Attitude Estimation with Feedback Particle Filter

Chi Zhang, Amirhossein Taghvaei and Prashant G. Mehta

Abstract—This paper presents theory, application, and comparisons of the feedback particle filter (FPF) algorithm for the problem of attitude estimation. The paper builds upon our recent work on the exact FPF solution of the continuous-time nonlinear filtering problem on compact Lie groups. In this paper, the details of the FPF algorithm are presented for the problem of attitude estimation—a nonlinear filtering problem on \( SO(3) \). The quaternions are employed for computational purposes. The algorithm requires a numerical solution of the filter gain function, and two methods are applied for this purpose. Comparisons are also provided between the FPF and some popular algorithms for attitude estimation on \( SO(3) \), including the invariant EKF, the multiplicative EKF, and the unscented Kalman filter. Simulation results are presented that help illustrate the comparisons.

I. INTRODUCTION

Attitude estimation is important to numerous fields including localization of mobile robots [7], [4], [24], visual tracking of objects [18], [28], and navigation of spacecrafts [22], [14]. The mathematical problem of attitude estimation is a nonlinear filtering problem on a matrix Lie group, in particular the special orthogonal group \( SO(3) \). The design of attitude filters thus requires consideration of the geometry of the manifold.

A number of attitude filters have been proposed and applied for the aerospace applications. A majority of these filters are based on the extended Kalman filter (EKF), e.g. the additive EKF [2], [19] and the multiplicative EKF [30], [32]. The EKF-based filters require a linearized model of the estimation error. Such a model is typically derived using one of the many three-dimensional attitude representations, e.g. the Euler angle [1], the rotation vector [35], and the modified Rodrigues parameter [21]. These representations have also been employed in the construction of unscented Kalman filters [22], [16]. More recently, group-theoretic methods for attitude estimation have been explored. Deterministic nonlinear observers that respect the intrinsic geometry of the Lie groups have appeared in [31], [27], [41], [8], [10]. A class of symmetry-preserving observers have been proposed to exploit certain invariance properties [11], [12], leading to the invariant EKF algorithm [13], [6], [3], the invariant ensemble EKF [6], and the invariant particle filter [5] within the stochastic filtering framework. Filters based on certain variational formulations on Lie groups have also been investigated [44], [9], [26]. Particle filters for attitude estimation include the bootstrap particle filter [15], [34], the marginalized particle filter [40], and the Rao-Blackwellized particle filter [38]. For more comprehensive review and performance comparison of the various attitude filters, c.f., [23], [43], [25]. Some of these filters are also described in Sec. V for the purpose of comparisons with the proposed FPF algorithm.

The feedback particle filter (FPF) is an exact algorithm for the solution of the continuous-time nonlinear filtering problem. The FPF algorithm was originally proposed in the Euclidean setting of \( \mathbb{R}^n \) [42]. In a recent paper from our group, the FPF was extended to filtering on compact matrix Lie groups [45]. The FPF is an intrinsic algorithm: The particle dynamics, expressed in their Stratonovich form, respect the geometric constraints of the manifold. The update step in FPF has a gain-feedback structure where the gain needs to be obtained numerically as a solution to a certain linear Poisson equation. When the gain function can be exactly computed, the FPF is an exact algorithm. In this case, in the limit of large number of particles, the empirical distribution of the particles exactly matches the posterior distribution of the hidden state.

The contributions of this paper are as follows:

- **FPF algorithm for attitude estimation.** The FPF algorithm is presented for the problem of attitude estimation. The explicit form of the filter is described with respect to both the rotation matrix and the quaternion coordinate, with the latter being demonstrated for computational purposes.

- **Numerical solution of the gain function.** The FPF algorithm requires numerical approximation of the gain function as a solution to a linear Poisson equation on the Lie group. For this purpose, two numerical methods are proposed: In a Galerkin scheme, the gain function is approximated with a set of pre-defined basis functions. The second scheme involves solving a fixed-point equation associated with the weighted Laplacian operator on the manifold.

- **Comparison of attitude filters.** For the purpose of comparison, the invariant EKF, the multiplicative EKF, and the UKF algorithms are briefly reviewed. Simulation studies are presented to compare performance between these filters and the proposed FPF algorithm.

The remainder of this paper is organized as follows: After a brief review in Sec. II of the relevant Lie group preliminaries, the problem of attitude estimation is formulated in Sec. III. The FPF algorithm on \( SO(3) \) is described in IV and some other attitude filters are briefly reviewed in Sec. V. Numerical simulations are contained in Sec. VI.
II. MATHEMATICAL PRELIMINARIES

Geometry of $SO(3)$: The special orthogonal group $SO(3)$ is the group of $3 \times 3$ matrices $R$ such that $RR^T = I$ and $\det(R) = 1$. The Lie algebra $so(3)$ is the 3-dimensional inner product space of skew-symmetric matrices. The inner product is denoted as $\langle \cdot, \cdot \rangle_{so(3)}$. Given an orthonormal basis $\{E_1, E_2, E_3\}$, a vector $ω = (ω_1, ω_2, ω_3) \in \mathbb{R}^3$ is uniquely mapped to an element in $so(3)$, denoted as $(ω)_x := ω_1 E_1 + ω_2 E_2 + ω_3 E_3$. The exponential map of $Ω ∈ so(3)$ is denoted as $\exp(Ω)$, and the space of smooth real-valued functions $f : SO(3) → \mathbb{R}$ is denoted as $C^∞(G)$, where we write $G$ interchangeably as $SO(3)$.

