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.

Financial support from the NSF CMMI grants 1334987 and 1462773 is gratefully acknowledged.

C. Zhang, A. Taghvaei and P. G. Mehta are with the Coordinated Science Laboratory and the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign (UIUC) {czhang54; taghvae2; mehtapg}@illinois.edu

II. Mathematical Preliminaries

Geometry of $SO(3)$: The special orthogonal group $SO(3)$ is the group of 3×3 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 $\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ is uniquely mapped to an element in $so(3)$, denoted as $[\,\omega\,]_\times := \omega_1 E_1 + \omega_2 E_2 + \omega_3 E_3$. The exponential map of $\Omega \in so(3)$ is denoted as $\exp(\Omega)$, and the space of smooth real-valued functions $f : SO(3) \to \mathbb{R}$ is denoted as $C^\infty(G)$, where we write G interchangeably as $SO(3)$.

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

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

with $v_n(R) \in C^\infty(G)$. We write $\mathscr{V} = RV$, 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 $\big(v_1(R), v_2(R), v_3(R)\big)$ are called *coordinates* of $\mathscr{V}$. The inner product of two vector fields is

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

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

$$V\cdot f(R) := \frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0} f\big(R\exp(tV(R))\big).$$

A smooth function, denoted as $\mathrm{div}\mathscr{V}$, is then defined as,

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

We also define the vector field $\mathrm{grad}(\phi)$ for $\phi \in C^\infty(G)$ as,

$$\mathrm{grad}(\phi)(R) = R\,\mathsf{K}(R),$$

where $\mathsf{K}(R) \in so(3)$, with coordinates $\big(\mathsf{k}_1(R),\ \mathsf{k}_2(R),\ \mathsf{k}_3(R)\big) := \big(E_1\cdot\phi(R),\ E_2\cdot\phi(R),\ E_3\cdot\phi(R)\big)$.

Apart from $C^\infty(G)$, we also consider the following function spaces: For a probability measure π on G, $L^2(G;\pi)$ denotes the Hilbert space of functions on G that satisfy $\pi(|f|^2) < \infty$ (here $\pi(|f|^2) := \int_G |f|^2\,\mathrm{d}\pi$); $H^1(G;\pi)$ denotes the Hilbert space of functions f such that f and $E_n\cdot f$ (defined in the weak sense) are all in $L^2(G;\pi)$.

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

$$\begin{aligned} q &= (q_0,\ q_1,\ q_2,\ q_3) \\ &= \Big(\cos(\frac{\theta}{2}),\ \sin(\frac{\theta}{2})\omega_1,\ \sin(\frac{\theta}{2})\omega_2,\ \sin(\frac{\theta}{2})\omega_3\Big), \end{aligned}$$

and represents rotation of angle θ about the axis defined by the unit vector $(\omega_1, \omega_2, \omega_3)$. As with $SO(3)$, the space of quaternions admits a Lie group structure: The identity quaternion is $q_I = (1,0,0,0)$, the inverse of q is $q^{-1} = (q_0, -q_1, -q_2, -q_3)$, and the multiplication is defined as,

$$p\otimes q = \begin{bmatrix} p_0q_0 - p_V\cdot q_V \\ p_0q_V + q_0p_V + p_V\times q_V \end{bmatrix},$$

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

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

$$R = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 2q_0^2 + 2q_2^2 - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}. \tag{1}$$

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,

$$\mathrm{d}R_t = R_t\Omega_t\,\mathrm{d}t + R_t \circ [\,\mathrm{d}B_t\,]_\times, \tag{2}$$

where $R_t \in SO(3)$ is the orientation of the rigid body at time t, expressed with respect to an inertial frame, $\Omega_t = [\,\omega_t\,]_\times$ represents the angular velocity expressed in the body frame, and B_t is a standard Wiener process in $\mathbb{R}^3$. Both Ω_t and $[\,\mathrm{d}B_t\,]_\times$ are elements of $so(3)$. The $\circ$ before $\mathrm{d}B_t$ indicates that the stochastic differential equation (sde) (2) is expressed in its Stratonovich form.

Using the quaternion coordinate, (2) is written as,

$$\mathrm{d}q_t = \frac{1}{2} q_t \otimes (\omega_t\,\mathrm{d}t + \mathrm{d}B_t), \tag{3}$$

where, by a slight abuse of notation, $\omega_t \in \mathbb{R}^3$ is interpreted as a quaternion $(0, \omega_t)$, and $\mathrm{d}B_t$ is interpreted similarly. The sde (3) is also interpreted in the Stratonovich sense.

