Feedback Particle Filter on Riemannian Manifolds and Matrix Lie Groups

Chi Zhang, Amirhossein Taghvaei and Prashant G. Mehta

***Abstract*—This paper is concerned with the problem of continuous-time nonlinear filtering of stochastic processes evolving on connected Riemannian manifolds without boundary. The main contribution of this paper is to derive the feedback particle filter (FPF) algorithm for this problem. In its general form, the FPF is shown to provide an intrinsic description of the filter that automatically satisfies the geometric constraints of the manifold. The particle dynamics are encapsulated in a Stratonovich stochastic differential equation (sde) that retains the feedback structure of the original (Euclidean) FPF. The implementation of the filter requires a solution of a Poisson equation on the manifold, and a numerical algorithm is described for this purpose. For the special case when the manifold is a matrix Lie group, explicit formulae for the filter are derived, using the matrix coordinates. Filters for two example problems are presented: the attitude estimation problem on $SO(3)$ and the robot localization problem in $SE(3)$. Comparisons are also provided between the FPF and popular algorithms for attitude estimation, namely the multiplicative EKF, the invariant EKF, the unscented quaternion estimator, the invariant ensemble Kalman filter, and the bootstrap particle filter. Numerical simulations are presented to illustrate these comparisons.**

Index Terms: Nonlinear filters, Particle filters, Poisson equation, Riemannian manifolds

I. Introduction

There is a growing interest in the nonlinear filtering community to develop geometric approaches for handling constrained systems. In many cases, the constraints are described by Riemannian manifolds. Engineering applications of filtering on manifolds include: (i) subspace tracking of signals [1], (ii) attitude estimation of aircrafts [2], [3], (iii) localization of mobile robots [4], [5], and (iv) visual tracking of humans and objects [6], [7]. In these applications, the manifolds of interest include matrix manifolds, in particular, the matrix Lie groups.

A. Problem Statement

Let M be a Riemannian manifold. We consider the following continuous-time stochastic differential equation (sde) model:

$$\text{Signal:} \qquad \mathrm{d}X_t = V_0(X_t)\,\mathrm{d}t + V_1(X_t)\circ \mathrm{d}B_t,\ \ X_0 \sim \pi_0^* \quad (1a)$$

$$\text{Observation:} \qquad \mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t \quad (1b)$$

where $X_t \in M$ is the state at time t, the initial condition X_0 at time $t=0$ is sampled from a given prior distribution π_0^* on M; $Z_t \in \mathbb{R}$ is the observation; V_0 and V_1 are given vector fields on M; B_t and W_t are mutually independent standard Wiener processes in $\mathbb{R}$, and they are also assumed to be independent of the initial state X_0; $h : M \to \mathbb{R}$ is a given real-valued function. The scalar observation is considered for notational ease: the extension to the general vector-valued observation is straightforward and described in Remark 3. The $\circ$ before $\mathrm{d}B_t$ indicates that the stochastic differential equation (sde) (1a) is expressed in its Stratonovich form, which provides an intrinsic description of the sde [8].

The nonlinear filtering problem is to numerically approximate the conditional distribution (posterior) of X_t given the time-history (filtration) of observations $\mathscr{Z}_t = \sigma(Z_s : s \le t)$. The posterior is denoted as π_t^*. It acts on a smooth function f as

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

The time-evolution of $\pi_t^*(f)$ is described by the Kushner-Stratonovich filtering equation ([9, Theorem 5.7]):

$$\pi_t^*(f) = \pi_0^*(f) + \int_0^t \pi_s^*(\mathscr{L}f)\,\mathrm{d}s + \int_0^t \big(\pi_s^*(fh) - \pi_s^*(h)\pi_s^*(f)\big)\big(\mathrm{d}Z_s - \pi_s^*(h)\,\mathrm{d}s\big) \quad (2)$$

where

$$\mathscr{L}f := V_0 \cdot f + \frac{1}{2} V_1 \cdot (V_1 \cdot f) \quad (3)$$

The notation $V_0 \cdot f$ is used to denote the action of a vector field V_0 on a function f. The definitions appear in Sec. II-A as part of an overview of the Riemannian manifolds.

B. Literature Review

Filtering of stochastic processes evolving on Riemannian manifolds has a rich history; c.f., [10], [11], [12]. In particular, matrix manifolds have been considered in many applications, e.g., the Stiefel manifold and the Grassmann manifold for subspace tracking [13], [1], and the manifold of symmetric positive definite matrices for covariance tracking [14], [15]. An intensively studied class of matrix manifolds are the matrix Lie groups. Filters for matrix Lie groups have been developed, e.g., by extending the classical extended Kalman filter (EKF). These extensions have appeared in discrete-time [16], [17], continuous-time [18], [19], and continuous-discrete-time [3], [20] settings. A number of EKF-based filters have been proposed and applied for attitude estimation, e.g., the additive EKF [21] and the multiplicative EKF [22]. The EKF-based attitude filters require a linearized model of the

C. Zhang was with the Coordinated Science Laboratory and the Department of Mechanical Science and Engineering at the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL 61801 USA. He is now with Experian DataLabs, San Diego, CA 92122 USA (e-mail: czhang54@illinois.edu).

A. Taghvaei and P. G. Mehta are with the Coordinated Science Laboratory and the Department of Mechanical Science and Engineering at UIUC (e-mails: taghvae2@illinois.edu; mehtapg@illinois.edu)

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

estimation error, typically derived using one of the many three-dimensional attitude representations, e.g. the Euler angle [23], the rotation vector [24], and the Rodrigues parameter [25].

Apart from the EKF, particle filters for matrix Lie groups have been an active area of research [26], [27], [28]. Typically, particle filters adopt discrete-time description of the dynamics and are based on importance sampling and resampling numerical procedures. For the attitude estimation problem, the unscented quaternion estimator [29] and the bootstrap particle filter [30], [31] have been developed, using one of the attitude representations. Other non-parametric approaches include filters based on variational formulations on the Lie groups [32], [33].

In recent years, geometric group-theoretic methods have been applied to develop deterministic nonlinear observers [34], [35], [36]. A class of symmetry preserving observers is proposed in [37] leading to the invariant EKF [3], [18], the invariant unscented Kalman filter [38], the invariant ensemble Kalman filter [3], and the invariant particle filter [39] algorithms within the stochastic filtering framework. A closely related theme is the use of non-commutative harmonic analysis for characterizing error propagation and Bayesian fusion on Lie groups [40], [41], [42]. More comprehensive surveys of filters for matrix Lie groups and attitude estimation can be found in [43], [44].

C. Overview of the Paper

The objective of this paper is to obtain a generalization of the feedback particle filter (FPF), originally developed in [45], [46], [47] in the Euclidean settings, to the filtering problem (1a)-(1b). The main result is that the update formula in the original Euclidean setting carries over to the manifold setting.

The contributions of this paper are as follows:

• **FPF on Riemannian manifolds:** The extension of the FPF for Riemannian manifolds is derived. The particle dynamics, expressed in their Stratonovich form, are shown to provide an intrinsic description of the filter that automatically satisfies the geometric constraints of the manifold. Even in the manifold setting, the FPF is i) shown to admit an error-correction gain-feedback structure, and ii) proved to be an exact algorithm. Exactness means that, in the limit of large number of particles, the empirical distribution of the particles exactly matches the posterior distribution.

• **FPF on matrix Lie groups:** In applications, matrix Lie groups is the most important example of the Riemannian manifolds. For the matrix Lie groups, explicit formulae for the filter are derived using the matrix coordinates[1]. Two example problems are considered, namely the attitude estimation problem on $SO(3)$[2] and the robot localization problem on $SE(3)$. For the attitude estimation problem, the filter formulae are described with respect to both the matrix (rotations) and the quaternion coordinates, with the latter being demonstrated for computational purposes. Comparisons are provided between the FPF and popular algorithms for attitude estimation, namely the multiplicative EKF, the invariant EKF, the unscented quaternion estimator, the invariant ensemble Kalman filter, and a bootstrap particle filter. Numerical simulations are presented to illustrate these comparisons.

• **FPF with a concentrated distribution:** For a certain special case, namely when the posterior distribution is concentrated, a closed-form approximation of the filter gain is derived. It is referred to as the *constant gain approximation*. With this approximation, evolution equations for the mean and the covariance are also derived and shown to be closely related to the left invariant EKF algorithm.

D. Organization of the Paper

The outline of the remainder of this paper is as follows: In Sec. II, the FPF algorithm for Riemannian manifolds is presented. Its application to matrix Lie groups appears in Sec. III: explicit formulae for the FPF are derived using the matrix coordinates and filters for $SO(3)$ and $SE(3)$ are presented. The special case of FPF with a concentrated distribution is discussed as part of Sec. IV. The numerical results and the conclusions appear in Sec. V and Sec. VI, respectively. All the proofs appear in the Appendix.

II. FPF on Riemannian Manifolds

A. Mathematical Preliminaries

Riemannian manifold: M is a d-dimensional smooth connected Riemannian manifold without boundary. A Riemannian metric $\langle\cdot,\cdot\rangle$ on M is a smoothly varying inner product $\langle\cdot,\cdot\rangle_x$ defined on each of the tangent (vector) space T_xM. In the local coordinates $(x_n)_{n=1}^d$, the inner product is a represented by a $d\times d$ positive-definite symmetric matrix $(g_{nk}(x))_{n,k=1}^d$. For two tangent vectors $v,w\in T_xM$ with coordinates $(v_n)_{n=1}^d$ and $(w_n)_{n=1}^d$, $\langle v,w\rangle_x=\sum_{n,k=1}^d g_{nk}(x)v_nw_k$.

Vector field: A vector field V is a smoothly varying map that assigns to each point $x\in M$ a tangent vector $V(x)\in T_xM$. The action of a vector field V on a function f is denoted as $V\cdot f$. It is defined according to

$$V\cdot f(x):=\frac{\mathrm{d}}{\mathrm{d}t}f(\gamma(t))\Big|_{t=0}$$

where $\gamma(t)$ is a curve on M with initial value $\gamma(0)=x$ and initial velocity $\frac{\mathrm{d}\gamma}{\mathrm{d}t}\big|_{t=0}=V(x)$. Expressed in the local coordinates, $V\cdot f=\sum_{n=1}^d v_n\frac{\partial f}{\partial x_n}$. The norm $|V|:=\sqrt{\langle V,V\rangle}$.

Gradient: The gradient of a function f is denoted as $\mathrm{grad}f$. It is defined as the vector field that satisfies the following identity:

$$\langle\mathrm{grad}\,f,V\rangle=V\cdot f$$

for all vector fields V, where $\langle\mathrm{grad}\,f,V\rangle$ is interpreted as a function on M with value $\langle\mathrm{grad}\,f(x),V(x)\rangle_x$ at $x\in M$. In the local coordinates, $(\mathrm{grad}f)_n=\sum_{k=1}^d g^{-1}_{nk}\frac{\partial f}{\partial x_k}$ where g^{-1} denotes the inverse of the matrix g.

[1] FPF for matrix Lie groups first appeared in our prior conference paper [48]. The filter derivation in this paper (special case of a Riemannian manifold) is more general and better highlights the geometric aspects of the filter.

[2] FPF for $SO(3)$ was described in our prior conference paper [49].

