IN-SPACE SPACECRAFT ALIGNMENT CALIBRATION USING THE UNSCENTED FILTER

Kok-Lam Lai,* John L. Crassidis†
Department of Aerospace & Mechanical Engineering
University at Buffalo, The State University of New York
Amherst NY 14260-4400

Richard R. Harman‡
Guidance, Navigation & Control Systems Engineering Branch
NASA-Goddard Space Flight Center
Greenbelt, MD 20771

ABSTRACT

A new spacecraft sensor alignment estimation approach based on the Unscented filter is derived. Basic six-state attitude estimation is widely implemented in actual spacecraft missions. However, more stringent spacecraft pointing accuracy requires sensor alignment calibration to be performed post-launch to accommodate in-space disturbances and launch shock vibration. Sequential filtering is preferred since the calibration parameters could drift over time. It also minimizes ground crew intervention and mission disruption. Simulated spacecraft data results shows that the Unscented filter is more robust and more suitable for on-board implementation than the traditional extended Kalman filter. Also, experimental results from the Microwave Anisotropy Probe using the Unscented filter are shown to test the performance of the algorithm using real data.

INTRODUCTION

Precise attitude estimation is crucial to most spacecraft missions today. The traditional six-state extended Kalman filter (EKF) estimates the current attitude and gyro biases simultaneously. This filter assumes unvarying alignment of the sensors involved in attitude estimation. However, sensor misalignment is inevitable and would contribute to unreliable attitude estimates.[1] More stringent attitude pointing accuracy requires misalignments to be estimated and implemented into the attitude estimator. It has been noted in some recent papers[2–4] of the importance of proper calibration for use in fault detection or rate derivation. Without in-space sequential calibration, the sensors have to be re-calibrated on the ground from time to time to improve residual characteristics.[3]

Misalignment calibration is usually performed prior to launch. However, launch shock often makes this pre-launch calibration somewhat irrelevant and requires post-launch alignment calibration before nominal mission mode. Thus in-flight alignment calibration is needed to accommodate these unanticipated changes in alignment. The ill effects of attitude estimation using misaligned sensors are shown in Ref. [5]. On-board real-time or sequential filtering is preferred since misalignment parameters could drift, either due to in-space disturbances (for example solar wind, aerodynamics, persistence thermal shock) or unattended changes in the attitude sensor's relative orientation, especially those with moving mechanisms. A sequential filter constantly outputs the best estimate of the calibration parameters and can tolerate parameter drifts in real-time with minimal ground crew intervention or mission disruption. In this paper, we propose a new real-time misalignment estimator using the Unscented Filter (UF).[6]

This in-space calibration scheme would enhance the degree of autonomy of the Attitude Determination and Control System to reduce ground crew intervention for an extended period of time. Specifically, autonomy refers to a mode of ground system operation in which manual human actions are not required to accomplish desired functions. Autonomy refers to self-acting, self-regulating systems on the spacecraft wherein functions are delegated to the spacecraft systems. Because of operational necessity, budget constraints, and technology push a number of automation initiatives are underway at NASA-Goddard Space Flight Center.[8]

The organization of this paper proceeds as follows. First, the Unscented filtering and attitude kinematics are briefly reviewed. Sensor models with misalignments are then derived. These include a three-axis orthogonal gyros, vector output star tracker and vector output payload. The gyro model also includes asymmetric and symmetric scale factor errors and bias drifts. Then, simulated spacecraft results and exper-

*Graduate Student, Student Member AIAA, klai2@eng.buffalo.edu
†Associate Professor, Associate Fellow AIAA, johnc@eng.buffalo.edu
‡Research Engineer, richard.r.harman.1@gsfc.nasa.gov

imental data results from the Microwave Anisotropy Probe (MAP) are presented.

UNSCENTED FILTERING

In this section the UF is reviewed. For more information, please refer to Ref. [9]. The filter presented in Ref. [10] is derived for discrete-time nonlinear equations, where the system model is given by

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + G_k \mathbf{w}_k \tag{1a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \tag{1b}$$

where $\mathbf{x}_k$ is the $n \times 1$ state vector and $\tilde{\mathbf{y}}_k$ is the $m \times 1$ measurement vector. Note that a continuous-time model can always be expressed in the form of Eq. (1a) through an appropriate numerical integration scheme. We assume that the process noise $\mathbf{w}_k$ and measurement-error noise $\mathbf{v}_k$ are zero-mean Gaussian noise processes with covariances given by Q_k and R_k, respectively. The standard Kalman Filter update equations are first rewritten as[11]

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k \boldsymbol{\upsilon}_k \tag{2a}$$

$$P_k^+ = P_k^- - K_k P_k^{\upsilon\upsilon} K_k^T \tag{2b}$$

where $\hat{\mathbf{x}}_k^-$ and P_k^- are the pre-update state estimate and covariance, respectively, and $\hat{\mathbf{x}}_k^+$ and P_k^+ are the post-update state estimate and covariance, respectively. The *innovation* $\boldsymbol{\upsilon}_k$ is given by

$$\boldsymbol{\upsilon}_k \equiv \tilde{\mathbf{y}}_k - \hat{\mathbf{y}}_k^- = \tilde{\mathbf{y}}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, k) \tag{3}$$

The covariance of $\boldsymbol{\upsilon}_k$ is denoted by $P_k^{\upsilon\upsilon}$. The gain K_k is computed by

$$K_k = P_k^{xy} (P_k^{\upsilon\upsilon})^{-1} \tag{4}$$

where P_k^{xy} is the cross-correlation matrix between $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{y}}_k^-$.

The UF uses a different propagation than the standard EKF. Given an $n \times n$ covariance matrix P, a set of $2n$ *sigma points* can be generated from the columns of the matrices $\pm\sqrt{(n+\lambda)P}$, where $\sqrt{M}$ is shorthand notation for a matrix Z such that $Z\,Z^T = M$. The set of points is zero-mean, but if the distribution has mean $\boldsymbol{\mu}$, then simply adding $\boldsymbol{\mu}$ to each of the points yields a symmetric set of $2n$ points having the desired mean and covariance.[10] Due to the symmetric nature of this set, its odd central moments are zero, so its first three moments are the same as the original Gaussian distribution. The scalar λ is a convenient parameter for exploiting knowledge (if available) about the higher moments of the given distribution.[11] In scalar systems (i.e., for $n = 1$), a value of $\lambda = 2$ leads to errors in the mean and variance that are sixth order. For higher-dimensional systems choosing $\lambda = 3 - n$ minimizes the mean-squared-error up to the fourth order.[10] However, caution should be exercised when λ is negative since a possibility exists that the predicted covariance can become non-positive semi-definite. If this is a major concern, then another approach can be used that allows for scaling of the sigma points, which guarantees a positive semi-definite covariance matrix. Also, it can be shown that when $n+\lambda$ tends to zero the mean tends to that calculated by the truncated second-order filter.[12] This is the foundation for the UF.

A method for incorporating process noise in the UF is shown in Ref. [13]. This approach generates a set of points in $[\mathbf{x}_k,\, \mathbf{w}_k]$ space that has the correct mean and covariance, and propagates these points through the model in Eq. (1a). The predicted mean and covariance are also augmented to included the process noise, but the basic structure of the their calculations remain unchanged (see Ref. [13] for more details). Although this approach may more fully utilize the capability of the unscented transformation, it will be more computationally costly due to the extra required calculations arising from the augmented system. For the basic six-state attitude estimation problem, a set of six more sigma points is required to implement this approach. This significantly increases the computational burden, which may prohibit its use for actual onboard implementations.