B. Measurement model

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

$$\mathrm{d}Z_t^g = R_t^T r^g\,\mathrm{d}t + \mathrm{d}W_t^g, \tag{4}$$

where $r^g \in \mathbb{R}^3$ is the unit vector in the inertial frame aligned with the gravity, and W_t^g is a standard Wiener process in $\mathbb{R}^3$.

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

$$\mathrm{d}Z_t^b = R_t^T r^b\,\mathrm{d}t + \mathrm{d}W_t^b, \tag{5}$$

where $r^b \in \mathbb{R}^3$ is the unit vector in the inertial frame aligned with the local magnetic field, and W_t^b is a standard Wiener process in $\mathbb{R}^3$.

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,

$$\mathrm{d}R_t = R_t\Omega_t\,\mathrm{d}t + R_t \circ [\,\mathrm{d}B_t\,]_\times, \tag{6a}$$
$$\mathrm{d}Z_t = h(R_t)\,\mathrm{d}t + \mathrm{d}W_t, \tag{6b}$$

where $\Omega_t = [\,\omega_t\,]_\times$ 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 (4) and (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 π_0^*.

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) $\mathscr{Z}_t = \sigma(Z_s : s \le t)$. The conditional distribution, denoted as π_t^*, acts on a function $f \in C^\infty(G)$ according to,

$$\pi_t^*(f) := \mathsf{E}[f(R_t)|\mathscr{Z}_t].$$

Remark 1: There are a number of simplifying assumptions implicit in the model defined in (6a) and (6b). In practice, ω_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 ω_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 N stochastic processes $\{R_t^i\}_{i=1}^N$ where $R_t^i \in SO(3)$ [1]. The conditional distribution of the particle R_t^i given $\mathscr{Z}_t$ is denoted by π_t, which acts on $f \in C^\infty(G)$ according to,

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

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

$$\mathrm{d}R_t^i = \underbrace{R_t^i\,\Omega_t\,\mathrm{d}t + R_t^i \circ [\,\mathrm{d}B_t^i\,]_\times}_{\text{propagation}} + \underbrace{R_t^i\,[\,\mathsf{K}(R_t^i,t) \circ \mathrm{dI}_t^i\,]_\times}_{\text{measurement update}}, \tag{7}$$

where $\{B_t^i\}_{i=1}^N$ are mutually independent standard Wiener processes in $\mathbb{R}^3$, and R_0^i is drawn from the initial distribution π_0^*. The i-th particle implements the Bayesian update step – to account for the conditioning due to the measurements – as gain $\mathsf{K}(R_t^i)$ times an error dI_t^i. The resulting control input to the i-th particle is an element of the Lie algebra $so(3)$.

The error dI_t^i is a modified form of the innovation process:

$$\mathrm{dI}_t^i = \mathrm{d}Z_t - \frac{1}{2}\big(h(R_t^i) + \hat{h}\big)\,\mathrm{d}t, \tag{8}$$

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

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 $\mathrm{grad}(\phi_j)$, where the function $\phi_j \in H^1(G;\pi)$ is a solution to the Poisson equation,

$$\begin{aligned} &\pi_t\big(\langle \mathrm{grad}(\phi_j), \mathrm{grad}(\psi)\rangle\big) = \pi_t\big((h_j - \hat{h}_j)\psi\big), \\ &\pi_t(\phi_j) = 0 \qquad \text{(normalization)}, \end{aligned} \tag{9}$$

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 $t \ge 0$. The existence-uniqueness of the solution of (9) requires additional assumptions on π_t; c.f., [29].