Vector field: The Lie algebra is identified with the tangent space at the identity matrix $I ∈ SO(3)$, and used to construct a basis $\{E_1^R, E_2^R, E_3^R\}$ for the tangent space at $R ∈ SO(3)$, where $E_n^R := RE_n$ for $n = 1, 2, 3$. Therefore, a smooth vector field, denoted as $\mathcal{V}$, is expressed as,

$$\mathcal{V}(R) = v_1(R)E_1^R + v_2(R)E_2^R + v_3(R)E_3^R,$$

with $v_n(R) ∈ C^∞(G)$. We write $\mathcal{V} = RV$, where $V(R) := v_1(R)E_1 + v_2(R)E_2 + v_3(R)E_3$ is an element of $so(3)$. The functions $(v_1(R), v_2(R), v_3(R))$ are called coordinates of $\mathcal{V}$. The inner product of two vector fields is

$$\langle \mathcal{V}, \mathcal{W} \rangle := \langle V, W \rangle_{so(3)}(R) = \sum_{n=1}^{3} v_n(R)w_n(R).$$

With a slight abuse of notation, the action of the vector field $\mathcal{V}$ on $f ∈ C^∞(G)$ is denoted as,

$$V(f)(R) := \frac{d}{dt} \bigg|_{t=0} f(R\exp(tV(R))).$$

A smooth function, denoted as $\text{div}\mathcal{V}$, is then defined as,

$$\text{div}\mathcal{V}(R) = \sum_{n=1}^{3} E_n \cdot v_n(R).$$

We also define the vector field $\text{grad}(\phi)$ for $\phi ∈ C^∞(G)$ as,

$$\text{grad}(\phi)(R) = RK(\phi),$$

where $K(R) ∈ so(3)$, with coordinates $(k_1(R), k_2(R), k_3(R)) := (E_1·\phi(R), E_2·\phi(R), E_3·\phi(R)).$

Apart from $C^∞(G)$, we also consider the following function spaces: For a probability measure $μ$ on $G$, $L^2(G, μ)$ denotes the Hilbert space of functions on $G$ that satisfy $\int |f|^2 dμ < ∞$ (here $\pi([f]) := \int_G |f|^2 dμ$; $H^1(G, μ)$ denotes the Hilbert space of functions $f$ such that $f$ and $E_n · f$ (defined in the weak sense) are all in $L^2(G, μ)$).

Quaternions: Quaternions provide a computationally efficient coordinate representation for $SO(3)$. A unit quaternion has the general form

$$q = (q_0, q_1, q_2, q_3) = \left(\cos\left(\frac{θ}{2}\right), \sin\left(\frac{θ}{2}\right)ω_1, \sin\left(\frac{θ}{2}\right)ω_2, \sin\left(\frac{θ}{2}\right)ω_3\right),$$

and represents rotation of angle $θ$ about the axis defined by the unit vector $(ω_1, ω_2, ω_3)$. As with $SO(3)$, the space of quaternions admits a Lie group structure: The identity quaternion is $q_1 = (1, 0, 0, 0)$, the inverse of $q$ is $q^{-1} = (q_0, -q_1, -q_2, -q_3)$, and the multiplication is defined as,

$$p ⊗ q = \left[ p_0q_0 - p_V · q_V, p_0q_V + p_0q_V + p_V × q_V \right],$$

where $p_V = (p_1, p_2, p_3)$, $q_V = (q_1, q_2, q_3)$, and $·$ and $×$ denote the dot product and the cross product of two vectors.

Given a unit quaternion $q$, the corresponding rotation matrix $R = R(q) ∈ SO(3)$ is calculated by,

$$R = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 2q_0^2 + 2q_2^2 - 1 & 2(q_2q_3 - q_1q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_1q_1) & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}.$$  

(1)

For more comprehensive introduction of Lie groups and quaternions, we refer the reader to [17], [37].

III. ATTITUDE ESTIMATION PROBLEM STATEMENT

A. Process model

A kinematic model of rigid body is given by,

$$dR_t = R_tΩ_t dt + R_t \circ [dB_t],$$

(2)

where $R_t ∈ SO(3)$ is the orientation of the rigid body at time $t$, expressed with respect to an inertial frame, $Ω_t = [ω_t]_x$ represents the angular velocity expressed in the body frame, and $B_t$ is a standard Wiener process in $R^3$. Both $Ω_t$ and $[dB_t]_x$ are elements of $so(3)$. The $o$ before $dB_t$ indicates that the stochastic differential equation (sde) is expressed in its Stratonovich form.

Using the quaternion coordinate, $q_t$ is written as,

$$dq_t = \frac{1}{2}q_t ⊗ (ω_t dt + dB_t),$$

(3)

where, by a slight abuse of notation, $ω_t ∈ R^3$ is interpreted as a quaternion $(0, ω_t)$, and $dB_t$ is interpreted similarly. The sde is also interpreted in the Stratonovich sense.