The general formulation for the propagation equations are given as follows. First, compute the following set of sigma points:

$$\boldsymbol{\sigma}_k \leftarrow 2n \text{ columns from } \pm\sqrt{(n+\lambda)[P_k^+ + \bar{Q}_k]} \tag{5a}$$

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+ \tag{5b}$$

$$\boldsymbol{\chi}_k(i) = \boldsymbol{\sigma}_k(i) + \hat{\mathbf{x}}_k^+ \tag{5c}$$

where the matrix $\bar{Q}_k$ is related to the process noise covariance, which will be discussed shortly. One efficient method to compute the matrix square root is the Cholesky decomposition. Alternatively, the sigma points can be chosen to lie along the eigenvectors of the covariance matrix. Note that there are a total of $2n$ values for $\boldsymbol{\sigma}_k$ (the positive and negative square roots). The transformed set of sigma points are evaluated for each of the points by

$$\boldsymbol{\chi}_{k+1}(i) = \mathbf{f}[\boldsymbol{\chi}_k(i), k] \tag{6}$$

The predicted mean is given by

$$\hat{\mathbf{x}}_{k+1}^- = \frac{1}{n+\lambda} \left\{ \lambda\, \boldsymbol{\chi}_{k+1}(0) + \frac{1}{2} \sum_{i=1}^{2n} \boldsymbol{\chi}_{k+1}(i) \right\} \tag{7}$$

The predicted covariance is given by

$$P_{k+1}^- = \frac{1}{n+\lambda} \left\{ \lambda\, [\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-]\, [\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^-]^T + \frac{1}{2} \sum_{i=1}^{2n} [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]\, [\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^-]^T \right\} + \bar{Q}_k \tag{8}$$

The mean observation is given by

$$\hat{\mathbf{y}}_{k+1}^{-} = \frac{1}{n+\lambda}\left\{\lambda\,\boldsymbol{\gamma}_{k+1}(0) + \frac{1}{2}\sum_{i=1}^{2n}\boldsymbol{\gamma}_{k+1}(i)\right\} \tag{9}$$

where

$$\boldsymbol{\gamma}_{k+1}(i) = \mathbf{h}[\boldsymbol{\chi}_{k+1}(i), k] \tag{10}$$

The output covariance is given by

$$P_{k+1}^{yy} = \frac{1}{n+\lambda}\left\{\lambda\left[\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^{-}\right]\left[\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^{-}\right]^T + \frac{1}{2}\sum_{i=1}^{2n}\left[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}\right]\left[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}\right]^T\right\} \tag{11}$$

Then, the innovation covariance is simply given by

$$P_{k+1}^{vv} = P_{k+1}^{yy} + R_{k+1} \tag{12}$$

Finally, the cross-correlation matrix is determined using

$$P_{k+1}^{xy} = \frac{1}{n+\lambda}\left\{\lambda\left[\boldsymbol{\chi}_{k+1}(0) - \hat{\mathbf{x}}_{k+1}^{-}\right]\left[\boldsymbol{\gamma}_{k+1}(0) - \hat{\mathbf{y}}_{k+1}^{-}\right]^T + \frac{1}{2}\sum_{i=1}^{2n}\left[\boldsymbol{\chi}_{k+1}(i) - \hat{\mathbf{x}}_{k+1}^{-}\right]\left[\boldsymbol{\gamma}_{k+1}(i) - \hat{\mathbf{y}}_{k+1}^{-}\right]^T\right\} \tag{13}$$

The filter gain is then computed using Eq. (4), and the state vector can now be updated using Eq. (2). Even though $2n+1$ propagations are required for the UF, the computations may be comparable to the EKF, especially if the continuous-time covariance equations need to be integrated and a numerical Jacobian matrix is evaluated. Since the propagations can be performed in parallel, the UF is ideally suited for parallel computation architectures.

Reference [13] states that if the process noise is purely additive in the model, then its covariance can simply be added using a simple addictive procedure. Next, we expand upon this concept by incorporating an approximation for the integration over the sampling interval, which more closely follows the actual process. Any process noise that is added to the state vector in the UF is effectively multiplied by the state transition matrix, $\Phi(\Delta t)$, which gives $\Phi(\Delta t)\,Q_k\Phi^T(\Delta t)$ at the end of the interval. This mapping is done automatically by the state propagation, and does not need to be explicitly accounted for in the propagation. However, adding equal process noise at the beginning and end of the propagation might yield better results. The total desired process noise follows

$$\Phi(\Delta t)\,\bar{Q}_k\Phi^T(\Delta t) + \bar{Q}_k = G_k Q_k G_k^T \tag{14}$$

where $\bar{Q}_k$ is used in Eq. (5a) and in the calculation of the predicted covariance in Eq. (8). This approach is similar to a trapezoid rule for integration. An explicit solution for $\bar{Q}_k$ in the attitude estimation problem depends on the attitude kinematics, which we next show.

Note that, in this paper we are not utilizing the $\bar{Q}_k$. Simulation results indicate that the UF performance is not greatly affected by this simple assumption. Instead, we would use the discrete process noise covariance Q_k only in Eq. (5a) and not Eq. (8).

ATTITUDE KINEMATICS

In this section a brief review of the attitude kinematics equation of motion using quaternions is shown. For more information, please refer to Ref. [9]. Also, a generalization of the Rodrigues parameters is briefly discussed. Finally, gyro and attitude-vector sensor models are shown. The quaternions is defined by $\mathbf{q} \equiv [\boldsymbol{\varrho}^T \;\; q_4]^T$, with $\boldsymbol{\varrho} \equiv [q_1 \;\; q_2 \;\; q_3]^T = \hat{\mathbf{e}}\sin(\vartheta/2)$, and $q_4 = \cos(\vartheta/2)$, where $\hat{\mathbf{e}}$ is the axis of rotation and ϑ is the angle of rotation.[14] Since a four-dimensional vector is used to describe three dimensions, the quaternion components cannot be independent of each other. The quaternion satisfies a single constraint given by $\mathbf{q}^T\mathbf{q} = 1$. The attitude matrix is related to the quaternion by

$$A(\mathbf{q}) = \Xi^T(\mathbf{q})\Psi(\mathbf{q}) \tag{15}$$

with

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} + [\boldsymbol{\varrho}\times] \\ -\boldsymbol{\varrho}^T \end{bmatrix} \tag{16}$$

$$\Psi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} - [\boldsymbol{\varrho}\times] \\ -\boldsymbol{\varrho}^T \end{bmatrix} \tag{17}$$

where $I_{3\times 3}$ is a 3×3 identity matrix and $[\boldsymbol{\varrho}\times]$ is a cross product matrix since $\mathbf{a}\times\mathbf{b} = [\mathbf{a}\times]\mathbf{b}$, with

$$[\mathbf{a}\times] \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{18}$$

Successive rotations can be accomplished using quaternion multiplication. Here we adopt the convention of Refs. [14] and [15] who multiply the quaternions in the same order as the attitude matrix multiplication: $A(\mathbf{q}')A(\mathbf{q}) = A(\mathbf{q}'\otimes\mathbf{q})$. The composition of the quaternions is bilinear, with

$$\mathbf{q}'\otimes\mathbf{q} = \begin{bmatrix} \Psi(\mathbf{q}') & \vdots & \mathbf{q}' \end{bmatrix}\mathbf{q} = \begin{bmatrix} \Xi(\mathbf{q}) & \vdots & \mathbf{q} \end{bmatrix}\mathbf{q}' \tag{19}$$

Also, the inverse quaternion is given by $\mathbf{q}^{-1} = [\boldsymbol{\varrho}^T \;\; q_4]^T$. The quaternion kinematics equation is given by

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\Xi[\mathbf{q}(t)]\boldsymbol{\omega}(t) \tag{20}$$

where $\boldsymbol{\omega}$ is the 3×1 angular velocity vector.