Assumption 1: The distribution π_t is absolutely continuous with respect to the uniform (Lebesgue) measure on $SO(3)$ with a positive density function ρ. ■

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 π_t to π_t^*:

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 $\pi_0 = \pi_0^*$, we have

$$\pi_t(f) = \pi_t^*(f),$$

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,

$$\mathrm{d}q_t^i = \frac{1}{2}\, q_t^i \otimes \mathrm{d}v_t^i, \tag{10}$$

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

$$\mathrm{d}v_t^i = \omega_t\,\mathrm{d}t + \mathrm{d}B_t^i + \mathsf{K}(q_t^i) \circ \Big(\mathrm{d}Z_t - \frac{h(q_t^i) + \hat{h}}{2}\,\mathrm{d}t\Big), \tag{11}$$

where $\mathsf{K}(q,t) = \mathsf{K}(R(q),t)$ 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 ϕ_j, h_j are denoted as ϕ, h). As the time t is fixed,

[1] Although the rotation matrix parameterization of $SO(3)$ is used, the filter is intrinsic. The FPF using the quaternion appears in Sec. IV-B.

the explicit dependence on t is suppressed (i.e., we denote π_t as π, R_t^i as R^i). This notation is also used in Sec. IV-D.

In a Galerkin scheme, the solution ϕ is approximated as,

$$\phi = \sum_{l=1}^{L} \kappa_l \, \psi_l,$$

where $\{\psi_l\}_{l=1}^L$ is a given (assumed) set of *basis functions* on $SO(3)$. The gain function $\mathsf{K} = (\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3)$, defined as the coordinates of $\text{grad}(\phi)$, is then given by,

$$\mathsf{k}_n = \sum_{l=1}^{L} \kappa_l \, E_n \cdot \psi_l, \quad n = 1,2,3.$$

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

$$\sum_{l=1}^{L} \kappa_l \, \pi\big(\langle \text{grad}(\psi_l), \text{grad}(\psi) \rangle\big) = \pi\big((h-\hat{h})\psi\big), \tag{12}$$

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

$$A\kappa = b, \tag{13}$$

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

$$\begin{aligned}
[A]_{kl} &= \pi\big(\langle \text{grad}(\psi_l), \text{grad}(\psi_k) \rangle\big) \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \langle \text{grad}(\psi_l)(R^i), \text{grad}(\psi_k)(R^i) \rangle \\
&= \frac{1}{N} \sum_{i=1}^{N} \sum_{n=1}^{3} (E_n \cdot \psi_l)(R^i)\, (E_n \cdot \psi_k)(R^i),
\end{aligned} \tag{14}$$

$$b_k = \pi\big((h-\hat{h})\psi_k\big) \approx \frac{1}{N} \sum_{i=1}^{N} (h(R^i) - \hat{h})\, \psi_k(R^i), \tag{15}$$

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

Note that both the Poisson equation (9) as well as its Galerkin finite-dimensional approximation (13) are coordinate-free representations. Particle-based approximation 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 τ,

$$\phi = e^{\tau\Delta_\rho} \phi + \int_0^\tau e^{s\Delta_\rho} (h - \hat{h}) \, \mathrm{d}s, \tag{16}$$

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

$$e^{\varepsilon\Delta_\rho}\phi(R) \approx \frac{\frac{1}{N}\sum_{i=1}^N k^{(\varepsilon,N)}(R,R^i)\phi(R^i)}{\frac{1}{N}\sum_{i=1}^N k^{(\varepsilon,N)}(R,R^i)}, \tag{17}$$

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

$$k^{(\varepsilon,N)}(R_1,R_2) = \frac{g^{(\varepsilon)}(R_1,R_2)}{\sqrt{\frac{1}{N}\sum_{i=1}^N g^{(\varepsilon)}(R_1,R^i)}\sqrt{\frac{1}{N}\sum_{i=1}^N g^{(\varepsilon)}(R_2,R^i)}}, \tag{18}$$

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

$$g^{(\varepsilon)}(R_1,R_2) := \frac{1}{(4\pi\varepsilon)^{3/2}} \exp\Big(-\frac{|R_1 - R_2|_F^2}{4\varepsilon}\Big), \tag{19}$$

where ε 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):

$$\Phi = T^{(N)}\Phi + \varepsilon H^{(N)}, \tag{20}$$

where $\Phi \in \mathbb{R}^N$ is the approximate solution that needs to be computed, $H^{(N)} = \big(h(R^1) - \hat{h}^{(N)}, h(R^2) - \hat{h}^{(N)}, ..., h(R^N) - \hat{h}^{(N)}\big)$, and $T^{(N)} \in \mathbb{R}^{N\times N}$ whose entries are given by,

$$T_{ij}^{(N)} = \frac{k^{(\varepsilon,N)}(R^i,R^j)}{\sum_{l=1}^N k^{(\varepsilon,N)}(R^i,R^l)}. \tag{21}$$

Note that $T^{(N)}$ 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 ϕ of (16), evaluated at the particles, is then approximated as $\phi(R^i) \approx \Phi_i$, the i-th entry of Φ.

