Contents

List of Figures

List of Tables

1 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. More stringent attitude pointing accuracy requires misalignments to be estimated and implemented into the attitude estimator. It has been noted in some recent papers [1]–[3] of the importance of proper calibration for use in fault detection or rate derivation. Usually, sensor alignments are calibrated regularly to account for the drifting nature of sensor alignment due to in-space disturbances or spacecraft operational mechanisms. This improves residual characteristics and consequently maintains proper attitude pointing accuracy.[2]. In-space disturbances include solar wind, aerodynamics, persistence thermal shock. Spacecraft mechanisms include momentum damping, thruster firing, sensors or payload reorientation and so on.

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 normal 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. [4].

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. However, most accurate sensor misalignment calibration comes from batch estimation that utilizes the whole measurement subset to obtain more accurate misalignment parameters estimation. Furthermore, sequential or real-time alignment calibration requires more computational power and the alignment drift might not occurring at constant "magnitude". Also, during nominal missions mode, some spacecraft might not produce enough maneuver to "feed" the attitude filter with enough diversified measurements, i.e., lack of observability for misalignment parameters.

In this project, we propose a batch attitude estimator and misalignment calibration. The particular batch estimator that will be used is that of Rauch-Tung-Striebel(RTS). Batch calibration usually performed by the ground crew with attitude sensors telemetry from the spacecraft. Batch estimation is usually much more computationally burdensome but poses little problem to modern computers on the ground. Because of operational necessity, budget constraints, and technology push a number of automation initiatives are underway at NASA-Goddard Space Flight Center (GSFC)[5]. When the on-board computational suffice, alignment calibration could be scheduled autonomously and regularly.

All alignment calibration are essentially relative alignment calibration. This project addresses alignment estimation [6] with respect to the gyros.

First, the attitude kinematics are briefly reviewed, follows by derivation that leads to the nonlinear continuous-time model, discrete-time measurements RTS batch estimator. Sensor models with misalignments are then derived. These include a three-axis orthogonal gyros and vector output star tracker. The gyro model also includes symmetric scale factor errors and bias drifts. The alignment filter follows next. Simulation is the performed with realistically simulated attitude measurement and lastly the conclusion is drawn.

2 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. [7]. 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.[8] 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

$$A(\mathbf{q}) = \Xi^T(\mathbf{q})\Psi(\mathbf{q}) \tag{2.1}$$

with

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} + [\boldsymbol{\varrho}\times] \\ -\boldsymbol{\varrho}^T \end{bmatrix} \tag{2.2}$$

$$\Psi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} - [\boldsymbol{\varrho}\times] \\ -\boldsymbol{\varrho}^T \end{bmatrix} \tag{2.3}$$

where $I_{3\times3}$ 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

$$[\mathbf{a}\times] \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2.4}$$

Successive rotations can be accomplished using quaternion multiplication. Here we adopt the convention of Refs. [9] and [8] 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

$$\mathbf{q}'\otimes\mathbf{q} = \begin{bmatrix} \Psi(\mathbf{q}') & \vdots & \mathbf{q}' \end{bmatrix}\mathbf{q} = \begin{bmatrix} \Xi(\mathbf{q}) & \vdots & \mathbf{q} \end{bmatrix}\mathbf{q}' \tag{2.5}$$

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

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\Xi[\mathbf{q}(t)]\boldsymbol{\omega}(t) \tag{2.6}$$

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

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

$$\tilde{\mathbf{b}}_i = A(\mathbf{q})\mathbf{r}_i + \boldsymbol{\nu}_i \tag{2.7}$$

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

$$E\{\boldsymbol{\nu}_i\} = 0 \tag{2.8}$$

$$E\{\boldsymbol{\nu}_i\boldsymbol{\nu}_i^T\} = \sigma_i^2 I \tag{2.9}$$

Where $E\{\}$ denotes expectation. Note that if unit measurement vectors are used then Eq. (2.9) should be appropriately modified. Multiple (N) vector measurements can be concatenated to form

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} A(\mathbf{q})\mathbf{r}_1 \\ A(\mathbf{q})\mathbf{r}_1 \\ \vdots \\ A(\mathbf{q})\mathbf{r}_N \end{bmatrix}_k + \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \vdots \\ \boldsymbol{\nu}_N \end{bmatrix}_k \tag{2.10}$$

$$R_k = \text{diag}[\sigma_1^2 \;\; \sigma_2^2 \;\; \dots \;\; \sigma_n^2] \tag{2.11}$$

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. (2.7).

A common sensor that measures the angular rate is a rate-integrating gyro. For this sensor, a widely used model is given by[10]

$$\tilde{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}(t) + \boldsymbol{\beta}(t) + \boldsymbol{\eta}_v(t) \tag{2.12}$$

$$\dot{\boldsymbol{\beta}}(t) = \boldsymbol{\eta}_u(t) \tag{2.13}$$

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

$$E\{\boldsymbol{\eta}_v(t)\boldsymbol{\eta}_v^T(\tau)\} = I_{3\times 3}\sigma_v^2\delta(t-\tau) \tag{2.14}$$

$$E\{\boldsymbol{\eta}_u(t)\boldsymbol{\eta}_u^T(\tau)\} = I_{3\times 3}\sigma_u^2\delta(t-\tau) \tag{2.15}$$

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

$$\begin{aligned} \hat{\boldsymbol{\omega}}_k^+ &= \tilde{\boldsymbol{\omega}}_k - \hat{\boldsymbol{\beta}}_k^+ & (2.16) \\ \hat{\boldsymbol{\beta}}_{k+1}^- &= \hat{\boldsymbol{\beta}}_k^- & (2.17) \end{aligned}$$

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. (2.6):

$$\hat{\mathbf{q}}_{k+1}^- = \Omega(\hat{\boldsymbol{\omega}}_k^+)\hat{\mathbf{q}}_k^+ \tag{2.18}$$

with

$$\Omega(\hat{\boldsymbol{\omega}}_k^+) \equiv \begin{bmatrix} Z_k & \hat{\varphi}_k^+ \\ -\hat{\varphi}_k^+ & \cos(0.5\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t) \end{bmatrix} \tag{2.19}$$

$$Z_k \equiv \cos(0.5\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t)I_{3\times3} - [\hat{\varphi}_k^+ \times] \tag{2.20}$$

$$\hat{\varphi}_k^+ \equiv \sin(0.5\|\hat{\boldsymbol{\omega}}_k^+\|\Delta t)\hat{\boldsymbol{\omega}}_k^+/\|\hat{\boldsymbol{\omega}}_k^+\| \tag{2.21}$$

where Δt is the sampling interval in the gyro.

