JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 4, July–August 2006

Attitude Estimation from Vector Observations Using Genetic-Algorithm-Embedded Quaternion Particle Filter

Yaakov Oshman* and Avishy Carmi†
Technion—Israel Institute of Technology, 32000 Haifa, Israel

A novel algorithm is presented for the estimation of spacecraft attitude quaternion from vector observations in gyro-equipped spacecraft. The new estimator is a particle filter that uses approximate numerical representation techniques for performing the otherwise exact time propagation and measurement update of potentially non-Gaussian probability density functions in inherently nonlinear systems. The new method can be applied using various kinds of vector observations. In this paper, the case of a low-Earth-orbit spacecraft, acquiring noisy geomagnetic field measurements via a three-axis magnetometer, is considered. A genetic algorithm is used to estimate the gyro bias parameters, avoiding the need to augment the particle filter's state and rendering the estimator computationally efficient. Contrary to conventional filters, which address the quaternion's unit norm constraint via special (mostly ad hoc) techniques, the new filter maintains this constraint naturally. An extensive simulation study is used to compare the new filter to three extended Kalman filters and to the unscented Kalman filter in Gaussian and non-Gaussian scenarios. The new algorithm is shown to be robust with respect to initial conditions and to possess a fast convergence rate. An evaluation of the Cramér–Rao estimation error lower bound demonstrates the filter's asymptotic statistical efficiency and optimality.

I. Introduction

THE use of a sequence of vector measurements for attitude determination has been intensively investigated over the last four decades. First proposed in 1965 by Wahba,[1] the problem is to estimate the attitude of a spacecraft based on a sequence of noisy vector observations, resolved in the body-fixed coordinate system and in a reference system. Body-fixed vector observations are typically obtained from onboard sensors, such as star trackers, sun sensors, or magnetometers. Corresponding reference observations are obtained by using an ephemeris routine (for a sun observation), or from orbit data and a magnetic field routine (for a magnetic field observation), or from a star catalog (for star observations).

Inertial reference systems typically utilize vector measurements in combination with strap-down gyros to estimate both the spacecraft attitude and the gyro drift rate biases. Several approaches have been proposed for the design of such systems, differing mainly in their choice of attitude representation method. The quaternion, a popular rotation specifier, was used in the framework of extended Kalman filtering (EKF) in Ref. 2 (using the so-called multiplicative approach) and in Ref. 3 (using the additive approach). The incorporation of the QUEST measurement model within a Kalman filter's measurement update stage was presented in Ref. 4. In Ref. 5, vector observations were used to estimate both the quaternion and the angular velocity of the spacecraft, in a gyroless attitude determination and control setting. The main advantage of using the quaternion representation is that it is not singular for any rotation. Moreover, its kinematic equation is linear, and the computation of the associated attitude matrix involves only algebraic expressions. However, the quaternion representation is not minimal because it is four dimensional. This leads to a normalization constraint that has to be addressed in filtering algorithms. Thus, assuming that the 4×4 quaternion estimation error covariance matrix must be singular, Ref. 2 surveys reduced-order algorithms that maintain this singularity. On the other hand, Ref. 3 assumes no such singularity, but incorporates a normalization stage within its EKF algorithm, that renders the resulting estimator strictly nonoptimal and increases its workload. In a recent paper,[6] an unscented Kalman filter (UKF) has been proposed for the estimation of the rotation quaternion. The UKF does possess a reported advantage over the EKF with regards to dealing with strongly nonlinear systems because it avoids the linearization associated with the EKF. However, because using the UKF directly with the quaternion attitude parameterization would also yield a nonunit norm quaternion estimate (after all, the UKF is still a Kalman filter), the authors of Ref. 6 chose to work with a generalized three-dimensional attitude representation, still using the quaternion for updates in order to maintain the normalization constraint. Also, as a Kalman filter mechanization, the UKF is also sensitive to the statistical distribution of the stochastic processes driving the dynamic model: non-Gaussian distributions guarantee nonoptimality of the estimates.

Consolidating and extending the results previously presented in conference papers,[7,8] this paper introduces a particle filter (PF) that sequentially and directly estimates the rotation quaternion from vector observations. Also known as sequential Monte Carlo (SMC) methods, particle filters refer to a set of algorithms implementing a recursive Bayesian model using simulation-based methods.[9] Avoiding the underlying assumptions of the Kalman filter, namely, that the state space is linear and Gaussian, these rather general and flexible methods enable solving for the posterior probability distributions of the unknown variables (on which all inference on these variables is based) within a Bayesian framework, exploiting the dramatic recent increase in computer power. PFs are not just smart implementations of the Kalman filter or its nonlinear variants/extensions; rather, they are entirely different algorithms that lead to entirely different solutions to the nonlinear, non-Gaussian filtering problem. Contrary to the Kalman filter extensions, the solutions obtained using PF algorithms are approximations to the optimal (in the Bayesian sense) solutions, which can be made arbitrarily close to the exact solutions by increasing the number of particles involved in the computation, thereby also increasing the computation workload.

In a related work,[10] a PF has been recently proposed for attitude estimation using star camera measurements. The modified Rodrigues parameters (MRP) are used for attitude representation, and the singularity associated with this representation is avoided by

Presented as Paper 2004-5340 at the AIAA Guidance, Navigation and Control Conference, Providence, RI, 16–19 August 2004; received 30 May 2005; revision received 13 December 2005; accepted for publication 14 December 2005.

*Professor, Department of Aerospace Engineering; yaakov.oshman@technion.ac.il. Associate Fellow AIAA.

†Doctoral Student, Department of Aerospace Engineering; avishy@aerodyne.technion.ac.il.

switching to an alternative set of MRPs or by using quaternions for time propagation. The authors do note that using the rotation quaternion is preferable because of its well-known attributes, but refrain from using it because of the need to maintain its unit norm constraint, and because of the quaternion ambiguity problem. The state of the suggested particle filter is six-dimensional, which accounts for the MRP and the three-axis gyro bias. Acquiring simultaneously up to five vector observations from the star camera once in every 2 s and using 2000 particles, this filter does yield better results than the UKF for large initial attitude errors. Nevertheless, the suggested filter's performance is unsatisfactory, and the authors propose that it be used as an initialization stage for some conventional Kalman filter variant.

In contradistinction to the algorithm of Ref. 10, this paper presents a computationally efficient particle filtering estimator that works directly with the quaternion, that is, its particles are all attitude quaternions. The quaternion ambiguity problem is eliminated via a special computation of the regularization intensity. Unlike KF variants, the quaternion PF (QPF) does not propagate and update the covariance, completely avoiding the question of singularity (although the covariance can be computed at any moment if needed for output monitoring purposes). Although a variety of vector observations can be utilized by the new filter to determine the spacecraft attitude, the problem considered here is that of estimating the attitude quaternion from three-axis magnetometer (TAM) measurements in gyro-equipped spacecraft. This attitude estimation problem is considerably more difficult than that of estimating the attitude from star camera readings. Because of the poor observability associated with TAM measurements, using an augmented, high-dimensional state vector (as used, e.g., in Ref. 10) would require a prohibitively large number of particles. This might, in turn, result in an impractical computational burden, which would limit the use of the estimator to offline applications. The estimator proposed herein alleviates the workload problem via decoupling the computation of the quaternion and gyro bias estimates. This allows for the implementation of the algorithm with a remarkably small number of particles (only 150), which renders the estimator fast and real-time implementable. To assess the statistical efficiency of the new filtering algorithm, the Cramér–Rao lower bound (CRLB) is computed for the problem addressed. A simulation study is then used to demonstrate the new algorithm's accuracy, rate of convergence, statistical efficiency, and robustness with respect to unknown initial conditions.

The remainder of this paper is organized as follows. The next section presents a brief introduction to SMC methods. Next, the mathematical model of the quaternion estimation problem is outlined. Section IV provides a detailed development of the quaternion particle filter for the ideal case where gyro biases are not present. In the next section this filter is augmented with a maximum likelihood estimator of the gyro biases, which is implemented via the use of a genetic algorithm. A derivation of the posterior CRLB for this problem is outlined in Sec. VI. Section VII presents the results of an extensive numerical simulation study that was carried out to assess the performance of the new algorithm and to compare it to existing, state-of-the-art quaternion estimators. Concluding remarks are offered in the last section.

II. Sequential Monte Carlo Methods

The optimal solution of the nonlinear estimation problem involves an accurate propagation of the optimal probability density function (PDF), namely, the conditional PDF of the state given the observation history. Because of the complex nature of nonlinear estimation problems, many estimation algorithms rely on various assumptions to ensure mathematical tractability. It is well known that the famous Kalman filter is the optimal estimator for linear Gaussian state-space models, but its performance is limited when the aforementioned assumptions do not hold. The optimal PDF admits a Bayesian recursion, which means that it is propagated in accordance with some prior distribution of the state and a likelihood function that relates the states to the incoming observations. In the case of linear Gaussian models, where the PDF can be characterized by its first two moments, the Bayesian approach yields the Kalman filter. In general, for nonlinear, non-Gaussian models, there is no explicit, closed-form solution. Several approximate methods have been proposed. These include the EKF, the Gaussian sum filter, and numerical integration over a state-space grid. Particle filters, or SMC methods,[9] refer to a set of algorithms implementing a recursive Bayesian model by simulation-based methods. This involves representing the required posterior PDF by a set of random samples with associated weights and deriving the estimates based on these samples. Unlike the other methods, SMC methods are very flexible, easy to implement and applicable in very general settings.

A. Bayesian Inference

It is assumed that the unobserved signal (i.e., the process) $\{x_k, k \in \mathbb{N}\}$ is an $\mathbb{R}^n$-valued Markov process with a given initial PDF p_{x_0}, which evolves according to a transition kernel $p_{x_k|x_{k-1}}$. Let the observations $\{y_k, k \in \mathbb{N}\}$ be an $\mathbb{R}^p$-valued stochastic process assumed to be conditionally independent given x_k, possessing the conditional PDF $p_{y_k|x_k}$. Let $\mathcal{X}^k \triangleq \{x_0, \ldots, x_k\}$ and $\mathcal{Y}_1^k \triangleq \{y_1, \ldots, y_k\}$ denote the process and observations time histories up to time k, respectively, and let $X^k \triangleq \{X_0, \ldots, X_k\}$ and $Y_1^k \triangleq \{Y_1, \ldots, Y_k\}$ be the realizations of $\mathcal{X}^k$ and $\mathcal{Y}_1^k$, respectively. In most nonlinear filtering problems the goal is to estimate the marginal PDF $p_{x_k|\mathcal{Y}_1^k}$ (filtering distribution) recursively in time. By applying the Chapman–Kolmogorov equation and Bayes formula, the following recursion is obtained for the filtering PDF:

$$p_{x_k|\mathcal{Y}_1^{k-1}}\left(X_k \middle| Y_1^{k-1}\right) = \int_{-\infty}^{+\infty} p_{x_k|x_{k-1}}(X_k|X_{k-1}) \times p_{x_{k-1}|\mathcal{Y}_1^{k-1}}\left(X_{k-1} \middle| Y_1^{k-1}\right) \mathrm{d}X_{k-1} \tag{1a}$$

$$p_{x_k|\mathcal{Y}_1^k}\left(X_k \middle| Y_1^k\right) = \frac{p_{y_k|x_k}(Y_k|X_k)}{\int_{-\infty}^{+\infty} p_{y_k|x_k}(Y_k|X_k) p_{x_k|\mathcal{Y}_1^{k-1}}\left(X_k \middle| Y_1^{k-1}\right) \mathrm{d}X_k} \times p_{x_k|\mathcal{Y}_1^{k-1}}\left(X_k \middle| Y_1^{k-1}\right) \tag{1b}$$

In most cases one cannot obtain the normalizing density $p_{y_k|\mathcal{Y}_1^{k-1}}$ and the marginals of the posterior $p_{\mathcal{X}^k|\mathcal{Y}_1^k}$. Thus, these expressions can rarely be used in a straightforward implementation. Instead, approximations should be utilized, using alternative methods. The key idea behind the PF method is to obtain a discrete support approximation for the distributions in Eq. (1) by means of simulation-based techniques.

B. Particle Approximation

Assume that it is possible to sample N independent and equally weighted random samples (called "particles") $\{X^k(i)\}_{i=1}^N$ from the posterior distribution. Then, it follows directly from the strong law of large numbers that, for any function f which is integrable with respect to $p_{\mathcal{X}^k|\mathcal{Y}_1^k}$ (Ref. 11),

$$\frac{1}{N}\sum_{i=1}^{N} f(X^k(i)) \to E\left[f(\mathcal{X}^k)\middle|Y_1^k\right] \tag{2}$$

where (here and in the sequel) the symbol $\to$ stands for almost sure convergence in N. Equation (2) means that the continuous posterior PDF $p_{\mathcal{X}^k|\mathcal{Y}_1^k}$ can be effectively approximated by its particles. Unfortunately, one cannot usually sample efficiently from the posterior. This problem can be alleviated by adopting a sampling technique called importance sampling. This technique consists of introducing an arbitrary so-called importance density, denoted by $\pi_{\mathcal{X}^k|\mathcal{Y}_1^k}$. If one can sample efficiently from $\pi_{\mathcal{X}^k|\mathcal{Y}_1^k}$, an approximation to the expectation in Eq. (2) is obtained as

$$\frac{1}{N}\sum_{i=1}^{N} w_k(i) f(X^k(i))$$

where

$$w_k(i) \triangleq \frac{p_{\mathcal{X}^k|\mathcal{Y}_1^k}\left(X^k(i)\middle|Y_1^k\right)}{\pi_{\mathcal{X}^k|\mathcal{Y}_1^k}\left(X^k(i)\middle|Y_1^k\right)} \tag{3}$$

is the importance weight of the ith particle. The selection of the importance function can consist of minimizing the variance of the estimate in Eq. (2). It can be shown that

$$\mathrm{Var}\left\{\frac{1}{N}\sum_{i=1}^{N} w_k(i) f(X^k(i))\right\}$$

is minimized by choosing $\pi(\cdot)$ such that $w_k(i)$ is nearly constant for all $i = 1, \ldots, N$.