The gain function is given by $\mathsf{K} = (\mathsf{k}_1, \mathsf{k}_2, \mathsf{k}_3)$, where $\mathsf{k}_n = E_n \cdot \phi$ for $n = 1,2,3$, and is evaluated at the particles according to,

$$E_n \cdot \phi(R^i) = -\varepsilon\, E_n \cdot h(R^i) + \frac{1}{2\varepsilon}\big[\big(S_n\Phi\big)_i - \big(S_n\mathbf{1}\big)_i \big(T^{(N)}\Phi\big)_i\big], \tag{22}$$

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

$$(S_n)_{ij} = T_{ij}^{(N)} \,\text{Tr}(R^i E_n R^j),$$

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, $\{q_t^i\}_{i=1}^N$, according to the sde's (10) and (11), with the initial conditions $\{q_0^i\}_{i=1}^N$ sampled i.i.d. from a given prior distribution π_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 $\{q_t^i\}_{i=1}^N$, its empirical mean is obtained as the eigenvector (with norm 1) of the 4×4 matrix $Q = \frac{1}{N}\sum_{i=1}^N q_t^i q_t^{i\,T}$, corresponding to its largest eigenvalue [33].

Algorithm 1 Feedback Particle Filter on $SO(3)$

initialization: sample $\{q_0^i\}_{i=1}^N$ from π_0^*
Assign $t = 0$
iteration: from t to $t + \Delta t$
Calculate $\hat{h}^{(N)} = (1/N)\sum_{i=1}^N h(q_t^i)$
for $i = 1$ to N **do**
    Generate a sample, ΔB_t^i, from $N(0, I)$
    Calculate the error

$$\Delta \mathrm{I}_t^i := \Delta Z_t - (1/2)(h(q_t^i) + \hat{h}^{(N)})\,\Delta t$$

    Calculate gain function $\mathsf{K}(q_t^i, t)$
    Calculate $\Delta \nu_t^i = \omega_t\,\Delta t + \sqrt{\Delta t}\,\Delta B_t^i + \mathsf{K}(q_t^i, t)\,\Delta \mathrm{I}_t^i$
    Propagate the particle q_t^i according to

$$q_{t+\Delta t}^i = q_t^i \otimes \begin{bmatrix} \cos\left(|\Delta \nu_t^i|/2\right) \\ \frac{\Delta \nu_t^i}{|\Delta \nu_t^i|}\sin\left(|\Delta \nu_t^i|/2\right) \end{bmatrix}$$

    ($|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^3$)
end for
Define matrix $Q = \frac{1}{N}\sum_{i=1}^N q_{t+\Delta t}^i q_{t+\Delta t}^{i\,T}$
return: empirical mean of $\{q_{t+\Delta t}^i\}_{i=1}^N$, i.e., the eigenvector of Q associated with its largest eigenvalue
Assign $t = t + \Delta t$

Algorithm 2 Galerkin gain function approximation

input: Particles $\{q^i\}_{i=1}^N$
Calculate $\hat{h}^{(N)} = (1/N)\sum_{i=1}^N h(q^i)$
for $k = 1$ to L **do**
    Calculate $b_k = \frac{1}{N}\sum_{i=1}^N \left(h(q^i) - \hat{h}^{(N)}\right)\psi_k(q^i)$
    for $l = 1$ to L **do**
        Calc. $A_{kl} = \frac{1}{N}\sum_{i=1}^N\sum_{n=1}^3 (E_n \cdot \psi_l)(q^i)\,(E_n \cdot \psi_k)(q^i)$
    end for
end for
Solve the matrix equation $A\kappa = b$, with $A = [A_{kl}]$, $b = [b_k]$

Calculate $\mathsf{k}_n(q^i) = \sum_{l=1}^L \kappa_l\,E_n \cdot \psi_l(q^i)$, for $n = 1, 2, 3$
return: $\left\{\mathsf{K}(q^i) = \left(\mathsf{k}_1(q^i), \mathsf{k}_2(q^i), \mathsf{k}_3(q^i)\right)\right\}_{i=1}^N$