The local error-quaternion, denoted by $\boldsymbol{\delta}\mathbf{q} \equiv \begin{bmatrix}\boldsymbol{\delta\varrho}^T & \delta q_4\end{bmatrix}^T$, which will be defined in the UF formulation, is represented using a vector of generalized

Rodrigues parameters:

$$\boldsymbol{\delta p} \equiv f\frac{\boldsymbol{\delta \varrho}}{a+\delta q_4} \tag{21}$$

where a is a parameter from 0 to 1, and f is a scale factor. Note when $a = 0$ and $f = 1$ then Eq. (21) gives the Gibbs vector, and when $a = f = 1$ then Eq. (21) gives the standard vector of Modified Rodrigues Parameters (MRPs). For small errors the attitude part of the covariance is closely related to the attitude estimation errors for any rotation sequence, given by a simple factor. For example, the Gibbs vector linearize to half angles, and the vector of MRPs linearize to quarter angles. We will choose $f = 2(a + 1)$ so that $\|\boldsymbol{\delta p}\|$ is equal to ϑ for small errors. The inverse transformation from $\boldsymbol{\delta p}$ to $\boldsymbol{\delta q}$ is given by

$$\delta q_4 = \frac{-a\|\boldsymbol{\delta p}\|^2 + f\sqrt{f^2+(1-a^2)\|\boldsymbol{\delta p}\|^2}}{f^2+\|\boldsymbol{\delta p}\|^2} \tag{22}$$

$$\boldsymbol{\delta \varrho} = f^{-1}(a+\delta q_4)\boldsymbol{\delta p} \tag{23}$$

Discrete-time attitude observations for a single sensor are given by

$$\tilde{\mathbf{b}}_i = A(\mathbf{q})\mathbf{r}_i + \boldsymbol{\nu}_i \tag{24}$$

where $\tilde{\mathbf{b}}_i$ denotes the i^{th} 3×1 measurement vector, $\mathbf{r}_i$ is the i^{th} known 3×1 reference vector, and the sensor error-vector $\boldsymbol{\nu}_i$ is Gaussian which satisfies

$$E\{\boldsymbol{\nu}_i\} = 0 \tag{25}$$

$$E\{\boldsymbol{\nu}_i\boldsymbol{\nu}_i^T\} = \sigma_i^2 I \tag{26}$$

Where $E\{\}$ denotes expectation. Note that if unit measurement vectors are used then Eq. (26) should be appropriately modified. Multiple (N) vector measurements can be concatenated to form

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} A(\mathbf{q})\mathbf{r}_1 \\ A(\mathbf{q})\mathbf{r}_1 \\ \vdots \\ A(\mathbf{q})\mathbf{r}_N \end{bmatrix}_k + \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \vdots \\ \boldsymbol{\nu}_N \end{bmatrix}_k \tag{27}$$

$$R_k = \mathrm{diag}[\sigma_1^2 \;\; \sigma_2^2 \;\; \ldots \;\; \sigma_n^2] \tag{28}$$

where diag denotes a diagonal matrix of appropriate dimension. We should note that any attitude sensor, such as a three-axis magnetometer, star tracker, sun sensor, etc., can be put into the form given by Eq. (24). However, most sensors only observe two quantities, such as two angles in star trackers. The resulting form in Eq. (24) for these type of sensors has a unity norm constraint in the observation. However, the mean observation given by Eq. (9) may not produce an estimate with unit norm. Therefore, it is recommended that the original two quantity observation model be used for these types of sensors in the UF.

A common sensor that measures the angular rate is a rate-integrating gyro. For this sensor, a widely used model is given by[16]

$$\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t) + \boldsymbol{\eta}_v(t) \tag{29}$$

$$\dot{\boldsymbol{\beta}}(t) = \boldsymbol{\eta}_u(t) \tag{30}$$

where $\tilde{\boldsymbol{\omega}}(t)$ is the continuous-time measured angular rate, and $\boldsymbol{\eta}_v(t)$ and $\boldsymbol{\eta}_u(t)$ are independent zero-mean Gaussian white-noise processes with

$$E\{\boldsymbol{\eta}_v(t)\boldsymbol{\eta}_v(\tau)\}^T = I_{3\times 3}\sigma_v^2\delta(t-\tau) \tag{31}$$

$$E\{\boldsymbol{\eta}_u(t)\boldsymbol{\eta}_u(\tau)\}^T = I_{3\times 3}\sigma_u^2\delta(t-\tau) \tag{32}$$

where $\delta(t-\tau)$ is the Dirac delta function.

In the standard, given a post-update estimate $\hat{\boldsymbol{\beta}}_k^+$, the post-update angular velocity and propagated gyro bias follow

$$\hat{\boldsymbol{\omega}}_k^+ = \tilde{\boldsymbol{\omega}}_k - \hat{\boldsymbol{\beta}}_k^+ \tag{33}$$

$$\hat{\boldsymbol{\beta}}_{k+1}^- = \hat{\boldsymbol{\beta}}_k^- \tag{34}$$

Given post-update estimates $\hat{\boldsymbol{\omega}}_k^+$ and $\hat{\mathbf{q}}_k^+$, the propagated quaternion is found from the discrete-time equivalent of Eq. (20):

$$\hat{\mathbf{q}}_{k+1}^- = \Omega(\hat{\boldsymbol{\omega}}_k^+)\hat{\mathbf{q}}_k^+ \tag{35}$$

with

$$\Omega(\hat{\boldsymbol{\omega}}_k^+) \equiv \begin{bmatrix} Z_k & \hat{\boldsymbol{\varphi}}_k^+ \\ -\hat{\boldsymbol{\varphi}}_k^{+T} & \cos(0.5\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t) \end{bmatrix} \tag{36}$$

$$Z_k \equiv \cos(0.5\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t)I_{3\times 3} - [\hat{\varphi}_k^+\times] \tag{37}$$

$$\hat{\boldsymbol{\varphi}}_k^+ \equiv \sin(0.5\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t)\hat{\boldsymbol{\omega}}_k^+/\|\hat{\boldsymbol{\omega}}_k^+\| \tag{38}$$

where Δt is the sampling interval in the gyro.

MISALIGNMENT SENSOR MODELS

In this section misalignment models for the gyros, star tracker, and payload are presented. The gyro model also includes bias and scale factor errors.

Gyro Model

The gyro model follows the development from Ref. [5] and Ref. [7] with some modifications. The gyro model from Eq. (29) with added misalignment and scale factor errors is given by

$$\tilde{\boldsymbol{\omega}} = (I - \tilde{\Lambda} - \tilde{U})(I - \tilde{\Delta})T_{g_0,b}\boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta}_v \tag{39}$$

where $\tilde{\Lambda} = \mathrm{diag}[\lambda_x, \lambda_y, \lambda_z]$ is a matrix of symmetric scale factor errors, $\tilde{U} = \mathrm{diag}[\mu_x\mathrm{sign}(\omega_x), \mu_y\mathrm{sign}(\omega_y), \mu_z\mathrm{sign}(\omega_z)]$ is a matrix of asymmetric scale factor errors, $I - \tilde{\Delta}$ is a non-orthogonal small angle misalignment matrix (refer to §4.2 of Ref. [17] for more information; note that the sign changes later as an approximation to the

inverse of the matrix) and $T_{g_0,b}$ is the assumed-known transformation matrix from body coordinates to nominal (assumed or designed) gyro coordinates. Note that $\boldsymbol{\beta}$ and $\boldsymbol{\eta}_v$ are now in the true (misaligned) gyro coordinate system. Thus the true angular rate in body coordinate is

$$\begin{aligned}\boldsymbol{\omega} &= T_{b,g_0}(I-\tilde{\Delta})^{-1}(I-\tilde{\Lambda}-\tilde{U})^{-1}(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v)\\ &\simeq T_{b,g_0}(I+\Delta)(I+\Lambda+U)(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v) \quad (40)\end{aligned}$$

with $\Delta \simeq \tilde{\Delta}$, $\Lambda \simeq \tilde{\Lambda}$ and $U \simeq \tilde{U}$ using a small angle approximation. Since each axis is assumed to be misaligned independently we can sum up their orthogonal small angle rotations of the angular rate about each axis.[5] Let's temporarily define $\bar{\boldsymbol{\omega}} = (I+\Lambda+U)(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v)$. From Eq. (40) we have

$$\begin{aligned}(I+\Delta)\bar{\boldsymbol{\omega}} &= \begin{bmatrix}1 & -\delta_{xz} & \delta_{xy}\\ \delta_{xz} & 1 & 0\\ -\delta_{xy} & 0 & 1\end{bmatrix}\begin{bmatrix}\bar{\omega}_x\\0\\0\end{bmatrix}\\ &\quad+\begin{bmatrix}1 & -\delta_{yz} & 0\\ \delta_{yz} & 1 & -\delta_{yx}\\ 0 & \delta_{yx} & 1\end{bmatrix}\begin{bmatrix}0\\\bar{\omega}_y\\0\end{bmatrix}\\ &\quad+\begin{bmatrix}1 & 0 & \delta_{zy}\\ 0 & 1 & -\delta_{zx}\\ -\delta_{zy} & \delta_{zx} & 1\end{bmatrix}\begin{bmatrix}0\\0\\\bar{\omega}_z\end{bmatrix}\\ &= \begin{bmatrix}1 & -\delta_{yz} & \delta_{zy}\\ \delta_{xz} & 1 & -\delta_{zx}\\ -\delta_{xy} & \delta_{yx} & 1\end{bmatrix}\bar{\boldsymbol{\omega}} \quad (41)\end{aligned}$$