3 Batch Estimation

3.1 Fixed Interval Smoother

Batch estimators are also called *smoothers* since they tend to “smooth” out the effect of measurement noise. Batch estimator uses all measurement in a subset to estimate the states of a system at certain time t. Thus, it provides a lower error-covariance than sequential estimator. The disadvantage of batch estimation is that it cannot be implemented in real-time. Lower error-covariance is desired when accuracy is an issue but real-time estimation is not required. This is especially useful for calibration. For more formal treatment of batch estimation, please refer to Ref. [11] and Ref. [12].

Two filters are used in batch estimator, a forward-time filter and a backward-time filter. The covariances and estimation from these two filter are then combined to produced an equal or better estimate than just a forward-filter. There are three types of batch estimators:-

- Fixed-Interval Smoothing – Uses the entire measurement on a fixed interval. Initial time, 0, and final time, T, are fixed and t runs from 0 to T. This produces the best estimation over the interval.

- Fixed-Point Smoothing – Estimates the states at a specific point in time as the final time, T, increases.

- Fixed-Lag Smoothing – Estimates the states at a constant interval that lags behind current measurement as T increases. This is often used to “refine” an optimal filter.

In this report, we will only consider the fixed-interval type smoother since it is the most widely used. Specifically, a Rauch-Tung-Striebel (RTS) version of the batch filter for continuous-time model with discrete-time measurements will be derived since most modern day applications fall into this category. Finally, this filter will be extended for nonlinear systems.

Let’s define

$$\hat{\mathbf{x}}(t) = w\hat{\mathbf{x}}_f(t) + w'\hat{\mathbf{x}}_b(t) \tag{3.1}$$

where $\hat{\mathbf{x}}(t)$ is the smoother estimate, $\hat{\mathbf{x}}_f(t)$ is the forward estimate, $\hat{\mathbf{x}}_b(t)$ is the backward estimate and w and w' are two optimal weighting parameters we sought. Let the over tilde denotes the error between the estimates and the truth,

$$\begin{aligned} \tilde{\mathbf{x}}(t) &= \hat{\mathbf{x}}(t) - \mathbf{x}(t) \\ \tilde{\mathbf{x}}_f(t) &= \hat{\mathbf{x}}_f(t) - \mathbf{x}(t) \\ \tilde{\mathbf{x}}_b(t) &= \hat{\mathbf{x}}_b(t) - \mathbf{x}(t) \end{aligned} \tag{3.2}$$

Using Eq. (3.2) in Eq. (3.1) yields

$$\tilde{\mathbf{x}}(t) = (w + w' - I)\mathbf{x}(t) + w\tilde{\mathbf{x}}_f(t) + w'\tilde{\mathbf{x}}_b(t) \tag{3.3}$$

Taking the expectation of Eq. (3.3) and obviously for an unbiased estimator $w' = I - w$, thus,

$$\tilde{\mathbf{x}}(t) = w\tilde{\mathbf{x}}_f(t) + (I - w)\tilde{\mathbf{x}}_b(t) \tag{3.4}$$

The definition of error covariances:

$$P(t) \equiv E\{\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t)\} \tag{3.5}$$

$$P_f(t) \equiv E\{\tilde{\mathbf{x}}_f(t)\tilde{\mathbf{x}}_f^T(t)\} \tag{3.6}$$

$$P_b(t) \equiv E\{\tilde{\mathbf{x}}_b(t)\tilde{\mathbf{x}}_b^T(t)\} \tag{3.7}$$

where $P(t)$ is the smoother error covariance, $P_f(t)$ is the forward filter error covariance and $P_b(t)$ is the backward filter error covariance. With $w' = I = w$ in Eq. (3.3) and Eq. (3.5), the optimal smoother error covariance becomes

$$P(t) = wP_f(t)w^T + (I - w)P_b(t)(I - w)^T \tag{3.8}$$

We choose to minimize the trace of $P(t)$

$$\begin{aligned} \frac{\partial}{\partial w} Tr[P(t)] &= 0 \\ 2wP_f(t) - 2(I - w)P_b(t) &= 0 \end{aligned} \tag{3.9}$$

or

$$w = P_b(t)[P_f(t) + P_b(t)]^{-1} \tag{3.10}$$

thus

$$\begin{aligned} I - w &= [P_f(t) + P_b(t)][P_f(t) + P_b(t)]^{-1} - P_b(t)[P_f(t) + P_b(t)]^{-1} \\ &= P_f(t)[P_f(t) + P_b(t)]^{-1} \end{aligned} \tag{3.11}$$

Substituting Eq. (3.10) and Eq. (3.11) into Eq. (3.8) and after some algebraic manipulations gives

$$\begin{aligned} P(t) &= [P_f^{-1}(t) + P_b^{-1}(t)]^{-1} && (3.12) \\ P^{-1}(t) &= P_f^{-1}(t) + P_b^{-1}(t) && (3.13) \end{aligned}$$

By inspection of Eq. (3.12), the smoother error covariance, $P(t)$, is at least equal or lower than the forward error covariance, $P_f(t)$. Thus, the estimated smoother states are at least as accurate as the forward filter estimates alone. Ref. [13] proved that only those states that are driven by process noise are *smoothable*. In other words, only time-varying states are smoothable.

