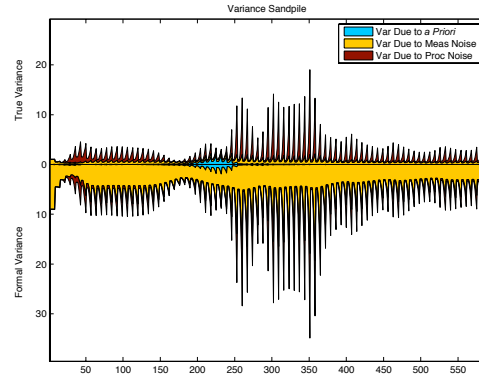
Figure 5 Gyro Angular Errors from 25-case Monte-Carlo Simulation.



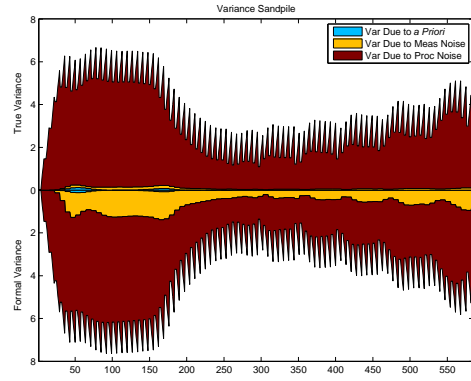
(a) *X*-component of Inertial Position Error



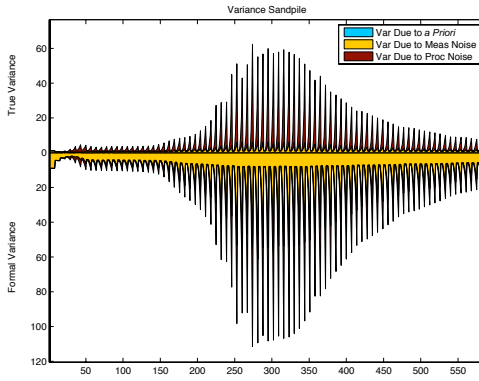
(b) *X*-component of Case-Fixed Gyro Angular Error



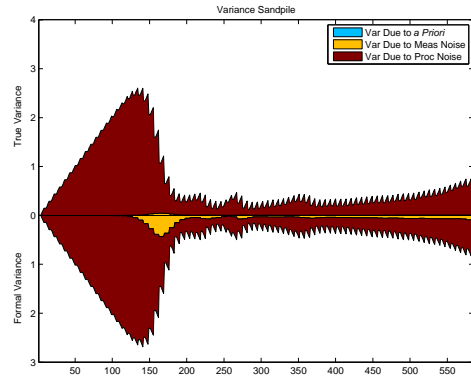
(c) *Y*-component of Inertial Position Error



(d) *Y*-component of Case-Fixed Gyro Angular Error

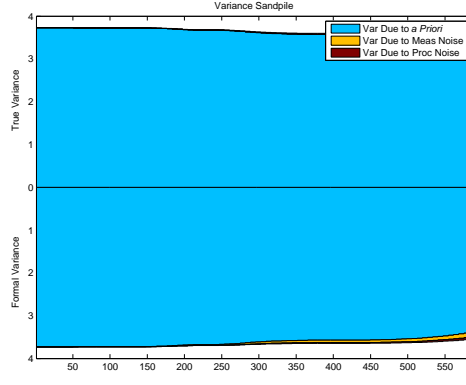


(e) *Z*-component of Inertial Position Error

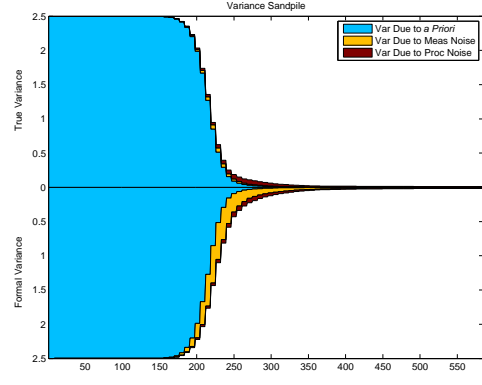


(f) *Z*-component of Case-Fixed Gyro Angular Error

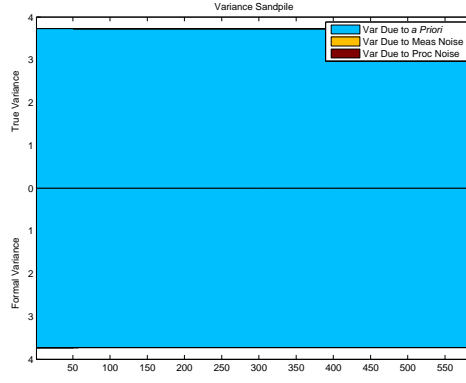
Figure 6 Position Error (a,c,e) and Gyro Angular Error (b,d,f) Variance Sandpiles (plotted vs. sample number).