We then perform QR factorization so that $I+\Delta = \mathsf{QR}$, where Q is orthogonal and R is upper triangle:

$$\begin{aligned}I+\Delta &= \mathsf{QR}\\ &= \{I+[\boldsymbol{\delta}\times]\}\,\mathsf{R}\\ &= \begin{bmatrix}1 & -\delta_z & \delta_y\\ \delta_z & 1 & -\delta_x\\ -\delta_y & \delta_x & 1\end{bmatrix}\begin{bmatrix}1 & \xi_z & \xi_y\\ 0 & 1 & \xi_x\\ 0 & 0 & 1\end{bmatrix}\\ &\simeq \begin{bmatrix}1 & -(\delta_z-\xi_z) & \delta_y+\xi_y\\ \delta_z & 1 & -(\delta_x-\xi_x)\\ -\delta_y & \delta_x & 1\end{bmatrix} \quad (42)\end{aligned}$$

Equating Eq. (42) with $I+\Delta$ from Eq. (41) leads to

$$\begin{aligned}\delta_x &= \delta_{yx} & \xi_x &= \delta_{yx}-\delta_{zx}\\ \delta_y &= \delta_{xy} & \xi_y &= \delta_{zy}-\delta_{xy}\\ \delta_z &= \delta_{xz} & \xi_z &= \delta_{xz}-\delta_{yz}\end{aligned} \quad (43)$$

Since the gyros are used as the "reference" sensors for the calibration of other the sensors, the orthogonal components of gyro misalignment are all set to zero.[7] Thus $\delta_x = \delta_y = \delta_z = 0$, which gives

$$\begin{aligned}\xi_x &= -\delta_{zx}\\ \xi_y &= \delta_{zy}\\ \xi_z &= -\delta_{yz}\end{aligned}$$

and

$$\begin{aligned}I+\Delta &= \mathsf{R}\\ &= \begin{bmatrix}1 & \xi_z & \xi_y\\ 0 & 1 & \xi_x\\ 0 & 0 & 1\end{bmatrix} \quad (44)\end{aligned}$$

From Eq. (40) we have

$$\begin{aligned}\boldsymbol{\omega} &= T_{b,g_0}(I+\Delta)(I+\Lambda+U)(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v)\\ &\simeq T_{b,g_0}(I+\Delta+\Lambda+U)(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v) \quad (45)\end{aligned}$$

Let's define $M = \Delta+\Lambda+U$ (which will be used later) and $\breve{\boldsymbol{\omega}} = \tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}$. With the latter definition, the previous equation becomes

$$\begin{aligned}\boldsymbol{\omega} &= T_{b,g_0}(I+\Delta+\Lambda+U)(\breve{\boldsymbol{\omega}}-\boldsymbol{\eta}_v)\\ &= T_{b,g_0}\breve{\boldsymbol{\omega}} - T_{b,g_0}(I+\Delta+\Lambda+U)\boldsymbol{\eta}_v\\ &\quad +T_{b,g_0}(\Delta+\Lambda+U)\breve{\boldsymbol{\omega}}\\ &= T_{b,g_0}\breve{\boldsymbol{\omega}} - T_{b,g_0}(I+\Delta+\Lambda+U)\boldsymbol{\eta}_v + T_{b,g_0}\\ &\quad\times\begin{bmatrix}0 & \breve{\omega}_z & \breve{\omega}_y & \breve{\omega}_x & 0 & 0 & |\breve{\omega}_x| & 0 & 0\\ \breve{\omega}_x & 0 & 0 & 0 & \breve{\omega}_y & 0 & 0 & |\breve{\omega}_y| & 0\\ 0 & 0 & 0 & 0 & 0 & \breve{\omega}_z & 0 & 0 & |\breve{\omega}_z|\end{bmatrix}\\ &\quad\times[\xi_x\ \xi_y\ \xi_z\ \lambda_x\ \lambda_y\ \lambda_z\ \mu_x\ \mu_y\ \mu_z]^T\\ &= T_{b,g_0}\breve{\boldsymbol{\omega}} + T_{b,g_0}\Omega_g\boldsymbol{\kappa}\\ &\quad -T_{b,g_0}(I+\Delta+\Lambda+U)\boldsymbol{\eta}_v \quad (46)\end{aligned}$$

The last three columns from the 3×9 matrix from the equation above, $|\breve{\omega}_i|$ where $i = x, y, z$, are approximations of $\text{sign}(\omega_i)\breve{\omega}_i \simeq |\breve{\omega}_i|$. Also, Ω_g and $\boldsymbol{\kappa}$ correspond to the previous matrix of gyro measurement and vector of gyro misalignments and scale factor errors. Thus our angular velocity estimate is

$$\hat{\boldsymbol{\omega}} = T_{b,g_0}\breve{\boldsymbol{\omega}} + T_{b,g_0}\hat{\Omega}_g\hat{\boldsymbol{\kappa}} \quad (47)$$

where the hats correspond to their respective estimated values. Note that $\breve{\boldsymbol{\omega}}$ here is a function of both $\tilde{\boldsymbol{\omega}}$ (the uncompensated, measured gyro rate) and $\tilde{\boldsymbol{\beta}}$ (the estimated gyro bias).

Star Tracker Model

A vector star tracker model with misalignments is presented here. This model is applicable to other vector-based sensors too, for example Sun and Earth sensors. The star tracker measurement model is given by

$$\mathbf{S} = \{I-[\boldsymbol{\varsigma}\times]\}\,T_{s_0,b}A(\mathbf{q})\mathbf{p}_s \quad (48)$$

where $\boldsymbol{\varsigma}$ is the tracker misalignment, $T_{s_0,b}$ is the transformation matrix from the body to the nominal tracker coordinate, and $\mathbf{p}_s$ is a known vector in ECI coordinates of the observed star from an ephemeris calculation. Here again we employ a small angle approximation for the star tracker misalignment matrix. The estimated star tracker vector is given by

$$\hat{\mathbf{S}} = \{I-[\hat{\boldsymbol{\varsigma}}\times]\}\,T_{s_0,b}A(\hat{\mathbf{q}})\mathbf{p}_s \quad (49)$$

Payload Model

The model for the payload measurement is given by

$$\mathbf{P} = T_{p,b} A(\mathbf{q}) \mathbf{p}_p \tag{50}$$

Again, $T_{p,b}$ is the transformation matrix from the body to the payload coordinate system, and $\mathbf{p}_p$ could be a known ground point described in ECI coordinates. Since our misalignment calibration is relative to the payload, there is no misalignment to the payload model. The estimated vector is given by

$$\hat{\mathbf{P}} = T_{p,b} A(\hat{\mathbf{q}}) \mathbf{p}_p \tag{51}$$

UNSCENTED CALIBRATION

The Unscented Alignment Calibration filter is developed in this section. The filter is an extension of the USQUE developed in Ref. [9] with inclusion of calibration parameters into its state vector. The continuous attitude kinematics equation including calibration parameters is given by

$$\begin{bmatrix} \boldsymbol{\delta\dot{\varrho}} \\ \boldsymbol{\delta\dot{\beta}} \\ \boldsymbol{\delta\dot{\kappa}} \\ \boldsymbol{\delta\dot{\varsigma}} \end{bmatrix} = \begin{bmatrix} -[\boldsymbol{\omega}\times] & -\frac{1}{2}T_{b,g_0} & -\frac{1}{2}T_{b,g_0}\Omega_g & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta\varrho} \\ \boldsymbol{\delta\beta} \\ \boldsymbol{\delta\kappa} \\ \boldsymbol{\delta\varsigma} \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}T_{b,g_0}(I+M) & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_v \\ \boldsymbol{\eta}_u \\ \boldsymbol{\eta}_\kappa \\ \boldsymbol{\eta}_\varsigma \end{bmatrix} \tag{52}$$

where the continuous process noise covariance is given by

$$Q = \text{diag}\left[\sigma_v^2 I_{3\times3} \;\; \sigma_u^2 I_{3\times3} \;\; \sigma_\kappa^2 I_{9\times9} \;\; \sigma_\varsigma^2 I_{3\times3}\right] \tag{53}$$