Substituting Eq. (3.10) and Eq. (3.11) into Eq. (3.1) and utilizing Eq. (3.12) yields

$$\mathbf{x}(t) = P(t)[P_f^{-1}(t)\hat{\mathbf{x}}_f(t) + P_b^{-1}(t)\hat{\mathbf{x}}_b(t)] \tag{3.14}$$

The forward filter follows the Kalman filter with states and error covariance given by

$$\frac{d}{dt}\hat{\mathbf{x}}_f(t) = F(t)\hat{\mathbf{x}}_f(t) + P(t)H^T(t)P^{-1}[\tilde{y}(t) - H(t)\hat{x}(t)] \qquad \hat{\mathbf{x}}_f(t_0) = \hat{\mathbf{x}}_{f0} \tag{3.15}$$

$$\begin{aligned} \frac{d}{dt}P_f(t) = F(t)P_f(t) + P_f(t)F^T(t) - P_f(t)H^T(t)R^{-1}(t)H(t)P_f(t) \\ +G(t)Q(t)G^T(t) \qquad P(t_0) = P_0 \end{aligned} \tag{3.16}$$

The backward filter is obtained by defining $\tau = T - t$ and thus

$$\frac{d}{d\tau}\mathbf{x}_b(t) = -\frac{d}{dt}\mathbf{x}(t) = -F(t)\mathbf{x}(t) - G(t)\mathbf{w}(t) \tag{3.17}$$

So we replace

$$\begin{aligned}\frac{d}{d\tau}\hat{\mathbf{x}}_b(t) &= -F(t)\hat{\mathbf{x}}_b(t) + P_b(t)H^T(t)R^{-1}(t)[\tilde{y} - H(t)\hat{\mathbf{x}}_b(t)] \qquad (3.18)\\ \frac{d}{d\tau}P_b(t) &= -F(t)P_b(t) - P_b(t)F^T(t) - P_b(t)H^T(t)R^{-1}(t)H(t)P_b(t)\\ &\quad +G(t)Q(t)G^T(t) \qquad (3.19)\end{aligned}$$

Since at $t = T$, the smoother error covariance should equal to the forward error covariance, thus from Eq. (3.13), the boundary condition for Eq. (3.19) is $P_b^{-1}(T) = 0$ but the boundary condition $\hat{\mathbf{x}}_b(T)$ is not known. Let's define

$$\boldsymbol{\chi}_b(t) = P_b^{-1}(t)\hat{\mathbf{x}}_b(t) \qquad (3.20)$$

Since $P_b^{-1}(T) = 0$, $\boldsymbol{\chi}_b(T) = 0$ too. Now we take the time derivative of this identity, $P_b^{-1}(t)P_b(t) = I$

$$\left[\frac{d}{d\tau}P_b^{-1}(t)\right]P_b(t) + P_b^{-1}(t)\left[\frac{d}{d\tau}P_b(t)\right] = 0 \qquad (3.21)$$

so

$$\frac{d}{d\tau}P_b^{-1}(t) = -P_b^{-1}(t)\left[\frac{d}{d\tau}P_b(t)\right]P_b^{-1}(t) \qquad (3.22)$$

Substituting Eq. (3.19) into Eq. (3.22) yields

$$\begin{aligned}\frac{d}{d\tau}P_b^{-1}(t) = P_b^{-1}(t)F(t) + F^T(t)P_b^{-1}(t) - P_b^{-1}(t)G(t)Q(t)G^T(t)P_b^{-1}(t)\\ +H^T(t)R^{-1}(t)H(t) \qquad (3.23)\end{aligned}$$

Taking the backward time derivative of Eq. (3.20) and plugging Eq. (3.18) and Eq. (3.23) into the resultant gives

$$\frac{d}{d\tau}\boldsymbol{\chi}_b(t) = [F^T(t) - P_b^{-1}(t)G(t)Q(t)G^T(t)]\boldsymbol{\chi}_b(t) \\ +H^T(t)R^{-1}(t)\tilde{y}(t) \qquad \boldsymbol{\chi}_b(T) = \mathbf{0} \tag{3.24}$$

Thus Eq. (3.14) with definition of Eq.(3.20) and Eq. (3.12) now becomes

$$\hat{\mathbf{x}}(t) = [I + P(t)P_b^{-1}(t)]^{-1}\mathbf{x}_f(t) + P(t)\boldsymbol{\chi}_b(t) \tag{3.25}$$

Applying Matrix Inversion Lemma, Eq. (8.1), with $A = P_f^{-1}(t)$, $B = P_b^{-1}(t)$ and $C = D = I$ yields

$$P(t) = P_f(t) - P_f(t)P_b^{-1}(t)[I + P(t)P_b^{-1}(t)]^{-1}P(t) \tag{3.26}$$

First, run the forward filter with Eq. (3.15) and Eq. (3.16) and backward filter with Eq. (3.24) and Eq. (3.23). Finally Eq. (3.25) and Eq. (3.26) are used to combine the forward and backward filters to obtain an even lower error covariance and states estimate.

3.2 RTS formulation

The RTS form that eliminate the one pass by combining the backward filter and smoother into one single backward recursion, thus offering great advantage in terms of computational requirement. It also reduce memory requirement by almost half before running the smoother since there is no need to store estimation and error covariance of the backward filter.