V. Review of Some Attitude Filters

In this section, we restrict our attention to the attitude estimation problem with linear observations of the form $h(R_t) = R_t^T r$ where r is a known reference vector in the inertial frame (see the models of accelerometer and magnetometer in (4), (5)). A majority of the literature deals with such linear models. For discrete-time filters, it is convenient to define $Y_t := \frac{\mathrm{d}Z_t}{\mathrm{d}t}$, 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 $\{t_n\}$, $n = 0, 1, 2, \ldots$, with uniform time step $\Delta t = t_{n+1} - t_n$. 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 ω_t. Furthermore, $\widehat{R}_n$ denotes the posterior filter estimate at time t_n, $\widehat{R}_{n|n-1}$ denotes the filter estimate after the propagation step but before the measurement update, and $\Sigma_{n|n-1}$, Σ_n denote the associated covariance matrices.

Algorithm 3 Kernel-based gain function approximation

input: Particles $\{q^i\}_{i=1}^N$, parameters ε, K
Calculate $\hat{h}^{(N)} = (1/N)\sum_{i=1}^N h(q^i)$
for $i = 1$ to N **do**
    Calculate $H_i^{(N)} = h(q^i) - \hat{h}^{(N)}$
    for $j = 1$ to N **do**
        Calculate $g^{(\varepsilon)}(q^i, q^j)$, $k^{(\varepsilon,N)}(q^i, q^j)$ by (19), (18)
        Calculate $T_{ij}^{(N)}$ according to (21)
        for $n = 1, 2, 3$ **do**
            Calculate $(S_n)_{ij} = T_{ij}^{(N)}\,\mathrm{Tr}\left(R(q^i)\,E_n\,R(q^j)\right)$
        end for
    end for
end for
Assign Φ^0 as solution of (20) in previous time step
for $k = 0$ to $K - 1$ **do**
    Calculate $\Phi^{k+1} = T^{(N)}\Phi^k + \varepsilon H^{(N)}$, with $T^{(N)} = [T_{ij}^{(N)}]$
end for
for $i = 1$ to N **do**
    for $n = 1, 2, 3$ **do**
        Calc. $\mathsf{k}_n(q^i)$ by (22) with $S_n = [(S_n)_{ij}]$ and $\Phi = \Phi^K$
    end for
end for
return: $\left\{\mathsf{K}(q^i) = \left(\mathsf{k}_1(q^i), \mathsf{k}_2(q^i), \mathsf{k}_3(q^i)\right)\right\}_{i=1}^N$

A. Invariant extended Kalman filter

The invariant EKF (IEKF) models the attitude at time t_n as the product

$$R_n = \delta R_n \widehat{R}_n, \tag{23}$$

where the estimation error, $\delta R_n \in SO(3)$, is represented as $\delta R_n = \exp([\eta_n]_\times)$ where $\eta_n \in \mathbb{R}^3$. At each time step, the estimate of η_n, denoted as $\hat{\eta}_n$, is obtained as follows:

(i) Propagation step:

$$\widehat{R}_{n|n-1} = \widehat{R}_{n-1}\exp([\omega_{n-1}\Delta t]_\times),$$
$$\Sigma_{n|n-1} = \Sigma_{n-1} + (\Delta t)I.$$

(ii) Update step: The innovation error is defined in the inertial frame,

$$\mathrm{I}_n = \widehat{R}_{n|n-1}Y_n - r,$$

and the gain matrix K_n is calculated according to,

$$\mathsf{K}_n = \Sigma_{n|n-1}H^T\left(H\Sigma_{n|n-1}H^T + I\right)^{-1},$$

where $H = [r]_\times$.

(iii) Posterior update:

$$\hat{\eta}_n = \mathsf{K}_n \mathrm{I}_n,$$
$$\widehat{R}_n = \exp\left([\hat{\eta}_n]_\times\right)\widehat{R}_{n|n-1},$$
$$\Sigma_n = (I - \mathsf{K}_n H)\,\Sigma_{n|n-1}.$$