Let

$$\tilde{w}_k(i) \triangleq \frac{w_k(i)}{\sum_{j=1}^{N} w_k(j)} \tag{4}$$

be the normalized importance weight of the ith particle, then it can be easily verified that

$$\sum_{i=1}^{N} \tilde{w}_k(i) f(X^k(i)) \to E\left[f(\mathcal{X}^k)\middle|Y_1^k\right] \tag{5}$$

The implementation of particle approximations using a finite number of samples raises some practical difficulties.[9] The so-called particle degeneracy problem refers to a phenomenon always encountered when using the importance sampling method, when after a few iterations of Eq. (1) all but usually one particle have zero weights. Over time, the distribution of the importance weights becomes more and more skewed, which consequently means that much computational effort is devoted to updating state trajectories whose contributions to the final estimate are close to zero. To alleviate the problem of degeneracy, a so-called selection/resampling step is introduced into the PF algorithm.

D. Practical Implementation

From a practitioner's point of view, the PF method consists of simply passing a finite set of samples, drawn from the prior distribution, through the process equation, and updating their likelihood weights based upon incoming observations. Thus, a single PF cycle is outlined in the following for the general state-space model.

Consider the following nonlinear discrete-time process and observation equations

$$x_k = f_k(x_{k-1}, u_{k-1}, v_k) \tag{6a}$$

$$y_k = g_k(x_k, n_k) \tag{6b}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k \in \mathbb{R}^p$ denote the state, the deterministic control, and the observation processes, respectively. In Eqs. (6), $v_k \in \mathbb{R}^l$ and $n_k \in \mathbb{R}^p$ are the mutually independent process and measurement white noises with known (not necessarily Gaussian) densities p_{v_k} and p_{n_k}, respectively. The initial state is independent of both noises and has a known PDF, denoted by p_{x_0}.

Let $\{X_{k-1}(i)\}_{i=1}^N$ be a particle approximation of the filtering distribution at time $k-1$, and let $\{w_{k-1}(i)\}_{i=1}^N$ be the weights associated with this approximation. Then, applying Eqs. (1) to $\{X_k(i)\}_{i=1}^N$ results in a new set of particles representing the filtering distribution at time k. First, the particle approximation of $p_{x_k|\mathcal{Y}_1^{k-1}}$ is obtained via Eq. (1a). In practice, Eq. (1a) is solved by passing the particles of time $k-1$ through the process equation along with the injection of artificially simulated process noise. (In most cases, though, the deliberate injection of artificial process noise during this stage is not necessary because of the ensuing regularization process, which amounts to naturally adding process noise.) Relaxing the requirement to evaluate the transition PDF, this method is applicable to irregular models (e.g., hybrid state-space models and models involving singular transition densities), as long as the transition distribution is specified in any of its forms.

Now, adopting the prior as the importance density in Eq. (3),

$$\pi_{\mathcal{X}^k|\mathcal{Y}_1^{k-1}} \triangleq p_{\mathcal{X}^k} \tag{7}$$

gives

$$w_k(i) \propto \frac{p_{y_k|x_k}(Y_k|X_k(i))\, p_{x_k|x_{k-1}}(X_k(i)|X_{k-1}(i))\, p_{\mathcal{X}^{k-1}|\mathcal{Y}_1^{k-1}}\left(X^{k-1}(i)\middle|Y_1^{k-1}\right)}{p_{x_k|x_{k-1}}(X_k(i)|X_{k-1}(i))\, p_{\mathcal{X}^{k-1}}(X^{k-1}(i))} = p_{y_k|x_k}(Y_k|X_k(i)) w_{k-1}(i) \tag{8}$$

which constitutes the measurement update stage.

Remark 1: The existence of a degeneracy situation can be easily detected. Hence, it is easy to provide a solution to this problem in a relatively straightforward manner.

C. Selection/Resampling

Selection/resampling consists of discarding state trajectories whose contributions to the final estimate are small and multiplying trajectories whose contributions are expected to be significant. This means regeneration of particles with large importance weights and eliminating those with small importance weights. The resampling procedure decreases the particle degeneracy algorithmically, but introduces some practical problems. During the resampling procedure, more likely particles are multiplied, so that the particle cloud is concentrated in regions of interest of the state space. This produces a new particle system in which several particles have the same location. Moreover, if the dynamical noise is small, the particle system ultimately concentrates in a single point in state space. This loss of diversity will eventually prevent the filter from correctly representing the posterior. One way of maintaining the particles' diversity is by injecting artificial process noise into the system. This technique is known as regularization, or roughening (see Ref. 12, p. 247).

III. Mathematical Model

In this section the problem of quaternion estimation from vector observations is mathematically defined.

A. Observation Model

Let $\boldsymbol{r}_k$ and $\boldsymbol{b}_k$ be a pair of corresponding vector measurements acquired at time k in the two Cartesian coordinate systems $\mathcal{R}$ and $\mathcal{B}$, respectively. Let A_k be the rotation matrix (also known as the attitude matrix or the direction cosine matrix) that brings the axes of $\mathcal{R}$ onto the axes of $\mathcal{B}$ at time k. In general, the reference vector $\boldsymbol{r}_k$ is known exactly, whereas the body vector $\boldsymbol{b}_k$ is measured. This results in the following attitude measurement model:

$$\boldsymbol{b}_k = A_k \boldsymbol{r}_k + \delta \boldsymbol{b}_k \tag{9}$$

where $\delta \boldsymbol{b}_k$ is the measurement noise process, with known PDF, denoted as $\delta \boldsymbol{b}_k \sim p_{\delta \boldsymbol{b}_k}$.

B. Quaternion Process Model

The discrete-time quaternion stochastic process satisfies the recurrence equation

$$\mathbf{q}_{k+1} = \Phi_k^o \mathbf{q}_k \tag{10}$$

where $\{\mathbf{q}_k\}_{k=1}^{\infty}$ denotes the quaternion of rotation from a given reference frame $\mathcal{R}$ onto the body frame $\mathcal{B}$ at times $k = 1, 2, \ldots \infty$, with some initial PDF $\mathbf{q}_0 \sim p_{\mathbf{q}_0}$. The quaternion process takes its values on the unit three-sphere $\mathbb{S}^3$ and is constructed from vector and scalar parts:

$$\mathbf{q}_k = \begin{bmatrix} \boldsymbol{\varrho}_k^T & \mathrm{q}_{4k} \end{bmatrix}^T \tag{11}$$

The orthogonal transition matrix Φ_k^o is expressed using $\boldsymbol{\omega}_k^o = [\omega_{1k}^o \;\; \omega_{2k}^o \;\; \omega_{3k}^o]^T$, the true angular velocity of $\mathcal{B}$ with respect to $\mathcal{R}$, resolved in $\mathcal{B}$. Assuming that $\boldsymbol{\omega}_k^o$ is constant during the sampling time interval Δt yields

$$\Phi_k^o \triangleq \Phi\left(\boldsymbol{\omega}_k^o\right) = \exp\left(\frac{1}{2}\begin{bmatrix} -\left[\boldsymbol{\omega}_k^o \times\right] & \boldsymbol{\omega}_k^o \\ -\boldsymbol{\omega}_k^{o^T} & 0 \end{bmatrix}\Delta t\right) \tag{12}$$

where $[\boldsymbol{\omega}_k^o \times]$ denotes the cross-product matrix associated with the vector $\boldsymbol{\omega}_k^o$.

In practice, the true angular velocity vector $\boldsymbol{\omega}_k^o$ is not known; rather, it is measured or estimated. Let $\{\boldsymbol{\omega}_k\}_{k=1}^{\infty}$ be the measured angular velocity stochastic process. Using $\boldsymbol{\omega}_k$ instead of $\boldsymbol{\omega}_k^o$ in Eq. (12) yields the following quaternion process equation:

$$\mathbf{q}_{k+1} = \Phi(\boldsymbol{\omega}_k)\mathbf{q}_k \tag{13}$$

where the process noise is incorporated through the transition matrix.

The relation between the process and observations is established by expressing the attitude matrix as a quadratic function of $\mathbf{q}$, that is,

$$A = A(\mathbf{q}) = [(\mathrm{q}_4)^2 - \boldsymbol{\varrho}^T\boldsymbol{\varrho}]I_{3\times 3} + 2\boldsymbol{\varrho}\boldsymbol{\varrho}^T - 2\mathrm{q}_4[\boldsymbol{\varrho}\times] \tag{14}$$

C. Rate Sensor Measurement Model

When the angular rate is measured, the characterization of the driving process noise depends upon the rate sensor. The most common angular rate sensor onboard spacecraft is the gyro triad. For this sensor, a widely used model is given by[2]

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}_k^o + \boldsymbol{\eta}_k + \boldsymbol{\epsilon}_k \tag{15}$$

where $\boldsymbol{\omega}_k$ denotes the measured angular velocity vector and $\boldsymbol{\epsilon}_k \sim p_{\epsilon_k}$ and $\boldsymbol{\eta}_k \sim p_{\eta_k}$ are the gyros' measurement white noise and bias vectors with their given PDFs, respectively. Usually the bias vector is modeled as a random-walk process, that is,

$$\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_k + \boldsymbol{\zeta}_k \tag{16}$$

where $\{\boldsymbol{\zeta}_k\}_{k=1}^{\infty}$ is a stationary zero-mean, white-noise process with covariance Q_ζ. Typically Q_ζ is very small (e.g., with entries on the order of $10^{-7}\ \mu\mathrm{rad}^2/\mathrm{s}^2$.)

IV. Quaternion Particle Filter

This section presents the novel QPF. The algorithm estimates the quaternion from pairs of vector observations. Within this particle filter, each particle is a unit norm quaternion, so that the norm constraint is inherently preserved. In the first stage, presented in this section, the QPF is derived assuming unbiased gyro measurements. Then, in the next section, the algorithm is expanded to incorporate gyro biases.

A. QPF Initialization

Large initial attitude errors require a large number of quaternion particles, at least until the zones of high likelihood are populated. A simple initialization procedure that demands a significantly smaller number of particles is used in this work. The idea is based on the fact that the first vector observation defines a quaternion of rotation up to one degree of freedom. This degree of freedom is used to generate the initial set of particles from the first observation only, $Y_0 = [\boldsymbol{b}_0; \boldsymbol{r}_0]$. The technique is detailed in Appendix B.

B. Measurement Update

Denote by $Y_1^k = \{[\boldsymbol{b}_1; \boldsymbol{r}_1], \ldots, [\boldsymbol{b}_k; \boldsymbol{r}_k]\}$ a set of measurements constructed from pairs of vector observations up to time k. Given a realization $\mathbf{q}_k$ of the quaternion $\mathbf{q}_k$ at time k, the measurement $Y_k = [\boldsymbol{b}_k; \boldsymbol{r}_k]$ is statistically independent of past observations. The likelihood of the measurement Y_k associated with a given quaternion is

$$p_{y_k|\mathbf{q}_k}\left(Y_k\middle|\boldsymbol{q}_k\right) = p_{\delta\mathbf{b}_k}(\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k) \tag{17}$$

When the measurement Y_k becomes available, the filtering PDF at time k satisfies

$$p_{\mathbf{q}_k|\mathcal{Y}_1^k}\left(\boldsymbol{q}_k\middle|Y_1^k\right) \propto p_{y_k|\mathbf{q}_k}(Y_k|\boldsymbol{q}_k)\, p_{\mathbf{q}_k|\mathcal{Y}_1^{k-1}}\left(\boldsymbol{q}_k\middle|Y_1^{k-1}\right) \tag{18}$$

Now let $\{\boldsymbol{q}_{k-1}(i)\}_{i=1}^N$ and $\{\tilde{w}_{k-1}(i)\}_{i=1}^N$ denote N independent unit quaternion samples from the filtering distribution at time $k-1$ and their associated weights, respectively. Setting the importance distribution to be the prior PDF yields the importance weights as

$$w_k(i) = p_{y_k|\mathbf{q}_k}(Y_k|\boldsymbol{q}_k(i))\tilde{w}_{k-1}(i) \tag{19}$$

Equation (19) is referred to as the update stage. Still, in accordance with Eqs. (18) and (19), one has to incorporate an evolution stage, as the samples need to represent the prior PDF.

C. Filtered Quaternion

At time k, N weighted unit quaternion samples are available. The optimal quaternion estimate can be computed in several ways depending on the objective. Two alternative methods, using the maximum a posteriori probability (MAP) approach and the minimum mean square error (MMSE) approach, are presented next.

MAP Approach

The simplest approach for obtaining the filtered quaternion is the MAP approach. In this method the filtered quaternion is obtained by solving the following maximization problem:

$$\hat{\boldsymbol{q}}_k = \arg\max_{\boldsymbol{q}\in\{\boldsymbol{q}_k(i)\}_{i=1}^N} p_{\mathbf{q}_k|\mathcal{Y}_1^k}\left(\boldsymbol{q}\middle|Y_1^k\right) \tag{20}$$

Given the weighted particle set $\{\boldsymbol{q}_k(i), \tilde{w}_k(i)\}_{i=1}^N$, the solution to this problem is easily obtained as the quaternion particle corresponding to the largest importance weight, that is,

$$\hat{\boldsymbol{q}}_k = \boldsymbol{q}_k\left(\arg\max_{1\le i\le N}\tilde{w}_k(i)\right) \tag{21}$$

Notice, that the MAP solution does not involve any manipulation of the particle stock.