Taking the backward time derivative of Eq. (3.13) and utilizing Eq. (3.22)

$$\frac{d}{d\tau}P^{-1}(t) = -P_f^{-1}(t)\left[\frac{d}{d\tau}P_f(t)\right]P_F^{-1}(t) + \frac{d}{d\tau}P_b^{-1}(t) \tag{3.27}$$

With $\frac{dP_f(t)}{dt} = -\frac{dP_f(t)}{d\tau}$ the above equation leads to

$$\frac{d}{d\tau}P^{-1}(t) = P_f^{-1}(t)\left[\frac{d}{dt}P_f(t)\right]P_F^{-1}(t) + \frac{d}{d\tau}P_b^{-1}(t) \tag{3.28}$$

Substituting Eq. (3.16) and Eq. (3.19) into Eq. (3.28) and utilizing Eq. (3.13) gives

$$\begin{aligned}\frac{d}{d\tau}P^{-1}(t) &= P^{-1}(t)F(t) + F^T(t)P^{-1}(t) + P_f^{-1}(t)G(t)Q(t)G^T(t)P_f^{-1}(t) \\ &\quad -[P^{-1}(t) - P_f^{-1}(t)]G(t)Q(t)G^T(t)[P^{-1}(t) - P_f^{-1}(t)]\end{aligned} \tag{3.29}$$

Using similar approached in obtaining Eq. (3.22)

$$\frac{d}{d\tau}P_f^{-1}(t) = P_f^{-1}(t)\left[\frac{d}{d\tau}P_f(t)\right]P_f^{-1}(t) \tag{3.30}$$

and substituting this resultant into Eq. (3.29) and multiplying both sides with $P(t)$ yields

$$\begin{aligned}\frac{d}{dt}P(t) &= [F(t) + G(t)Q(t)G^T(t)P_f^{-1}(t)]P(t) \\ &\quad +P(t)[F(t) + G(t)Q(t)G^T(t)P_f^{-1}(t)]^T - G(t)Q(t)G^T(t)\end{aligned} \tag{3.31}$$

as before, $P_b^{-1}(t) = 0$, thus Eq. (3.31) is integrated backward in time with boundary condition $P(T) = P_f(T)$. Notice also Eq.(3.31) does not contain the backward error covariance and only one matrix inverse is required, thus making the RTS for more desired in batch filtering.

Now we turn our attention to the RTS states estimate. Rearranging Eq. (3.14) gives

$$P^{-1}(t)\mathbf{x}(t) = P_f^{-1}(t)\hat{\mathbf{x}}_f(t) + P_b^{-1}(t)\hat{\mathbf{x}}_b(t) \tag{3.32}$$

Taking the time derivative of the equation above leads to

$$\begin{aligned} P^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}(t)\right] &= \left[\frac{d}{dt}P_f^{-1}(t)\right]\hat{\mathbf{x}}_f(t) + P_f^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}_f(t)\right] \\ &\quad + \frac{d}{dt}\boldsymbol{\chi}_b(t) - \frac{d}{dt}\left[P^{-1}(t)\right]\hat{\mathbf{x}}(t) \end{aligned} \tag{3.33}$$

Using similar approached in obtaining Eq. (3.22)

$$\frac{d}{d\tau}P^{-1}(t) = -P^{-1}(t)\left[\frac{d}{d\tau}P(t)\right]P^{-1}(t) \tag{3.34}$$

Substituting Eq. (3.30) and Eq. (3.34) into Eq. (3.33)

$$\begin{aligned} P^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}\right] &= -P_f^{-1}(t)\left[\frac{d}{dt}P_f(t)\right]P_f^{-1}(t)\hat{\mathbf{x}}_f(t) + P_f^{-1}(t)\left[\frac{d}{dt}\hat{\mathbf{x}}_f(t)\right] \\ &\quad + \frac{d}{dt}\hat{\chi}_b(t) + P^{-1}(t)\left[\frac{d}{dt}P(T)\right]P^{-1}(t)\hat{\mathbf{x}}(t) \end{aligned} \tag{3.35}$$

Substituting Eq. (3.16) for $\frac{d}{dt}P_f(t)$, Eq. (3.15) for $\frac{d}{dt}\hat{\mathbf{x}}_f(t)$, Eq. (3.24) for $\frac{d}{dt}\hat{\chi}_b(t)$ and Eq. (3.31) for $\frac{d}{dt}P(t)$ into Eq. (3.35) and after considerable algebra gives

$$\frac{d}{dt}\hat{\mathbf{x}} = F(t)\hat{\mathbf{x}}(t) + B(t)\mathbf{u}(t) + G(t)Q(t)G^T(t)P_f^{-1}(t)[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)] \tag{3.36}$$

Eq. (3.36) is integrated backward in time with boundary condition $\hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$. As before, first the forward Kalman filter is executed for t from 0 to T and then Eq. (3.31) and Eq. (3.36) are integrated backward in time. Notice an interesting fact that Eq. (3.31) is not used in finding the smoother states, in other words, Eq. (3.31) is solely to for finding the smoother error covariance. Also, Eq. (3.36) does not involve any measurement directly.

3.3 Nonlinear Smoothing

In this section, the RTS smoother is extended to include nonlinear model. Extended Kalman filter is used however direct implementation of EKF on backward integration and measurement updates is undesired. This is because the linearization for the backward filter has to be about the forward filter estimate and not backward filter estimate. Thus, traditional EKF algorithm is only used for the forward filter. The backward filter need to be rederived.

First, we linearize $\mathbf{f}(\hat{\mathbf{x}}(t), \mathbf{u}(t), t)$ about $\hat{\mathbf{x}}_f(t)$. Then with $d\mathbf{x}/dt = -d\mathbf{x}/d\tau$, Eq. (3.36) becomes

$$\frac{d}{d\tau}\hat{\mathbf{x}}(t) = -[F(\hat{\mathbf{x}}_f(t), t) + K(t)][\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_f(t)] - \mathbf{f}(\hat{\mathbf{x}}_f(t), \mathbf{u}(t), t) \tag{3.37}$$

where

$$K(t) \equiv G(t)Q(t)G^T(t)P_f^{-1}(t) \tag{3.38}$$

$$F(\hat{\mathbf{x}}_f(t), t) \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_f(t)} \tag{3.39}$$

Notice that Eq .(3.37) is integrated backward in time with initial condition given by the forward filter estimate at $t = T$, $\hat{\mathbf{x}}(T) = \hat{\mathbf{x}}_f(T)$. The smoother error covariance follows from Eq. (3.31) again with $\frac{d}{d\tau}P(t) = -\frac{d}{dt}P(t)$

$$\frac{d}{d\tau}P(t) = -[F(\hat{\mathbf{x}}_f(t), t) + K(t)]P(t) - P(t)[F(\hat{\mathbf{x}}_f(t), t) + K(t)]^T + G(t)Q(t)G^T(t) \tag{3.40}$$

As before, Eq. (3.40) is integrated backward in time with initial condition $P(T) = P_f(T)$.