B. Measurement model

Accelerometer: In the absence of translational motion, the accelerometer is modeled as,

$$dZ^a_t = R_t^T r^a dt + dW^a_t,$$

(4)

where $r^a ∈ R^3$ is the unit vector in the inertial frame aligned with the gravity, and $W^a_t$ is a standard Wiener process in $R^3$.

Magnetometer: The model of the magnetometer is of a similar form,

$$dZ^b_t = R_t^T r^b dt + dW^b_t,$$

(5)

where $r^b ∈ R^3$ is the unit vector in the inertial frame aligned with the local magnetic field, and $W^b_t$ is a standard Wiener process in $R^3$. 

C. Nonlinear filtering problem on SO(3)

In terms of the process and measurement models, the nonlinear filtering problem for attitude estimation is succinctly expressed as,

\[ \begin{align*}
    dR_t &= R_t \Omega_t \, dt + R_t \circ \left[ dB_t \right]_x, \quad (6a) \\
    dZ_t &= h(R_t) \, dt + dW_t, \quad (6b)
\end{align*} \]

where \( \Omega_t = [\omega_t]_x \) is the angular velocity, \( h : SO(3) \to \mathbb{R}^m \) is a given nonlinear function whose \( j \)-th coordinate is denoted as \( h_j \) (i.e. \( h = (h_1, h_2, ..., h_m) \)), and \( W_t \) is a standard Wiener process in \( \mathbb{R}^m \). Note that \( (6b) \) encapsulates the sensor models given in \( \textbf{4} \) and \( \textbf{5} \) with a single equation. For the purpose of this paper, it is not necessary to assume that the models are linear. It is assumed that \( B_t \) and \( W_t \) are mutually independent, and independent of the initial condition \( R_0 \) which is drawn from a known initial distribution, denoted as \( \pi_0^i \).

The objective of the attitude estimation problem, described by \( (6a) \) and \( (6b) \), is to compute the conditional distribution of \( R_t \) given the history of measurements (filtration) \( \mathcal{Z}_t = (Z_s)_{s \leq t} \). The conditional distribution, denoted as \( \pi_t^x \), acts on a function \( f \in C^\infty(G) \) according to,

\[ \pi_t^x (f) := E[f(R_t) | \mathcal{Z}_t]. \]

**Remark 1:** There are a number of simplifying assumptions implicit in the model defined in \( (6a) \) and \( (6b) \). In practice, \( \omega_t \) needs to be estimated from noisy gyroscope measurements and there is translational motion as well. This will require additional models which can be easily incorporated within the proposed filtering framework.

The purpose of this paper is to elucidate the geometric aspects of the FPF in the simplest possible setting of \( SO(3) \). More practical FPF-based filters that also incorporate models for translational motion, measurements of \( \omega_t \) from gyroscope, effects of translational motion on accelerometer, and effects of sensor bias are subject of separate publication. ■

IV. FEEDBACK PARTICLE FILTER ON SO(3)

A. FPF on SO(3)

The feedback particle filter is a controlled system with \( \mathcal{N} \) stochastic processes \( \{R_t^i \}_{i=1}^\mathcal{N} \) where \( R_t^i \in SO(3) \). The conditional distribution of the particle \( R_t^i \) given \( \mathcal{Z}_t^i \) is denoted by \( \pi_t^i \), which acts on \( f \in C^\infty(G) \) according to,

\[ \pi_t^i (f) := E[f(R_t^i) | \mathcal{Z}_t^i]. \]

The dynamics of the \( i \)-th particle is defined by,

\[ \begin{align*}
    dR_t^i &= R_t^i \Omega_t \, dt + R_t^i \circ \left[ dB_t^i \right]_x + R_t^i \circ \left[ K(R_t^i, t) \circ dt \right]_x, \quad (7) \\
    d\mathcal{Z}_t^i &= h(R_t^i) \, dt + dW_t^i
\end{align*} \]

where \( \{B_t^i \}_{i=1}^\mathcal{N} \) are mutually independent standard Wiener processes in \( \mathbb{R}^3 \), and \( R_0^i \) is drawn from the initial distribution \( \pi_0^i \). The \( i \)-th particle implements the Bayesian update step – to account for the conditioning due to the measurements – as gain \( K(R_t^i) \) times an error \( d\mathcal{I}_t^i \). The resulting control input to the \( i \)-th particle is an element of the Lie algebra so(3).

The error \( d\mathcal{I}_t^i \) is a modified form of the innovation process:

\[ d\mathcal{I}_t^i = d\mathcal{Z}_t^i - \frac{1}{2}(h(R_t^i) + \hat{h}) \, dt, \quad (8) \]

where \( \hat{h} := \pi_t(h) \). In a numerical implementation, we approximate \( \hat{h} = \frac{1}{N} \sum_{i=1}^{\mathcal{N}} h(R_t^i) = \hat{h}(\mathcal{N}) \).