Riemannian measure: The Riemannian measure on M is denoted by m. In the local coordinates, $m(\mathrm{d}x) = \sqrt{\det(g(x))}\,\mathrm{d}x$ where $\det(g(x))$ is the determinant of the matrix $g(x)$ and $\mathrm{d}x$ is the Lebesgue (volume) measure.

Function spaces: The vector space of smooth real-valued functions $f : M \to \mathbb{R}$ with compact support is denoted as $C_c^\infty(M)$. For a probability distribution π on M, $L^2(M;\pi)$ denotes the Hilbert space of functions on M that satisfy $\pi(|f|^2) < \infty$ (here $\pi(|f|^2) := \int_M |f|^2 \,\mathrm{d}\pi$); $H^1(M;\pi)$ denotes the Hilbert space of functions f such that f and $|\mathrm{grad} f|$ (defined in the weak sense) are both in $L^2(M;\pi)$; and its subspace $H_0^1(M;\pi) := \{f \in H^1(M;\pi) \,|\, \pi(f) = 0\}$.

B. Particle Dynamics and Control Architecture

The FPF on a Riemannian manifold M is a controlled system comprising of N stochastic processes $\{X_t^i\}_{i=1}^N$ with $X_t^i \in M$. The dynamics of the i^{th} particle are modeled by the Stratonovich sde

$$\mathrm{d}X_t^i = \big(V_0(X_t^i) + \mathsf{u}_t(X_t^i)\big)\,\mathrm{d}t + V_1(X_t^i) \circ \mathrm{d}B_t^i + \mathsf{K}_t(X_t^i) \circ \mathrm{d}Z_t \quad (4)$$

where B_t^i is a standard Wiener processes in $\mathbb{R}$. The time-varying vector fields u_t, K_t are referred to as the *control* and *gain*, respectively. These vector fields need to be chosen. The following admissibility requirement is imposed on u_t and K_t:

Definition 1: (Admissible Input): The vector fields u_t and K_t are *admissible* if they are $\mathscr{Z}_t-$measurable and $\mathsf{E}[|\mathsf{u}_t \cdot f(X_t^i)||\mathscr{Z}_t] < \infty$, $\mathsf{E}[|\mathsf{K}_t \cdot f(X_t^i)|^2|\mathscr{Z}_t] < \infty$ and $\mathsf{E}[|\mathsf{K}_t \cdot (\mathsf{K}_t \cdot f)(X_t^i)||\mathscr{Z}_t] < \infty$ for all functions $f \in C_c^\infty(M)$ and all $t \in [0,T]$. □

The conditional distribution of the particle X_t^i given $\mathscr{Z}_t$ is denoted by π_t. It acts on a function $f \in C_c^\infty(M)$ as

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

The evolution equation for π_t is given in the proposition below. The proof appears in Appendix A.

Proposition 1: Consider the particle X_t^i with dynamics described by (4). The forward evolution equation of the conditional distribution π_t is given by

$$\begin{aligned}\pi_t(f) = \pi_0(f) + \int_0^t \pi_s(\mathscr{L}f)\,\mathrm{d}s + \int_0^t \pi_s(\mathsf{u}_s \cdot f)\,\mathrm{d}s \\ + \int_0^t \pi_s(\mathsf{K}_s \cdot f)\,\mathrm{d}Z_s + \frac{1}{2}\int_0^t \pi_s(\mathsf{K}_s \cdot (\mathsf{K}_s \cdot f))\,\mathrm{d}s\end{aligned} \quad (5)$$

for any $f \in C_c^\infty(G)$, where the operator $\mathscr{L}$ is defined in (3). □

Problem statement: There are two types of conditional distributions of interest:

- π_t^*: The conditional distribution of X_t given $\mathscr{Z}_t$
- π_t: The conditional distribution of X_t^i given $\mathscr{Z}_t$

The objective is to choose u_t and K_t such that, given $\pi_0 = \pi_0^*$, the evolution of the two conditional distributions are identical (i.e., $\pi_t = \pi_t^*$ for all $t \in [0,T]$). In this case, the filter is said to be *exact*.

Solution: The FPF represents the following choice of the gain K_t and the control u_t:

1) Gain: For a fixed $t \in [0,T]$, let $\phi \in H^1(M;\pi_t)$ be an admissible solution of the linear Poisson equation:

$$\mathscr{PE}: \quad \begin{aligned}&\pi_t\big(\mathrm{grad}\,\phi \cdot \psi\big) = \pi_t\big((h - \hat{h}_t)\psi\big) \\ &\pi_t(\phi) = 0\end{aligned} \quad (6)$$

for all $\psi \in H^1(M;\pi_t)$, where $\hat{h}_t := \pi_t(h)$. The gain is chosen as

$$\mathsf{K}_t = \mathrm{grad}\,\phi \quad (7)$$

2) Control: The control u_t is chosen as

$$\mathsf{u}_t = -\frac{1}{2}\big(h + \hat{h}_t\big)\,\mathsf{K}_t \quad (8)$$

In the remainder of this paper, $\phi = \mathscr{PE}(h)$ means ϕ is the solution of the Poisson equation (6) with the righthand-side as the (given) real-valued function h.

Feedback particle filter: Using these choices of u_t and K_t, the i-th particle in the FPF has the following representation:

$$\mathrm{d}X_t^i = \underbrace{V_0(X_t^i)\,\mathrm{d}t + V_1(X_t^i) \circ \mathrm{d}B_t^i}_{\text{propagation}} + \underbrace{\mathsf{K}_t(X_t^i) \circ \mathrm{d}\mathrm{I}_t^i}_{\text{observation update}} \quad (9)$$

where the error $\mathrm{d}\mathrm{I}_t^i \in \mathbb{R}$ is a modified form of the innovation process defined as follows:

$$\mathrm{d}\mathrm{I}_t^i := \mathrm{d}Z_t - \frac{h(X_t^i) + \hat{h}_t}{2}\,\mathrm{d}t \quad (10)$$

The i-th particle implements the Bayesian update step – to account for the conditioning due to the observations – as gain times an error, which is akin to the feedback structure in a classical Kalman filter.

Note that the Poisson equation (6) must be solved for each time $t \in [0,T]$.

The exactness is asserted in the following theorem. The proof is contained in Appendix B.

Theorem 1: Consider the two evolution equations for π_t^* and π_t, defined according to the solution of the forward evolution equations (2) and (5), respectively. Suppose that the gain K_t is a solution of (6)-(7), and the control u_t is given by (8). Suppose also that u_t and K_t are admissible. Then, provided $\pi_0 = \pi_0^*$, we have

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

for all $t \in [0,T]$ and all functions $f \in C_c^\infty(M)$. □

Remark 1 (Intrinsic nature of FPF): A numerical implementation of the filter requires a collection of coordinate charts $\{\varphi : U \subset M \to \mathbb{R}^d\}$ defined on the manifold. A particle is represented by its (local) coordinates $x_t^i = \varphi(X_t^i)$ according to the local chart. Numerical integration of the sde (9) is carried out using these coordinates. The Stratonovich form of the sde is intrinsic [50, Sec. 3.3]: suppose $x_{t,1}^i$ and $x_{t,2}^i$ are the (coordinate-form) solutions of the sde (9) using charts φ_1 and φ_2 respectively. Then the two solutions are related according to $x_{t,2}^i = \varphi_2(\varphi_1^{-1}(x_{t,i}^i))$.

The gain $\text{grad}\phi$ – obtained as a solution of the Poisson equation (6) – is also intrinsic that furthermore does not depend upon the choice of the Riemannian metric. The latter property is surprising at first glance because the definition of the gradient depends upon the choice of the metric. The proof of the property is straightforward: Suppose ϕ_1 and ϕ_2 be two solutions to the Poisson equation (6) with respect to the metrics g_1 and g_2, respectively. Then

$$\pi_t\big(\text{grad}_1\phi_1\cdot\psi\big)=\pi_t\big(\text{grad}_2\phi_2\cdot\psi\big)=\pi_t\big((h-\hat{h})\psi\big)$$

for all $\psi\in H^1(M;\pi_t)$. Therefore, $\text{grad}_1\phi_1=\text{grad}_2\phi_2$ a.e. (π_t). Note that the solution ϕ will in general depend upon the metric but $\text{grad}\phi$ does not. □

Remark 2 (Strong form of the Poisson equation): The equation (6) is referred to as the weak form of the Poisson equation. A strong form is obtained under certain additional assumptions: Suppose π_t admits an everywhere positive density, denoted as ρ_t, with respect to the Riemannian measure. Then the strong form of the Poisson equation (6) is obtained as

$$\Delta_{\rho_t}\phi=-(h-\hat{h}_t) \tag{11}$$

where $\Delta_\rho\phi:=\frac{1}{\rho}\text{div}\big(\rho\,\text{grad}\phi\big)$ is the weighted Laplacian on the manifold [51], and $\text{div}(\cdot)$ denotes the divergence operator.

There are a number of equivalent approaches to define the divergence operator, e.g., in terms of the Lie derivative of the Riemannian measure ([52, page 423]) or in terms of the Levi-Civita connection ([53, page 83]). With respect to the local coordinates, $\text{div}(V)(x)=\sum_{n=1}^d\sqrt{\det g^{-1}(x)}\frac{\partial}{\partial x_n}(\sqrt{\det g(x)}v_n(x))$. On a manifold without boundary, the following integration-by-parts formula holds ([52, page 436]):

$$\int_M f\,\text{div}(V)\,\text{d}m=\int_M V\cdot f\,\text{d}m$$

for all smooth functions f and vector fields V (m is the Riemannian measure).

Formally, the weak form (6) is obtained by multiplying both sides of (11) by $\psi(x)\rho_t(x)$ and using the integration-by-parts formula.

For both theoretical and computational reasons, the weak form is the preferred form in the remainder of this paper. □

Remark 3 (FPF with vector-valued observations): Consider the filtering problem for the sde model (1a)-(1b) where $Z_t\in\mathbb{R}^m$ is now a (column) vector, the observation function $h=(h^1,\dots,h^m):M\to\mathbb{R}^m$ is vector-valued function, and $W_t\in\mathbb{R}^m$ is a Wiener process with a $m\times m$ covariance matrix Q_W that is assumed to be positive-definite. For this problem, the FPF is a straightforward extension of (9):

$$\text{d}X_t^i=V_0(X_t^i)\,\text{d}t+V_1(X_t^i)\circ\text{d}B_t^i+\mathsf{K}_t(X_t^i)\,Q_W^{-1}\circ\text{d}\mathrm{I}_t^i \tag{12}$$

where the error $\text{d}\mathrm{I}_t^i=\text{d}Z_t-\frac{1}{2}(h(X_t^i)+\hat{h}_t)\,\text{d}t\in\mathbb{R}^m$ is a (column) vector and the gain $\mathsf{K}_t(x)=[\mathsf{K}_t^1(x)\,|\,\mathsf{K}_t^2(x)\,|\,\dots\,|\,\mathsf{K}_t^m(x)]\in\mathbb{R}^{d\times m}$ is a matrix whose j^{th} column $\mathsf{K}_t^j=\text{grad}\,\phi^j$ where $\phi^j=\mathscr{P}\mathscr{E}(h^j)$. The exactness of the filter (12) is shown by extending the proof of Theorem 1 (see [46]). The extension is conceptually straightforward but notationally somewhat cumbersome. □

C. Well-posedness and Admissibility of the Gain

The admissibility of the gain and control requires a well-posedness analysis of the Poisson equation. As in the original Euclidean setting [46], we make the following assumptions:

Assumption 1: The distribution π_t is absolutely continuous with respect to the Riemannian measure, with density $\rho_t(x)=e^{-\mathscr{U}_t(x)}$. π_t admits a uniform spectral gap (or Poincaré inequality) with constant $\bar{\lambda}$ ([54, Sec. 4.2]): That is for all $t\in[0,T]$,

$$\pi_t\big(|\psi|^2\big)\le\frac{1}{\bar{\lambda}}\pi_t\big(|\text{grad}\psi|^2\big)$$

for all $\psi\in H_0^1(M;\pi_t)$.