4 Sensor Models with Misalignment

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

4.1 Gyro

The gyro model follows the development from Ref. [4] and Ref. [14] with some modifications. The gyro model from Eq. (2.12) is modified with added misalignment and scale factor errors is given by

$$\tilde{\boldsymbol{\omega}} = (I - \tilde{\Lambda})(I - \tilde{\Delta})T_{g_0,b}\boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\eta}_v \tag{4.1}$$

where $\tilde{\Lambda} = \text{diag}(\lambda_x, \lambda_y, \lambda_z)$ is symmetric scale factor errors and $\tilde{\Delta}$ is a non-orthogonal small angle misalignment matrix and $T_{g_0,b}$ is a 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

$$\begin{aligned} \boldsymbol{\omega} &= T_{b,g_0}(I - \tilde{\Delta})^{-1}(I - \tilde{\Lambda})^{-1}(\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v) \\ &\simeq T_{b,g_0}(I + \Delta)(I + \Lambda)(\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v) \end{aligned} \tag{4.2}$$

with $\Delta \simeq \tilde{\Delta}$ and $\Lambda \simeq \tilde{\Lambda}$ for small angles 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.[4]. Let's temporarily define $\bar{\boldsymbol{\omega}} = (I + \Lambda)(\tilde{\boldsymbol{\omega}} - \boldsymbol{\beta} - \boldsymbol{\eta}_v)$ and from Eq. (4.2) we have

$$\begin{aligned}
(I+\Delta)\bar{\boldsymbol{\omega}} &= \begin{bmatrix} 1 & -\delta_{xz} & \delta_{xy} \\ \delta_{xz} & 1 & 0 \\ -\delta_{xy} & 0 & 1 \end{bmatrix}\begin{bmatrix} \bar{\omega}_x \\ 0 \\ 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 1 & -\delta_{yz} & 0 \\ \delta_{yz} & 1 & -\delta_{yx} \\ 0 & \delta_{yx} & 1 \end{bmatrix}\begin{bmatrix} 0 \\ \bar{\omega}_y \\ 0 \end{bmatrix} \\
&\quad + \begin{bmatrix} 1 & 0 & \delta_{zy} \\ 0 & 1 & -\delta_{zx} \\ -\delta_{zy} & \delta_{zx} & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \bar{\omega}_z \end{bmatrix} \\
&= \begin{bmatrix} 1 & -\delta_{yz} & \delta_{zy} \\ \delta_{yz} & 1 & -\delta_{zx} \\ -\delta_{xy} & \delta_{yx} & 1 \end{bmatrix}\bar{\boldsymbol{\omega}} \qquad (4.3)
\end{aligned}$$

We then perform QR factorization so that $I + \Delta = QR$ where Q is orthogonal and R is upper triangle.

$$\begin{aligned}
I + \Delta &= \mathsf{QR} \\
&= [I + \boldsymbol{\delta}\times]\,\mathsf{R} \\
&= \begin{bmatrix} 1 & -\delta_z & \delta_y \\ \delta_z & 1 & -\delta_x \\ -\delta_y & \delta_x & 1 \end{bmatrix} \begin{bmatrix} 1 & \xi_z & \xi_y \\ 0 & 1 & \xi_x \\ 0 & 0 & 1 \end{bmatrix} \\
&\simeq \begin{bmatrix} 1 & -(\delta_z - \xi_z) & \delta_y + \xi_y \\ \delta_z & 1 & -(\delta_x - \xi_x) \\ -\delta_y & \delta_x & 1 \end{bmatrix} \qquad (4.4)
\end{aligned}$$

Equating Eq. (4.4) with $I + \Delta$ from Eq. (4.3) we found that

$$\begin{aligned}
\delta_x &= \delta_{yx} & \xi_x &= \delta_{yx} - \delta_{zx} \\
\delta_y &= \delta_{xy} & \xi_y &= \delta_{zy} - \delta_{xy} \qquad (4.5) \\
\delta_z &= \delta_{xz} & \xi_z &= \delta_{xz} - \delta_{yz}
\end{aligned}$$

Since we are using gyro as "reference" sensors to calibration of other sensors, we equate the orthogonal components of gyro misalignment to zeros. Thus $\delta_z = \delta_y = \delta_z = 0$ and we obtained

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

and

$$\begin{aligned} I+\Delta &= \mathsf{R} \\ &= \begin{bmatrix} 1 & \xi_z & \xi_y \\ 0 & 1 & \xi_x \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.6}$$

From Eq. (4.2)

$$\begin{aligned} \boldsymbol{\omega} &= T_{b,g_0}(I+\Delta)(I+\Lambda)(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v) \\ &\simeq T_{b,g_0}(I+\Delta+\Lambda)(\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}-\boldsymbol{\eta}_v) \end{aligned} \tag{4.7}$$

Let's define $\breve{\boldsymbol{\omega}}=\tilde{\boldsymbol{\omega}}-\boldsymbol{\beta}$. With this, the previous equation becomes

$$\begin{aligned} \boldsymbol{\omega} &= T_{b,g_0}(I+\Delta+\Lambda)(\breve{\boldsymbol{\omega}}-\boldsymbol{\eta}_v) \\ &= T_{b,g_0}\breve{\boldsymbol{\omega}}-T_{b,g_0}(I+\Delta+\Lambda)\boldsymbol{\eta}_v \\ &\quad +T_{b,g_0}(\Delta+\Lambda)\breve{\boldsymbol{\omega}} \\ &= T_{b,g_0}\breve{\boldsymbol{\omega}}-T_{b,g_0}(I+\Delta+\Lambda)\boldsymbol{\eta}_v+T_{b,g_0}\times \\ &\quad \begin{bmatrix} 0 & \breve{\omega}_z & \breve{\omega}_y & \breve{\omega}_x & 0 & 0 \\ \breve{\omega}_x & 0 & 0 & 0 & \breve{\omega}_y & 0 \\ 0 & 0 & 0 & 0 & 0 & \breve{\omega}_z \end{bmatrix} \\ &\quad \times[\xi_x\ \xi_y\ \xi_z\ \lambda_x\ \lambda_y\ \lambda_z\ \mu_x\ \mu_y\ \mu_z]^T \\ &= T_{b,g_0}\breve{\boldsymbol{\omega}}+T_{b,g_0}\Omega_g\boldsymbol{\kappa} \\ &\quad -T_{b,g_0}(I+\Delta+\Lambda)\boldsymbol{\eta}_v \end{aligned} \tag{4.8}$$

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

$$\hat{\boldsymbol{\omega}} = T_{b,g_0}\breve{\boldsymbol{\omega}} + T_{b,g_0}\hat{\Omega}_g\hat{\boldsymbol{\kappa}} \tag{4.9}$$

where the hats correspond to their respective estimation. 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).