Although the MAP solution is very easy to implement, it usually yields noisier estimates than the MMSE approach, to be presented next. This is because of the effect of the resampling procedure. Several smoothing methods to overcome this problem are suggested in Ref. 12 (pp. 296–299). (Smoother MAP estimates can be obtained by reducing the effective sample size threshold N_{th}, consequently demanding less resampling procedures to take place, as explained in Sec. IV.D.)

MMSE Approach

To obtain the filtered quaternion via the MMSE approach, the following minimization problem is solved:

$$\min_{\hat{A}_k}\sum_{i=1}^N \tilde{w}_k(i)\|A_k(\boldsymbol{q}_k(i)) - \hat{A}_k\|_F^2 \quad \text{subject to} \quad \hat{A}_k^T\hat{A}_k = I_{3\times 3} \tag{22}$$

where $\hat{A}_k$ denotes the orthogonal attitude matrix associated with the filtered quaternion, and $\|\cdot\|_F$ is the Frobenius norm. Because $\|C\|_F^2 = \mathrm{tr}(C^TC)$ for any $C \in \mathbb{R}^{m\times m}$, it follows that if $\hat{A}_k$ is orthogonal then the constrained minimization problem (22) is equivalent

to the unconstrained minimization problem

$$\min_{\hat{A}_k} \sum_{i=1}^{N} \tilde{w}_k(i)\Big(\mathrm{tr}\big[A_k(\boldsymbol{q}_k(i))^T A_k(\boldsymbol{q}_k(i))\big] + 3 - 2\,\mathrm{tr}\big[\hat{A}_k^T A_k(\boldsymbol{q}_k(i))\big]\Big) \tag{23}$$

which leads to the maximization problem

$$\max_{\hat{A}_k} \sum_{i=1}^{N} \tilde{w}_k(i)\,\mathrm{tr}\big[\hat{A}_k^T A_k(\boldsymbol{q}_k(i))\big] = \max_{\hat{A}_k} \mathrm{tr}\left[\hat{A}_k^T \sum_{i=1}^{N} \tilde{w}_k(i) A_k(\boldsymbol{q}_k(i))\right] \tag{24}$$

This maximization problem is equivalent to the orthogonal Procrustes problem (see Ref. 13, p. 601). Letting

$$B_k \triangleq \sum_{i=1}^{N} \tilde{w}_k(i) A_k(\boldsymbol{q}_k(i)) \tag{25}$$

and denoting the singular value decomposition of B_k by

$$B_k = U_k \Sigma_k V_k^T \tag{26}$$

where U_k and V_k are the orthogonal singular vector matrices and Σ_k is the singular value matrix of B_k, the solution of the Procrustes problem is obtained as

$$\hat{A}_k = U_k V_k^T \tag{27}$$

The filtered quaternion is then obtained from $\hat{A}_k$.

An equivalent approach for minimizing the cost in Eq. (22) consists of writing Eq. (24) in quadratic form as follows:

$$\max_{\hat{A}_k} \mathrm{tr}\left[\hat{A}_k^T \sum_{i=1}^{N} \tilde{w}_k(i) A_k(\boldsymbol{q}_k(i))\right] = \max_{\hat{\boldsymbol{q}}_k} \hat{\boldsymbol{q}}_k^T K \hat{\boldsymbol{q}}_k \tag{28}$$

with

$$K \triangleq \begin{bmatrix} B_k + B_k^T - I_{3\times 3}\,\mathrm{tr}(B_k) & z \\ z & \mathrm{tr}(B_k) \end{bmatrix}, \qquad z \triangleq \begin{bmatrix} B_{k_{32}} - B_{k_{23}} \\ B_{k_{13}} - B_{k_{31}} \\ B_{k_{21}} - B_{k_{12}} \end{bmatrix} \tag{29}$$

Now, similarly to Davenport's well-known q method (see Ref. 14 and Sec. 12.2.3 of Ref. 15), the quaternion that solves the maximization problem of Eq. (28) is the normalized eigenvector corresponding to the largest eigenvalue of K.

D. Particle Maintenance

To avoid particle degeneracy, a resampling procedure is implemented. The measure of degeneracy adopted here is the effective sample size. Introduced by Kong et al.,[16] this criterion is defined using the variance of the importance weights. The effective sample size N_{eff} is defined as

$$N_{\mathrm{eff}} \triangleq N/(1 + \mathrm{Var}\, w_k) = N/E_\pi\big[w_k^2\big] < N \tag{30}$$

An empirical estimate of N_{eff} is given by[9]

$$\hat{N}_{\mathrm{eff}} = 1\Big/\sum_{i=1}^{N} \tilde{w}_k(i)^2 \tag{31}$$

The resampling procedure is used whenever $\hat{N}_{\mathrm{eff}}$ becomes less than a predetermined threshold N_{th}. The new set of samples is generated by resampling each particle $\boldsymbol{q}_k(i)$ with probability $\tilde{w}_k(i)$. This consists of multiplying each sample according to its associated normalized weight. The number of offspring for each sample is computed as

$$N_k(i) = \mathrm{int}(N\tilde{w}_k(i)) \qquad i = 1, \ldots, N \tag{32}$$

where $\mathrm{int}(x)$ denotes the integer nearest to x for any $x \in \mathbb{R}$. To compensate for the loss of particle diversity, an artificial perturbation scheme based on regularization is introduced into the algorithm. The adopted measure of regularization intensity is introduced next.

Regularization Intensity

The rotation quaternion is defined on the unit three-sphere. Thus, to obtain the regularized set of quaternion particles, a measure for regularization intensity needs to be defined that allows for producing particles with unit norm. In this work this measure is based on computing the second moment of the quaternion multiplicative error, defined as

$$\delta\mathbf{q}_k \triangleq \mathbf{q}_k \otimes \hat{\mathbf{q}}_k^{-1} \tag{33}$$

where $\otimes$ denotes the usual quaternion product operator. The conditional second moment of $\delta\mathbf{q}_k$ is

$$P_{q_k} = E\Big[\big(\mathbf{q}_k \otimes \hat{\mathbf{q}}_k^{-1}\big)\big(\mathbf{q}_k \otimes \hat{\mathbf{q}}_k^{-1}\big)^T \big| \mathcal{Y}_1^k\Big] \tag{34}$$

An empirical estimate of the conditional second moment is obtained using weighted unit quaternion samples, as

$$\hat{P}_{\mathbf{q}_k} = \sum_{i=1}^{N} \tilde{w}_k(i)\big[\boldsymbol{q}_k(i) \otimes \hat{\boldsymbol{q}}_k^{-1}\big]\big[\boldsymbol{q}_k(i) \otimes \hat{\boldsymbol{q}}_k^{-1}\big]^T \tag{35}$$

Remark 2: The regularization intensity measure $\hat{P}_{\mathbf{q}_k}$ is invariant under change of the quaternion particles sign. Thus, $+\boldsymbol{q}_k(i)$ and $-\boldsymbol{q}_k(i)$, which represent the same attitude, are treated equivalently. This completely avoids the quaternion ambiguity problem, which might arise when computing a covariance-based regularization intensity measure.[10]

Regularization Scheme

Let $\{\boldsymbol{q}_k(i)\}_{i=1}^N$ with $\boldsymbol{q}_k(i) = [\boldsymbol{\rho}_k^T(i)\ \ q_{4k}(i)]^T$ be the stock of N unit quaternion samples at time k. Using Eq. (32), the number of offspring for each particle $N_k(i)$ is determined. The quaternion offspring are then computed in the following manner.

Sample a set of vectors from a three-dimensional zero-mean, unit covariance Gaussian (or some other) kernel denoted by $\mathcal{K}$, to get

$$\delta\boldsymbol{\xi}_0(j) \sim \mathcal{K}(0, I_{3\times 3}) \qquad j = 1, 2, \ldots, N_k(i)/2 \tag{36}$$

When $N_k(i)$ is odd, one of the values $\delta\boldsymbol{\xi}_0(j)$ is taken to be $0_{3\times 1}$, thus ensuring a symmetric set around $\boldsymbol{q}_k(i)$. The next step consists of rescaling and rotating the three-dimensional space according to the regularization intensity measure previously obtained. Thus, each vector is rescaled and rotated according to

$$\delta\boldsymbol{\theta}_k(j) = \pm\sqrt{h\hat{P}_{\theta_k}}\,\delta\boldsymbol{\xi}_0(j) \tag{37}$$

where h is some predetermined bandwidth and $\sqrt{\hat{P}_{\theta_k}}$ is the matrix square root of the 3×3 matrix obtained by eliminating the last row and last column of $\hat{P}_{q_k}$. In this work, the bandwidth h is set as suggested in Ref. 12 (p. 253), that is,

$$h = \{4/[N(n+2)]\}^{1/(n+4)} \tag{38}$$

with $n = 4$, corresponding to the quaternion's dimension. Notice that Eq. (37) produces a symmetric set of $N_k(i)$ vectors $\{\delta\boldsymbol{\theta}_k(j)\}_{j=1}^{N_k(i)}$. The quaternion offspring of the ith particle $\boldsymbol{q}_k(i)$ are then computed as

$$\bar{\boldsymbol{q}}_k(j) = \delta\boldsymbol{q}_k(j) \otimes \boldsymbol{q}_k(i) \qquad j = 1, \ldots, N_k(i) \tag{39a}$$

where

$$\delta \boldsymbol{q}_k(j) = \frac{\begin{bmatrix}\delta\boldsymbol{\theta}_k(j)^T & 1\end{bmatrix}^T}{\left\|\begin{bmatrix}\delta\boldsymbol{\theta}_k(j)^T & 1\end{bmatrix}\right\|_2} \tag{39b}$$

After obtaining the quaternion offspring, each one should be properly weighted. This second-stage weighting is crucial for the overall performance of the filter. The resampling procedure is, in fact, an external interference that injects random particles which have no past trajectories. A proper weighting of the offspring will reduce the effect of this contamination, whereas improper weighting of the offspring will degrade the posterior representation and, in some extreme cases, might cause divergence. Reweighting can be carried out based upon regularization, so that the new particles are treated as if they were sampled from a continuous PDF. Another idea, which tends to give better results, is to reweight the offspring proportionally to their likelihood. Thus, the second-stage importance weights are computed as

$$\bar{w}_k(j) = (1/c)\, p_{y_k|q_k}(Y_k|\bar{\boldsymbol{q}}_k(j)) \qquad j = 1, \ldots, N_k(i) \tag{40}$$

where the normalizing constant c is selected based on numerical considerations. A particle stock with skewed importance weights can be improved in the next time step by properly choosing c. In this work the value $c = p_{y_k|\mathbf{q}_k}(Y_k|\hat{\boldsymbol{q}}_k)$ is used, where $\hat{\boldsymbol{q}}_k$ denotes the filtered quaternion at time k.

E. Particle Evolution

Passing the unit quaternion samples at time $k-1$ through the process equation results in a new set of samples. This is almost equivalent to applying Eq. (1a) to the samples, that is,

$$\mathbf{p}_{q_k|\mathcal{Y}_1^{k-1}}\left(\boldsymbol{q}_k \middle| Y_1^{k-1}\right) = \int_{-\infty}^{+\infty} p_{\mathbf{q}_k|\mathbf{q}_{k-1}}(\boldsymbol{q}_k|\boldsymbol{q}_{k-1})\, p_{\mathbf{q}_{k-1}|\mathcal{Y}_1^{k-1}}\left(\boldsymbol{q}_{k-1} \middle| Y_1^{k-1}\right) \mathrm{d}\boldsymbol{q}_{k-1} \tag{41}$$

The slight difference is caused by the process noise distribution, which forms the transition kernel $p_{\mathbf{q}_k|\mathbf{q}_{k-1}}$. In the case of a relatively low-intensity process noise, such as the noise characterizing the quaternion evolution model, the new quaternion samples thus obtained represent $p_{\mathbf{q}_k|\mathcal{Y}_1^{k-1}}$ quite adequately. In other cases, the injection of an additional, artificial noise might be required.

V. Incorporating Gyro Biases

Properly accounting for biased gyro measurements via state augmentation results in a higher dimensional model: a seven-state model is required to account for the dynamics of both the quaternion and the three-component bias process. Whereas state augmentation is the common solution in Kalman filtering-based algorithms, straightforward implementation of a particle filtering algorithm for a high dimensional state would require a prohibitively large number of samples to properly represent the posterior PDF, as shown in Ref. 17 using the notion of particle survival rate.