Assumption 2: The functions $h,|\text{grad}(h)|\in L^2(M;\pi_t)$. For any chart $\varphi:U\to\mathbb{R}^d$, the Hessian of $\mathscr{U}_t\circ\varphi^{-1}$ is uniformly bounded.

The proof of the following well-posedness theorem is adapted from the proof in Euclidean case [46, Theorem 2 and Corollary 1]. It is omitted here on account of space.

Theorem 2 (Theorem 2 in [46]): Under Assumptions 1-2, for each fixed $t\in[0,T]$, the Poisson equation (6) possesses a unique solution $\phi\in H_0^1(G;\pi_t)$, satisfying

$$\pi_t\big(|\text{grad}\phi|^2\big)\le\frac{1}{\bar{\lambda}}\pi_t\big(|h|^2\big)$$

The following bounds hold:

$$\begin{aligned}\pi_t\big(|\mathsf{K}_t\cdot f|^2\big)&\le C\,\pi_t\big(|h|^2\big)\\ \pi_t\big(|\mathsf{u}_t\cdot f|\big)&\le C\,\pi_t\big(|h|^2\big)\\ \pi_t\big(|\mathsf{K}_t\cdot(\mathsf{K}_t\cdot f)|\big)&\le C\,\big(\pi_t\big(|h|^2\big)+\pi_t\big(|\text{grad}h|^2\big)\big)\end{aligned}$$

for all $f\in C_c^\infty(M)$, where the constant C depends on $\bar{\lambda}$, sup norm of the second derivative of $\mathscr{U}_t$, and the sup norm of function f and its derivatives. □

Remark 4 (Assumption 1): For compact manifolds, the existence of a smooth density ρ_t is sufficient for Assumptions 1-2 to hold. For non-compact manifolds, the assumptions hold if the density has a Gaussian tail as $|x|\to\infty$ [46, Remark 2]. □

The main challenge to implement the FPF algorithm is to numerically approximate the gain function. Our group has investigated two types of algorithms[3]:

Galerkin algorithm: The Galerkin algorithm involves projecting the infinite-dimensional equation (6) onto a finite-dimensional subspace, defined as span of a (given) set of basis functions. The details of the Galerkin algorithm appear in the conference version of this paper [48]. An issue with the Galerkin algorithm is how to select the basis functions[4], particularly for high-dimensional problems. As the dimension

[3] In Euclidean settings, other algorithms to approximate the gain function include: the constant gain approximation which is a special case of the Galerkin algorithm with a choice of linear basis functions [46]; a continuation-based scheme for numerical approximation of the Poisson equation [55], and certain probabilistic approaches involving dynamic programming [56].

[4] A data-driven approach for selecting basis functions, based on the use of proper orthogonal decomposition (POD), appears in [57].

increases, the number of basis functions can scale poorly (exponential in the dimension).

Kernel algorithm: The kernel algorithm is described next. Since the time t is fixed, the explicit dependence on t is suppressed in the proof (i.e., π_t, X_t^i, K_t are denoted as π, X^i, K etc.).

D. Kernel Algorithm for Gain Approximation

The objective is to approximate the gain $\{\mathsf{K}(X^i)\}_{i=1}^N$ for the ensemble $\{X^i\}_{i=1}^N$. The kernel algorithm proceeds in 5 steps:

Step 1: The manifold is embedded in a Euclidean space $\mathbb{R}^{d_1}$ with a smooth isometric embedding $\iota : M \to \mathbb{R}^{d_1}$. For a fixed element $z \in M$, define a smooth vector field

$$x \mapsto \operatorname{grad} |\iota(x) - \iota(z)|^2 =: s(x;z)$$

where $|\cdot|$ is the Euclidean norm in $\mathbb{R}^{d_1}$.

Step 2: The ensemble $\{X^i\}_{i=1}^N$ is used to define a graph in $\mathbb{R}^{d_1}$. For any two particles X^i and X^j compute (see [58, Assumption 19])

$$\begin{aligned} \mathsf{d}_{ij} &= |\iota(X^i) - \iota(X^j)| \\ \mathsf{s}_{ij} &= s(X^i; X^j) \end{aligned}$$

Step 3: Select a parameter $\varepsilon > 0$. Assemble an $N \times N$ Markov matrix $\mathsf{T}^{(\varepsilon,N)}$ on the graph. The (i,j)-th element of $\mathsf{T}^{(\varepsilon,N)}$ is:

$$\mathsf{T}_{ij}^{(\varepsilon,N)} = \frac{1}{n^\varepsilon(X^i)} \frac{k^{(\varepsilon)}(\mathsf{d}_{ij})}{\sqrt{\sum_{l=1}^N k^{(\varepsilon)}(\mathsf{d}_{il})}\sqrt{\sum_{l=1}^N k^{(\varepsilon)}(\mathsf{d}_{jl})}} \tag{13}$$

where $k^{(\varepsilon)}(z) = \exp(-\frac{z^2}{4\varepsilon})$ is a Gaussian kernel in $\mathbb{R}$ and $n^\varepsilon(X^i)$ is a normalization factor chosen such that the matrix row-sum is 1.

Step 4: Compute the zero-mean solution, $N \times 1$ vector $\phi^{(\varepsilon,N)}$, by solving the fixed-point equation:

$$\phi^{(\varepsilon,N)} = \mathsf{T}^{(\varepsilon,N)}\phi^{(\varepsilon,N)} + \varepsilon(h^{(N)} - \hat{h}^{(N)}) \tag{14}$$

where $h^{(N)} := \big(h(X^1), \ldots, h(X^N)\big)$ and $\hat{h}^{(N)} := \frac{1}{N}\sum_{i=1}^N h(X^i)$. In terms of this solution, compute

$$\mathsf{r}_j = \phi^{(\varepsilon,N)}(X^j) + \varepsilon h(X^j) - \sum_{l=1}^N \mathsf{T}_{il}^{(\varepsilon,N)}(\phi^{(\varepsilon,N)}(X^l) + \varepsilon h(X^l)) \tag{15}$$

Step 5: The gain at the particle X^i is approximated as

$$\mathsf{K}(X^i) = -\frac{1}{4\varepsilon}\sum_{j=1}^N \mathsf{r}_j \mathsf{T}_{ij}^{(\varepsilon,N)} \mathsf{s}_{ij} \tag{16}$$

Formula (16) is referred to as the *kernel approximation* of the gain.

The justification for the kernel algorithm appears in Appendix C.

Remark 5 (Computational issues): The computational complexity of the kernel algorithm is $O(N^2)$, whereas the Galerkin algorithm and some other particle filters scale as $O(N)$ to achieve comparable performance measured using various error metrics (see Sec. V-A and in particular Fig. 6). This may not be favorable for real-time applications of filtering. In practice, one can exploit the sparsity structure (because of the rapid decay of the Gaussian kernel) to improve the computational efficiency. In contrast to the Galerkin algorithm, the computational complexity does not depend upon the dimension of the problem. Such approximations are computationally attractive whenever $N \ll d$, i.e., when the dimension of state space is high but the dynamics is confined to a low-dimensional subset which, however, is not a priori known. Fig. 6 in Sec. V-A includes comparative results on the complexity of the kernel algorithm compared to other types of particle filtering algorithms.

The tuning parameters of the kernel algorithm are the Gaussian kernel bandwidth parameter ε and the number of particles N. In Euclidean settings, it is shown in [59] that for small values of the parameter ε, the error scales as $O(\varepsilon) + O(\frac{1}{\varepsilon^{1+d/2}\sqrt{N}})$. Therefore, for a fixed number of particles N, there is an optimal value for ε where the error is minimum (computing this parameter value is difficult though). A remarkable feature of the kernel algorithm is that the error remains bounded even for very large ε. In fact, in Euclidean settings, as $\varepsilon \to \infty$, one obtains the constant gain approximation (equal to the Kalman gain) of the gain function [46]. FPF in this case reduces to the ensemble Kalman filter. A useful rule of thumb is that ε is chosen large enough that most particles have about 10% of the population in their ε-neighborhood. □

III. FPF on Matrix Lie Groups

A. Mathematical Preliminaries

Matrix Lie group: A matrix Lie group G is a closed subgroup of invertible $n \times n$ matrices. In addition to being a d-dimensional Riemannian manifold, it is also a group. The group operations are matrix multiplication and matrix inversion. The identity element is the identity matrix I. The tangent space at I is referred to as the Lie algebra $\mathscr{G}$. The Lie algebra is equipped with an inner product $\langle\cdot,\cdot\rangle_{\mathscr{G}}$ and an orthonormal basis $\{E_1, \ldots, E_d\}$ with respect to this inner product. A vector $a = (a_1, \ldots, a_d) \in \mathbb{R}^d$ is uniquely mapped to an element in $\mathscr{G}$, denoted as $[a]_\times := \sum_{n=1}^d a_n E_n$. Left translation yields a (left-invariant) *global frame* whereby $\{xE_1, \ldots, xE_d\}$ span T_xG for each matrix $x \in G$. The Riemannian metric $\langle\cdot,\cdot\rangle$ is chosen as a smoothly varying inner product $\langle xE_n, xE_l\rangle_x := \langle E_n, E_l\rangle_{\mathscr{G}} := \frac{1}{2}\operatorname{tr}(E_n E_l^\top)$ for all $x \in G$, where $\operatorname{tr}(\cdot)$ is the matrix trace and the superscript $^\top$ denotes the matrix transpose.

Vector field: The global frame is used to represent any vector field V as $V(x) = x[a(x)]_\times = \sum_{n=1}^d (a(x))_n xE_n$ where $a : G \to \mathbb{R}^d$ is a vector-valued map. The functions $(a(x))_{n=1}^d$ are referred to as the *coordinate functions*. The construction is depicted in Fig. 1.

Gradient: For a smooth function f, the coordinate functions of $\operatorname{grad} f$ are given by the action of xE_n on f:

$$(\operatorname{grad} f)_n = (xE_n)\cdot f = \frac{\mathrm{d}}{\mathrm{d}t} f\big(x\exp(tE_n)\big)\Big|_{t=0}$$

where $\exp(\cdot)$ is the standard matrix exponential.