4.2 Star Tracker

A vector star tracker model with misalignments is presented here. Note that this model is applicable to other vector-based sensors too, for example Sun and Earth sensors. The star tracker measurement is assumed to be

$$\mathbf{S} = [I - \boldsymbol{\varsigma}\times]T_{s_0,b}A(\mathbf{q})\mathbf{p}_s \tag{4.10}$$

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 known vector in ECI coordinates of the observed star from ephemeris calculation. Here again we employ small angle approximation for the star tracker misalignment matrix. The estimation star tracker measurement is obviously

$$\hat{\mathbf{S}} = [I - \hat{\boldsymbol{\varsigma}}\times]T_{s_0,b}A(\hat{\mathbf{q}})\mathbf{p}_s \tag{4.11}$$

4.3 Payload

The model for the payload measurement is given by

$$\mathbf{P} = T_{p,b}A(\mathbf{q})\mathbf{p}_p \tag{4.12}$$

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 coordinate. Since our misalignment calibration is relative to the payload, there is no misalignment to the payload model. The estimated measurement is simply

$$\hat{\mathbf{P}} = T_{p,b} A(\hat{\mathbf{q}}) \mathbf{p}_p \tag{4.13}$$

5 Alignment Filter

The alignment filter is developed in a way similar to Ref. [9]. The continuous attitude kinematics equation including calibration parameters is given by

$$
\begin{aligned}
\frac{d}{dt}\begin{bmatrix} \boldsymbol{\delta\varrho} \\ \boldsymbol{\delta\beta} \\ \boldsymbol{\delta\kappa} \\ \boldsymbol{\delta\varsigma} \end{bmatrix} &= \begin{bmatrix} -\boldsymbol{\omega}\times & -\frac{1}{2}T_{b,g_0} & -\frac{1}{2}T_{b,g_0}\Omega_g & \mathbf{0} \\ & \mathbf{0} & & \\ & & \mathbf{0} & \\ & & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta\varrho} \\ \boldsymbol{\delta\beta} \\ \boldsymbol{\delta\kappa} \\ \boldsymbol{\delta\varsigma} \end{bmatrix} \\
&\quad \times \begin{bmatrix} -\frac{1}{2}T_{b,g_0}(I+M) & & & \\ & I & & \\ & & I & \\ & & & I \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_v \\ \boldsymbol{\eta}_u \\ \boldsymbol{\eta}_\kappa \\ \boldsymbol{\eta}_\varsigma \end{bmatrix}
\end{aligned} \tag{5.1}
$$

with the continuous process noise covariance is given by

$$
Q = \text{diag}\left[\sigma_v^2 I_{3\times3} \;\; \sigma_u^2 I_{3\times3} \;\; \sigma_\mu^2 I_{9\times9} \;\; \sigma_\varsigma^2 I_{3\times3}\right] \tag{5.2}
$$

The first square matrix on the right hand side of Eq. (5.1) is traditionally referred to as $F(t)$ and the second square matrix as $G(t)$.

When integrating Eq. (3.37) and Eq. (3.40), the necessary parameters are flipped in the

RTS smoother and integrated “forward” until the last measurement time. And the smoother estimated states and error covariance are then flipped again so that they correspond to time increment.

The observation sensitivity matrix of the vector star tracker is given by

$$H_{st} = \begin{bmatrix} 2\hat{T}_{b,st}[A(\hat{\mathbf{q}})\mathbf{p}_s] \times & \mathbf{0}_{3\times 9} & [\hat{T}_{b,st}A(\hat{\mathbf{q}})\mathbf{p}_s]\times \end{bmatrix} \tag{5.3}$$

where $\hat{T}_{b,st}$ is the estimated alignment matrix from body to true sensor coordinate.

The observation sensitivity matrix of the payload is given by

$$H_p = [2T_{b,p}[A(\hat{\mathbf{q}})\mathbf{p}_r] \times \quad \mathbf{0}_{3\times 9} \quad \mathbf{0}_{3\times 3}] \tag{5.4}$$

6 Simulation Results

The simulation is performed with greater flexibility and is divided into different subcomponents. Various simulation parameters could be changed easily to evaluate different scenarios that are more accurate representations of the real world applications.

Simulation 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: $\delta_{xz} = \delta_{xy} = \delta_{yz} = 0, \delta_{zx} = 400, \delta_{zy} = 300, \delta_{yz} = 200$ arc-s
- Gyro symmetric scale factor error: $\lambda_x = 500, \lambda_y = 500, \lambda_z = 500$ ppm
- Calibration maneuver: 0.09 des/s sinusoidal in each axis at (0.0006, 0.0007, 0.0008) Hz
- Gyro measurement is simulated with $\sigma_u = 1.3036 \times 10^{-3} \mu rad/sec^{3/2}$ and $\sigma_v = 1.45444 \mu rad/sec^{1/2}$

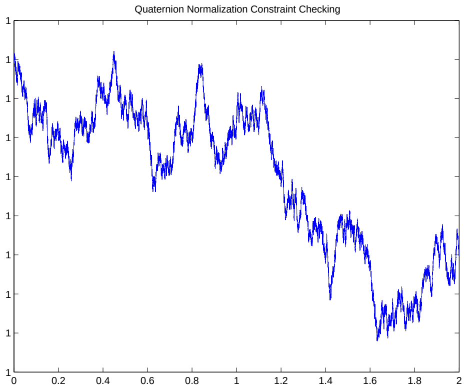

Figure 6.1: Quaternion Constraint Check

- 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

- Payload measurement accuracy: 0.5, 0.5, 0.5 arc-s

In our simulation, we assume no *a priori* knowledge of the calibration parameters.