To alleviate the potential computation load problem, the technique proposed herein is based on interlacing the PF with a maximum-likelihood (ML) estimator. In the proposed procedure, the particle filter algorithm is used for the representation of $p_{\mathbf{q}_k|\eta_k,\mathcal{Y}_1^k}$ instead of $p_{\mathbf{q}_k,\eta_k|\mathcal{Y}_1^k}$, thus keeping the dimension of the state low, whereas $\boldsymbol{\eta}_k$ is estimated via an external ML estimator assuming the knowledge of $\hat{\boldsymbol{q}}_k$. It can be easily shown that implementing the PF algorithm for $p_{\mathbf{q}_k|\eta_k,\mathcal{Y}_1^k}$ does not require any modifications in the computation of the importance weights as a result of the formulation of the observation model in Eq. (9), which leads to the following equivalence of PDFs:

$$p_{y_k|\mathbf{q}_k} = p_{y_k|\mathbf{q}_k,\eta_k} \tag{42}$$

Let $\mathcal{Y}_{k_1}^{k_2} \triangleq \{y_{k_1}, \ldots, y_{k_2}\}$ be a batch of measurements corresponding to the time interval $[k_1, k_2]$, with the corresponding realization $Y_{k_1}^{k_2} \triangleq \{Y_{k_1}, \ldots, Y_{k_2}\}$. Based on the previously stated fact that Q_ζ is typically very small, the following proposition is central to the development of the gyro bias ML estimator, as it facilitates the implementation of a computationally efficient ML estimator based on maximizing the likelihood function $\mathcal{L}(\boldsymbol{\eta}_{k_1}|Y_{k_1}^{k_2}) \triangleq p_{\mathcal{Y}_{k_1}^{k_2}|\boldsymbol{\eta}_{k_1}}(Y_{k_1}^{k_2}|\boldsymbol{\eta}_{k_1})$ over a finite time interval $[k_1, k_2]$.

Proposition V.1: The gyro bias process can be approximated by a random process having piecewise-constant sample functions.

Proof: Let k be any point interior to a time interval $[k_1, k_2]$. Using the dynamics of the bias process [Eq. (16)], and the fact that $\{\boldsymbol{\zeta}_k\}_{k=1}^{\infty}$ is white, gives

$$E\left\|\boldsymbol{\eta}_k - \boldsymbol{\eta}_{k_1}\right\|_2^2 = E\left\|\sum_{i=k_1+1}^{k} \boldsymbol{\zeta}_i\right\|_2^2 = \sum_{i=k_1+1}^{k} E\|\boldsymbol{\zeta}_i\|_2^2 = (k-k_1)\,\mathrm{tr}(Q_\zeta) \le (k_2-k_1)\,\mathrm{tr}(Q_\zeta) \tag{43}$$

Using the Markov inequality, it then follows from Eq. (43) that, for any $\varepsilon > 0$,

$$\mathrm{Prob}\left(\left\|\boldsymbol{\eta}_k - \boldsymbol{\eta}_{k_1}\right\|_2^2 \ge \varepsilon\right) \le \frac{E\left\|\boldsymbol{\eta}_k - \boldsymbol{\eta}_{k_1}\right\|_2^2}{\varepsilon} \le \frac{(k_2-k_1)}{\varepsilon}\,\mathrm{tr}(Q_\zeta) \qquad k \in [k_1, k_2] \tag{44}$$

Because Q_ζ is typically very small, an interval limit $k_2(\varepsilon) > k_1$ can be determined for any practically small ε, such that

$$\boldsymbol{\eta}_k \approx \boldsymbol{\eta}_{k_1} \qquad \forall k \in [k_1, k_2] \tag{45}$$

□

A numerical demonstration of the validity of proposition V.1, using realistic values, is given in Sec. VII.

In the remainder of this section, a technique is presented for maximizing the likelihood $\mathcal{L}(\boldsymbol{\eta}_{k_1}|Y_{k_1}^{k_2})$ by applying a genetic search algorithm. The resulting interlaced filtering algorithm is termed GA-QPF.

A. Approximation of the Likelihood

The approximation (45) allows writing the likelihood function as

$$\mathcal{L}\left(\boldsymbol{\eta}_{k_1}|Y_{k_1}^{k_2}\right) = \prod_{j=k_1}^{k_2} p_{y_j|\mathcal{Y}_{k_1}^{j-1},\boldsymbol{\eta}_{k_1}}\left(Y_j \middle| Y_{k_1}^{j-1}, \boldsymbol{\eta}_{k_1}\right) \tag{46}$$

where

$$p_{y_{k_1}|\mathcal{Y}_{k_1}^{k_1-1},\boldsymbol{\eta}_{k_1}}\left(Y_{k_1} \middle| Y_{k_1}^{k_1-1}, \boldsymbol{\eta}_{k_1}\right) \triangleq p_{y_{k_1}|\boldsymbol{\eta}_{k_1}}\left(Y_{k_1} \middle| \boldsymbol{\eta}_{k_1}\right) \tag{47}$$

Equation (46) yields

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{\eta}_{k_1} \middle| Y_{k_1}^{k_2}\right) &= \prod_{j=k_1}^{k_2} \int_{-\infty}^{+\infty} p_{y_j|\mathbf{q}_j,\mathcal{Y}_{k_1}^{j-1},\boldsymbol{\eta}_{k_1}}\left(Y_j \middle| \boldsymbol{q}_j, Y_{k_1}^{j-1}, \boldsymbol{\eta}_{k_1}\right) \\ &\quad \times p_{\mathbf{q}_j|\mathcal{Y}_{k_1}^{j-1},\boldsymbol{\eta}_{k_1}}\left(\boldsymbol{q}_j \middle| Y_{k_1}^{j-1}, \boldsymbol{\eta}_{k_1}\right) \mathrm{d}\boldsymbol{q}_j \\ &= \prod_{j=k_1}^{k_2} \int_{-\infty}^{+\infty} p_{y_j|\mathbf{q}_j}\left(Y_j \middle| \boldsymbol{q}_j\right) \\ &\quad \times p_{\mathbf{q}_j|\mathcal{Y}_{k_1}^{j-1},\boldsymbol{\eta}_{k_1}}\left(\boldsymbol{q}_j \middle| Y_{k_1}^{j-1}, \boldsymbol{\eta}_{k_1}\right) \mathrm{d}\boldsymbol{q}_j \\ &= \prod_{j=k_1}^{k_2} E_{\mathbf{q}_j|\mathcal{Y}_{k_1}^{j-1},\boldsymbol{\eta}_{k_1}}\left[p_{y_j|q_j}(Y_j|\boldsymbol{q}_j)|Y_{k_1}^{j-1}, \boldsymbol{\eta}_{k_1}\right]\end{aligned} \tag{48}$$

An empirical approximation of the likelihood in Eq. (48) can be obtained as

$$\hat{\mathcal{L}}\left(\boldsymbol{\eta}_{k_1}\middle|Y_{k_1}^{k_2}\right)=\prod_{j=k_1}^{k_2}\sum_{i=1}^{N}\tilde{w}_j(i)\,p_{y_j|\mathbf{q}_j}\left(Y_j|\Phi\left(\boldsymbol{\omega}_j-\boldsymbol{\eta}_{k_1}\right)\boldsymbol{q}_{j-1}(i)\right) \tag{49}$$

where $\Phi(\boldsymbol{\omega}_j-\boldsymbol{\eta}_{k_1})$ is the quaternion process transition matrix, evaluated at $\boldsymbol{\eta}_{k_1}$. Equation (49) is very computationally intensive because it has to be evaluated over the entire particle set for a given $\boldsymbol{\eta}_{k_1}$. This computational effort can be dramatically reduced by further assuming that the quaternion particles are concentrated near the filtered quaternion. In this case the likelihood can be approximated by

$$\begin{aligned}\hat{\mathcal{L}}\left(\boldsymbol{\eta}_{k_1}\middle|Y_{k_1}^{k_2}\right)&\approx\prod_{j=k_1}^{k_2}p_{y_j|\mathbf{q}_j}\left(Y_j|\Phi\left(\boldsymbol{\omega}_j-\boldsymbol{\eta}_{k_1}\right)\hat{\boldsymbol{q}}_{j-1}\right)\\&=\prod_{j=k_1}^{k_2}p_{\delta\boldsymbol{b}_k}\left(\boldsymbol{b}_j-A\left(\Phi\left(\boldsymbol{\omega}_j-\boldsymbol{\eta}_{k_1}\right)\hat{\boldsymbol{q}}_{j-1}\right)\boldsymbol{r}_j\right)\end{aligned} \tag{50}$$

Equation (50) requires the evaluation of the likelihood only once per time interval for a given realization of $\boldsymbol{\eta}_{k_1}$, thus enabling a real-time bias estimator to be implemented. The ML estimate of $\boldsymbol{\eta}_{k_1}$ maximizes $\hat{\mathcal{L}}(\boldsymbol{\eta}_{k_1}|Y_{k_1}^{k_2})$. Computing this estimate using a genetic search algorithm is presented next.

B. Genetic Search Algorithms

Sophisticated but simple, genetic algorithms (GA) are search methods based upon the mechanization of natural selection and natural genetics (see Ref. 18, Chap. 1). In much the same manner that the particle filter maintains the posterior, namely, by eliminating particles with zero weights and producing offspring by resampling, GAs combine survival of the fittest among string structures (called chromosomes) with randomized information exchange.

The common GA uses a coding of the parameter set, usually via a string, as opposed to working with the parameters themselves. This enables the algorithm to exchange information between string elements in the population. Given an initial population of chromosomes and applying a single GA iteration results in an improved generation of chromosomes by means of likelihood. A simple GA consists of the following stages: 1) reproduction, 2) crossover, and 3) mutation. Reproduction is the process by which individual chromosomes are being reproduced according to their fitness function (or likelihood function). Thus, more likely chromosomes will have higher probability of contributing offspring in the next generation. Crossover is the process of exchanging genetic information between two reproduced chromosomes. A simple crossover is carried out by selecting two chromosomes randomly and swapping all of their characters from a randomly selected position to the total string length. Even though reproduction and crossover improve the population, they can become overzealous and lose potentially important genetic information. Mutation protects against such an irrecoverable loss by simply altering a character with small probability every once in a while.

C. Implementation of the GA Algorithm

The QPF is modified to process gyro biases by introducing a simple GA into the algorithm. The proposed GA maximizes the likelihood in Eq. (50) sequentially in time and includes some modifications in order to cope with the varying nature of the biases.

Coding Scheme

The coding scheme used by the GA is the following. Denote a bias realization by $\boldsymbol{\Upsilon}=[\Upsilon^1\ \ \Upsilon^2\ \ \Upsilon^3]^T$, where Υ^j is the jth component of the vector $\boldsymbol{\Upsilon}$. The corresponding chromosome of length $l+1$ of the jth component, denoted by $\mathcal{B}(\Upsilon^j)$, is a string of length $l+1$ containing the binary coding of $|C\Upsilon^j|$ and a sign bit. The predetermined constant C is used for fitting the largest value of Υ^j into the maximal length of the string. Having the initial set of N_η bias parameters $\{\boldsymbol{\Upsilon}(i)\}_{i=1}^{N_\eta}$ drawn from p_{η_0}, the constant C is determined by

$$C=\frac{2^{l-1}-1}{\max_{\substack{1\le i\le N_\eta\\ j=1,2,3}}|\Upsilon^j(i)|} \tag{51}$$

The chromosome of $\mathcal{B}(\boldsymbol{\Upsilon})$ is, therefore, a concatenated string of $\mathcal{B}(\Upsilon^j)$ for $j=1,2,3$, that is,

$$\mathcal{B}(\boldsymbol{\Upsilon})=\{\mathcal{B}(\Upsilon^1),\mathcal{B}(\Upsilon^2),\mathcal{B}(\Upsilon^3)\} \tag{52a}$$

with

$$\mathcal{B}(\Upsilon^j)=\{\mathrm{sgn}(\Upsilon^j),\mathrm{bin}|C\Upsilon^j|\} \tag{52b}$$

where $\mathrm{bin}|C\Upsilon^j|$ represents the binary coding of $|C\Upsilon^j|$.

Remark 3: The variables l and N_η are predetermined parameters of the GA step. They are tuned to agree with the required accuracy and convergence rate of the GA step, as well as with the available computing power.

Sequential Likelihood Buildup

The purpose of the GA step is to produce an improved population of bias parameters by means of maximizing the likelihood. This requirement identifies the GA fitness function to be the likelihood approximation in Eq. (50). Therefore,

$$\arg\max_{1\le i\le N_\eta}\varphi_k(\boldsymbol{\Upsilon}_k(i))=\arg\max_{1\le i\le N_\eta}\hat{\mathcal{L}}\left(\boldsymbol{\Upsilon}_k(i)|Y_1^k\right) \tag{53}$$

where $\varphi_k(\boldsymbol{\Upsilon}_k(i))$ denotes the fitness value of the parameter $\boldsymbol{\Upsilon}_k(i)$ at time k. Furthermore, recalling that every GA iteration produces a new population of parameters, clearly, for the likelihood of some $\boldsymbol{\Upsilon}_k(i)$ to be computed sequentially, it is essential to specify time intervals where the population does not change through generations. (In those time intervals there are no reproduction/crossover/mutation operations.) Maintaining $\boldsymbol{\Upsilon}_k(i)=\boldsymbol{\Upsilon}_{k-1}(i)$ for any $k\in[k_1,k_2]$ consequently yields the following recursion for the fitness function of any parameter in the set $\{\boldsymbol{\Upsilon}_k(i)\}_{i=1}^{N_\eta}$:

$$\varphi_k(\boldsymbol{\Upsilon}_k(i))=p_{\delta\boldsymbol{b}_k}(\boldsymbol{b}_k-A(\Phi(\boldsymbol{\omega}_k-\boldsymbol{\Upsilon}_k(i))\hat{\boldsymbol{q}}_{k-1})\boldsymbol{r}_k)\varphi_{k-1}(\boldsymbol{\Upsilon}_{k-1}(i))\qquad k\in[k_1,k_2] \tag{54}$$

Reproduction

At time $k-1$ the set $\{\boldsymbol{\Upsilon}_{k-1}(i),\varphi_{k-1}(\boldsymbol{\Upsilon}_{k-1}(i))\}_{i=1}^{N_\eta}$, consisting of the parameters and their corresponding fitness values, is available. When a new measurement is acquired, the fitness value of each parameter is updated using Eq. (54). The fitness function is then normalized according to

$$\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i))=\frac{\varphi_k(\boldsymbol{\Upsilon}_k(i))}{\sum_{j=1}^{N_\eta}\varphi_k(\boldsymbol{\Upsilon}_k(j))}\qquad i=1,\ldots,N_\eta \tag{55}$$

so that the number of offspring of each parameter is simply its fitness function multiplied by the total number of parameters {which means that $\boldsymbol{\Upsilon}_k(i)$ is reproduced with probability $\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i))$}. Each parameter $\boldsymbol{\Upsilon}_k(i)$ is then reproduced $N_k(i)$ times, where the corresponding number of offspring is obtained by

$$N_k(i)=\mathrm{int}(\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i))N_\eta)\qquad i=1,\ldots,N_\eta \tag{56}$$

The reproduced offspring for the parameter $\boldsymbol{\Upsilon}_k(i)$ are

$$\bar{\boldsymbol{\Upsilon}}_k(j)_i=\boldsymbol{\Upsilon}_k(i)\qquad j=1,\ldots,N_k(i) \tag{57}$$

with the fitness

$$\tilde{\varphi}_k(\bar{\boldsymbol{\Upsilon}}_k(j)_i)=\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i)) \tag{58}$$

Usually the reproduced population has slightly less than N_η parameters because $N_\eta\times\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i))$ is not an integer. In that case one can reproduce the most likely parameter several more times.