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

$$R_n = \widehat{R}_n \, \delta R_n, \tag{24}$$

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:

$$\begin{aligned}
\widehat{R}_{n|n-1} &= \widehat{R}_{n-1}\exp([\,\omega_{n-1}\Delta t\,]_\times),\\
\Sigma_{n|n-1} &= \Lambda\Sigma_{n-1}\Lambda^T + (\Delta t)I,
\end{aligned}$$

where $\Lambda = I - [\,\omega_{n-1}\Delta t\,]_\times$.
(ii) Update step:

$$\mathrm{I}_n = Y_n - \widehat{R}^T_{n|n-1} r,$$

$$\mathsf{K}_n = \Sigma_{n|n-1} H^T \left(H\Sigma_{n|n-1}H^T + I\right)^{-1},$$

where $H = [\widehat{R}^T_{n|n-1} r]_\times$. In contrast to the IEKF, the innovation error in the MEKF is defined in the body frame.
(iii) Posterior estimate:

$$\begin{aligned}
\hat{a}_n &= \mathsf{K}_n \mathrm{I}_n,\\
\widehat{R}_n &= \widehat{R}_{n|n-1}\, \delta R(\hat{a}_n),\\
\Sigma_n &= (I - \mathsf{K}_n H)\,\Sigma_{n|n-1}.
\end{aligned}$$

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.,

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

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,

$$\begin{aligned}
\mathrm{d}q_t &= \frac{1}{2} q_t \otimes \left(\omega_t \,\mathrm{d}t + \Sigma_B \,\mathrm{d}B_t\right),\\
\mathrm{d}Z_t &= \begin{bmatrix} R(q_t)^T & 0 \\ 0 & R(q_t)^T \end{bmatrix} \begin{bmatrix} r^g \\ r^b \end{bmatrix} \mathrm{d}t + \begin{bmatrix} \Sigma_W & 0 \\ 0 & \Sigma_W \end{bmatrix} \mathrm{d}W_t,
\end{aligned}$$

where the angular velocity is given by [43],

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

$r^g = (0,0,-1)$ and $r^b = (1/\sqrt{2}, 0, 1/\sqrt{2})$ represent the direction of the gravity and the local magnetic field, and Σ_B and Σ_W are 3×3 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 $\varepsilon = 1$ and $K = 10$.

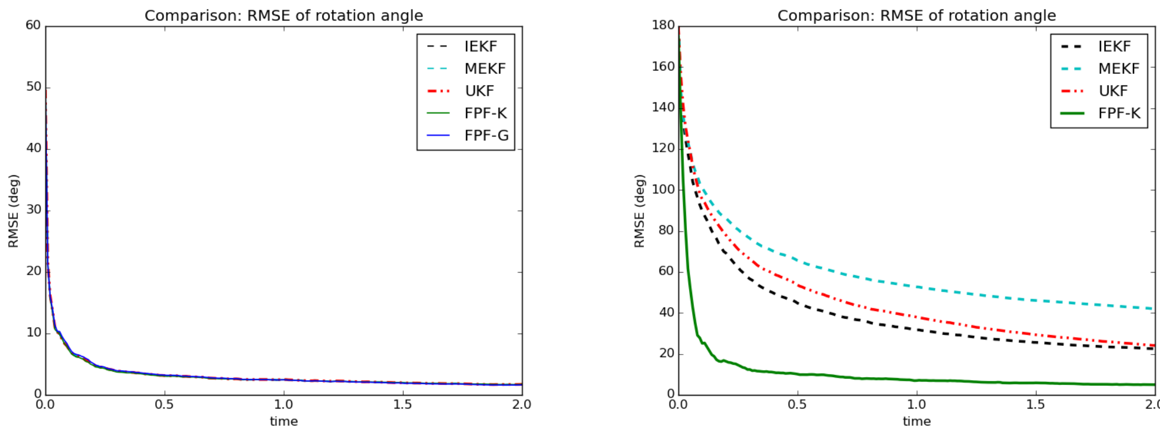
The performance metric is the root-mean-squared error (RMSE) [43], [25]:

$$\mathrm{RMSE}_t = \sqrt{(1/M)\sum_{j=1}^{M}\left(\delta\phi_t^j\right)^2},$$

where $\delta\phi_t^j$ 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_t and $\hat{q}_t$ denote the true and estimated attitude at time t, and let $\delta q_t := \hat{q}_t^{-1} \otimes q_t$ represent the estimation error, then $\delta\phi_t = 2\arccos(|\delta q_0|)$, where δq_0 is the first component of δq_t.