The gain function \( K \) is a \( 3 \times m \) matrix whose entries are obtained as follows: For \( j = 1, 2, ..., m \), the \( j \)-th column of \( K \) is the coordinate of the vector field \( \nabla (\phi_j) \), where the function \( \phi_j \in H^1(G; \pi) \) is a solution to the Poisson equation,

\[ \pi_t \left( \left( \nabla (\phi_j), \nabla (\psi) \right) \right) = \pi_t \left( (h_j - \hat{h}_j) \psi \right), \]

\[ \pi_t (\phi_j) = 0 \quad \text{(normalization)}, \]

for all \( \psi \in H^1(G; \pi) \). This linear partial differential equation (pde) has to be solved for each \( j = 1, 2, ..., m \), and for each time \( t \geq 0 \). The existence-uniqueness of the solution of \( (9) \) requires additional assumptions on \( \pi_t \); c.f., [29].

**Assumption 1:** The distribution \( \pi_t \) is absolutely continuous with respect to the uniform (Lebesgue) measure on \( SO(3) \) with a positive density function \( \rho \).

Two numerical schemes for approximating the solution of \( (9) \) appear in Sec. IV-C and Sec. IV-D, respectively. For the FPF \( (7), (9) \), the following result is proved in [45] that relates \( \pi_t \) to \( \pi_t^i \):

**Theorem 1:** Consider the particle system that evolves according to \( (7) \), where the gain function is obtained as solution to the Poisson equation \( (9) \), and the error is defined as in \( (8) \). Suppose that Assumption 1 holds. Then assuming \( \pi_0 = \pi_0^i \), we have

\[ \pi_t (f) = \pi_t^i (f), \]

for all \( t > 0 \) and all function \( f \in C^\infty(G) \).

B. Quaternion representation

For numerical purposes, it is convenient to express the FPF with respect to the quaternion coordinate. In this coordinate, the dynamics of the \( i \)-th particle evolves according to,

\[ dq_t^i = \frac{1}{2} q_t^i \odot dq_t^i, \]

where \( q_t^i \) is the quaternion state of the \( i \)-th particle, and \( dq_t^i \in \mathbb{R}^3 \) evolves according to,

\[ dq_t^i = \omega_t \, dt + d\mathcal{Z}_t^i + K(q_t^i) \circ \left( d\mathcal{Z}_t^i - \frac{h(q_t^i) + \hat{h}}{2} \, dt \right), \]

where \( K(q, t) = K(R(q), t) \) and \( h(q) = h(R(q)) \), with \( R = R(q) \) given by the formula \( (1) \).

C. Galerkin gain function approximation

In this section, a Galerkin scheme is presented to approximate the solution of the Poisson equation \( (9) \). Since the equations for each \( j = 1, 2, ..., m \) are uncoupled, without loss of generality, a scalar-valued measurement is assumed (i.e., \( m = 1 \) and \( \phi_j, h_j \) are denoted as \( \phi, h \)). As the time \( t \) is fixed,
the explicit dependence on $t$ is suppressed (i.e., we denote $\pi_t$ as $\pi, R_t^i$ as $R^i$). This notation is also used in Sec. [IV-D]

In a Galerkin scheme, the solution $\phi$ is approximated as,

$$\phi = \sum_{i=1}^{L} \kappa_i \psi_i,$$

where $\{\psi_i\}_{i=1}^{L}$ is a given (assumed) set of basis functions on $SO(3)$. The gain function $K = (k_1, k_2, k_3)$, defined as the coordinates of $\text{grad}(\phi)$, is then given by,

$$k_n = \sum_{i=1}^{L} \kappa_i E_n \cdot \psi_i, \quad n = 1, 2, 3.$$

The finite-dimensional approximation of the Poisson equation (9) is to choose coefficients $\{\kappa_i\}_{i=1}^{L}$ such that,

$$\sum_{i=1}^{L} \kappa_i (\langle \text{grad}(\psi_i), \text{grad}(\psi) \rangle) = \pi (\langle h - \hat{h} \rangle \psi),$$

for all $\psi \in \text{span}\{\psi_1, ..., \psi_L\} \subset \mathbb{H}^1(G; \pi)$. On taking $\psi = \psi_1, ..., \psi_L$, (12) is compactly written as a linear matrix equation,

$$A \kappa = b,$$

where $\kappa := (\kappa_1, ..., \kappa_L)$. The $L \times L$ matrix $A$ and the $L \times 1$ vector $b$ are defined and approximated as,

$$[A]_{kl} = \pi (\langle \text{grad}(\psi_l), \text{grad}(\psi_k) \rangle)$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \langle \text{grad}(\psi_l)(R^i), \text{grad}(\psi_k)(R^i) \rangle,$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{L} (E_n \cdot \psi_l)(R^i)(E_n \cdot \psi_k)(R^i),$$

$$b_k = \pi (\langle h - \hat{h} \rangle \psi_k) \approx \frac{1}{N} \sum_{i=1}^{N} \langle h(R^i) - \hat{h} \rangle \psi_k(R^i),$$

where recall $\hat{h} \approx \frac{1}{2} \sum_{i=1}^{N} h(R^i) =: \hat{h}_N$.

Note that both the Poisson equation (9) as well as its Galerkin finite-dimensional approximation (13) are coordinate-free representations. Particle-based approximations of (13), viz. (14) and (15), may be obtained using $R$ or $q$, or any other coordinate representation.

The choice of basis function is crucial in the Galerkin scheme, and one choice appears in Appendix A.