Crossover

Coding the reproduced set yields a population of corresponding N_η chromosomes. Moving some chromosomes to a so-called mating pool, the process proceeds by choosing in random two chromosomes from this mating pool and by choosing another uniformly distributed random number between 2 and $l+1$, denoted by n. The crossover is carried out by swapping each and every bit between two corresponding bias components of the mated chromosomes, from the nth place to the $l+1$. This process is applied to each bias string component. Implementing a so-called sign rule into this inbreeding refines each generation and, consequently, improves the convergence of the GA. Thus, half of the mated chromosomes preserve their sign character, whereas the other half adopt the sign character of the fittest of the two mated chromosomes. Finally, any pair of mated chromosomes is taken out of the mating pool.

Mutation

After the crossover the mutation process walks through the string space (i.e., all of the strings within the population) and changes a bit (from 1 to 0 and from 0 to 1) every N_{mut} positions. Recommended values for N_{mut} are rather large; thus, this process plays only a secondary role in the GA. In this work N_{mut} was set equal to N_η as recommended in Chapter 3 of Ref. 18.

Finally, decoding the reproduced chromosomes yields a new generation of parameters. The fitness weights of the new parameters are then set to

$$\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i)) = 1/N_\eta \qquad i = 1, \ldots, N_\eta \tag{59}$$

because some of the reproduced parameters have no past likelihood weights.

D. GA–QPF Algorithm

Incorporating the GA into the QPF is carried out by first setting the times during which sequential buildup of the likelihood takes place. Recalling that the resampling stage degrades the posterior representation, thus typically yielding worse estimates, it is reasonable to implement this stage whenever resampling is not needed. This leads to executing the GA iteration every resampling.

A single GA–QPF cycle is illustrated in Fig. 1.

VI. Cramér–Rao Lower Bound

To assess the statistical efficiency of the QPF, the posterior CRLB is computed. Because of the complex computations that follow, the ensuing development is constrained to the case of unbiased gyro measurements.

Denoting by $\hat{\mathbf{q}}_k$ any unbiased estimator of $\mathbf{q}_k$, it is well known from estimation theory that the estimation error covariance is bounded from below by

$$E\left[(\hat{\mathbf{q}}_k - \mathbf{q}_k)(\hat{\mathbf{q}}_k - \mathbf{q}_k)^T\right] \geq J_k^{-1} \tag{60}$$

where J_k is the Fisher information matrix for k measurements, defined as

$$J_k \triangleq E\left[-\Delta_{\mathbf{q}_k}^{\mathbf{q}_k} \log p_{\mathbf{q}_k, \mathcal{Y}_1^k}\left(\mathbf{q}_k, \mathcal{Y}_1^k\right)\right] \tag{61}$$

where $\Delta_x^y \triangleq \nabla_y \nabla_x^T$ is the Laplacian operator. It has been recently shown[19] that the posterior information matrix can be computed recursively as

$$J_{k+1} = D_k^{22} - D_k^{12^T}\left(J_k + D_k^{11}\right)^{-1} D_k^{12} \tag{62}$$

where

$$D_k^{11} = E\left[-\Delta_{\mathbf{q}_{k+1}}^{\mathbf{q}_k} \log p_{\mathbf{q}_{k+1}|\mathbf{q}_k}(\mathbf{q}_{k+1}|\mathbf{q}_k)\right] \tag{63a}$$

$$D_k^{12} = E\left[-\Delta_{\mathbf{q}_k}^{\mathbf{q}_{k+1}} \log p_{\mathbf{q}_{k+1}|\mathbf{q}_k}(\mathbf{q}_{k+1}|\mathbf{q}_k)\right] \tag{63b}$$

$$D_k^{22} = E\left[-\Delta_{\mathbf{q}_{k+1}}^{\mathbf{q}_{k+1}} \log p_{\mathbf{q}_{k+1}|\mathbf{q}_k}(\mathbf{q}_{k+1}|\mathbf{q}_k)\right] + E\left[-\Delta_{\mathbf{q}_{k+1}}^{\mathbf{q}_{k+1}} \log p_{y_{k+1}|\mathbf{q}_{k+1}}(y_{k+1}|\mathbf{q}_{k+1})\right] \tag{63c}$$

Notice that, whereas the computation of the CRLB using Eq. (61) implies the knowledge of the joint PDF of the state and entire measurement history, Eqs. (63) use only the transition and the likelihood PDFs, which are easily computed from the system's governing equations.

A. CRLB for the Quaternion Estimation Problem

From Eqs. (63) it is clear that the CRLB recursion depends mostly upon the transition density $p_{\mathbf{q}_{k+1}|\mathbf{q}_k}$. However, because of the quaternion norm constraint, this transition density is singular. Thus, the CRLB is computed for the quaternion's vector part only. The transition density $p_{\boldsymbol{\varrho}_{k+1}|\boldsymbol{\varrho}_k}$ and the likelihood $p_{y_k|\boldsymbol{\varrho}_k}$ are evaluated as follows.

Equation (13) can be used to express the quaternion vector part at time $k+1$, $\boldsymbol{\varrho}_{k+1}$, in terms of $\boldsymbol{\varrho}_k$ and $\boldsymbol{\epsilon}_k$, the gyro white-noise process. Assuming that $\boldsymbol{\varrho}_k$ is known, this expression takes the general form

$$\boldsymbol{\varrho}_{k+1} = \boldsymbol{f}_k(\boldsymbol{\epsilon}_k) \tag{64}$$

where $\boldsymbol{f}_k$ has $\boldsymbol{\varrho}_k$ as its parameter. Assuming that the process noise PDF $p_{\boldsymbol{\epsilon}_k}$ is known, the required transition density can be computed

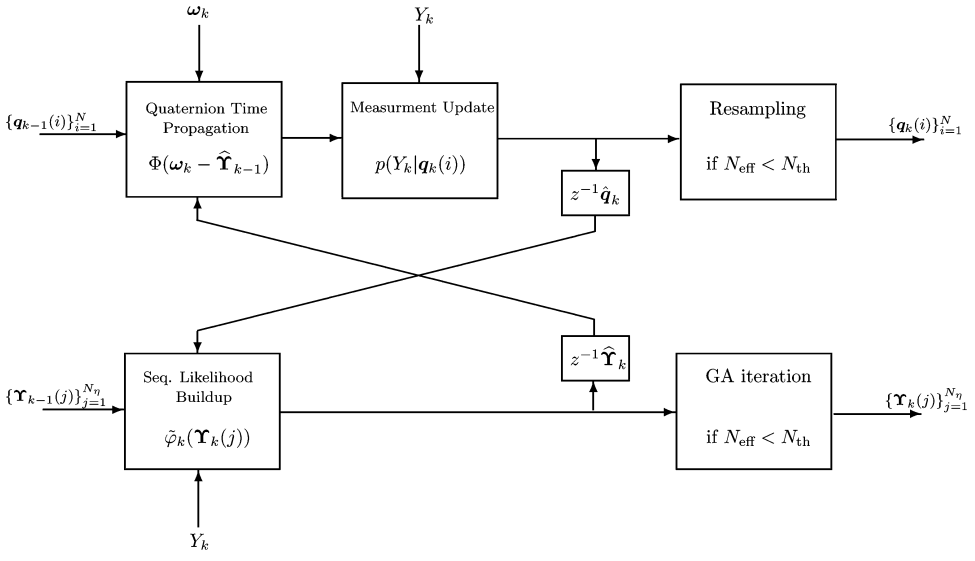

Fig. 1 GA–QPF scheme.

as

$$p_{\varrho_{k+1}|\varrho_k}(\boldsymbol{\rho}_{k+1}|\boldsymbol{\rho}_k) = p_{\epsilon_k}\left(\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})\right)\det\left(\nabla_{\boldsymbol{\rho}_{k+1}}\boldsymbol{f}_k^{-T}(\boldsymbol{\rho}_{k+1})\right)^T \tag{65}$$

where $(\nabla_{\varrho_{k+1}}\boldsymbol{f}_k^{-T}(\boldsymbol{\rho}_{k+1}))^T$ is the Jacobian matrix of $\boldsymbol{f}_k^{-1}$ with respect to $\boldsymbol{\varrho}_{k+1}$, and the function $\boldsymbol{f}_k^{-1}$ is the inverse function of $\boldsymbol{f}_k$ such that $\boldsymbol{\epsilon}_k = \boldsymbol{f}_k^{-1}(\boldsymbol{\varrho}_{k+1})$. From Eq. (15), the random variable $\boldsymbol{\epsilon}_k$ is related to the measured angular rate by

$$\boldsymbol{\epsilon}_k = \boldsymbol{\omega}_k - \boldsymbol{\omega}_k^o \tag{66}$$

Observing Eq. (66), $\boldsymbol{f}_k^{-1}$ can be found by extracting the random variable $\boldsymbol{\omega}_k$ from Eq. (13). An easy way to accomplish this is by using an alternative form for the quaternion transition matrix in Eq. (12) (Ref. 6)

$$\Phi_k = \begin{bmatrix} \cos\left(\frac{1}{2}\|\boldsymbol{\omega}_k\|\Delta t\right)I_{3\times 3} - [\boldsymbol{\psi}_k\times] & \boldsymbol{\psi}_k \\ -\boldsymbol{\psi}_k^T & \cos\left(\frac{1}{2}\|\boldsymbol{\omega}_k\|\Delta t\right) \end{bmatrix} \tag{67}$$

where $\boldsymbol{\psi}_k = \sin(\frac{1}{2}\|\boldsymbol{\omega}_k\|\Delta t)(\boldsymbol{\omega}_k/\|\boldsymbol{\omega}_k\|)$. Taking the quaternion product of both sides of Eq. (13) with the inverse of $\mathbf{q}_k$ and using Eq. (67) yields

$$\mathbf{q}_{k+1}\otimes\mathbf{q}_k^{-1} = (\Phi_k\mathbf{q}_k)\otimes\mathbf{q}_k^{-1} = \begin{bmatrix} \boldsymbol{\psi}_k \\ \cos\left(\frac{1}{2}\|\boldsymbol{\omega}_k\|\Delta t\right) \end{bmatrix} \tag{68}$$

Let $\delta\mathbf{q}_k \triangleq \mathbf{q}_{k+1}\otimes\mathbf{q}_k^{-1}$, and let $\delta\boldsymbol{\varrho}_k$ denote the vector part of $\delta\mathbf{q}_k$. Using Eq. (68), the angular rate is easily obtained in terms of $\delta\boldsymbol{\varrho}_k$. Finally, using the resulting expression for $\boldsymbol{\omega}_k$ in Eq. (66) yields

$$\boldsymbol{\epsilon}_k = \boldsymbol{f}_k^{-1}(\boldsymbol{\varrho}_{k+1}) = 2\arccos\left(\sqrt{1-\|\delta\boldsymbol{\varrho}_k\|^2}\right)\frac{\delta\boldsymbol{\varrho}_k}{\|\delta\boldsymbol{\varrho}_k\|\Delta t} - \boldsymbol{\omega}_k^o \tag{69}$$

Now using Eq. (65) and assuming a zero-mean, white Gaussian process noise $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0_{3\times 1}, R_\epsilon)$ yields

$$\begin{aligned}\log p_{\varrho_{k+1}|\varrho_k}(\boldsymbol{\rho}_{k+1}|\boldsymbol{\rho}_k) &= \log C_0 - \tfrac{1}{2}\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})^T R_\epsilon^{-1}\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1}) \\ &\quad + \log\det(\nabla_{\boldsymbol{\rho}_{k+1}}\boldsymbol{f}_k^{-T}(\boldsymbol{\rho}_{k+1}))^T \\ &\approx \log C_0 - \tfrac{1}{2}\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})^T R_\epsilon^{-1}\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1}) \\ &\quad - 3\log\left(\tfrac{1}{2}\Delta t\right) - \tfrac{1}{2}\log\left(1-\|\boldsymbol{\rho}_k\|^2\right)\end{aligned} \tag{70}$$

where C_0 is the normalization constant of the PDF of $\boldsymbol{\epsilon}_k$, and the approximate form in Eq. (70) is based on using the first-order expansion of the transition matrix Φ_k.