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 Σ_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 $\{v^i\}_{i=1}^N$ from the Gaussian distribution $N(0,\Sigma_0)$ in $\mathbb{R}^3$. Then, the particles $\{R_0^i\}_{i=1}^N$ are obtained by $R_0^i = \exp([\,v^i\,]_\times)$, and converted to the quaternions $\{q_0^i\}_{i=1}^N$.

The simulations are conducted over a finite time-horizon $t \in [0,T]$ with fixed time step Δt. The process noise Σ_B has standard deviation (std. dev.) of $5\,(^\circ/\mathrm{s})$. 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 Δt with N_f uniform subintervals. For FPF-G, $N_f = 100$; For other filters, $N_f = 20$ when Σ_W is large, and $N_f = 30$ when Σ_W is small. The relevant parameters are listed in Table 1.



(a) Initial distribution: 30°, sensor noise: 10°

(b) Initial distribution: 60°, sensor noise: 30°

Fig. 1: Comparison of filter performance with different initial distribution error and sensor noise.

TABLE 1: SIMULATION PARAMETERS

Σ_B	T	Δt	N	M
$0.008727^2 I$	2	0.01	200	100

Fig. 1 illustrates the filter performance with different initial distribution and sensor noise. In Fig. 1 (a), $\Sigma_0 = 0.5236^2 I$, corresponding to the std. dev. of 30°, 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°. 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°, and the target is initialized with fixed attitude – rotation of 180° about the axis $(3,1,4)$. The sensor noise $\Sigma_W = 0.05236^2 I$, i.e., the std. dev. is 30°. 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.

VII. Conclusion

In this paper, the feedback particle filter was presented for the problem of attitude estimation. The FPF is an intrinsic algorithm, possesses a gain-feedback structure and automatically respects the geometric constraint of the manifold. The algorithm was described using both the rotation matrix and the quaternion coordinate. The performance of FPF and its comparison with other attitude filters was illustrated by numerical simulations.

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.

Appendix

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

	$E_1\cdot$	$E_2\cdot$	$E_3\cdot$
ψ_1	$q_0^2 - q_1^2$	$-q_1q_2 - q_3q_0$	$-q_1q_3 + q_2q_0$
ψ_2	$-q_1q_2 + q_3q_0$	$q_0^2 - q_2^2$	$-q_2q_3 - q_1q_0$
ψ_3	$-q_1q_3 - q_2q_0$	$-q_2q_3 + q_1q_0$	$q_0^2 - q_3^2$
ψ_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.

[5] A. Barrau and S. Bonnabel. Invariant particle filtering with application to localization. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 5599–5605, 2014.

[6] A. Barrau and S. Bonnabel. Intrinsic filtering on Lie groups with applications to attitude estimation. *IEEE Transactions on Automatic Control*, 60(2):436–449, 2015.

[7] A. Barrau and S. Bonnabel. Invariant filtering for pose EKF-SLAM aided by an IMU. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 2133–2138, 2015.

[8] P. Batista, C. Silvestre, and P. Oliveira. Attitude and earth velocity estimation-part II: Observer on the special orthogonal group. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 127–132, 2014.

[9] J. Berger, A. Neufeld, F. Becker, F. Lenzen, and C. Schnörr. Second order minimum energy filtering on SE(3) with nonlinear measurement equations. In *Scale Space and Variational Methods in Computer Vision*, pages 397–409. 2015.

[10] J. Bohn and A. K. Sanyal. Almost global finite-time stable observer for rigid body attitude dynamics. In *Proceedings of the American Control Conference*, pages 4949–4954, 2014.

[11] S. Bonnabel, P. Martin, and P. Rouchon. Symmetry-preserving observers. *IEEE Transactions on Automatic Control*, 53(11):2514–2526, 2008.

[12] S. Bonnabel, P. Martin, and P. Rouchon. Non-linear symmetry-preserving observers on Lie groups. *IEEE Transactions on Automatic Control*, 54(7):1709–1713, 2009.

[13] S. Bonnabel, P. Martin, and E. Salaün. Invariant extended Kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 1297–1304, 2009.

[14] A. Carmi and Y. Oshman. Adaptive particle filtering for spacecraft attitude estimation from vector observations. *Journal of Guidance, Control, and Dynamics*, 32(1):232–241, 2009.

[15] Y. Cheng and J. L. Crassidis. Particle filtering for attitude estimation using a minimal local-error representation. *Journal of Guidance, Control, and Dynamics*, 33(4):1305–1310, 2010.