Embedding: Using the matrix coordinates, a smooth isometric embedding $\iota : G \to \mathbb{R}^{n^2}$ is obtained as $\iota(x) = (x_{11}, x_{12}, \ldots, x_{nn})$

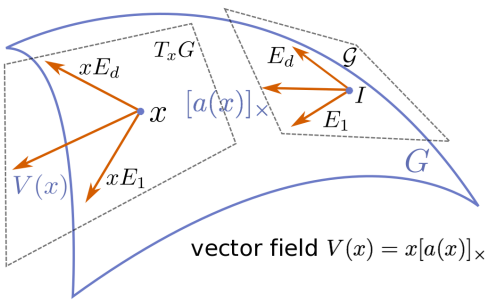


Fig. 1: Construction of a vector field on a matrix Lie group G: Given a basis $\{E_1,\dots,E_d\}$ of the Lie algebra $\mathscr{G}$, $\{xE_1,\dots,xE_d\}$ defines a global frame on G. A vector field $V(x) = x[a(x)]_\times = \sum_{n=1}^d (a(x))_n xE_n$, where $[a(x)]_\times = \sum_{n=1}^d (a(x))_n E_n \in \mathscr{G}$ and $(a(x))_{n=1}^d$ are referred to as the coordinate functions of V.

where the Euclidean norm $|\imath(x)-\imath(z)| := |x-z|_F$, the Frobenius norm for matrices $x,z \in G$. It is a straightforward calculation to obtain the coordinate functions of the vector field

$$(s(x;z))_n = (xE_n)\cdot|x-z|_F^2 = 2\,\mathrm{tr}(xE_n(x-z)^\top) \tag{17}$$

used in the kernel algorithm (see Step 1 in Sec. II-D).

Example 1: The special orthogonal group $SO(3)$ is the group of 3×3 matrices R such that $RR^\top = I$ and $\det(R) = 1$. The Lie algebra $so(3)$ is the 3-dimensional vector space of skew-symmetric matrices. An orthonormal basis $\{E_1,E_2,E_3\}$ of $so(3)$ is given by

$$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix},\ E_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},\ E_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

These matrices have the physical interpretation of generating rotations about the three canonical axes in $\mathbb{R}^3$.

Example 2: Unit quaternions provide a computationally efficient coordinate representation for $SO(3)$. It has the form

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

and has a physical interpretation of a 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 $q_I := (1,0,0,0)$, the inverse $q^{-1} := (q_0,-q_1,-q_2,-q_3)$, and the multiplication operation 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 respectively between two vectors in $\mathbb{R}^3$.

Given a unit quaternion q, the corresponding rotation matrix

$$R(q) = \begin{bmatrix} 2q_0^2+2q_1^2-1 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 2q_0^2+2q_2^2-1 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 2q_0^2+2q_3^2-1 \end{bmatrix} \tag{18}$$

For a comprehensive introduction of matrix Lie groups and quaternions, we refer the reader to [60], [61], [62].

B. FPF on Matrix Lie Groups

Problem Statement: The general form of the nonlinear filtering problem on a matrix Lie group is as follows:

$$\begin{aligned} &\text{Signal:} && \mathrm{d}X_t = X_t\,[a_0(X_t)]_\times\,\mathrm{d}t + X_t\,[a_1(X_t)]_\times \circ \mathrm{d}B_t \\ &\text{Observation:} && \mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t \end{aligned}$$

where (an $n\times n$ matrix) $X_t \in G$ is the state at time t, $a_0 : G\to\mathbb{R}^d$ and $a_1 : G\to\mathbb{R}^d$ are $\mathbb{R}^d$-valued functions (process model), $h : G\to\mathbb{R}$ is a real-valued function on G (observation model), and B_t, W_t are mutually independent standard Wiener processes in $\mathbb{R}$. The problem is to numerically approximate the conditional distribution of X_t.

Solution: The FPF has the following general form:

$$\mathrm{d}X_t^i = X_t^i\,[a_0(X_t^i)]_\times\,\mathrm{d}t + X_t^i\,[a_1(X_t^i)]_\times \circ \mathrm{d}B_t^i + X_t^i\,[l_t(X_t^i)]_\times \circ \mathrm{d}\mathrm{I}_t^i \tag{19}$$

where the formula for the error $\mathrm{d}\mathrm{I}_t^i = \mathrm{d}Z_t - \frac{1}{2}(h(X_t^i)+\hat{h}_t)\,\mathrm{d}t$ is the same as before (10). The gain $\mathsf{K}_t(x) = x[l_t(x)]_\times$ where the coordinates of the map $l_t : G\to\mathbb{R}^d$ are obtained as

$$(l_t(x))_n = (xE_n)\cdot\phi(x)$$

where $\phi = \mathscr{P}\mathscr{E}(h)$ is a solution of the Poisson equation (6). The kernel approximation of these coordinates is obtained as (see (16)):

$$(l_t(X_t^i))_n = -\frac{1}{2\varepsilon}\sum_{l=1}^N \mathsf{r}_l\,\mathsf{T}_{il}^{(\varepsilon,N)}\mathrm{tr}\big(X_t^iE_n(X_t^i - X_t^l)^\top\big) \tag{20}$$

where $\mathsf{T}^{(\varepsilon,N)}$ is the $N\times N$ Markov matrix (Step 2 in Sec. II-D) and the formula for r_l is given by (15) (Step 4 in Sec. II-D).

The pseudo-code for FPF on matrix Lie groups appears as Algorithm 1 in the accompanying table. In a numerical implementation, N particles (matrices) $\{X_t^i\}_{i=1}^N$ are simulated according to a discretization of the sde (19). The initial condition $\{X_0^i\}_{i=1}^N$ is sampled i.i.d. from a given (prior) distribution π_0^* on G. Matrix exponential is used to approximate the discrete-time evolution $X_t^i \to X_{t+\Delta t}^i$ of the particles (Line 10 in Algorithm 1). In examples such as $SO(3)$ and $SE(3)$, the Rodrigues formula ([61, Sec. 10.6.6]) can be used to efficiently compute the exponential map in closed-form.

The use of exponential map for numerical integration is referred to as the geometric Euler-Maruyama scheme on a matrix Lie group [63]. It ensures that the particles stay on the manifold. The discretization is consistent with the Stratonovich form of the sde: In the limit as $\Delta t\to 0$, the approximate solution converges to the continuous-time solution obtained from the Stratonovich form of the sde [64, Sec. VIII-E]. More advanced algorithms for numerical integration of a sde on matrix Lie groups and manifolds appear in [65], [66].

Remark 6 (FPF on manifolds and matrix Lie groups): Compared to the general Riemannian manifolds, the FPF on matrix Lie groups has a more tractable explicit form. The simplification arises on account of the following reasons:

Algorithm 1 Feedback Particle Filter on matrix Lie groups

Initialization: Sample $\{X_0^i\}_{i=1}^N$ i.i.d. from π_0^*
Assign $t = 0$
Iteration: from t to $t+\Delta t$
Calc. $\hat{h}_t^{(N)} = \frac{1}{N}\sum_{i=1}^N h(X_t^i)$
for $i = 1$ to N **do**
  Generate samples ΔB_t^i from $\mathscr{N}(0, \Delta t)$
  Calc. the error $\Delta \mathrm{I}_t^i = \Delta Z_t - \frac{1}{2}\big(h(X_t^i) + \hat{h}_t^{(N)}\big)\Delta t$
  Calc. $l_t(X_t^i)$ (see Algorithm 2)
  Calc. $\Delta a_t^i = a_0(X_t^i)\Delta t + a_1(X_t^i)\Delta B_t^i + l_t(X_t^i)\Delta \mathrm{I}_t^i$
  Propagate the particle $X_{t+\Delta t}^i = X_t^i \exp\big([\Delta a_t^i]_\times\big)$
end for
Assign $t = t + \Delta t$

Algorithm 2 Kernel-based gain function approximation

Input: Particles $\{X_t^i\}_{i=1}^N$, parameters ε, MaxIter
Calc. $h^{(N)} = \big(h(X^i), \dots, h(X^N)\big)$ and $\hat{h}^{(N)} = \frac{1}{N}\sum_{i=1}^N h(X^i)$
Calc. the distance $\mathsf{d}_{ij} = |\imath(X^i) - \imath(X^j)|$ for all i, j
Calc. $k^{(\varepsilon)}(\mathsf{d}_{ij})$ and $\mathsf{T}_{ij}^{(\varepsilon,N)}$ according to (13) for all i, j
Assign initial condition ϕ_0
for $k = 0$ to MaxIter **do**
  Calc. $\phi_{k+1} = \mathsf{T}^{(\varepsilon,N)}\phi_k + \varepsilon(h^{(N)} - \hat{h}^{(N)})$
  Assign $\phi_{k+1} = \phi_{k+1} - \frac{1}{N}\sum_{i=1}^N (\phi_{k+1})_i$
end for
Calc. $(l_t(X^i))_n$ for $n = 1, \dots, d$ according to (20) with r_l given by (15) and $\phi^{(\varepsilon,N)} = \phi_{\text{MaxIter}}$
Return: $\{l_t(X_t^1), \dots, l_t(X_t^N)\}$

(i) Use of matrices to represent and manipulate elements of the matrix Lie groups is computationally convenient in scientific programming environments such as Python or Matlab.

(ii) Matrix Lie groups are parallelizable, i.e., there exists a (left-invariant) global frame. Consequently, it is convenient to represent vector-fields, e.g., the gain vector-field as $\mathsf{K} = x[l(x)]_\times$ where $l(x) \in \mathbb{R}^d$.

(iii) The infinitesimal generator is an element of the Lie algebra. This is useful in several ways, e.g., to derive an explicit formula for the gradient.

(iv) For the purposes of numerical integration, the exponential map is easily computed in terms of the matrix exponential. In important examples such as $SO(2)$, $SO(3)$ and $SE(3)$, there is a closed form formula for the matrix exponential. The numerical solutions thus obtained automatically satisfy the manifold constraints.

(v) In the kernel approximation, matrix coordinates provide an isometric embedding, the matrix Frobenius norm is used as an Euclidean norm, and certain expressions (e.g. (17)) involving the gradient can be computed in closed-form.

For both manifolds and matrix Lie groups, the hard part is to approximate the gain function. For the Galerkin algorithm, there is often a standard choice of basis functions for matrix Lie groups: basis functions for $SO(3)$ appear in [49]. Some of the simplifications for the kernel algorithm on matrix Lie groups are listed above. However, the core computational problem of assembling a Markov matrix on a graph, and using it to define and solve a fixed-point problem is no easier in matrix Lie groups or harder in more general settings. □

C. Filtering on $SO(3)$: The Attitude Estimation Problem

Process model: A kinematic model of rigid body attitude is given by

$$\mathrm{d}R_t = R_t\,[\,\omega_t\,]_\times\,\mathrm{d}t + R_t \circ [\,\mathrm{d}B_t\,]_\times \tag{21}$$

where $R_t \in SO(3)$ is the attitude (state) at time t, expressed with respect to an inertial frame; $\omega_t \in \mathbb{R}^3$ represents the angular velocity expressed in the body frame; B_t is a Wiener process in $\mathbb{R}^3$. Both $[\,\omega_t\,]_\times$ and $[\,\mathrm{d}B_t\,]_\times$ are elements of the Lie algebra $so(3)$.