For a Gaussian white observation noise $\delta\boldsymbol{b}_k \sim \mathcal{N}(0, R_b)$, it can be shown that the log-likelihood function and its gradient with respect to $\boldsymbol{\varrho}_{k+1}$ are

$$\log p_{y_{k+1}|\varrho_{k+1}}(Y_{k+1}|\boldsymbol{\rho}_{k+1}) = \log C_1 - \tfrac{1}{2}\left[\boldsymbol{b}_{k+1} - A(\boldsymbol{\rho}_{k+1})\boldsymbol{r}_{k+1}\right]^T \times R_b^{-1}\left[\boldsymbol{b}_{k+1} - A(\boldsymbol{\rho}_{k+1})\boldsymbol{r}_{k+1}\right] \tag{71a}$$

$$\nabla_{\boldsymbol{\varrho}_{k+1}}\log p_{y_{k+1}|\varrho_{k+1}}(Y_{k+1}|\boldsymbol{\rho}_{k+1}) = -\nabla_{\boldsymbol{\varrho}_{k+1}}\left[A(\boldsymbol{\rho}_{k+1})\boldsymbol{r}_{k+1}\right]^T \times R_b^{-1}\left[\boldsymbol{b}_{k+1} - A(\boldsymbol{\rho}_{k+1})\boldsymbol{r}_{k+1}\right] \tag{71b}$$

where C_1 is a normalization constant and $A(\boldsymbol{\rho}_{k+1})$ denotes the attitude matrix corresponding to the quaternion with $\boldsymbol{\rho}_{k+1}$ as its vector part. The final step in the computation of the CRLB consists of computing the Hessians in Eqs. (63). The highly complicated analytic expressions for these matrices were obtained using the Maple[‡] symbolic algebra package. These expressions are omitted here for conciseness.

[‡]Data available online at http://www.maplesoft.com.

VII. Simulation Study

A simulation study has been performed to evaluate the performance of the new algorithm and to compare it to standard EKFs and to the recently proposed USQUE filter of Ref. 6. Two cases were investigated. In the first case, the filters were run in a synthetic noise example, taken from Ref. 6. In the second case, real measurements from the Technion's TechSAT satellite were used to examine the performance of the algorithms in a non-Gaussian measurement noise case.

A. Synthetic Noise Case

The GA–QPF was applied to a realistic spacecraft model and compared to the additive quaternion extended Kalman filter (AEKF) of Ref. 3, to the multiplicative quaternion extended Kalman filter (MEKF) of Ref. 2, and to the unscented quaternion estimator (USQUE) recently presented in Ref. 6. (The actual USQUE code of Ref. 6 was used with the kind permission of its authors.) In addition to these algorithms, an iterated version of the AEKF has also been examined. The iterated EKF algorithm was proposed for improving the EKF performance in strongly nonlinear systems.[20] In all simulations, the iterated AEKF used eight iterations at each measurement update.

The simulation scenario, as well as all USQUE settings, are taken from Ref. 6. The spacecraft is in an Earth-pointing, near-circular 90-min orbit with an inclination of 35 deg. The spacecraft is equipped with a TAM and gyroscopic rate sensors. The TAM's noise is modeled as a zero-mean Gaussian white process with a standard deviation of 50 nT. The rate-integrating gyros' output is contaminated with a measurement noise having two components: a white, zero-mean Gaussian process with intensity 0.1 $(\mu\text{rad})^2/\text{s}$ and a drift bias modeled as an integrated Gaussian white noise with intensity 1×10^{-7} $(\mu\text{rad})^2/\text{s}^3$. The RIGs' initial bias is set to 0.1 deg/h on each axis. Using these values in Eq. (44) gives

$$\text{Prob}(\|\boldsymbol{\eta}_k - \boldsymbol{\eta}_{k_1}\|_2 \geq 0.1\text{ deg/h}) \leq 6\times 10^{-3} \qquad k\in[k_1, k_1+T_\eta] \tag{72}$$

for a time interval of length $T_\eta = 30$ s starting at any time k_1 (see Sec. V). The Earth magnetic field is modeled using the eighth-order international geomagnetic reference field.

Constant Initial Attitude

In this example, the initial attitude errors are set to -50, 50, and 160 deg on all three axes, amounting to a norm attitude error of 120 deg. The initial attitude covariance of the USQUE is set to $50\text{ deg}^2 I_{3\times 3}$ (which seemed to be the most successful choice based on tuning runs made by the authors of Ref. 6). The GA–QPF is initialized with $N = 1500$ particles and $N_\eta = 200$ bias parameters, using the initialization scheme of Appendix B. The initial bias parameters are simulated using a zero-mean Gaussian kernel with the same covariance as in the Kalman filter variants, and the chromosome length was set to $l = 30$ bit. The crossover procedure of the GA is applied to only 40% of the chromosomes. After the first two measurement updates, the filter's particle set is reduced to the $N = 150$ unit quaternion particles corresponding to the largest importance weights. Each particle is weighted using the likelihood

$$\begin{aligned}p_{y_k|\mathbf{q}_k}(Y_k|\boldsymbol{q}_k) &= \{1/[(2\pi)^{\frac{3}{2}}(\det R_b)^{\frac{1}{2}}]\} \\ &\quad\times\exp\left\{-\tfrac{1}{2}[\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k]^T R_b^{-1}[\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k]\right\}\end{aligned} \tag{73}$$

with the covariance $R_b = 50^2 I_{3\times 3}$ $(\text{nT})^2$. The resampling threshold is set to $N_{\text{th}} = \frac{2}{3}N$ based on tuning runs. (Decreasing N_{th} can reduce the resampling frequency, consequently introducing less Monte Carlo variations into the estimates; however, this might also increase the algorithm's sensitivity to heavy-tailed measurement noise PDFs.) The sampling rate of the TAM is 1 per 10 s, whereas the gyros are sampled at 1 Hz. The attitude estimation error (in degrees) is computed as

$$\delta\alpha = 2\arccos(\delta q_4) \tag{74}$$

where δq_4 is the scalar component of the error quaternion $\delta\boldsymbol{q}$.

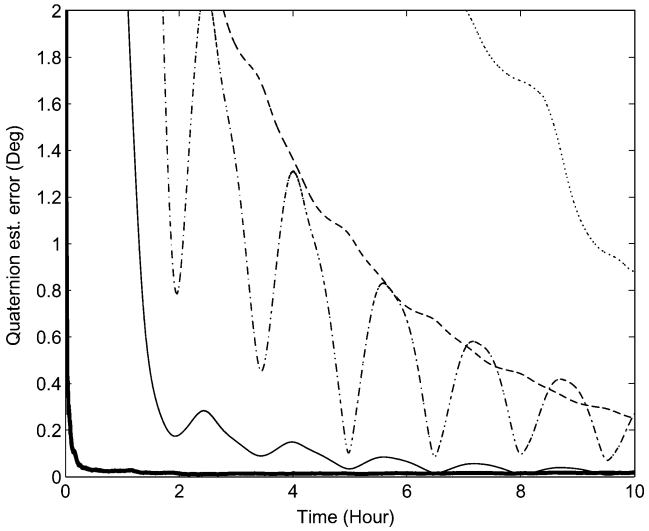

Fig. 2 Mean quaternion estimation error for the GA–QPF (thick ——), USQUE (thin ——), MEKF (–·–·–), IAEKF (- - - -), and AEKF (· · ·), filters: 100 Monte Carlo runs, synthetic noise case.

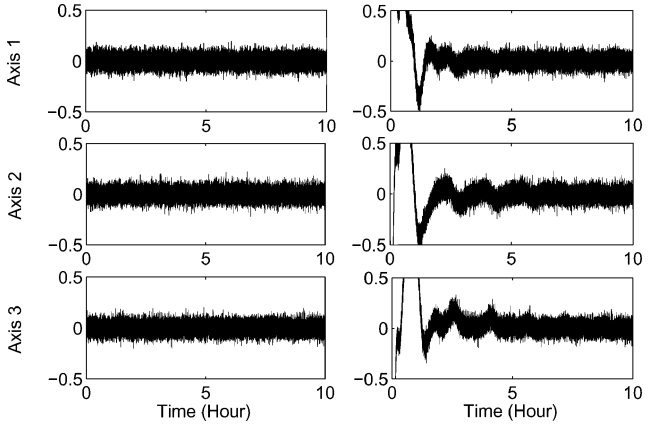

Fig. 3 Innovations process of the GA–QPF filter (left panel) and the USQUE filter (right panel).

Figure 2 presents the results of a 100-run Monte Carlo study in this case. As can be seen from Fig. 2, the GA–QPF reaches its steady-state attitude estimation error of less than 0.1 deg in about 15 min, whereas it takes the USQUE filter about 10 h to reach the same accuracy level. All three Kalman filter variants perform worse.

The innovations processes of both the GA–QPF filter and the USQUE filter are shown in Fig. 3. Recalling that the innovations process of an optimal filter is a white process, this figure clearly shows that the GA–QPF filter better processes the information embedded in the measurements.

Uncertain Initial Attitude

In this example the initial attitude quaternion is randomly generated according to a uniform distribution on the unit three-sphere. Two filters are examined: GA–QPF and USQUE. In the USQUE filter the initial estimate is set to $\hat{\boldsymbol{q}}_0 = [0, 0, 0, 1]^T$. The GA–QPF algorithm is initialized as described earlier. In all runs, the USQUE's initial covariance is set to

$$P_0 = 1.2 \times \operatorname{diag}\left\{\boldsymbol{\xi}_1^2, \boldsymbol{\xi}_2^2, \boldsymbol{\xi}_3^2\right\} \tag{75}$$

where $\boldsymbol{\xi}_i$, $i = 1, 2, 3$ are the modified Rodrigues parameter components corresponding to the initial attitude quaternion $\boldsymbol{q}_0$. The simulation consists of 1300 Monte Carlo runs.

The statistical distribution of the quaternion estimation errors of the GA–QPF and the USQUE algorithms are presented in Fig. 4 (notice the logarithmic scale). The lines in Fig. 4, top to bottom,

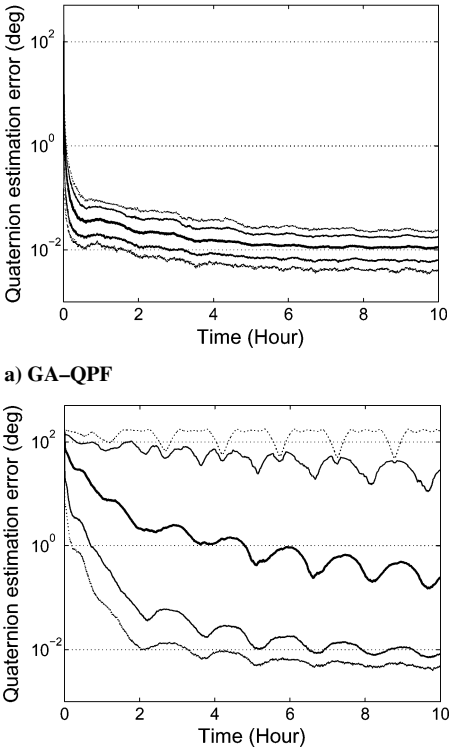

Fig. 4 Statistical distribution of quaternion estimation errors of the GA–QPF and USQUE filters, based on 1300 Monte Carlo runs, synthetic noise case. Lines, top to bottom: 95th, 85th, 50th, 15th, and 5th percentiles.

correspond to the 95th, 85th, 50th, 15th, and 5th percentiles of the Monte Carlo runs, respectively.

Figure 4a shows that after 10 hours, 70% of the GA–QPF quaternion estimation errors reach values between 0.0060 and 0.020 deg. Moreover, 95% of the GA–QPF's estimation errors are smaller than 0.050 deg after about two hours. In comparison, Fig. 4b shows the statistical performance of the USQUE for this scenario. As can be seen from Fig. 4b, after 10 hours, 70% of the USQUE quaternion estimation errors are between 0.010 and 30 deg. The 95th percentile line exhibits oscillations near 100 deg, which indicates that in about 5% of the runs the USQUE filter did not converge. This example clearly demonstrates the superior global convergence of the GA–QPF.

The statistical distribution of the normed gyro bias estimation errors of both the GA–QPF and the USQUE is shown in Fig. 5, which demonstrates the benefit of decoupling the quaternion and the bias estimation procedures (notice the logarithmic scale). Starting from the same initial bias errors, the USQUE estimation error increases dramatically at the beginning of the estimation process, before sufficient data are collected for the filter to start estimating the bias. This erroneous bias estimate is then used in the time propagation of the quaternion, which, consequently, leads to unsatisfactory quaternion estimates. On the other hand, the GA–QPF bias estimation suffers from no such phenomenon, and its bias estimation error decreases monotonically throughout the process.