The robustness of using quaternion as singular-free attitude representation is shown in Figure (6.1). Notice that the slight deviation from quaternion normalization constraint does not cause the divergence of both EKF and RTS filter. Furthermore, EKF attitude estimation algorithm employs small angle approximation of quaternions and it works well.

From our experience, sampling interval of 0.2 sec and below works well but not much larger than 0.2 sec. This is due to linearization nature of extended Kalman filter and the

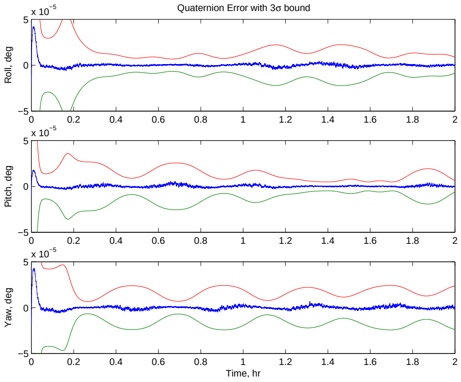

Figure 6.2: EKF Attitude Estimation Error with 3σ bound

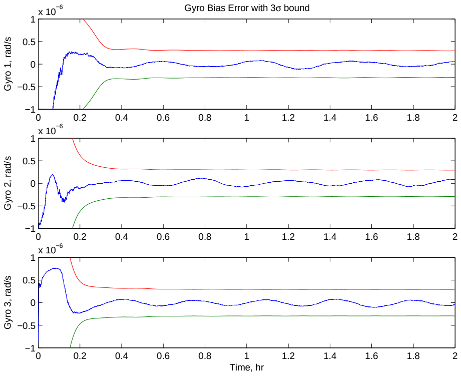

Figure 6.3: EKF Gyro Bias Estimation Error with 3σ bound

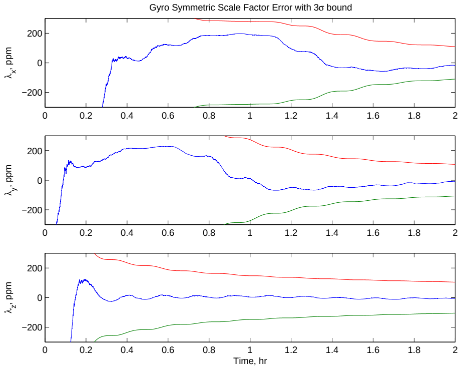

Figure 6.4: EKF Gyro Symmetric Scale Factor Error Estimation Error with 3σ bound

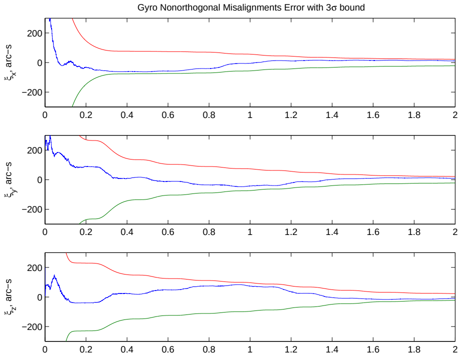

Figure 6.5: EKF Gyro Nonorthogonal Misalignment Estimation Error with 3σ bound

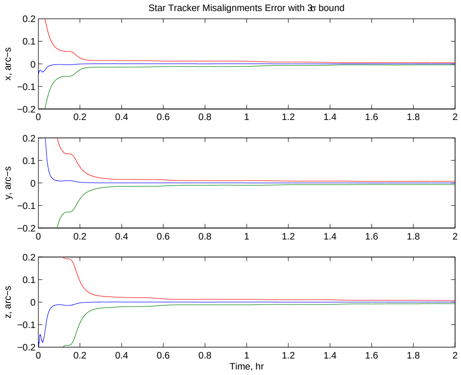

Figure 6.6: EKF Star Tracker Misalignment Estimation Error with 3σ bound

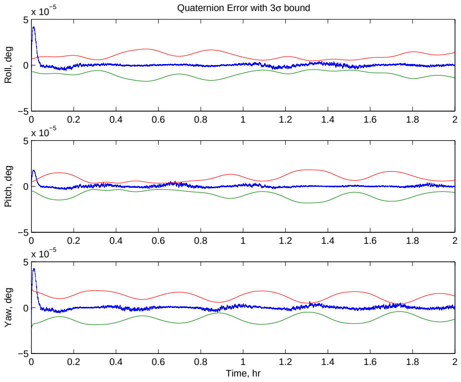

Figure 6.7: RTS Attitude Estimation Error with 3σ bound

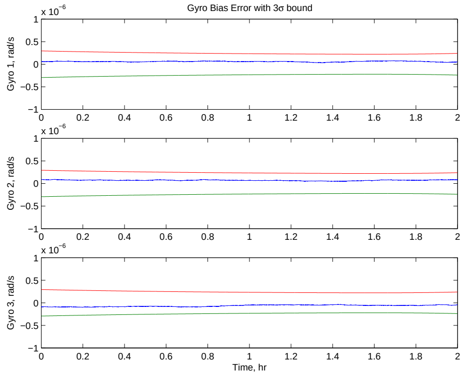

Figure 6.8: RTS Gyro Bias Estimation Error with 3σ bound

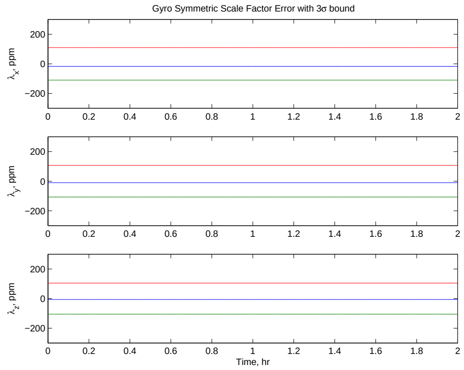

Figure 6.9: RTS Gyro Symmetric Scale Factor Error Estimation Error with 3σ bound

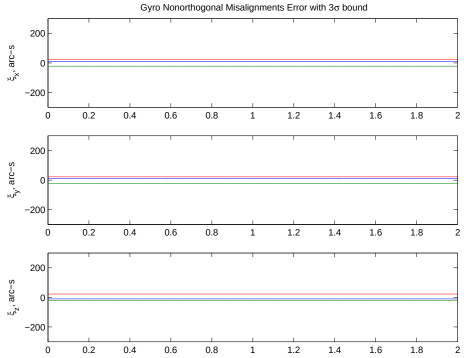

Figure 6.10: RTS Gyro Nonorthogonal Misalignment Estimation Error with 3σ bound

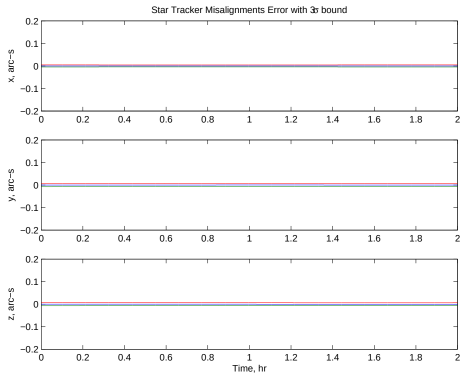

Figure 6.11: RTS Star Tracker Misalignment Estimation Error with 3σ bound

fast spacecraft attitude maneuver.

Figure (6.2) to (6.6) shows results from using EKF alone. These are also results from the forward pass of RTS batch estimation. Figure (6.7) to (6.11) shows results from using RTS filtering after the first EKF pass. The results after running the RTS clearly far superior than just the forward EKF alone. Attitude and gyro bias drift is “more” driven by white noise (in other words, they are more “time-varying”) and thus results in better estimation in RTS than EKF alone. Other calibration parameters do not drift as much and this the RTS results does not improve much from the final estimation of EKF alone.

7 Conclusion

Alignment Calibration with RTS smoother has been shown to be able to converge to lower covariance and more accurate estimation of alignment parameters. Thus, whenever highest accuracy is sought and real-time calibration is not an issue, batch filtering is preferred. It is recommended that batch filtering be performed at regular basis even real-time estimation of misalignments is implemented to maintain proper pointing accuracy.

Future works include trying the algorithm on attitude telemetry data from real spacecraft missions. This is particular useful to spacecraft with constant "irregular" nominal mission maneuvers. Chosen spacecraft attitude telemetry data are that of the Microwave Anisotropy Probe (MAP) and Wide-Field Infrared Explorer (WIRE). Another possible improvement to attitude estimation accuracy is the inclusion of asymmetry scale factor error into gyro calibration model.