The first square matrix on the right hand side of Eq. (52) is traditionally referred to as $F(t)$ and the second square matrix as $G(t)$. This equation is generally not used to generate simulation data or for filter propagation; however, we need the $F(t)$ and $G(t)$ matrices to compute the discrete process noise covariance. The discrete version of Eq. (52) with small angle approximation is given by

$$\begin{bmatrix} \boldsymbol{\delta\varrho} \\ \boldsymbol{\delta\beta} \\ \boldsymbol{\delta\kappa} \\ \boldsymbol{\delta\varsigma} \end{bmatrix}_{k+1} \simeq \begin{bmatrix} I & -\frac{1}{2}t\,T_{b,g_0} & -\frac{1}{2}t\,T_{b,g_0}\Omega_g & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \times \begin{bmatrix} \boldsymbol{\delta\varrho} \\ \boldsymbol{\delta\beta} \\ \boldsymbol{\delta\kappa} \\ \boldsymbol{\delta\varsigma} \end{bmatrix}_k + \Gamma_k \boldsymbol{\eta}_k \tag{54}$$

where $\boldsymbol{\eta}_k$ is a discrete white process noise with $E(\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T) = I$, and $\Gamma_k(t)$ is a square root of the discrete process noise that we are going to derive. The first square matrix on the right hand side of the above equation is traditionally referred to as the state transition matrix or $\Phi(t)$ matrix.

The discrete process noise is related to the continuous process noise by

$$Q_d = \int_0^{\Delta T} \Phi(t) G(t) Q(t) G^T(t) \Phi^T(t)\, \mathbf{d}t \tag{55}$$

where ΔT is the filter update interval. Substituting $F(t)$, $G(t)$ and $\Phi(t)$ with $M \simeq 0$ (small angle approximation) into the above equation yields:[5]

$$Q_d = \Re \begin{bmatrix} \xi_{11} & \xi_{12} & \xi_{13} & 0 \\ \xi_{21} & \xi_{22} & 0 & 0 \\ \xi_{31} & \xi_{32} & \xi_{33} & 0 \\ 0 & 0 & 0 & \xi_{44} \end{bmatrix} \Re^T \tag{56}$$

where

$$\begin{aligned} \xi_{11} &= \frac{1}{4}\Delta T \sigma_v^2 + \frac{1}{12}\Delta T^3 \sigma_u^2 + \frac{1}{12}\Delta T^3 \Omega_g \sigma_\kappa^2 \Omega_g^T \\ \xi_{12} &= -\frac{1}{4}\Delta T^2 \sigma_u^2 \\ \xi_{13} &= -\frac{1}{4}\Delta T^2 \Omega_g \sigma_\kappa^2 \\ \xi_{21} &= -\frac{1}{4}\Delta T^2 \sigma_u^2 \\ \xi_{22} &= \Delta T \sigma_u^2 \\ \xi_{31} &= -\frac{1}{4}\Delta T^2 \sigma_\kappa^2 \Omega_g^T \\ \xi_{33} &= \Delta T \sigma_\kappa^2 \\ \xi_{44} &= \Delta T \sigma_\varsigma^2 \end{aligned} \tag{57}$$

$$\Re = \begin{bmatrix} T_{b,g_0} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{58}$$

For the reasons described in Ref. [9], we choose to propagate and update the quaternion using MRPs. We begin with

$$\boldsymbol{\chi}_k(0) = \hat{\mathbf{x}}_k^+ \equiv \begin{bmatrix} \boldsymbol{\delta}\hat{\mathbf{p}}_k^+ \\ \hat{\boldsymbol{\beta}}_k^+ \\ \hat{\boldsymbol{\kappa}}_k^+ \\ \hat{\boldsymbol{\varsigma}}_k^+ \end{bmatrix} \tag{59}$$

Equation (5) is partitioned into two parts, the attitude-error part and calibration parameters part:

$$\boldsymbol{\chi}_k(i) \equiv \begin{bmatrix} \boldsymbol{\chi}_k^{\delta p}(i) \\ \cdots\cdots \\ \boldsymbol{\chi}_k^{\beta}(i) \\ \boldsymbol{\chi}_k^{\kappa}(i) \\ \boldsymbol{\chi}_k^{\varsigma}(i) \end{bmatrix}, \qquad i = 0, 1, ..., 36 \tag{60}$$

where $\boldsymbol{\chi}_k^{\beta}$ is the gyro bias, $\boldsymbol{\chi}_k^{\kappa}$ is the misalignment and scale factor error vector, and $\boldsymbol{\chi}_k^{\varsigma}$ is the star tracker

misalignment. First, a new quaternion is generated by multiplying an error quaternion by the current estimate:

$$\begin{aligned} \hat{\mathbf{q}}_k^+(0) &= \hat{\mathbf{q}}_k^+ && (61)\\ \hat{\mathbf{q}}_k^+(i) &= \boldsymbol{\delta}\hat{\mathbf{q}}_k^+(i) \otimes \hat{\mathbf{q}}_k^+, \quad i = 1,2,...,36 && (62)\end{aligned}$$

where $\boldsymbol{\delta}\mathbf{q}_k^+(i) \equiv [\boldsymbol{\delta}\boldsymbol{\varrho}_k^{+T}(i) \;\; \delta q_{4,k}^+(i)]^T$ is represented by Eqs. (22) and (23):

$$\begin{aligned} \delta q_{4,k}^+(i) &= \frac{-\alpha\|\boldsymbol{\chi}_k^{\delta p}(i)\|^2 + f\sqrt{f^2 + (1-\alpha^2)\|\boldsymbol{\chi}_k^{\delta p}(i)\|^2}}{f^2 + \|\boldsymbol{\chi}_k^{\delta p}(i)\|^2} \\ i &= 1,2,...,36 && (63)\\ \boldsymbol{\delta\varrho}_k^+(i) &= f^{-1}[\alpha + \delta q_{4,k}^+(i)]\boldsymbol{\chi}_k^{\delta p}(i) \\ i &= 1,2,...,36 && (64)\end{aligned}$$

From Eq. (61), it clearly requires that $\boldsymbol{\chi}_k^{\delta p}(0)$ be zero. Equation (62) is the propagated with

$$\hat{\mathbf{q}}_{k+1}^-(i) = \Omega[\hat{\boldsymbol{\omega}}_k^+(i)]\hat{\mathbf{q}}_k^+(i), \quad i = 0,1,...,36 \qquad (65)$$

where the estimated angular velocity from Eq. (46):

$$\begin{aligned} \hat{\boldsymbol{\omega}}_k^+(i) &= T_{b,g_0}\tilde{\boldsymbol{\omega}}_k - T_{b,g_0}\Omega_g\boldsymbol{\chi}_k^{\kappa}(i) - (I + \hat{M}(i))\boldsymbol{\chi}_k^{\beta}(i) \\ i &= 0,1,...,36 && (66)\end{aligned}$$

The error quaternions are then propagated with

$$\hat{\boldsymbol{\omega}}_k^+(i) = T_{b,g0}[I_{3\times3} - \hat{M}(i)][\tilde{\boldsymbol{\omega}}_k - \boldsymbol{\chi}_k^{\beta}(i)], \quad i = 0,1,...,36 \qquad (67)$$

Note that $\boldsymbol{\delta}\mathbf{q}_{k+1}^-(0)$ is the identity quaternion. We then convert these propagated quaternions back into sigma points with the MRPs representation using Eq. (21):

$$\begin{aligned} \boldsymbol{\chi}_{k+1}^{\delta p}(0) &= \mathbf{0} && (68)\\ \boldsymbol{\chi}_{k+1}^{\delta p}(i) &= f\frac{\boldsymbol{\delta\varrho}_{k+1}^-(i)}{a + \delta q_{4,k+1}^-(i)} \\ i &= 1,2,...,36 && (69)\end{aligned}$$