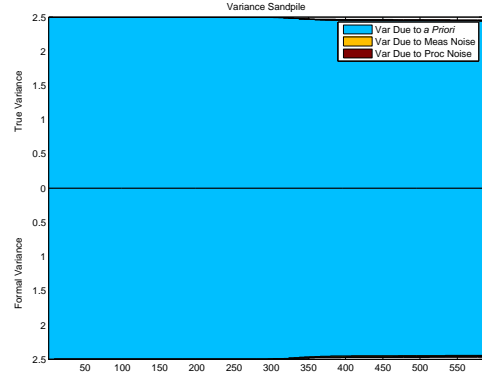
(a) X -component of Case-Fixed Accelerometer Bias



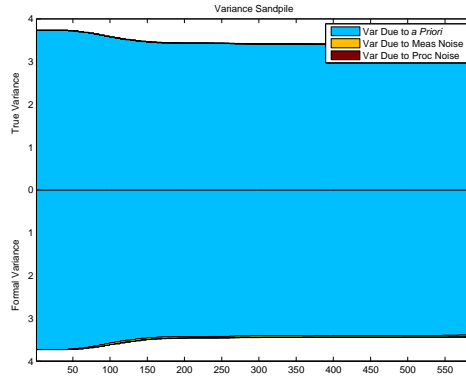
(b) X -component of Case-Fixed Accelerometer Scale Factor



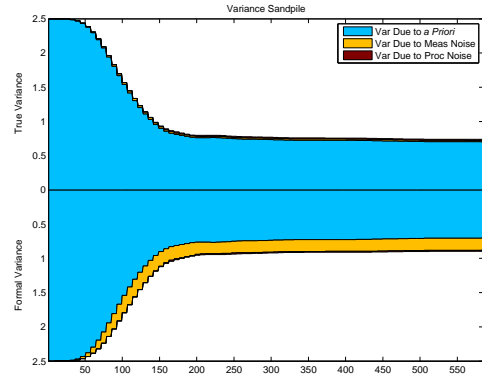
(c) Y -component of Case-Fixed Accelerometer Bias



(d) Y -component of Case-Fixed Accelerometer Scale Factor



(e) Z -component of Case-Fixed Accelerometer Bias



(f) Z -component of Case-Fixed Accelerometer Scale Factor

Figure 7 Accelerometer Bias (a,c,e) and Scale Factor (b,d,f) Variance Sandpiles (plotted vs. sample number).

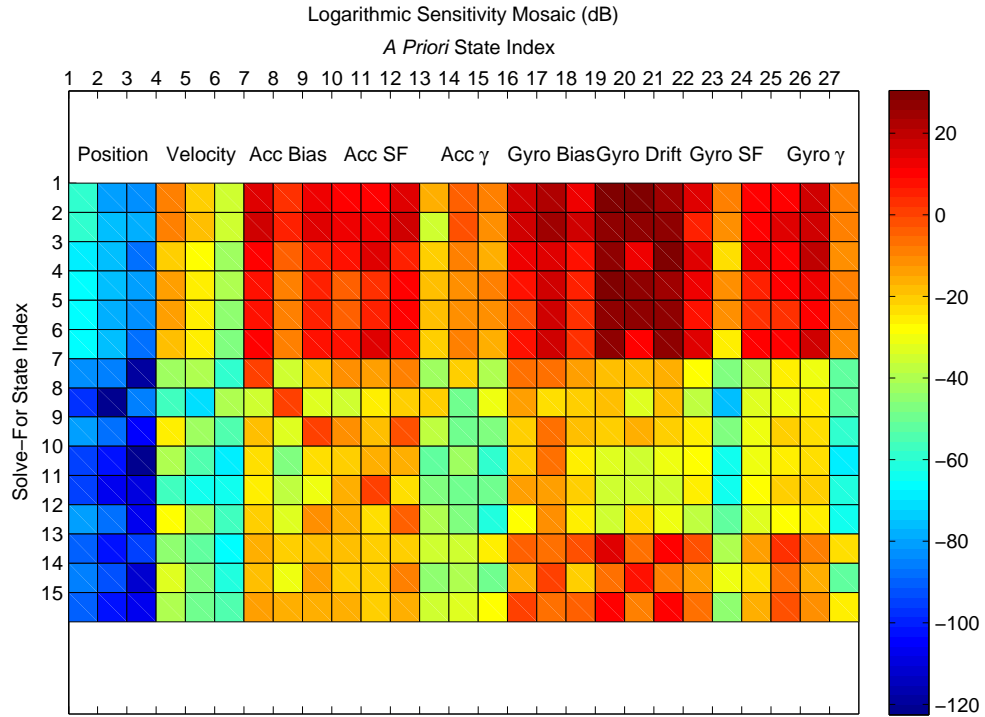


Figure 8 Sensitivity of solve-for errors to variations in *a priori* parameter specifications. The solve-for indices are as follows: 1–3, position; 4–6, velocity; 7–9, accelerometer bias; 10–12, accelerometer scale factor; 13–15, gyro angular error.