Proper alignment calibration is of prime importance. It reduces fault detection and correction (FDC) that would cause inadvertent trip into Sun Acquisition or Safehold mode and disrupt nominal spacecraft mission.

8 Appendix

8.1 Matrix Inversion Lemma

This section will present the result derived by Sherman-Morrison-Woodbury [15]. Let

$$F = (A + BCD)^{-1} \tag{8.1}$$

where

$$\begin{aligned} F &= \text{an arbitrary } n \times n \text{ matrix} \\ A &= \text{an arbitrary } n \times n \text{ matrix} \\ B &= \text{an arbitrary } n \times m \text{ matrix} \\ C &= \text{an arbitrary } m \times m \text{ matrix} \\ D &= \text{an arbitrary } m \times n \text{ matrix} \end{aligned}$$

Assuming all the inverses of F,A,B,C and D exist, then

$$F = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1} \tag{8.2}$$

which can be easily proved by showing that $F^{-1}F = I$. From inspection, this lemma does not seem to help much in terms of computation requirements or simplifying the original equation. However, in most application, the subset, m, is usually smaller than n and Eq. (8.2) leads to a reduction in the computational requirement.

Bibliography

[1] AIAA Guidance, Navigation, and Control Conference and Exhibit, *The Microwave Anisotropy Probe Guidance, Navigation and Control Hardware Suite*, August 2002.

[2] AIAA Guidance, Navigation, and Control Conference and Exhibit, *Recent Flight Results of the TRMM Kalman Filter*, August 2002.

[3] American Institute of aeronautics and Astronautics, *Design, Implementation, Testing, And Flight Results of the TRMM Kalman Filter*, August 1998.

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

[5] Third International Symposium on Ground Data Systems for Space Mission Operations (SpaceOps 96), *Automation of Satellite Operations: Experiences and Future Directions at NASA GSFC*, Munich, Germany, September 1996.

[6] Shuster, M. D., Pitone, D. S., and Bierman, G. J., "Batch Estimation of Spacecraft Sensor Alignments I. Relative Alignment Estimation," *Journal of the Astronautical Sciences*, Vol. 39, No. 4, Oct.-Dec. 1991.

[7] Crassidis, J. L. and Markley, F. L., "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 11, 11.-11. 2003.

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

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

[10] 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.

[11] Gelb, A., editor, *Applied Optimal Estimation*, The MIT Press, Cambridge, MA, 1974.

[12] Crassidis, J. and Junkins, J., *Optimal Estimation of Dynamic Systems*, 2nd ed.

[13] Fraser, D., *A New Technique for the Optimal Smoothing of Data*, Ph.D. thesis, Massachusetts Institute of Technology, 1967.

[14] American Institute of Aeronautics and Astronautics, *Everything is Relative in System Alignment Calibration*, April 2000.

[15] Golub, G. H. and Van Loac, C. F., *Matrix Computations*, The John Hopkins University Press, Baltimore, MD, 2nd ed., 1989.