The calibration parameters are expected to stay (due to zero-mean process noises) at their previous values, thus

$$\begin{aligned} \boldsymbol{\chi}_{k+1}^{\beta}(i) &= \boldsymbol{\chi}_k^{\beta}(i) \\ \boldsymbol{\chi}_{k+1}^{\kappa}(i) &= \boldsymbol{\chi}_k^{\kappa}(i) \\ \boldsymbol{\chi}_{k+1}^{\varsigma}(i) &= \boldsymbol{\chi}_k^{\varsigma}(i) \\ i &= 0,1,...,36 && (70)\end{aligned}$$

The procedure in UF alignment calibration is as follows. First, we are given initial estimates of attitude ($\hat{\mathbf{q}}_0^+$), gyro bias ($\boldsymbol{\beta}_0^+$), gyro misalignment and scale factor ($\boldsymbol{\kappa}_0^+$), and star tracker misalignment ($\boldsymbol{\varsigma}_0^+$) with their respective initial error covariances. The UF initial state vector is set to $\hat{\mathbf{x}}_0^+ = [\mathbf{0}_{3\times1}^T \;\; \boldsymbol{\beta}_0^{+T} \;\; \boldsymbol{\kappa}_0^{+T} \;\; \boldsymbol{\varsigma}_0^{+T}]^T$. The sigma points are then calculated with Eq. (5). The attitude part of the sigma points (the first 3 components of each column of sigma points) is then converted back into quaternion with Eqs. (23) and (22) and propagated with Eq. (65). The propagated error quaternion is calculated using Eq. (67) and then transformed to propagated attitude sigma points using Eqs. (68) and (69). The propagated quaternions are used again later to find the mean observations using Eqs. (9) and (10) with

$$\begin{aligned} \gamma_{k+1}(i) &= \begin{bmatrix} [I - \boldsymbol{\chi}_{k+1}^{\varsigma}(i)\times]T_{s_0,b}A(\hat{\mathbf{q}}_{k+1}^-(i))\mathbf{p}_s \\ T_{p,b}A(\hat{\mathbf{q}}_{k+1}^-(i))\mathbf{p}_p \end{bmatrix} \\ i &= 0,1,...,36 && (71)\end{aligned}$$

The output covariance, innovation covariance and cross-correlation matrix are computed using Eqs. (11), (12), and (13). The predicted mean and error covariance can now be computed using Eqs. (7) and (8). The state vector and covariance are then updated using Eq. (2) with $\hat{\mathbf{x}}_{k+1}^+ \equiv [\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1}^{+T} \;\; \hat{\boldsymbol{\beta}}_{k+1}^{+T} \;\; \hat{\boldsymbol{\kappa}}_{k+1}^{+T} \;\; \hat{\boldsymbol{\varsigma}}_{k+1}^{+T}]^T$. Then, $\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1}^{+T}$ is then converted to $\boldsymbol{\delta}\hat{\mathbf{q}}_{k+1}^{+T}$ using Eqs. (22) and (23), and the updated quaternion using

$$\hat{\mathbf{q}}_{k+1}^+ = \boldsymbol{\delta}\mathbf{q}_{k+1}^+ \otimes \hat{\mathbf{q}}_{k+1}^-(0) \qquad (72)$$

Finally, $\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1}^+$ is reset to zero for the next propagation.

RESULTS

In this section we compare the performance of the EKF to the UF using simulated data. Then, results with actual data from the MAP spacecraft are shown using the UF. The parameters used in this simulation are: Star tracker misalignment: $-20, -20, +20$ arc-s; Initial gyro bias: 0.2, 0.3, 0.2 deg/hr for each axis; Gyro misalignment: $\delta_{xz} = \delta_{xy} = \delta_{yx} = 0, \delta_{zx} = 400, \delta_{zy} = 300, \delta_{yz} = 200$ arc-s; Gyro symmetric scale factor error: $\lambda_x = 500, \lambda_y = 500, \lambda_z = 500$ ppm; Gyro asymmetric scale factor error: $\mu_x = 100, \mu_y = 100, \mu_z = 100$ ppm; Calibration maneuver: 0.09 des/s sinusoidal in each axis at (0.0006, 0.0007, 0.0008) Hz; Gyro measurement is simulated with $\sigma_u = 1.3036 \times 10^{-3}\mu\text{rad/sec}^{3/2}$ and $\sigma_v = 1.45444\mu\text{rad/sec}^{1/2}$; Initial estimate of all calibration parameters are zero with standard deviation of 5 deg for attitude, 0.5 deg/hr for gyro bias, 500 arc-s for gyro misalignments, 500 ppm for scale factors, 50 arc-s for star tracker misalignments; Star tracker accuracy: 5, 5, 5 arc-s; and Payload measurement accuracy: 0.5, 0.5, 0.5 arc-s.

The simulation is performed on a Intel Pentium III 933 Mhz with 256MB RDRAM system. In our simulation, we assume no *a priori* knowledge of the calibration parameters. The parameter λ in the UF is chosen to be a small positive number, because a negative λ yields a non-positive semi-definite predicted covariance. Also, $a = 1$ and $f = 4$ are chosen as in USQUE.[9]

Table 1 Filter Interval and Completion Time

Filter	Filter Interval, sec	Completion Time, sec
UF	0.2	4231
UF	2.0	486
EKF	0.2	1141
EKF	0.5	472
EKF	1.0	260
EKF	2.0	120

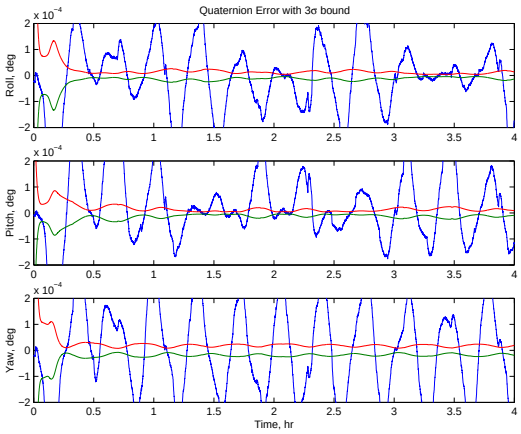


Fig. 1 EKF, $\Delta t = 2.0$, Attitude Error

Table 1 shows the time required for a few simulation runs with the UF and EKF for relative performance comparisons. Clearly, the UF requires more stringent computational power for the same filter update interval compared to the EKF. For the filter update interval of 0.2 sec, filter convergence of UF and EKF are comparable. However, due to linearization, the EKF does not work well with a higher filter update interval. The UF is shown to be able to converge even at an update interval of 2 sec with slight sacrifice to the filter performance. Compared to the EKF running at 0.2 sec interval, the UF running at 2 sec is even less computational intensive. Furthermore, the UF paves way to multiprocessing possibility that could further decrease the computational time. Lower filter update rates also lighten the burden on data telemetry and memory consumption in between update times. At higher update interval of 2.0 sec, Figures (1) to (5) clearly show that the EKF does not produce acceptable performance. However, as shown in Figures (6) to (10), the UF with a 2.0 sec update interval is still able to converge reasonably well.

Note that the simulations are performed without misalignments to the payload when it was used as an attitude sensor. Since we are performing absolute alignment calibration with respect to the gyros, the payload may be treated as an misaligned attitude sensor using an appropriate modification to Eqs. (50)