### D. Kernel-based gain function approximation

In a kernel-based scheme, the solution to the Poisson equation (9) is the solution of the following fixed-point equation for fixed positive $\tau$,

$$\phi = e^{t\Delta_p} \phi + \int_0^{t} e^{s\Delta_p} (h - \hat{h}) ds,$$

where $e^{t\Delta_p}$ is the semigroup associated with the weighted Laplacian on $SO(3)$, defined as $\Delta_p := (1/\rho) \text{div}(\rho \text{grad}(\phi))$, where $\rho$ is the density of $\pi$. For small time $\tau = \varepsilon$, the operator $e^{\varepsilon\Delta_p}$ is approximated using the matrix as,

$$e^{\varepsilon\Delta_p} \phi(R) \approx \frac{1}{N} \sum_{i=1}^{N} k^{(\varepsilon,N)}(R, R_i) \phi(R_i),$$

where the kernel $k^{(\varepsilon,N)}: \text{SO}(3) \times \text{SO}(3) \to \mathbb{R}$ is given by,

$$k^{(\varepsilon,N)}(R_1, R_2) = \frac{g^{(\varepsilon)}(R_1, R_2)}{\sqrt{\frac{1}{N} \sum_{i=1}^{N} g^{(\varepsilon)}(R_1, R_i)} \sqrt{\frac{1}{N} \sum_{i=1}^{N} g^{(\varepsilon)}(R_2, R_i)}},$$

and the Gaussian kernel $g^{(\varepsilon)}$ is defined as,

$$g^{(\varepsilon)}(R_1, R_2) := \frac{1}{(4\pi \varepsilon)^{3/2}} \exp \left( -\frac{|R_1 - R_2|_F^2}{4\varepsilon} \right),$$

where $\varepsilon$ is a small positive parameter, and $| \cdot |_F$ denotes the Frobenius norm of a matrix. The justification for the approximation (17) appears in (20).

The approximation (17) yields a finite-dimensional approximation of the fixed-point equation (16):

$$\Phi = T^{(\varepsilon)} \Phi + \varepsilon H^{(\varepsilon)},$$

where $\Phi \in \mathbb{R}^N$ is the approximate solution that needs to be computed, $H^{(\varepsilon)} = (h(R^i) - \hat{h}_N, h(R^2) - \hat{h}_N, ..., h(R^N) - \hat{h}_N)$, and $T^{(\varepsilon)} \in \mathbb{R}^{N \times N}$ whose entries are given by,

$$T^{(\varepsilon)}_{ij} = \frac{k^{(\varepsilon,N)}(R_i, R_j)}{\sum_{i=1}^{N} k^{(\varepsilon,N)}(R_i, R_i)}.$$

Note that $T^{(\varepsilon)}$ is a stochastic matrix with positive entries, and as a result, the fixed-point equation (20) is a contraction on the space of normalized vectors. The solution can be obtained by successive approximations. The solution of (16), evaluated at the particles, is then approximated as $\Phi(R) \approx \Phi_i$, the $i$-th entry of $\Phi$.

The gain function is given by $K = (k_1, k_2, k_3)$, where $k_n = E_n \cdot \phi$ for $n = 1, 2, 3$, and is evaluated at the particles according to,

$$E_n \cdot \phi(R) = -\varepsilon E_n \cdot h(R) + \frac{1}{2\varepsilon} \left[ (S_n \Phi)_i - (S_n 1)_i (T^{(\varepsilon)} \Phi) \right],$$

where $1 = (1, 1, ..., 1) \in \mathbb{R}^N$, and the entries of the $N \times N$ matrix $S_n$ are given by,

$$(S_n)_{ij} = T^{(\varepsilon)}_{ij} \text{Tr}(R_i E_n R^i),$$

where $\text{Tr}(\cdot)$ denotes the trace of a matrix.

**Remark 2:** The theory for the kernel-based gain function approximation, together with its convergence analysis and numerical illustration, appears in a companion paper [36].

### E. FPF algorithm

The FPF algorithm is numerically implemented using the quaternion coordinate, and is described in Algorithm 1. The algorithm simulates $N$ particles, $\{q_i^N\}_{i=1}^{N}$, according to the sde’s (10) and (11), with the initial conditions $\{q_i^0\}_{i=1}^{N}$, sampled i.i.d. from a given prior distribution $\pi_0$. The gain function is approximated using either the Galerkin scheme (see Sec. [IV-C] and Algorithm 2), or the kernel-based scheme (see Sec. [IV-D] and Algorithm 3).

Given a particle set $\{q_i^N\}_{i=1}^{N}$, its empirical mean is obtained as the eigenvector (with norm 1) of the $4 \times 4$ matrix $Q = \frac{1}{N} \sum_{i=1}^{N} q_i^N q_i^N^T$, corresponding to its largest eigenvalue [33].
Algorithm 1 Feedback Particle Filter on $SO(3)$