Accelerometer: In the absence of translational motion, the accelerometer is modeled as (see [34])

$$\mathrm{d}Z_t^g = -R_t^\top r^g\,\mathrm{d}t + \mathrm{d}W_t^g \tag{22}$$

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

Magnetometer: The model of the magnetometer is of a similar form (see [34])

$$\mathrm{d}Z_t^b = R_t^\top r^b\,\mathrm{d}t + \mathrm{d}W_t^b \tag{23}$$

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 Wiener process in $\mathbb{R}^3$.

Observation model: Concatenating the observations (22) and (23) as $Z_t := (Z_t^g, Z_t^b)$, $h(R_t) := (-R_t^\top r^g, R_t^\top r^b)$, and $W_t := (W_t^g, W_t^b)$, the observation model is succinctly expressed as

$$\mathrm{d}Z_t = h(R_t)\,\mathrm{d}t + \mathrm{d}W_t \tag{24}$$

The observation vector is thus 6-dimensional (3-axes accelerometer and 3-axes gyroscope). The 6×6 covariance matrix of the measurement noise W_t is denoted as Q_W. It is assumed to be positive-definite.

The problem is to estimate the attitude given the accelerometer and the gyroscope measurements.

FPF using rotation matrices: The FPF for the attitude estimation problem (21), (24) is given by

$$\mathrm{d}R_t^i = R_t^i\,[\,\omega_t\,]_\times\,\mathrm{d}t + R_t^i \circ [\,\mathrm{d}B_t^i\,]_\times + R_t^i\,[\,l_t(R_t^i)Q_W^{-1} \circ \mathrm{d}\mathrm{I}_t^i\,]_\times \tag{25}$$

where the error $\mathrm{d}\mathrm{I}_t^i = \mathrm{d}Z_t - \frac{1}{2}(h(R_t^i) + \hat{h}_t)\,\mathrm{d}t$ is a 6×1 (column) vector and $l_t(R) = [\,l_t^1(R)\,|\,l_t^2(R)\,|\,\cdots\,|\,l_t^6(R)\,]$ is a 3×6 matrix whose j^{th} column is a 3×1 vector with coordinates $(l_t^j(R))_n = (RE_n) \cdot \phi^j(R)$ for $n = 1, 2, 3$, where $\phi^j = \mathscr{P}\mathscr{E}(h^j)$. The kernel approximation of these coordinates is given by (see (20))

$$(l_t^j(R_t^i))_n = \frac{1}{2\varepsilon}\sum_{l=1}^N \mathsf{r}_l\,\mathsf{T}_{il}^{(\varepsilon,N)}\,\mathrm{tr}\big(R_t^i E_n R_t^{l\top}\big)$$

where $\mathsf{T}^{(\varepsilon,N)}$ and r_l are defined in Sec. II-D.

FPF using quaternions: Quaternions provide a computationally efficient representation of the filter as

$$\mathrm{d}q_t^i = \frac{1}{2}\,q_t^i \otimes (0, \mathrm{d}\nu_t^i) \tag{26}$$

Algorithm 3 Feedback Particle Filter for attitude estimation

1: **Initialization:** Sample $\{q_0^i\}_{i=1}^N$ i.i.d. from π_0^*
2: Assign $t=0$
3: **Iteration:** from t to $t+\Delta t$
4: Calc. $\hat{h}_t^{(N)} = \frac{1}{N}\sum_{i=1}^N h(R(q_t^i))$
5: **for** $i=1$ to N **do**
6: Generate a sample, ΔB_t^i, from $\mathscr{N}\big(0,(\Delta t)I\big)$
7: Calc. the error $\Delta \mathrm{I}_t^i = \Delta Z_t - \frac{1}{2}\big(h(R(q_t^i)) + \hat{h}_t^{(N)}\big)\Delta t$
8: Calc. $l_t^j(R(q_t^i))$ according to Algorithm 2 with $h=h^j$ for $j=1,\dots,6$.
9: Calc. $l_t(R(q_t^i)) = [l_t^1(R(q_t^i))\,|\,l_t^2(R(q_t^i))\,|\cdots|\,l_t^6(R(q_t^i))]$
10: Calc. $\Delta v_t^i = \omega_t\,\Delta t + \Delta B_t^i + l_t(R(q_t^i))Q_W^{-1}\,\Delta \mathrm{I}_t^i$
11: Propagate the particle q_t^i according to (see [62], and $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^3$)

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

12: **end for**
13: Assign $t = t+\Delta t$

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

$$\mathrm{d}v_t^i = \omega_t\,\mathrm{d}t + \mathrm{d}B_t^i + l_t(R(q_t^i))Q_W^{-1} \circ \mathrm{d}\mathrm{I}_t^i \tag{27}$$

where l_t is the 3×6 matrix (same as before) and the formula for $R(q)$ is given in (18).

The pseudo-code for numerically implementing the FPF-based attitude filter using quaternions appears in Algorithm 3. A closed-form formula for the exponential is used to approximate the discrete-time evolution $q_t^i \to q_{t+\Delta t}^i$ of the particles (Line 11 in Algorithm 3).

D. Filtering on $SE(3)$: The Robot Localization Problem

The special Euclidean group $SE(3)$ is the semi-direct product of $SO(3)$ and $\mathbb{R}^3$. Its Lie algebra is denoted as $se(3)$. Arbitrary elements $X \in SE(3)$ and $A \in se(3)$ are represented as follows:

$$X = \begin{bmatrix} R & p \\ 0 & 1\end{bmatrix}, \qquad A = \begin{bmatrix} [\omega]_\times & v \\ 0 & 0\end{bmatrix}$$

where $R \in SO(3)$, $p \in \mathbb{R}^3$, $\omega \in \mathbb{R}^3$, and $v \in \mathbb{R}^3$. The Lie algebra is a 6-dimensional vector space.

Process model: A kinematic model of rigid body attitude and translation is given by

$$\mathrm{d}R_t = R_t\,[\,\omega_t\,]_\times\,\mathrm{d}t + R_t \circ [\,\mathrm{d}B_t^\omega\,]_\times \tag{28a}$$
$$\mathrm{d}p_t = R_t v_t\,\mathrm{d}t + R_t \circ \mathrm{d}B_t^v \tag{28b}$$

where $R_t \in SO(3)$ is the attitude and $p_t \in \mathbb{R}^3$ is the position of the rigid body at time t, ω_t is the angular velocity and v_t is the translational velocity, both expressed in the body frame, and B^ω, B^v are Wiener processes in $\mathbb{R}^3$.

Observation model: The landmarks are observed in the body frame (see [36], [67]):

$$\mathrm{d}Z_t^\beta = R_t^\top(p_\beta - p_t)\,\mathrm{d}t + \mathrm{d}W_t^\beta \tag{29}$$

where Z_t^β denotes the observation due to the β^{th} landmark with a known position $p_\beta \in \mathbb{R}^3$, and W_t^β is a Wiener process in $\mathbb{R}^3$. Upon concatenating the observations due to r landmarks, the observation model is succinctly expressed as

$$\mathrm{d}Z_t = h(R_t,p_t)\,\mathrm{d}t + \mathrm{d}W_t$$

where $Z_t, W_t \in \mathbb{R}^{3r}$. The $3r\times 3r$ covariance matrix of the measurement noise W_t is denoted as Q_W. It is assumed to be positive-definite.

The robot localization problem is to estimate the attitude and the position of the rigid body given the landmark measurements.

FPF using the matrix coordinates: The FPF for the robot localization problem (28a)-(29) is given by

$$\mathrm{d}R_t^i = R_t^i\,[\,\omega_t\,]_\times\,\mathrm{d}t + R_t^i \circ [\,\mathrm{d}B_t^{\omega,i}\,]_\times + R_t^i\,[\,l_t^\omega(R_t^i,p_t^i)Q_w^{-1} \circ \mathrm{d}\mathrm{I}_t^i\,]_\times$$
$$\mathrm{d}p_t^i = R_t^i v_t\,\mathrm{d}t + R_t^i \circ \mathrm{d}B_t^{v,i} + R_t^i\,l_t^v(R_t^i,p_t^i)Q_w^{-1} \circ \mathrm{d}\mathrm{I}_t^i$$

where $\mathrm{d}\mathrm{I}_t^i = \mathrm{d}Z_t - \frac{1}{2}(h(R_t^i,p_t^i)+\hat{h}_t)\,\mathrm{d}t$ is a $3r\times 1$ column vector. The gain is obtained in terms of two $3\times 3r$ matrices:

$$l_t^\omega(R,p) := [l_t^{\omega,1}(R,p)\,|\,l_t^{\omega,2}(R,p)\,|\cdots|\,l^{\omega,3r}(R,p)]$$
$$l_t^v(R,p) := [l_t^{v,1}(R,p)\,|\,l_t^{v,2}(R,p)\,|\cdots|\,l^{v,3r}(R,p)]$$

whose j^{th} column is a 3×1 vector with coordinates

$$(l_t^{\omega,j}(R,p))_n = (RE_n)\cdot\phi^j(R,p), \quad (l_t^{v,j}(R,p))_n = \frac{\partial\phi^j}{\partial p_n}(R,p)$$

where $\phi^j = \mathscr{P}\mathscr{E}(h^j)$. The kernel approximation of these coordinates is given by (see (20))

$$(l_t^{\omega,j}(R_t^i,p_t^i))_n = \frac{1}{2\varepsilon}\sum_{l=1}^N \mathsf{r}_l\,\mathsf{T}_{il}^{(\varepsilon,N)}\,\mathrm{tr}\big(R_t^i E_n R_t^{l\top}\big)$$
$$(l_t^{v,j}(R_t^i,p_t^i))_n = -\frac{1}{2\varepsilon}\sum_{l=1}^N \mathsf{r}_l\,\mathsf{T}_{il}^{(\varepsilon,N)} e_n^\top R_t^{i\top}(p_t^i - p_t^l)$$

where $\mathsf{T}^{(\varepsilon,N)}$ and r_l are defined in Sec. II-D, and e_n for $n=1,2,3$ are the canonical basis vectors in $\mathbb{R}^3$.

IV. FPF with Concentrated Distributions

In its original Euclidean setting [46], the FPF algorithm is shown to represent a generalization of the Kalman filter in the following sense: Suppose the signal and observation models are linear and the prior distribution is Gaussian. Then

(i) The gain K_t is a constant for each t whose value equals the Kalman gain;

(ii) The conditional distribution of X_t^i is Gaussian whose mean and covariance evolve according to the Kalman filter.

For the general nonlinear non-Gaussian (Euclidean) case, the gain K_t is no longer a constant and must be numerically approximated. However, the expected value of the gain, $\pi_t(\mathsf{K}_t)$, admits a closed-form expression which can furthermore be approximated using only the particles. The resulting approximation is referred to as the constant gain approximation. This approximation reduces to the Kalman gain in the linear Gaussian case. For the general case, this approximation often suffices in practice particularly so when the conditional distribution is unimodal [46], [68], [69].

On a Riemannian manifold, unfortunately, even the state space does not possess a linear structure. However, under the additional assumption that the posterior distribution is "concentrated" (see [41], [42]), one can expect the results to be close to the Euclidean case. In this section, the following is shown for the special case of concentrated distributions on a manifold:

(i) A closed-form formula for the constant gain approximation is derived and shown to equal the Kalman gain;

(ii) The equation for the mean and covariance are derived and shown to be closely related to the continuous-time left invariant EKF algorithm in [18].