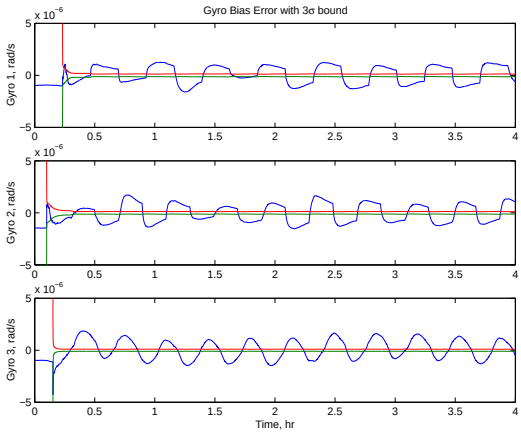


Fig. 2 EKF, $\Delta t = 2.0$, Bias Error

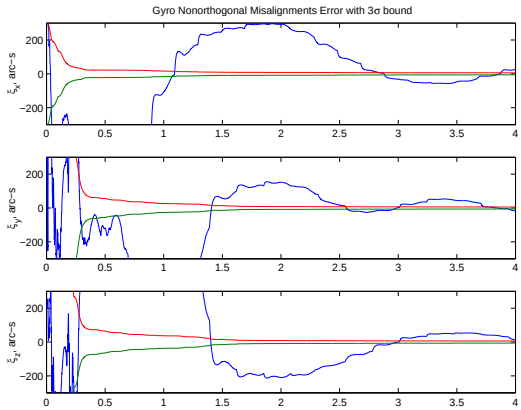


Fig. 3 EKF, $\Delta t = 2.0$, Gyro Nonorthogonal Misalignment Error

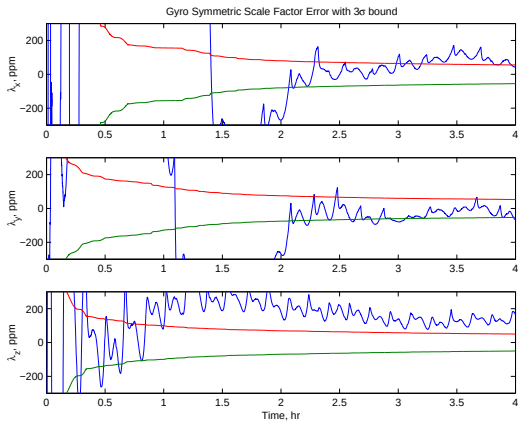


Fig. 4 EKF, $\Delta t = 2.0$, Gyro Symmetric Scale Factor Error

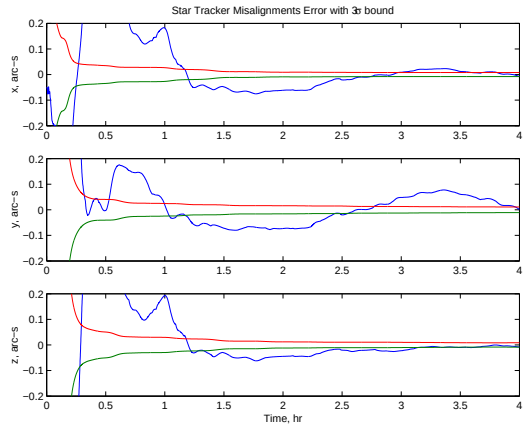


Fig. 5 EKF, $\Delta t = 2.0$, Star Tracker Misalignment Error

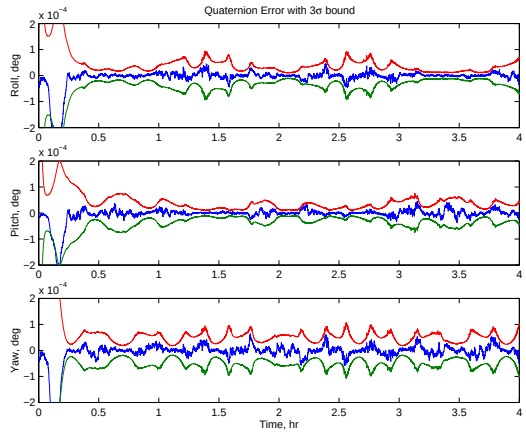


Fig. 6 UF, dt=2.0, Attitude Error

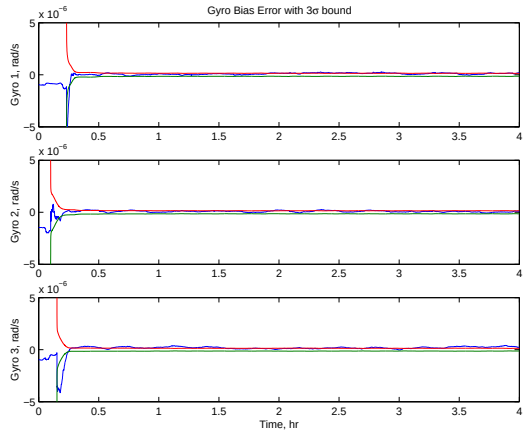


Fig. 7 UF, dt=2.0, Bias Error

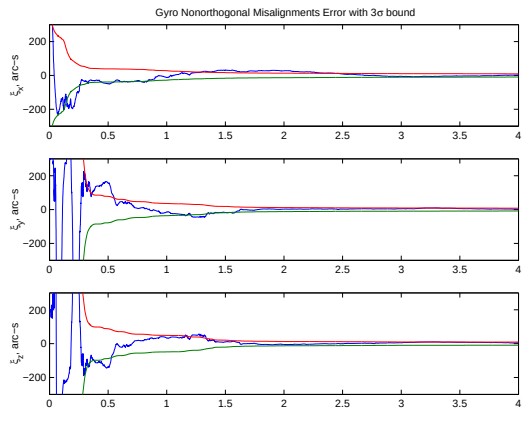


Fig. 8 UF, dt=2.0, Gyro Misalignment Error

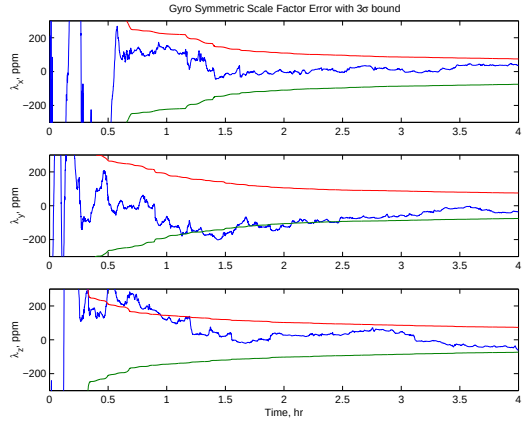


Fig. 9 UF, dt=2.0, Gyro Symmetric Scale Factor Error

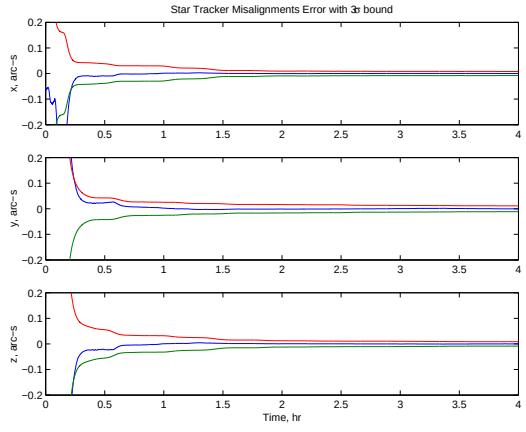


Fig. 10 UF, dt=2.0, Star Tracker Misalignment Error

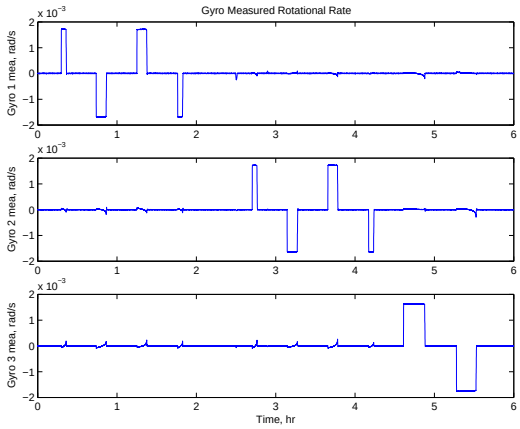


Fig. 11 Gyro Measurements

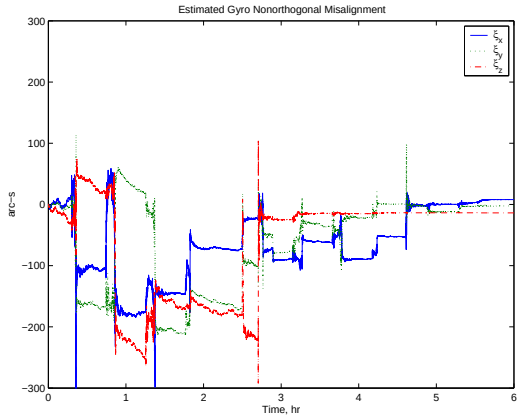


Fig. 12 Estimated Gyro Nonorthogonal Misalignment

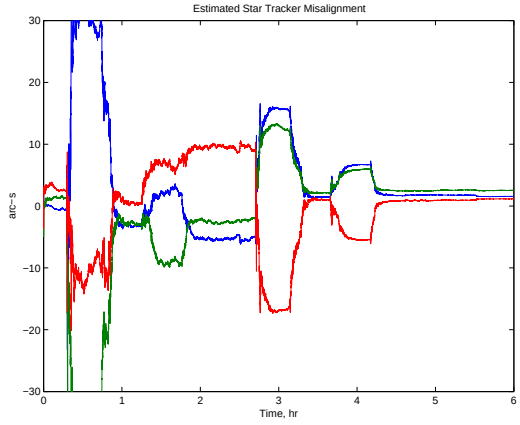


Fig. 13 Estimated Star Tracker Misalignments

and (51) with the addition of misalignment parameters. Also, Eq. (71) should be appropriately modified. As a result, the size of the state vector and the sigma points propagation become larger too.