1: initialization: sample $\{q_i^0\}_{i=1}^N$ from $\pi_0^e$
2: Assign $t = 0$
3: iteration: from $t$ to $t + \Delta t$
4: Calculate $\hat{h}^{(N)} = \left(1/N\right)\sum_{i=1}^Nh(q_i^t)$
5: for $i = 1$ to $N$
6: Generate a sample, $\Delta q_i^t$, from $N(0, I)$
7: Calculate the error
$$\Delta q_i^t := \Delta Z_i - (1/2)(b(q_i^t) + \hat{h}^{(N)}) \Delta t$$
8: Calculate gain function $K(q_i^t, t)$
9: Calculate $\Delta q_i^t = \psi_i \Delta h + \sqrt{\Delta t} \Delta B_i^t + K(q, i) \Delta q_i^t$
10: Propagate the particle $q_i^t$ according to
$$q_i^{t+\Delta t} = q_i^t \otimes \left[\begin{array}{c}
\cos(\Delta \nu_i^t/2) \\
\sin(\Delta \nu_i^t/2)
\end{array}\right]$$
(\cdot|\cdot denotes the Euclidean norm in $\mathbb{R}^3$)
11: end for
12: Define matrix $Q = \frac{1}{N} \sum_{i=1}^N q_i^{t+\Delta t} q_i^{t+\Delta t, T}$
13: return: empirical mean of $\{q_i^{t+\Delta t}\}_{i=1}^N$, i.e., the eigenvector of $Q$ associated with its largest eigenvalue
14: Assign $t = t + \Delta t$

Algorithm 2 Galerkin gain function approximation

1: input: Particles $\{q_{i}^{(k)}\}_{i=1}^{N}$
2: Calculate $\hat{h}^{(N)} = \left(1/N\right)\sum_{i=1}^Nh(q_{i}^{(k)})$
3: for $k = 1$ to $K$
4: Calculate $b_k = \frac{1}{N} \sum_{i=1}^N \left(h(q_{i}^{(k)}) - \hat{h}^{(N)}\right) \psi_k(q_{i}^{(k)})$
5: for $l = 1$ to $L$
6: Calc. $A_{kl} = \frac{1}{N} \sum_{i=1}^N \sum_{n=1}^3 (E_n \cdot \psi_l)(q_{i}^{(k)}) (E_n \cdot \psi_l)(q_{i}^{(k)})$
7: end for
8: end for
9: Solve the matrix equation $A_k b = b$, with $A = [A_{kl}]$, $b = [b_k]$
10: Calculate $k_n(q^{(k)}) = \sum_{l=1}^L k_l E_n \cdot \psi_l(q^{(k)})$, for $n = 1, 2, 3$
11: return: $\{K(q^{(k)}) = \{k_1(q^{(k)}), k_2(q^{(k)}), k_3(q^{(k)})\}\}_{i=1}^N$

V. REVIEW OF SOME ATTITUDE FILTERS

In this section, we restrict our attention to the attitude estimation problem with linear observations of the form $h(R_i) = R_i^T r$ where $r$ is a known reference vector in the inertial frame (see the models of accelerometer and magnetometer in [1], [5]). A majority of the literature deals with such linear models. For discrete-time filters, it is convenient to define $Y_t := \frac{dY_t}{dt}$, whose model is formally expressed as,
$$Y_t = R_t^T r + \dot{W}_t,$$
where $\dot{W}_t$ is a white noise process in $\mathbb{R}^3$. In this section, we assume without loss of generality that the covariance matrix associated with $\dot{W}_t$ is the identity matrix.

The sequence of sampling instants is denoted as $\{\tau_n\}$, $n = 0, 1, 2, \ldots$, with uniform time step $\Delta t = \tau_{n+1} - \tau_n$. The discrete-time sampled measurements are denoted as $\{Y_n\}$. Similarly, $\{R_n\}$ and $\{\omega_n\}$ denote the discrete-time samples of $R_t$ and $\omega_t$. Furthermore, $\hat{R}_n$ denotes the posterior filter estimate at time $\tau_n$, $\hat{R}_{n|n-1}$ denotes the filter estimate after the propagation step but before the measurement update, and $\Sigma_{n|n-1}$, $\Sigma_n$ denote the associated covariance matrices.

A. Invariant extended Kalman filter

The invariant EKF (I-EKF) models the attitude at time $\tau_n$ as the product
$$R_n = \delta R_n \hat{R}_n,$$
(23)
where the estimation error, $\delta R_n \in SO(3)$, is represented as $\delta \Sigma_{n|n-1} = \exp([\eta_n]_x)$ where $\eta_n \in \mathbb{R}^3$. At each time step, the estimate of $\eta_n$, denoted as $\hat{\eta}_n$, is obtained as follows:
(i) Propagation step:
$$\hat{\eta}_{n|n-1} = \hat{\eta}_{n-1} \Delta \tau_x,$$
$$\Sigma_{n|n-1} = \Sigma_{n-1} + (\Delta \tau) I.$$
(ii) Update step: The innovation error is defined in the inertial frame,
$$I_n = \hat{R}_{n|n-1} Y_n - r,$$
and the gain matrix $K_n$ is calculated according to,
$$K_n = \Sigma_{n|n-1} H^T (H \Sigma_{n|n-1} H^T + I)^{-1},$$
where $H = [r]_x$.
(iii) Posterior update:
$$\hat{\eta}_n = K_n I_n,$$
$$\hat{R}_n = \exp([\hat{\eta}_n]_x) \hat{R}_{n|n-1}$$
$$\Sigma_n = (I - K_n H) \Sigma_{n|n-1}.$$
For more details of the IEKF algorithm, we refer the reader to [6].