[16] Y. Cheon and J. Kim. Unscented filtering in a unit quaternion space for spacecraft attitude estimation. In *IEEE International Symposium on Industrial Electronics*, pages 66–71, 2007.

[17] G. S. Chirikjian and A. B. Kyatkin. *Engineering applications of noncommutative harmonic analysis: with emphasis on rotation and motion groups*. CRC press, 2000.

[18] C. Choi and H. Christensen. Robust 3D visual tracking using particle filtering on the SE (3) group. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4384–4390, 2011.

[19] D. Choukroun, I. Y. Bar-Itzhack, and Y. Oshman. Novel quaternion Kalman filter. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):174–190, 2006.

[20] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[21] J. L. Crassidis and F. L. Markley. Attitude estimation using modified Rodrigues parameters. 1996.

[22] J. L. Crassidis and F. L. Markley. Unscented filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542, 2003.

[23] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.

[24] M. Hua, G. Ducard, T. Hamel, R. Mahony, and K. Rudin. Implementation of a nonlinear attitude estimator for aerial robotic vehicles. *IEEE Transactions on Control Systems Technology*, 22(1):201–213, 2014.

[25] M. Izadi, E. Samiei, A. K. Sanyal, and V. Kumar. Comparison of an attitude estimator based on the Lagrange-d'Alembert principle with some state-of-the-art filters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2848–2853, 2015.

[26] M. Izadi and A. K. Sanyal. Rigid body attitude estimation based on the Lagrange-d'Alembert principle. *Automatica*, 50(10):2570–2577, 2014.

[27] A. Khosravian, J. Trumpf, R. Mahony, and T. Hamel. State estimation for invariant systems on Lie groups with delayed output measurements. *Automatica*, 68:254–265, 2016.

[28] J. Kwon, M. Choi, F. C. Park, and C. Chun. Particle filtering on the Euclidean group: framework and applications. *Robotica*, 25(06):725–737, 2007.

[29] R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. Poisson's equation in nonlinear filtering. *SIAM Journal on Control and Optimization*, 53(1):501–525, 2015.

[30] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, 1982.

[31] R. Mahony, T. Hamel, and J. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.

[32] F. L. Markley. Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003.

[33] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1197, 2007.

[34] Y. Oshman and A. Carmi. Attitude estimation from vector observations using a genetic-algorithm-embedded quaternion particle filter. *Journal of Guidance, Control, and Dynamics*, 29(4):879–891, 2006.

[35] M. E. Pittelkau. Rotation vector in attitude estimation. *Journal of Guidance, Control, and Dynamics*, 26(6):855–860, 2003.

[36] A. Taghvaei and P. G. Mehta. Gain function approximation in the feedback particle filter. arXiv preprint: 1603.05496.

[37] N. Trawny and S. I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2, 2005.

[38] P. Vernaza and D. D. Lee. Rao-Blackwellized particle filtering for 6-DOF estimation of attitude and position via GPS and inertial sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1571–1578, 2006.

[39] Y. Wang and G. S. Chirikjian. Error propagation on the Euclidean group with applications to manipulator kinematics. *IEEE Transactions on Robotics*, 22(4):591–602, 2006.

[40] Y. Wang, D. Wai, and M. Tomizuka. Steady-state marginalized particle filter for attitude estimation. In *Proceedings of the ASME Dynamic Systems and Control Conference*, pages 1–10, 2014.

[41] T. Wu and T. Lee. Hybrid attitude observer on SO(3) with global asymptotic stability. *arXiv preprint:1509.01754*, 2015.

[42] T. Yang, P. G. Mehta, and S. P. Meyn. Feedback particle filter. *IEEE Transactions on Automatic Control*, 58(10):2465–2480, 2013.

[43] M. Zamani. *Deterministic attitude and pose filtering, an embedded Lie groups approach*. PhD thesis, Australian National University, 2013.

[44] M. Zamani, J. Trumpf, and R. Mahony. Minimum-energy filtering for attitude estimation. *IEEE Transactions on Automatic Control*, 58(11):2917–2921, 2013.

[45] C. Zhang, A. Taghvaei, and P. G. Mehta. Feedback particle filter on matrix Lie groups. To appear in the Proceedings of the American Control Conference, 2016. arXiv preprint:1510.01259.