Experimental results from the MAP spacecraft using the UF are now shown in order to test the performance of the algorithm using real data. The spacecraft per-

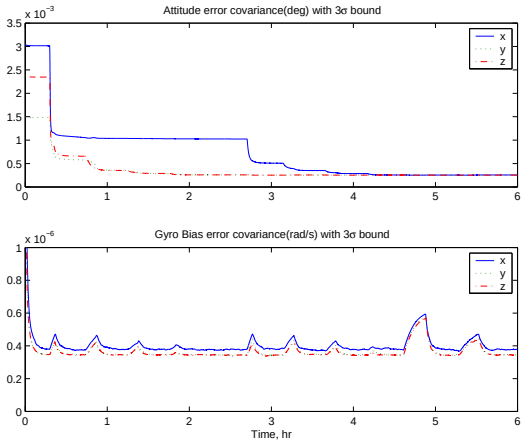


Fig. 14 3σ Bounds for the Attitude Errors

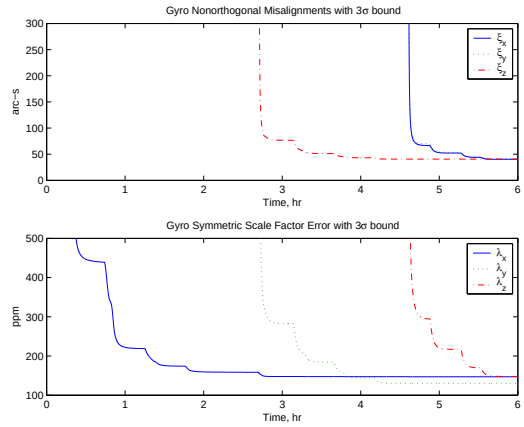


Fig. 15 3σ Bounds for the Gyro Misalignment Errors

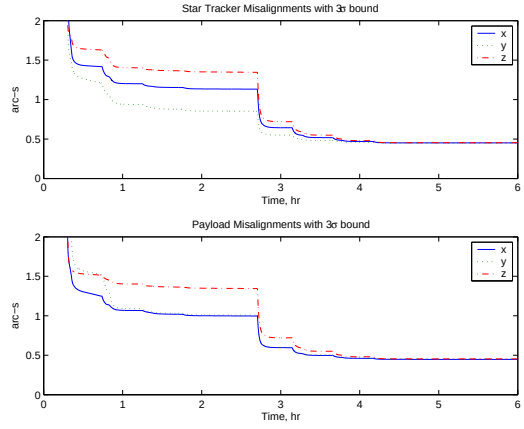


Fig. 16 3σ Bounds for the Star Tracker Misalignment Errors

forms several spacecraft maneuvers during a 6 hour data run, as shown in Figure 11. The measurements are given with a sampling interval of 1 sec. As with the simulation runs, it is assumed that no a priori information exists in the initial misalignment parameters. Plots of the estimated gyro nonorthogonal misalignments and star tracker misalignments are shown in Figures 12 and 13, respectively. Clearly, the UF converges to reasonable steady-state values. A plot of the 3σ bounds for the attitude errors is shown in Figure 14, which indicates that the attitude solutions converge to about 0.005 deg. Plots of the 3σ bounds for the gyro and star tracker misalignment errors are shown in Figures 15 and 15, respectively. These indicate that the estimated solutions converge to accurate values. Although a "truth" is not available with experimental data, these results indicate that the UF provides good convergence with reasonable alignment estimation.

CONCLUSIONS

Proper alignment calibration is of prime importance. It reduces fault detection and correction that may cause an inadvertent trip into Sun Acquisition or Safehold mode, disrupting nominal spacecraft mission capabilities. Using an onboard filter to determine misalignments is useful since parameter estimation can be accomplished in real time. In this paper, an Unscented filter has been developed from real-time alignment calibration. Simulated data results indicated that the Unscented filter proves to be more robust with respect to larger sampling intervals than the extended Kalman filter. Prolong loss of signal from attitude sensors can easily cause divergence in the extended Kalman filter. Experimental data from the Microwave Anisotropy Probe shows good convergence characteristics of the Unscented filter with accurate alignment estimation.

ACKNOWLEDGEMENT

This research was supported by NASA-Goddard Space Flight Center Grant NAG5-12179. The first author's graduate studies are supported by this grant. This author greatly appreciates the support.

REFERENCES

[1]Shuster, M. D. and Pitone, D. S., "Batch Estimation of Spacecraft Sensor Alignments II. Absolute Alignment Estimation," *Journal of the Astronautical Sciences*, Vol. 39, No. 4, Oct.-Dec. 1991.

[2]Ward, D. K., Davis, G. T., and Donnell, Jr, J. R. O., "The Microwave Anisotropy Probe Guidance, Navigation and Control Hardware Suite," *AIAA GN&C Conference & Exhibit*, Monterey, CA, Aug. 2002, AIAA-2002-4579.

[3]Andrews, S. F. and Bilanow, S., "Recent Flight Results of the TRMM Kalman Filter," *AIAA GN&C Conference & Exhibit*, Monterey, CA, Aug. 2002, AIAA-2002-5047.

[4]Andrews, S. F. and Morgenstern, W. M., "Design, Implementation, Testing, and Flight Results of the TRMM Kalman Filter," *Proceedings of the AIAA GN&C Conference & Exhibit*, Boston, MA, Aug. 1998, AIAA-1998-4509, pp. 1911–1918.

[5]Pittelkau, M. E., "Kalman Filtering for Spacecraft System Alignment Calibration," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 6, Nov.-Dec. 2001.

[6]Julier, S. J. and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," Vol. 3068 of *Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, Florida, April 1997, pp. 477–482.

[7]Pittelkau, M. E., "Everything is Relative in System Alignment Calibration," *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Denver, CO, Aug. 2000 AIAA-2000-4246, pp. 361–368.

[8]Hartley, J. B. and Hughes, P. M., "Automation of Satellite Operations: Experiences and Future Directions at NASA GSFC," *Third International Symposium on Ground Data Systems for Space Mission Operations*, Munich, Germany, Sept. 1996, SO96.8.007.

[9]Crassidis, J. L. and Markley, F. L., "Unscented Filtering for Spacecraft Attitude Estimation," *AIAA GN&C Conference & Exhibit*, Austin, TX, Aug. 2003 AIAA-2003-5484.

[10]Julier, S. J., Uhlann, J. K., and Durrant-Whyte, H. F., "A New Approach for Filtering Nonlinear Systems," *Proceedings of the American Control Conference*, Seattle, Washington, June 1995, pp. 1628–1632.

[11]Bar-Shalom, Y. and Fortmann, T. E., *Tracking and Data Association*, Academic Press, Boston, MA, 1988.

[12]Mayback, P. S., *Stochastic Models, Estimation, and Control*, Vol. 2, Academic Press, New York, NY, 1982.

[13]Wan, E. and van der Merwe, R., "The Unscented Kalman Filter," *Kalman Filtering and Neural Networks*, edited by S. Haykin, chap. 7, Wiley, 2001.

[14]Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, Oct.-Dec. 1993, pp. 439–517.

[15]Leffert, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estmation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sept.-Oct. 1982.

[16]Farenkopf, R. L., "Analytic Steady-State Accuracy Solutions for Two Common Spacecraft Attitude Estimators," *Journal of Guidance and Control*, Vol. 1, No. 4, July-Aug. 1978, pp. 282–284.

[17]Crassidis, J. L. and Junkins, J. L., *Optimal Estimal of Dynamic Systems*, CRC Press, in progress, Boca Raton, FL, 2003.