In this section, we restrict our attention to the attitude estimation problem on $SO(3)$ modeled by (21) and (24). The restriction to $SO(3)$ is not necessary but leads to a simpler presentation without undue notational burden. Also, it allows us to make comparisons with the literature on filters for attitude estimation.

A. Constant Gain Approximation of FPF

Definition 2: The distribution of the random variable R on $SO(3)$ is $(\mu,\bar{\Sigma})$-*concentrated* if it is parametrized as:

$$R = \mu \exp(\varepsilon\,[\,\chi\,]_\times)$$

where $\mu \in SO(3)$ is constant, $\chi \in \mathbb{R}^3$ is a random variable with mean 0 and covariance matrix $\bar{\Sigma}$, and ε is a small parameter. □

The concentrated distribution is illustrated in Fig. 2. Formally, most of the probability mass of a concentrated distribution is supported in a small neighborhood of μ, and the analysis pertains to the consideration of the asymptotic limit as $\varepsilon \to 0$. The following proposition provides an approximate formula for the gain in this special case. The proof appears in Appendix D.

Proposition 2: Consider the Poisson equation (6) on $SO(3)$ where the distribution π_t of the random variable $R_t = \mu_t \exp(\varepsilon\,[\,\chi_t\,]_\times)$ is $(\mu_t,\bar{\Sigma}_t)$-concentrated and $h : SO(3) \to \mathbb{R}$ is a real-valued function. Then, in the asymptotic limit as $\varepsilon \to 0$, the gain is $\mathsf{K}_t(R) = R[\,l_t(R)\,]_\times$ with

$$\pi_t(l_t) = \underbrace{\varepsilon\,\pi_t\big((h-\hat{h}_t)\chi_t\big)}_{\text{dominant term}} + \text{ h.o.t.} \tag{30}$$

where h.o.t. is an abbreviation for the "higher order terms" in the small paramater ε. □

For the attitude estimation problem with a 6-dimensional observation model (24) with $h = (h^1,\dots,h^6)$, the gain is defined in terms of a 3×6 matrix also denoted as $l_t(R)$: the j^{th}-column of the matrix is obtained from the solution $\phi^j = \mathscr{P}\mathscr{E}(h^j)$ for $j = 1,\dots,6$ (see Sec. III-C). In Appendix D, the dominant term of the asymptotic formula (30) is explicitly evaluated for the observation model (24) of the attitude estimation problem and shown to be $O(\varepsilon^2)$.

The *constant gain approximation* is obtained through setting l_t in FPF to its expected value (a constant matrix) which

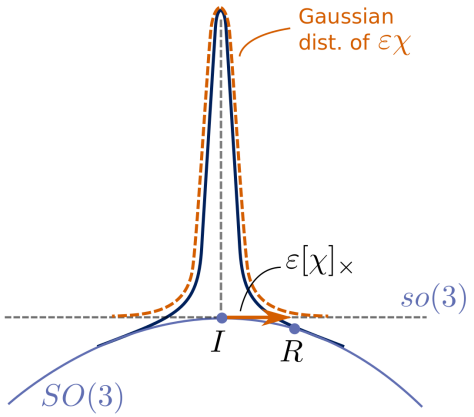


Fig. 2: Concentrated distribution on $SO(3)$ with mean at the identity. The random rotation matrix $R = \exp(\varepsilon\,[\,\chi\,]_\times)$ where $[\,\chi\,]_\times$ is a Gaussian random vector in the Lie algebra $so(3) \cong \mathbb{R}^3$. A concentrated distribution with an arbitrary mean $\mu \in SO(3)$ is obtained by left-translation whereby $R = \mu \exp(\varepsilon\,[\,\chi\,]_\times)$; c.f., [20].

for small ε is approximated by the dominant term in the asymptotic formula:

$$\pi_t(l_t) \;\approx\; \underbrace{\varepsilon^2\bar{\Sigma}_t\mathsf{H}_t}_{\text{dominant term}} \;=:\; \varepsilon^2\bar{l}_t \tag{31}$$

where $\mathsf{H}_t := \begin{bmatrix} -[\,\mu_t^\top r^g\,]_\times & [\,\mu_t^\top r^b\,]_\times \end{bmatrix}$ is a 3×6 matrix.

B. FPF with a Constant Gain Approximation

Assumption 3: The prior $R_0 = \mu_0 \exp(\varepsilon\,[\,\chi_0\,]_\times)$ where μ_0 is known and $\chi_0 \sim \mathcal{N}(0,\bar{\Sigma}_0)$. The process covariance $Q_B = \varepsilon^2\bar{Q}_B$ and the measurement covariance $Q_W = \varepsilon^2\bar{Q}_W$.

Based on Assumption 3 and some related results on error propagation and Bayesian fusion in matrix Lie groups [41], [70], [42], the conditional distribution of R_t^i is $(\mu_t,\bar{\Sigma}_t)$-concentrated for small time $t \in [0,\varepsilon T]$. That is, $R_t^i = \mu_t \exp(\varepsilon\,[\,\chi_t^i\,]_\times)$ where $\chi_t^i \in \mathbb{R}^3$ has mean zero and covariance $\bar{\Sigma}_t$. The $SO(3)$ FPF (25) with the constant gain approximation (upon setting l_t to $\varepsilon^2\bar{l}_t = \varepsilon^2\bar{\Sigma}_t\,\mathsf{H}_t$) is given by

$$\mathrm{d}R_t^i = R_t^i\,[\,\omega_t\,]_\times\,\mathrm{d}t + R_t^i \circ [\,\mathrm{d}B_t^i\,]_\times + R_t^i\,[\,\bar{l}_t\,\bar{Q}_W^{-1} \circ \mathrm{dI}_t^i\,]_\times \tag{32}$$

In the following theorem, it is shown that μ_t and $\bar{\Sigma}_t$ evolve according to the equations that are closely related to the left invariant EKF. The proof is contained in Appendix E.

Theorem 3: Consider the FPF (32) with the constant gain approximation. Suppose also Assumption 3 and a time-horizon $t \in [0,\varepsilon T]$ such that $R_t^i = \mu_t \exp(\varepsilon\,[\,\chi_t^i\,]_\times)$ where $\chi_t^i \in \mathbb{R}^3$ has mean zero and covariance $\bar{\Sigma}_t$. Then, in the asymptotic limit as $\varepsilon \to 0$, μ_t and $\bar{\Sigma}_t$ evolve according to the respective sdes:

$$\mathrm{d}\mu_t = \mu_t\,[\,\omega_t\,]_\times\,\mathrm{d}t + \mu_t\,[\,\bar{l}_t\,\bar{Q}_W^{-1} \circ \mathrm{dI}_t\,]_\times \tag{33}$$

$$\begin{aligned}\mathrm{d}\bar{\Sigma}_t = {} & -([\,\omega_t\,]_\times\,\mathrm{d}t)\,\bar{\Sigma}_t - \bar{\Sigma}_t\,([\,\omega_t\,]_\times\,\mathrm{d}t)^\top + \bar{Q}_B\,\mathrm{d}t \\ & - ([\,\bar{l}_t\,\bar{Q}_W^{-1}\,\mathrm{dI}_t\,]_\times)\,\bar{\Sigma}_t - \bar{\Sigma}_t\,([\,\bar{l}_t\,\bar{Q}_W^{-1}\,\mathrm{dI}_t\,]_\times)^\top \\ & - \bar{\Sigma}_t\,\mathsf{H}_t\,\bar{Q}_W^{-1}\,\mathsf{H}_t^\top\,\bar{\Sigma}_t\,\mathrm{d}t\end{aligned} \tag{34}$$

where $\mathrm{dI}_t = \mathrm{d}Z_t - h(\mu_t)\,\mathrm{d}t$. □

The equation for the mean (33) is identical to the left invariant EKF [18]. The equation of the covariance (34) includes additional terms that depend on the innovation process I_t. Analogous stochastic terms for updating the covariance, though in a discrete-time setting, have also appeared in [20].

V. Numerics

In this section, results of two numerical studies are presented for filters on $SO(3)$: (i) in Sec. V-A, an attitude estimation problem; and (ii) in Sec. V-B, a filtering problem for a bimodal prior supported on a subgroup $SO(2) \subset SO(3)$. In all cases, the FPF is implemented using the quaternion coordinates (Algorithm 3).

A. Attitude Estimation

Consider an attitude estimation problem with observations from both accelerometer and magnetometer:

$$\mathrm{d}q_t = \frac{1}{2} q_t \otimes \big(\omega_t \,\mathrm{d}t + \sigma_B \,\mathrm{d}B_t\big) \tag{35a}$$

$$\mathrm{d}Z_t = \begin{bmatrix} -R(q_t)^\top & 0 \\ 0 & R(q_t)^\top \end{bmatrix} \begin{bmatrix} r^g \\ r^b \end{bmatrix} \mathrm{d}t + \sigma_W \,\mathrm{d}W_t \tag{35b}$$

where the model for angular velocity is taken from [44]:

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

and $r^g = (0,0,1)$, $r^b = (1/\sqrt{2}, 0, 1/\sqrt{2})$ are assumed to be aligned with the gravity and the local magnetic field, respectively. B_t and W_t are standard Wiener processes, and σ_B, σ_W are scalars.

The following filters are simulated for the comparison:

(i) **MEKF:** the multiplicative EKF algorithm described in [25], [62] using the modified Rodrigues parameter.

(ii) **IEKF:** the invariant EKF algorithm described in [3].

(iii) **USQUE:** the unscented quaternion estimator described in [29] also using the modified Rodrigues parameter.

(iv) **IEnKF:** the invariant ensemble Kalman filter described in [3].

(v) **BPF:** the bootstrap particle filter described in [30] also using the modified Rodrigues parameter.

(vi) **FPF-G:** the FPF using the Galerkin gain approximation described in [49] with the basis functions provided therein.

(vii) **FPF-K:** the FPF using the kernel-based gain approximation (Algorithm 3 in Sec. II-D) with the parameter $\varepsilon = 1$.

(viii) **FPF-C:** the FPF using the constant gain approximation described in Sec. IV-A.

Except for FPF, the other filters are discrete-time filters. They require a discrete-time filtering model that is consistent with the continuous-time model (35a)-(35b): For the discrete-time filters, the sampled observations, denoted as $\{Y_{t_n}\}$, are made at discrete times $\{t_n\}$, whose model is formally expressed as $Y_{t_n} := \frac{\Delta Z_{t_n}}{\Delta t} = h(q_{t_n}) + W^\Delta_{t_n}$ where $\{W^\Delta_{t_n}\}$ are i.i.d. with the distribution $\mathcal{N}(0, \frac{\sigma_W^2}{\Delta t} I)$. Such a model leads to the correct scaling between the continuous and the discrete-time filter implementations. To provide a fair comparison, the same set of observations are used for all the continuous-time and the discrete-time algorithms.