B. TechSAT Real Noise Data

The performance of the GA–QPF was also examined using a realistic TAM noise model, taken from the Technion's TechSAT microsatellite. The TechSAT orbit is inclined at 98 deg, and its period is 101 min. As in the previous case, the satellite performs an Earth-pointing mission. The TechSAT's TAM measurement noise joint PDF was estimated using 75 hours of raw TAM data (acquired at a rate of once per 10 s) by computing the three-dimensional histogram of the measurement residuals (TAM measurements minus predicted observations as computed by the geomagnetic field model). This

PDF is shown in Fig. 6. Clearly, Fig. 6 shows that the TechSAT TAM's joint PDF is non-Gaussian. The double-peaked distribution is caused by a parasitic magnetic dipole moment along the Y body-frame axis, which was encountered during the momentum wheel slowdown.[21]

The results for this case were obtained by simulating the TAM noise via sampling from the PDF estimated from the real noise data. In all runs, the initial attitude errors and bias were set as in the preceding example. The likelihood function $p_{y_k|\mathbf{q}_k}$ was approximated by the Gaussian mixture

$$p_{y_k|\mathbf{q}_k}(Y_k|\boldsymbol{q}_k) = \kappa\Big(\exp\Big\{-\tfrac{1}{2}[\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k - \bar{\boldsymbol{n}}]^T \times R^{-1}[\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k - \bar{\boldsymbol{n}}]\Big\} + \exp\Big\{-\tfrac{1}{2}[\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k + \bar{\boldsymbol{n}}]^T \times R^{-1}[\boldsymbol{b}_k - A(\boldsymbol{q}_k)\boldsymbol{r}_k + \bar{\boldsymbol{n}}]\Big\}\Big) \tag{76}$$

where κ is a normalization constant, $\bar{\boldsymbol{n}} = [0\ -0.6\ 0]^T$, and $R = \mathrm{diag}\{\sigma_x^2, \sigma_y^2/5.7, \sigma_z^2\}$, where $\sigma_x, \sigma_y, \sigma_z$ denote the TAM noise data sample standard deviation along the X, Y, and Z axes, respectively.

The GA–QPF is initialized as described before. The measurement noise covariance matrices of the USQUE and the EKF variants are set to the sample variance of the TAM noise data.

Figure 7 shows the results of a 100-run Monte Carlo study in this case. It can be seen that the GA–QPF's mean estimation error reaches values under 0.1 deg in about 20 min. In comparison, the USQUE reaches twice the same values only after about six orbits

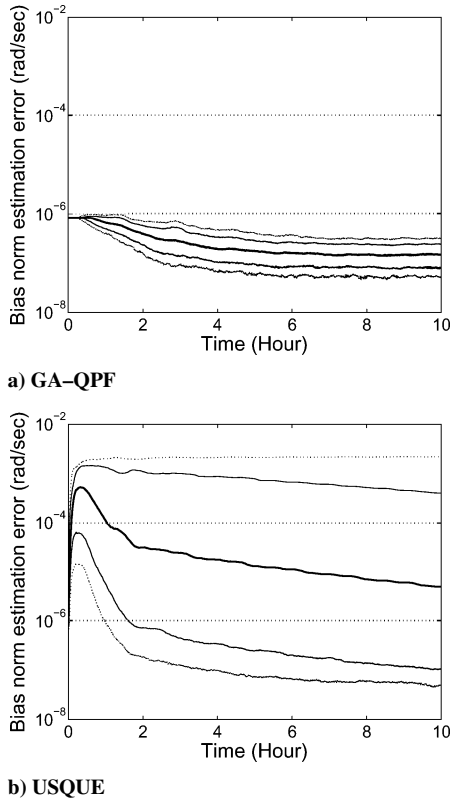

Fig. 5 Statistical distribution of bias estimation errors of the GA–QPF and USQUE filters, based on 1300 Monte Carlo runs, synthetic noise case. Lines, top to bottom: 95th, 85th, 50th, 15th, and 5th percentiles.

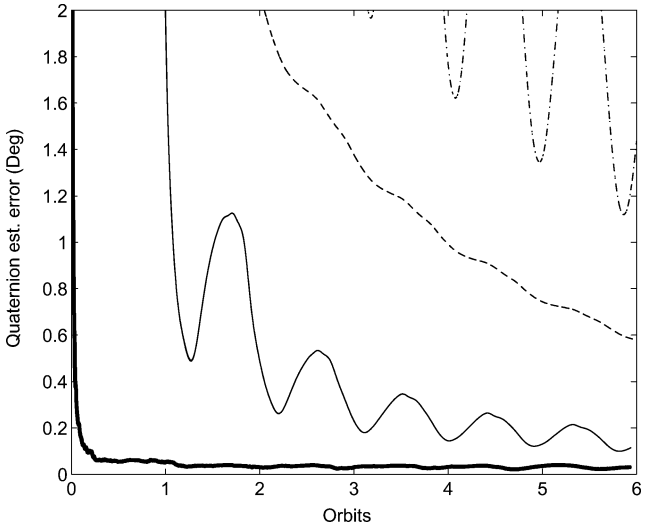

Fig. 7 Mean attitude estimation error of the GA–QPF (thick ——), USQUE (thin ——), IAEKF (- - - -), and MEKF (–·–·–) filters: 100 Monte Carlo runs, TechSAT real noise data.

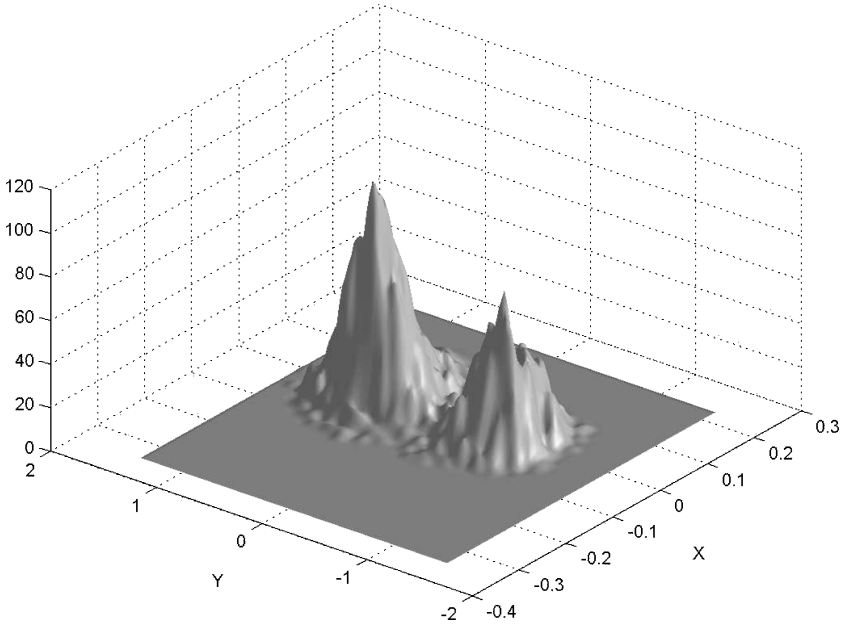

Fig. 6 TAM noise PDF for $Z = 0$ μT, TechSAT real data.

(about 10 hours). The mean estimation errors of all EKF variants are higher than 0.5 deg after six orbits. (The AEKF's error is out of scale and, hence, it is not shown in Fig. 7.)

C. Monte Carlo Approximation of the CRLB

To enable a numerical assessment of the algorithm's statistical efficiency, the observed estimation error covariance (not an integral part of the PF algorithm) is compared to the CRLB. To that end, the mathematical expectations in Eqs. (63) are estimated by a Monte Carlo simulation, that is, every expectation is computed using the ensemble mean, computed over all Monte-Carlo runs.

The expectations associated with the computation of the matrices D_k^{11} and D_k^{12} involve the random variables $\boldsymbol{\varrho}_{k+1}$ and $\boldsymbol{\varrho}_k$ only. To properly approximate these specific expectations, the joint PDF $p_{\boldsymbol{\varrho}_{k+1},\boldsymbol{\varrho}_k}$ is recast in a convenient form, as follows.

Recalling that $\{\boldsymbol{\epsilon}_k\}_{k=1}^{\infty}$ is a white sequence and that $\boldsymbol{\varrho}_k$ is independent of $\boldsymbol{\epsilon}_k$ yields

$$\begin{aligned}
&p_{\boldsymbol{\varrho}_{k+1},\boldsymbol{\varrho}_k}(\boldsymbol{\rho}_{k+1},\boldsymbol{\rho}_k)\\
&\quad=\int_{-\infty}^{+\infty} p_{\boldsymbol{\varrho}_{k+1}|\boldsymbol{\varrho}_k,\boldsymbol{\epsilon}_k}(\boldsymbol{\rho}_{k+1}|\boldsymbol{\rho}_k,\boldsymbol{\epsilon}_k)\,p_{\boldsymbol{\varrho}_k|\boldsymbol{\epsilon}_k}(\boldsymbol{\rho}_k|\boldsymbol{\epsilon}_k)\,p_{\boldsymbol{\epsilon}_k}(\boldsymbol{\epsilon}_k)\,\mathrm{d}\boldsymbol{\epsilon}_k\\
&\quad=p_{\boldsymbol{\varrho}_k}(\boldsymbol{\rho}_k)\int_{-\infty}^{+\infty} p_{\boldsymbol{\varrho}_{k+1}|\boldsymbol{\varrho}_k,\boldsymbol{\epsilon}_k}(\boldsymbol{\rho}_{k+1}|\boldsymbol{\rho}_k,\boldsymbol{\epsilon}_k)\,p_{\boldsymbol{\epsilon}_k}(\boldsymbol{\epsilon}_k)\,\mathrm{d}\boldsymbol{\epsilon}_k\\
&\quad=p_{\boldsymbol{\varrho}_k}(\boldsymbol{\rho}_k)\int_{-\infty}^{+\infty}\delta\left(\boldsymbol{\epsilon}_k-\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})\right)p_{\boldsymbol{\epsilon}_k}(\boldsymbol{\epsilon}_k)\,\mathrm{d}\boldsymbol{\epsilon}_k\\
&\quad=p_{\boldsymbol{\varrho}_k}(\boldsymbol{\rho}_k)\,p_{\boldsymbol{\epsilon}_k}\left(\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})\right)
\end{aligned}\tag{77}$$

where $\delta(\,)$ denotes the Dirac delta distribution. Using Eq. (77), the expectation of any $p_{\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}$ integrable function $g(\,)$ over the joint probability space of $\boldsymbol{\varrho}_{k+1}$, $\boldsymbol{\varrho}_k$ can be evaluated by

$$\begin{aligned}
&E[g(\boldsymbol{\varrho}_{k+1},\boldsymbol{\varrho}_k)]\\
&\quad=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(\boldsymbol{\rho}_{k+1},\boldsymbol{\rho}_k)\,p_{\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}(\boldsymbol{\rho}_k,\boldsymbol{\rho}_{k+1})\,\mathrm{d}\boldsymbol{\rho}_k\,\mathrm{d}\boldsymbol{\rho}_{k+1}\\
&\quad=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(\boldsymbol{\varrho}_{k+1},\boldsymbol{\varrho}_k)\,p_{\boldsymbol{\varrho}_k}(\boldsymbol{\rho}_k)\,p_{\boldsymbol{\epsilon}_k}\left(\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})\right)\mathrm{d}\boldsymbol{\rho}_k\,\mathrm{d}\boldsymbol{\rho}_{k+1}
\end{aligned}\tag{78}$$

Using Eq. (78), the following simulation scheme is adopted for D_k^{11} and D_k^{12}. First, an independent, identically distributed set of samples is drawn from $p_{\boldsymbol{\epsilon}_k}$ to get $\{\boldsymbol{\epsilon}_k(j)\}_{j=1}^{N_\epsilon}$. Next, for any sample of the given particle set $\{\boldsymbol{q}_k(i)\}_{i=1}^{N}$, the quaternion sample $\boldsymbol{q}_k(i)$ along with $\boldsymbol{\epsilon}_k(j)$ are passed through the process equation to yield $\{\{\boldsymbol{q}_{k+1}(i)_j\}_{i=1}^{N}\}_{j=1}^{N_\epsilon}$. An empirical estimate of the corresponding expectation is then computed as

$$E[g(\boldsymbol{\varrho}_{k+1},\boldsymbol{\varrho}_k)]\approx\frac{1}{NN_\epsilon}\sum_{j=1}^{N_\epsilon}\sum_{i=1}^{N} g(\boldsymbol{\rho}_{k+1}(i)_j,\boldsymbol{\rho}_k(i))\tag{79}$$

The only term in Eqs. (63) that depends on the measurement is the likelihood $p_{\mathbf{y}_{k+1}|\mathbf{q}_{k+1}}$, which appears in D_k^{22}. To properly approximate the expectation in that case, the joint density $p_{\mathbf{y}_{k+1},\boldsymbol{\varrho}_{k+1}}$ is written as

$$\begin{aligned}
p_{\mathbf{y}_{k+1},\boldsymbol{\varrho}_{k+1}}(Y_{k+1},\boldsymbol{\rho}_{k+1})&=\int_{-\infty}^{+\infty} p_{\mathbf{y}_{k+1}|\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}(Y_{k+1}|\boldsymbol{\rho}_k,\boldsymbol{\rho}_{k+1})\\
&\quad\times p_{\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}(\boldsymbol{\rho}_k,\boldsymbol{\rho}_{k+1})\,\mathrm{d}\boldsymbol{\rho}_k\\
&=\int_{-\infty}^{+\infty} p_{\mathbf{y}_{k+1}|\boldsymbol{\varrho}_{k+1}}(Y_{k+1}|\boldsymbol{\rho}_{k+1})\\
&\quad\times p_{\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}(\boldsymbol{\rho}_k,\boldsymbol{\rho}_{k+1})\,\mathrm{d}\boldsymbol{\rho}_k
\end{aligned}\tag{80}$$