B. Multiplicative extended Kalman filter

The multiplicative EKF (MEKF) models the attitude at time $t_n$ as the product

$$ R_n = \tilde{R}_n \delta R_n, $$

where the estimation error, $\delta R_n \in SO(3)$, is parameterized by some coordinate, e.g., the modified Rodrigues parameter [21]. The coordinate is denoted as $a_n \in \mathbb{R}^3$, and $\delta R_n = \delta R(a_n)$. At each time step, the estimate of $a_n$, denoted as $\hat{a}_n$, is obtained as follows:

(i) Propagation step:

$$ \tilde{R}_{n|n-1} = \tilde{R}_{n-1} \exp((\omega_{n-1} \Delta t) \times), $$

$$ \Sigma_{n|n-1} = \Lambda \Sigma_{n-1} \Lambda^T + (\Delta t) I, $$

where $\Lambda = I - [\omega_{n-1} \Delta t] \times$.

(ii) Update step:

$$ I_n = Y_n - \tilde{R}_{n|n-1} r, $$

$$ K_n = \Sigma_{n|n-1} H^T (H \Sigma_{n|n-1} H^T + I)^{-1}, $$

where $H = [\tilde{R}_{n|n-1} r] \times$. In contrast to the IEKF, the innovation error in the MEKF is defined in the body frame.

(iii) Posterior estimate:

$$ \hat{a}_n = K_n I_n, $$

$$ \tilde{R}_n = \tilde{R}_{n|n-1} \delta R(\hat{a}_n), $$

$$ \Sigma_n = (I - K_n H) \Sigma_{n|n-1}. $$

For more details of the MEKF algorithm, we refer the reader to [32], [37].

C. Unscented Kalman filter

The unscented Kalman filter (UKF) for attitude estimation, presented in [22], also uses the parameterization of the MEKF, i.e.,

$$ R_n = \tilde{R}_n \delta R(a_n), $$

where $a_n \in \mathbb{R}^3$ is the chosen coordinate. The estimate of $a_n$ is obtained by using a standard UKF in $\mathbb{R}^3$. For equations of the algorithm, we refer the reader to [22].

D. Other filters

Apart from the above, other types of attitude filters include the continuous-time IEKF [13], the geometric approximate minimum-energy (GAME) filter [44], and the bootstrap particle filter [15]. These filters are not included in the simulation-based comparisons that are presented next.

VI. SIMULATIONS

For numerical simulations of the filters, we consider the following attitude estimation problem,

$$ \text{d}q_t = \frac{1}{\Delta t} q_t \otimes (\omega_t + \Sigma_B \text{d}B_t), $$

$$ \text{d}Z_t = \begin{bmatrix} R(q_t)^T & 0 \\ 0 & R(q_t)^T \end{bmatrix} \begin{bmatrix} \text{r}^e \\ \text{r}^b \end{bmatrix} \text{d}t + \begin{bmatrix} \Sigma_W & 0 \\ 0 & \Sigma_W \end{bmatrix} \text{d}W_t, $$

where the angular velocity is given by [43],

$$ \omega_t = \left( \sin \left( \frac{2\pi}{15} t \right), -\sin \left( \frac{2\pi}{18} t + \frac{\pi}{20} \right), \cos \left( \frac{2\pi}{17} t \right) \right), $$

$r^e = (0, 0, -1)$ and $r^b = (1/\sqrt{2}, 0, 1/\sqrt{2})$ represent the direction of the gravity and the local magnetic field, and $\Sigma_B$ and $\Sigma_W$ are $3 \times 3$ diagonal matrices associated with the process noise and the sensor noise, respectively.

The following filters are implemented for the comparison:

1) IEKF: the algorithm is described in Sec. V-A.
2) MEKF: the algorithm is described in Sec. V-B, using the modified Rodrigues parameter.
3) UKF: the algorithm is described in Sec. V-C and [22], using the modified Rodrigues parameter.
4) FPF-G: the FPF using the Galerkin gain functions, as described in Sec. IV-C and Algorithm 2, with fixed basis functions defined in Appendix A.
5) FPF-K: the FPF using the kernel-based gain functions, as described in Sec. IV-D and Algorithm 3, with $\epsilon = 1$ and $K = 10$.

The performance metric is the root-mean-squared error (RMSE) [43], [25]:

$$ \text{RMSE} = \sqrt{(1/M) \sum_{j=1}^{M} (\delta \phi_j^f)^2}, $$

where $\delta \phi_j^f$ is the rotation angle error at time $t$ for the $j$-th Monte Carlo run, $j = 1, 2, ..., M$. The rotation angle error is defined as follows: Let $q_i$ and $\hat{q}_i$ denote the true and estimated attitude at time $t$, and let $\delta q_i := \hat{q}_i - q_i$ represent the estimation error, then $\delta \phi_i = 2 \arccos(|\delta q_0|)$, where $\delta q_0$ is the first component of $\delta q_i$.