The simulations are carried out over a finite time-horizon $t \in [0, T]$ with fixed time step Δt. The performance metric is evaluated in terms of the *rotation angle error* defined as follows: Let q_t and $\hat{q}_t$ denote the true and the estimated attitude, respectively, at time t. The estimation error is defined as $\delta q_t := \hat{q}_t^{-1} \otimes q_t$ and the rotation angle error $\delta\alpha_t := 2\arccos(|\delta q_t^0|) \in [0, \pi]$, where δq_t^0 denotes the first component of δq_t.

For particle filters, $\hat{q}_t$ is the empirical mean of the particles $\{q_t^i\}_{i=1}^N$ obtained as the eigenvector (with norm 1) of the 4×4 matrix $Q = \frac{1}{N}\sum_{i=1}^N q_t^i q_t^{i\top}$ corresponding to its largest eigenvalue [71].

In an experiment, each filter is simulated over J independent Monte Carlo runs. For the j-th Monte Carlo run, $\delta\alpha_t^j$ denotes the rotation angle error as a function of time. The average error of the J Monte Carlo runs as a function of time is defined as

$$\widehat{\delta\alpha}_t := \frac{1}{J}\sum_{j=1}^J \delta\alpha_t^j \tag{36}$$

The time-averaged error for the j-th run is defined as

$$\langle \delta\alpha^j \rangle_T := \frac{1}{T}\int_0^T \delta\alpha_t^j \,\mathrm{d}t \tag{37}$$

and the time-averaged error of the J runs is defined as

$$\langle \widehat{\delta\alpha} \rangle_T := \frac{1}{J}\sum_{j=1}^J \langle \delta\alpha_t^j \rangle_T \tag{38}$$

For the FPF implementation, the initial set of particles are sampled as follows: First, $\{v^i\}_{i=1}^N$ are sampled i.i.d. from the Gaussian distribution $\mathcal{N}(0, \sigma_0^2 I)$ in $\mathbb{R}^3$. Next, the particles $\{q_0^i\}_{i=1}^N$ are obtained by

$$q_0^i = q_0 \otimes \begin{bmatrix} \cos\big(|v^i|/2\big) \\ \frac{v^i}{|v^i|}\sin\big(|v^i|/2\big) \end{bmatrix}$$

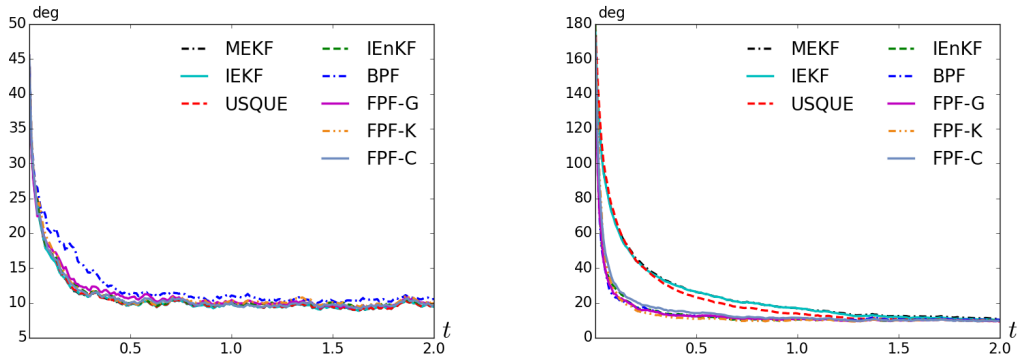
The IEnKF and BPF use the same initial particles as FPF.

In numerical simulations, it was observed that the continuous-time filters, especially the FPF-G, are susceptible to numerical instabilities due to high gain during the initial transients. In order to mitigate the numerical issues observed during the implementation of the FPF-G algorithm, the discrete time-step during the initial transients is further sub-divided. Specifically, for $t < T_f$, the time interval $[t, t+\Delta t]$ is uniformly divided into N_f sub-intervals. The update step in the FPF (specifically Line 4 – Line 11 in Algorithm 3) is implemented on each sub-interval by replacing ΔZ_t with $\frac{\Delta Z_t}{N_f}$ and Δt with $\frac{\Delta t}{N_f}$.

The nominal parameter values are chosen as: $T = 2$, $\Delta t = 0.01$, $N = 100$, $J = 100$, $T_f = 0.2$, $N_f = 100$. The choice of T_f and N_f may vary according to the severity of numerical issues encountered in practice.

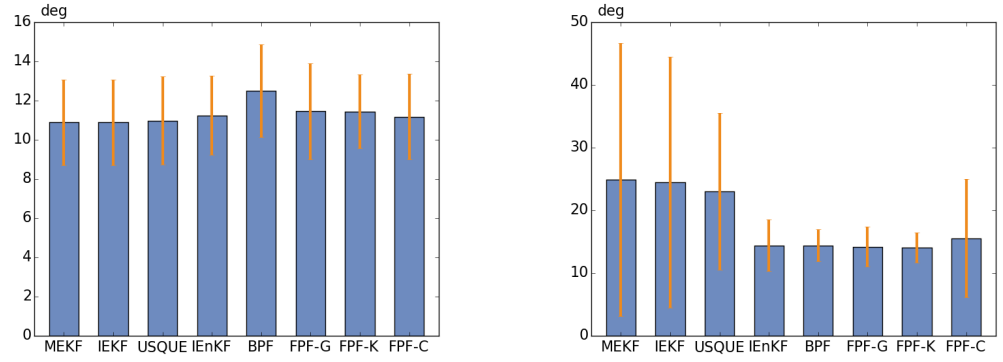
The simulation results are discussed next:

(i) **The average error $\widehat{\delta\alpha}_t$ as a function of the initial uncertainty:** Fig. 3 depicts the average error $\widehat{\delta\alpha}_t$ (obtained using (36)) of the filters over $J = 100$ simulation runs, with two choices of initial variance: (a) $\Sigma_0 = 0.5236^2 I$ and (b) $\Sigma_0 = 1.0472^2 I$. The two cases correspond to a standard deviation of 30° and 60°, respectively. For the two priors, the



(a) Initial distribution with $\sigma_0 = 30^\circ$ (b) Initial distribution with $\sigma_0 = 60^\circ$

Fig. 3: Comparison of the average error $\widehat{\delta\alpha}_t$: The performance is nearly identical across filters for case (a) when the initial uncertainty (of the prior) is small. For case (b) when the initial uncertainty is large, the particle-based filters exhibit superior performance compared to the Kalman filters and the unscented filter.



(a) Initial distribution with $\sigma_0 = 30^\circ$ (b) Initial distribution with $\sigma_0 = 60^\circ$

Fig. 4: Variability in the filter performance across multiple ($J = 100$) Monte-Carlo runs: The bars indicate the mean and the bounds indicate the ± 1 standard deviation of the error $\{\langle \delta\alpha^j \rangle_T\}_{j=1}^J$. In case (b) when the initial uncertainty (of the prior) is large, the Kalman filters and the unscented filter exhibit larger variability than the particle-based filters.

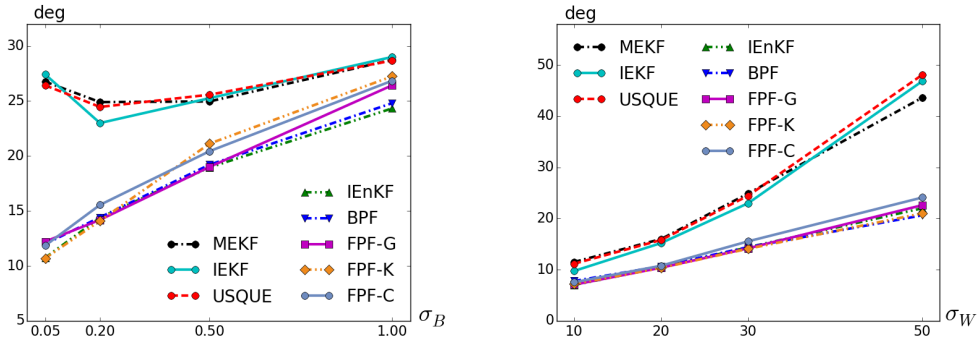
mean is the same, given by identity quaternion $q_I = (1,0,0,0)$. For case (a), the target is initialized by sampling from the prior distribution. For case (b), the target is initialized with a fixed attitude – rotation of 180° about the axis $(3,1,4)$. These parameters indicate large estimation error initially for case (b).

The results depicted in Fig. 3 show that the performance is nearly identical across filters for case (a) when the initial uncertainty is small. For case (b) when the initial uncertainty is large, the particle-based filters (IEnKF, BPF and FPF) exhibit superior performance compared to the Kalman filters (MEKF, IEKF) and the unscented filter (USQUE). The differences are exhibited in the speed of convergence of the estimate to the target with the particle-based filters converging quickly compared to the Kalman filters and the unscented filter.

As the results in Fig. 3 are averaged over multiple Monte-Carlo runs, statistical analysis was also carried out to assess the variability in performance across runs. The results of this analysis are presented in Fig. 4, which depicts the mean and standard deviation of $\{\langle \delta\alpha^j \rangle_T\}_{j=1}^J$ (see (37)). Apart from poorer performance on average, the Kalman filters also exhibit a greater variability in performance across the Monte-Carlo runs. For some trajectories, the Kalman filters exhibit slow convergence because the gain becomes very small.

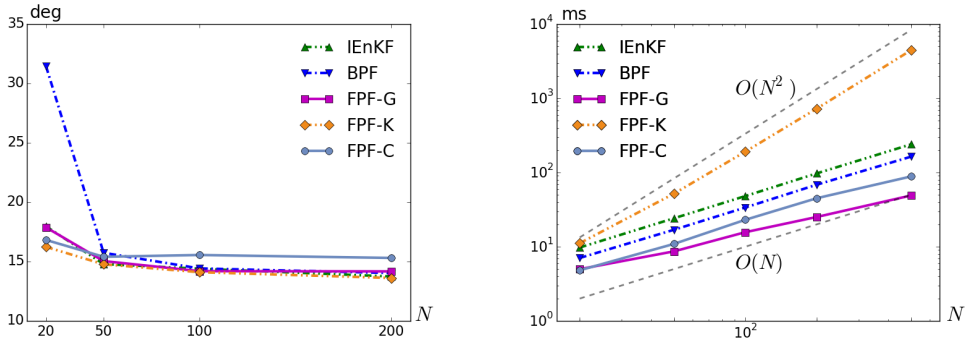
(ii) **The time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$ as a function of the process noise:** In this simulation, the process noise parameter $\sigma_B \in \{0.05, 0.2, 0.5, 1.0\}$ for fixed $\sigma_W = 0.05236$ and prior distribution according to case (b) in Fig. 3. Fig. 5 (a) depicts the time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$ (see (38)) across filters as the process noise parameter is varied. One would have expected the error to increase monontonically with the σ_B value. The fact that such is not the case for the Kalman filters indicates that the relatively poor performance of the Kalman filters for small values of process noise is an artifact of the linearization assumption that leads to overly small gains. These small gains adversely effect the filter performance during the initial transients.