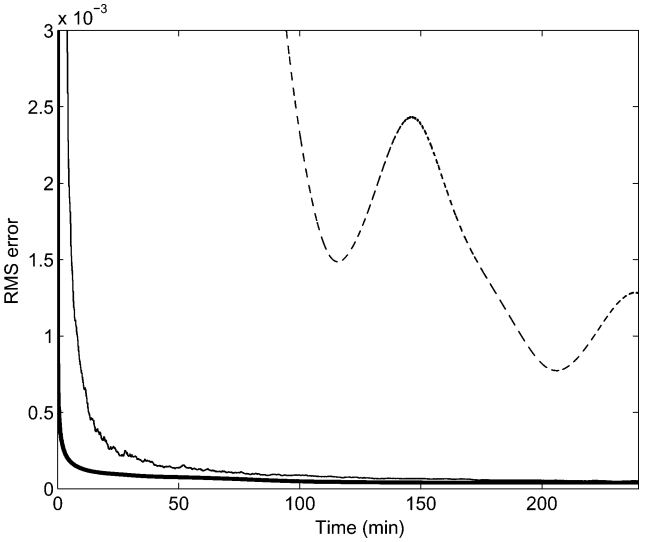

Fig. 8 RMS estimation errors of the QPF (thin ——) and USQUE (- - - -) and the CRLB (thick ——) for the attitude estimation problem.

where the equivalence $p_{\mathbf{y}_{k+1}|\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}=p_{\mathbf{y}_{k+1}|\boldsymbol{\varrho}_{k+1}}$ follows from the Markovian nature of the observation model [see Eq. (9)]. Now, using Eqs. (80) and (77), the expectation of any $p_{\mathbf{y}_{k+1},\boldsymbol{\varrho}_{k+1}}$ integrable function $g(\,)$ over the joint probability space of $\mathbf{y}_{k+1}$, $\boldsymbol{\varrho}_{k+1}$ is evaluated as

$$\begin{aligned}
&E[g(\mathbf{y}_{k+1},\boldsymbol{\varrho}_{k+1})]\\
&\quad=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(Y_{k+1},\boldsymbol{\rho}_{k+1})\,p_{\mathbf{y}_{k+1},\boldsymbol{\varrho}_{k+1}}(Y_{k+1},\boldsymbol{\rho}_{k+1})\,\mathrm{d}\boldsymbol{\rho}_{k+1}\,\mathrm{d}Y_{k+1}\\
&\quad=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(Y_{k+1},\boldsymbol{\rho}_{k+1})\,p_{\mathbf{y}_{k+1}|\boldsymbol{\varrho}_{k+1}}(Y_{k+1}|\boldsymbol{\rho}_{k+1})\\
&\qquad\times p_{\boldsymbol{\varrho}_k,\boldsymbol{\varrho}_{k+1}}(\boldsymbol{\rho}_k,\boldsymbol{\rho}_{k+1})\,\mathrm{d}\boldsymbol{\rho}_k\,\mathrm{d}\boldsymbol{\rho}_{k+1}\,\mathrm{d}Y_{k+1}\\
&\quad=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(Y_{k+1},\boldsymbol{\rho}_{k+1})\,p_{\mathbf{y}_{k+1}|\boldsymbol{\varrho}_{k+1}}(Y_{k+1}|\boldsymbol{\rho}_{k+1})\\
&\qquad\times p_{\boldsymbol{\varrho}_k}(\boldsymbol{\rho}_k)\,p_{\boldsymbol{\epsilon}_k}\left(\boldsymbol{f}_k^{-1}(\boldsymbol{\rho}_{k+1})\right)\mathrm{d}\boldsymbol{\rho}_k\,\mathrm{d}\boldsymbol{\rho}_{k+1}\,\mathrm{d}Y_{k+1}
\end{aligned}\tag{81}$$

Thus, the corresponding empirical estimate of the expectation in Eq. (81) has to be computed over all observations in the simulated i.i.d. set $\{Y_{k+1}(l)\}_{l=1}^{N_m}$ as well, that is,

$$E[g(\mathbf{y}_{k+1},\boldsymbol{\varrho}_{k+1})]\approx\frac{1}{NN_\epsilon N_m}\sum_{l=1}^{N_m}\sum_{j=1}^{N_\epsilon}\sum_{i=1}^{N} g(\boldsymbol{\rho}_{k+1}(i)_j,Y_{k+1}(l))\tag{82}$$

Using the same parameters and conditions as used in the synthetic noise case, the rms estimation error of 100 Monte Carlo runs of both the QPF and USQUE algorithms is compared to $\sqrt{[\mathrm{tr}(J_k^{-1})]}$ in Fig. 8. The Monte Carlo simulation of the CRLB is performed using $N=100$ state particles, in a random scenario involving $N_\epsilon=30$ process noise samples and $N_m=100$ measurement trajectories, amounting to 3000 Monte Carlo runs. Figure 8 shows that the rms estimation error achieved by the QPF practically attains the lower bound after about 200 min. It can be concluded, therefore, that this estimator is asymptotically efficient and optimal.

VIII. Conclusions

A new algorithm is presented for estimating the rotation quaternion from vector observations. The new attitude estimator is a particle filter that belongs to the class of sequential Monte Carlo methods. As such, this filter copes naturally with non-Gaussian driving processes and with the inherent nonlinearity of the attitude estimation problem. Moreover, because the attitude is represented in this filter

via quaternion particles the norm constraint is naturally maintained, avoiding the need for ad hoc and external procedures. A genetic algorithm is used to generate a maximum likelihood estimate of the gyro biases, thus alleviating the potential computational burden problem associated with the number of required particles. In fact, running with as few as 150 quaternion particles and a 200-element population for the genetic-algorithm bias estimation scheme, the algorithm is shown in simulations to be amenable to real-time implementations.

The resulting combined algorithm is shown via an extensive simulation study to be superior to both the (regular and iterated) additive and the multiplicative versions of the extended Kalman quaternion estimators and to the recently proposed unscented Kalman quaternion estimator. In particular, this superiority is manifested in the new filter's faster convergence rate and enhanced robustness to initial conditions uncertainty. The simulations include a case where a real measurement noise model, derived from the Technion's TechSAT satellite data, is used. Characterized by a non-Gaussian measurement noise, this case is used to demonstrate the ability of the new algorithm to cope with conditions where all Kalman filter variants (including the unscented filter) fail. A comparison of the estimation error covariance of the quaternion particle filter to the Cramér–Rao lower bound shows that the new estimator is asymptotically efficient in the statistical sense, thus corroborating its optimality.

Appendix A: GA–QPF Algorithm

A single cycle of the GA–QPF algorithm is summarized in the following informal pseudocode:

1) Obtain the bias estimate $\hat{\boldsymbol{\Upsilon}}_{k-1}$ from the previous time step.

2) Perform particle evolution to obtain a new particle set

$$\boldsymbol{q}_k(i) = \Phi(\boldsymbol{\omega}_k - \hat{\boldsymbol{\Upsilon}}_{k-1})\boldsymbol{q}_{k-1}(i) \qquad i = 1, \ldots, N \tag{A1}$$

3) IF {new measurement Y_k is available}

a) Update the importance weights using Eq. (19), and normalize for all $i = 1, \ldots, N$.

b) Compute the filtered quaternion as described in Sec. IV.C.

c) Update $\varphi_k((\boldsymbol{\Upsilon}_{k-1}(i))$ using Eq. (54), and normalize using Eq. (55), for all $i = 1, \ldots, N_\eta$.

d) Derive the bias estimate by computing the weighted average of all parameters in the population

$$\hat{\boldsymbol{\Upsilon}}_k = \sum_{i=1}^{N_\eta} \tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i))\boldsymbol{\Upsilon}_k(i) \tag{A2}$$

e) IF $\{N_{\text{eff}} < N_{\text{th}}\}$

i) Compute the regularization intensity using Eq. (35).

ii) Compute the number of offspring $N_k(i)$ for each particle $\boldsymbol{q}_k(i)$, $i = 1, \ldots, N$.

iii) Generate regularized weighted offspring as described in Sec IV.D.

iv) GA Iteration: Apply reproduction, crossover, and mutation to obtain a new bias parameter set $\{\boldsymbol{\Upsilon}_k(i)\}_{i=1}^{N_\eta}$.

v) Set the fitness of the new bias parameter set as $\tilde{\varphi}_k(\boldsymbol{\Upsilon}_k(i)) = 1/N_\eta$ for all $i = 1, \ldots, N_\eta$.

f) END IF

4) ELSE {Time propagation only: Compute the filtered quaternion as described in Sec. IV.C}

5) END IF

Appendix B: QPF Initialization Scheme

The QPF initialization algorithm is described in the following pseudocode:

1) Set z, an integer number, such that N/z is also integer.

2) Set $\Delta\beta = 2\pi/(N/z)$, $\beta = 0$, $i = 1$. Go through the following double recursion:

a) FOR $m = 1, \ldots, N/z$

i) FOR $j = 1, \ldots, z$

ii) Draw $\boldsymbol{\zeta}_1 \sim \mathcal{N}(0_{3\times 1}, \text{Cov}(\delta\boldsymbol{b}_0))$, $\zeta_2 \sim \mathcal{N}(0, \Delta\beta^2)$

iii) $\bar{\boldsymbol{b}} = \boldsymbol{b}_0 + \boldsymbol{\zeta}_1$, $\beta_0 = \frac{1}{2}\arccos(\bar{\boldsymbol{b}}^T\boldsymbol{r}_0/(\|\bar{\boldsymbol{b}}\|\|\boldsymbol{r}_0\|))$

iv) $\boldsymbol{e} = [(\bar{\boldsymbol{b}} \times \boldsymbol{r}_0)^T/\|\bar{\boldsymbol{b}} \times \boldsymbol{r}_0\| \sin\beta_0 \quad \cos\beta_0]^T$

v) $\delta\boldsymbol{e} = [\bar{\boldsymbol{b}}^T/\|\bar{\boldsymbol{b}}\| \sin\frac{1}{2}(\zeta_2 + \beta) \quad \cos\frac{1}{2}(\zeta_2 + \beta)]^T$

vi) $\boldsymbol{q}_0(i) = \delta\boldsymbol{e} \otimes \boldsymbol{e}$

vii) $i = i + 1$

viii) END FOR

b) $\beta = \beta + \Delta\beta$

c) END FOR

Acknowledgments

This research was supported by the Israel Science Foundation (Grant 1032/04) and by the Technion's Asher Space Research Fund. The kind permission by John L. Crassidis and F. Landis Markley to use their original USQUE computer code is deeply appreciated. The authors express their gratitude to M. Guelman and A. Shiryaev of the Technion's Asher Space Research Institute for providing the TechSAT data.

References

[1]Wahba, G., "A Least-Squares Estimate of Satellite Attitude. Problem 65-1," *SIAM Review*, Vol. 7, No. 3, 1965, p. 409.

[2]Lefferts, E. J., Markley, F. L., and Shuster, M. D., "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, 1982, pp. 417–429.

[3]Bar-Itzhack, I. Y., and Oshman, Y., "Attitude Determination from Vector Observations: Quaternion Estimation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-21, No. 1, 1985, pp. 128–136.

[4]Shuster, M., "Kalman Filtering of Spacecraft Attitude and the Quest Model," *The Journal of the Astronautical Sciences*, Vol. 38, No. 3, 1990, pp. 377–393.

[5]Gai, E., Daly, K., Harrison, J., and Lemos, L., "Star-Sensor-Based Satellite Attitude/Attitude Rate Estimator," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 5, 1985, pp. 560–565.

[6]Crassidis, J. L., and Markley, F. L., "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, 2003, pp. 536–542.

[7]Oshman, Y., and Carmi, A., "Spacecraft Attitude Estimation from Vector Observations Using a Fast Particle Filter," *Proceedings of 14th AAS/AIAA Space Flight Mechanics Conference*, Vol. 119, Univelt, Inc., San Diego, CA, 2005, pp. 593–608; also AAS Paper AAS 04-141, 2004.

[8]Oshman, Y., and Carmi, A., "Estimating Attitude from Vector Observations Using a Genetic Algorithm-Embedded Quaternion Particle Filter," AIAA Paper 2004-5340, 2004.

[9]Doucet, A., Godsill, S., and Andrieu, C., "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, Vol. 10, No. 3, 2000, pp. 197–208.

[10]Cheng, Y., and Crassidis, J., "Particle Filtering for Sequential Spacecraft Attitude Estimation," AIAA Paper 2004-5337, 2004.

[11]Geweke, J., "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica*, Vol. 57, No. 6, 1989, pp. 1317–1339.

[12]Doucet, A., de Freitas, N., and Gordon, N. (eds.), *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, Springer-Verlag, New York, 2001.

[13]Golub, G. H., and Van Loan, C. F., *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[14]Davenport, P. D., "A Vector Approach to the Algebra of Rotations with Applications," NASA TN D-4696, Aug. 1968.

[15]Wertz, J. R. (ed.), *Spacecraft Attitude Determination and Control*, D. Reidel, Dordrecht, The Netherlands, 1984.

[16]Kong, A., Liu, J., and Wong, W. H., "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, Vol. 89, No. 425, 1994, pp. 278–288.

[17]MacCormick, J., and Isard, M., "Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking," *Proceedings of 6th European Conference on Computer Vision*, Vol. 2, edited by D. Vernon, Lecture Notes in Computer Science, Vol. 1843, Springer-Verlag, London, 2000, pp. 3–19.

[18]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989.

[19]Tichavský, P., Muravchik, C. H., and Nehorai, A., "Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering," *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, 1998, pp. 1386–1396.

[20]Denham, W. F., and Pines, S., "Sequential Estimation When Measurement Function Nonlinearity Is Comparable to Measurement Error," *AIAA Journal*, Vol. 4, No. 6, 1966, pp. 1071–1076.

[21]Shiryaev, A., Waller, R., Roznov, M., and Guelman, M., "Design And Experimental Testing of Three-Axis Satellite Attitude Control Systems That Use Only Magnetic Torquers," Norman and Helen Asher Space Research Inst., Technion—Israel Institute of Technology, Tech. Rept. 2002.07, Haifa, June 2002.