The filters are initialized with a “concentrated Gaussian distribution” [39], denoted as $N(q_i, \Sigma_0)$, whose mean $q_i$ is the identity quaternion, and $\Sigma_0$ is a diagonal matrix representing the variance in each axis of the Lie algebra. The particles in the FPF algorithms are sampled from this distribution as follows: First, one generates samples $\{\nu^i\}_{i=1}^{N}$ from the Gaussian distribution $N(0, \Sigma_0)$ in $\mathbb{R}^3$. Then, the particles $\{R_{q_i}^0\}_{i=1}^{N}$ are obtained by $R_{q_i}^0 = \exp(\nu^i \times)$, and converted to the quaternions $\{q_{i0}\}_{i=1}^{N}$.

The simulations are conducted over a finite time-horizon $t \in [0, T]$ with fixed time step $\Delta t$. The process noise $\Sigma_B$ has standard deviation (std. dev.) of $5 \circ/s$. To avoid numerical instability due to large gain values, each of the first three measurement updates in all filters is implemented sequentially on a partition of the time step $\Delta t$ with $N_f$ uniform sub-intervals. For FPF-G, $N_f = 100$; For other filters, $N_f = 20$ when $\Sigma_W$ is large, and $N_f = 30$ when $\Sigma_W$ is small. The relevant parameters are listed in Table 1.
The algorithm was described using both the rotation matrix algorithm, possesses a gain-feedback structure and automatically respects the geometric constraint of the manifold.

In this paper, the feedback particle filter was presented for the problem of attitude estimation. The FPF is an intrinsic scheme for one measurement update is approximately linear in this case. The computational complexity of the Galerkin scheme yields significant error in computing the gain function.

The continuing research includes improving the computational efficiency of the gain function approximation, and application of FPF for attitude estimation with more complicated models with e.g., translation and sensor bias.

**TABLE 1: SIMULATION PARAMETERS**

| $\Sigma_0$ | $T$ | $\Delta t$ | $N$ | $M$ |
|------------|-----|------------|-----|-----|
| $0.00872^2 I$ | 2 | 0.01 | 200 | 100 |

Fig. 1 illustrates the filter performance with different initial distribution error and sensor noise. In Fig. 1 (a), $\Sigma_0 = 0.5236^2 I$, corresponding to the std. dev. of $30^\circ$, and the target is initialized from the same distribution. The sensor noise $\Sigma_w = 0.01745^2 I$, i.e., the std. dev. is $10^\circ$. In this case, all the filters have nearly identical performance.

In Fig. 1(b), $\Sigma_0 = 1.0472^2 I$, corresponding to the std. dev. of $60^\circ$, and the target is initialized with fixed attitude rotation of $180^\circ$ about the axis $(3,1,4)$. The sensor noise $\Sigma_w = 0.05236^2 I$, i.e., the std. dev. is $30^\circ$. These parameters indicate larger initial estimation error and uncertainty of the filters, and larger sensor noise. In this case, the FPF-K converges significantly faster than the other filters.

When the initial estimation error is large, the Galerkin scheme yields significant error in computing the gain functions, and thus FPF-G is not included in Fig. 1(b). It is expected that one would require additional basis functions in this case. The computational complexity of the Galerkin scheme for one measurement update is approximately linear in the number of particles, whereas it is approximately quadratic for computing the kernel-based gain functions.

**A. Basis functions in Galerkin scheme**

For the Galerkin scheme presented in Sec. IV-C, the following basis functions on $SO(3)$ are considered, expressed using the quaternion:

$$
\psi_1(q) = 2q_1q_0, \quad \psi_2(q) = 2q_1q_0, \\
\psi_3(q) = 2q_1q_0, \quad \psi_4(q) = 2q_0^2 - 1.
$$

In order to compute the matrix $A$ and the vector $b$ in the Galerkin scheme, the formulae for the action of $E_1$, $E_2$, $E_3$ on these basis functions are provided in Table 2.

**TABLE 2: ACTION OF $E_n$ ON BASIS FUNCTIONS**

| $E_1$ | $E_2$ | $E_3$ |
|-------|-------|-------|
| $\psi_1$ | $q_0^4 - q_1^4$ | $-q_1q_2 - q_3q_0$ | $-q_1q_3 + q_2q_0$ |
| $\psi_2$ | $-q_1q_2 + q_3q_0$ | $q_0^4 - q_1^4$ | $-q_2q_3 - q_1q_0$ |
| $\psi_3$ | $-q_1q_3 - q_2q_0$ | $-q_2q_3 + q_1q_0$ | $q_0^4 - q_1^4$ |
| $\psi_4$ | $-2q_1q_0$ | $-2q_2q_0$ | $-2q_3q_0$ |

**REFERENCES**

[1] I. Y. Bar-Itzhack and M. Idan. Recursive attitude determination from vector observations: Euler angle estimation. *Journal of Guidance, Control, and Dynamics*, 10(2):152–157, 1987.

[2] I. Y. Bar-Itzhack and Y. Oshman. Attitude determination from vector observations: quaternion estimation. *IEEE Transactions on Aerospace and Electronic Systems*, (1):128–136, 1985.

[3] M. Barczyk, S. Bonnabel, J. Deschaud, and F. Goulette. Invariant EKF design for scan matching-aided localization. *IEEE Transactions on Control Systems Technology*, 23(6):2440–2448, 2015.

[4] M. Barczyk and A. F. Lynch. Invariant observer design for a helicopter UAV aided inertial navigation system. *IEEE Transactions on Control Systems Technology*, 21(3):791–806, 2013.