(iii) **The time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$ as a function of the observation noise:** In this simulation, the observation noise parameter σ_W varies in the range $\{0.01745, 0.03491, 0.05236, 0.08727\}$ for fixed $\sigma_B = 0.2$ and prior distribution according to case (b) in Fig. 3. The σ_W parameter values correspond to the choice of the standard deviation of $10^\circ, 20^\circ, 30^\circ$ and 50° in the discrete-time model. Fig. 5 (b) depicts the time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$. As expected, the error deteriorates as the observation noise increases. The particle filters not only continue to exhibit better performance



(a) (b)

Fig. 5: Time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$ of filters as a function of the process noise std. dev. parameter σ_B (part (a)) and the measurement noise std. dev. parameter σ_W (part (b)). While the error deteriorates as the noise increases, the particle-based filters consistently exhibit better performance compared to EKF and the unscented Kalman filter.



(a) (b)

Fig. 6: Comparison of particle-based filter as the number of particles N is varied. (a): Time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$. The BPF exhibits poor performance for very small value of N. (b): Mean computational time of a single propagation-update step. The computational cost scale linearly with N except FPF-K which scales quadratically.

but also the performance deterioration is more graceful for larger values of σ_W.

(iv) **The time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$ as a function of N:** In this simulation, $N \in \{20, 50, 100, 200\}$ in the particle filters, for a fixed $\sigma_B = 0.2$, $\sigma_W = 0.05236$, and prior distribution according to case (b) in Fig. 3. Fig. 6 (a) depicts the time-averaged error $\langle \widehat{\delta\alpha} \rangle_T$. For all the particle-based filters, $N = 50$ particles is seen to be sufficient. For fewer than 50 particles, the BPF exhibits severe performance deterioration due to large approximation error in the importance sampling. Other filters exhibit slight performance deterioration as insufficient number of particles leads to larger error in the gain computation.

(v) **Computational times as a function of N:** In this simulation, $N \in \{20, 50, 100, 200, 500\}$. The mean computational time (per propagation-update step of the algorithm, averaged over 500 propagation-update steps) is depicted as a function of N in Fig. 6 (b). The $O(N)$ and $O(N^2)$ lines are included to aid the comparison. The computational cost of particle-based filters scale linearly with N except the kernel method which scales quadratically. For online computations, both FPF-G and FPF-C have lower computational burden compared with IEnKF and BPF. However, for the IEnKF algorithm, the gain computation – which contributes to most of the computation load – can be implemented offline [3]. The experiments were conducted on a workstation with an Intel i3-2120 3.3GHz CPU. The code was written in Python 2.7. The simulation time was measured using the `time` module in Python.

B. Filtering with a Bimodal Distribution

In this section, we consider the following static model in $SO(3)$:

$$\mathrm{d}q_t = \frac{1}{2}\, q_t \otimes \omega_t \, \mathrm{d}t$$

where $\omega_t = (0,0,0)$. The prior distribution is assumed to be supported on the subgroup $SO(2)$, parametrized by the angle $\theta \in [-\pi, \pi)$. An arbitrary element in $SO(2)$ is represented as $q = \big(\cos(\frac{\theta}{2}), 0, 0, \sin(\frac{\theta}{2})\big)$.

The observation model is of the following form:

$$\mathrm{d}Z_t = h(\theta_t)\, \mathrm{d}t + \sigma_W \, \mathrm{d}W_t$$

where $h(\theta) = \big(\cos(\theta), -\sin(\theta)\big)$, and W_t is the standard Wiener process in $\mathbb{R}^2$.

Since the process is static, the density of the posterior distribution has a closed-form Bayes' formula:

$$\rho^*(\theta, t) = (\text{const.})\, \exp\Big(\frac{1}{\sigma_W^2}\, h^\top(\theta) Z_t - \frac{1}{2\sigma_W^2}\, |h(\theta)|^2 t \Big)\, \rho_0^*(\theta) \tag{39}$$

where ρ_0^* is a given prior density.

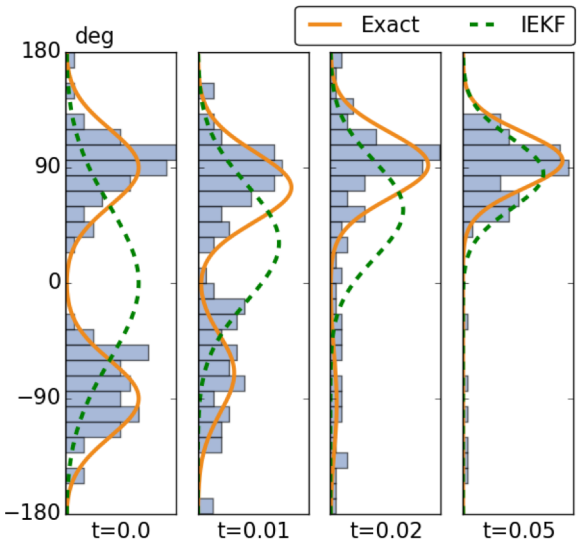


Fig. 7: Density evolution on $SO(2)$ with a bimodal prior distribution. Compared with the IEKF, the FPF is capable of capturing general non-Gaussian posterior distributions.

For the numerical results described next, the FPF is simulated according to (26) and (27):

$$\mathrm{d}q_t^i = \frac{1}{2} q_t^i \otimes \Big[\frac{1}{\sigma_W^2} l_t(q_t^i) \circ \big(\mathrm{d}Z_t - \frac{h(q_t^i) + \hat{h}_t}{2} \mathrm{d}t \big) \Big]$$

where q_0^i are sampled i.i.d. from the prior ρ_0^*.

The simulation parameters are as follows: The prior is a mixture of two Gaussians, $\mathcal{N}\big(-\mu_0, \sigma_0^2\big)$ and $\mathcal{N}\big(\mu_0, \sigma_0^2\big)$, with equal weights, where $\mu_0 = 90^\circ$ and $\sigma_0 = 30^\circ$. The observation noise parameter $\sigma_W = 0.12$, and the unknown state is initialized as $q_0 = (1/\sqrt{2}, 0, 0, 1/\sqrt{2})$, which corresponds to $\theta_0 = 90^\circ$. The simulations are carried out over $t \in [0, 0.2]$ with fixed time step $\Delta t = 0.01$. The FPF-K with $N = 100$ and $\varepsilon = 0.2$ is simulated together with the IEKF for a comparison.

Fig. 7 depicts the simulation results at selected time instants including the exact posterior (see (39)), the histogram of FPF particles, and the IEKF solution. This example shows that the FPF algorithm can handle a general class of non-Gaussian distributions.

VI. Conclusion

In this paper, the FPF algorithm is extended to the Riemannian manifolds and matrix Lie groups. The main result is that, expressed in its Stratonovich form, the gain times error update formula carries over to the manifold settings. To implement the filter requires a numerical approximation of the gain function for which a kernel algorithm is presented in this paper.

For the special case of the matrix Lie groups, explicit formulae for the gain and the FPF are derived, using the matrix coordinates. These formulae are applied to obtain filters for two example problems: (i) the attitude estimation problem in $SO(3)$ and (ii) the robot localization problem in $SE(3)$. For the $SO(3)$ problem, filter formula with respect to the computationally efficient quaternion coordinates is also given.

In order to better relate the filter to the EKF algorithm, certain approximations for the gain function and the filter are described under the additional assumption that the posterior is a concentrated distribution. In this case, it is shown that (i) the gain is approximated by the Kalman gain and (ii) the mean and the variance evolve according to formulae that are closely related to the left invariant EKF.

Detailed numerical comparisons between the FPF and other popular filtering algorithms are presented. These studies show that the particle filter is able to handle multimodal distributions as well as large uncertainty in noise and prior. In contrast to the importance sampling-based bootstrap particle filter, the FPF was found to require fewer particles.

There are a number of future research directions: (i) development of computationally efficient approximation schemes based on the kernel algorithm; (ii) error analysis of the kernel algorithm and the finite-N filter; (iii) investigation of other types of gain function approximation algorithms including an intrinsic version of the kernel algorithm; and (iv) extension of the FPF to handle manifold-valued observations. Development of computationally efficient approaches for approximation of the gain function are particularly important to be able to apply these algorithms in real-time settings.

Appendix A
Proof of Proposition 1

For any function $f \in C_c^\infty(G)$, $f(X_t^i)$ is a continuous semi-martingale that satisfies [8]

$$\begin{aligned}\mathrm{d}f(X_t^i) = {} & (V_0 + \mathsf{u}_t) \cdot f(X_t^i)\,\mathrm{d}t + V_1 \cdot f(X_t^i) \circ \mathrm{d}B_t^i \\ & + \mathsf{K}_t \cdot f(X_t^i) \circ \mathrm{d}Z_t \end{aligned} \tag{40}$$

For the ease of taking the expectation, we convert (40) to its Itô form [72, Theorem 1.2]: For real-valued continuous semi-martingales A, B, C

$$A \circ \mathrm{d}B = A\,\mathrm{d}B + \frac{1}{2}\mathrm{d}A\,\mathrm{d}B \tag{41}$$

$$(A \circ \mathrm{d}B)\,\mathrm{d}C = A(\mathrm{d}B\,\mathrm{d}C) \tag{42}$$

For the second term on the right hand side of (40), taking A in (41) to be $V_1 \cdot f(X_t^i)$ and B to be B_t^i

$$V_1 \cdot f(X_t^i) \circ \mathrm{d}B_t^i = V_1 \cdot f(X_t^i)\,\mathrm{d}B_t^i + \frac{1}{2}\mathrm{d}(V_1 \cdot f)(X_t^i)\,\mathrm{d}B_t^i \tag{43}$$

Replacing f by $V_1 \cdot f$ in (40)

$$\begin{aligned}\mathrm{d}(V_1 \cdot f) = {} & (V_0 + \mathsf{u}) \cdot (V_1 \cdot f)\,\mathrm{d}t + V_1 \cdot (V_1 \cdot f) \circ \mathrm{d}B_t^i \\ & + \mathsf{K}_t \cdot (V_1 \cdot f) \circ \mathrm{d}Z_t \end{aligned}$$

Using (42) and Itô's rule ($\mathrm{d}B_t^i\,\mathrm{d}B_t^i = \mathrm{d}t$, $\mathrm{d}B_t^i\,\mathrm{d}t = 0$, and $\mathrm{d}B_t^i\,\mathrm{d}Z_t = 0$)

$$\mathrm{d}(V_1 \cdot f)(X_t^i)\,\mathrm{d}B_t^i = V_1 \cdot (V_1 \cdot f)(X_t^i)\,\mathrm{d}t$$

which when substituted in (43) yields

$$V_1 \cdot f(X_t^i) \circ \mathrm{d}B_t^i = V_1 \cdot f(X_t^i)\,\mathrm{d}B_t^i + \frac{1}{2} V_1 \cdot (V_1 \cdot f)(X_t^i)\,\mathrm{d}t$$

The third term on the right hand side of (40) is similarly converted. The Itô form of (40) is then given by

$$\begin{aligned}\mathrm{d}f(X_t^i) = {} & \mathscr{L} f(X_t^i)\,\mathrm{d}t + \mathsf{u}_t \cdot f(X_t^i)\,\mathrm{d}t + V_1 \cdot f(X_t^i)\,\mathrm{d}B_t^i \\ & + \mathsf{K}_t \cdot f(X_t^i)\,\mathrm{d}Z_t + \frac{1}{2}\mathsf{K}_t \cdot (\mathsf{K}_t \cdot f)(X_t^i)\,\mathrm{d}t \end{aligned}$$

In its integral form

$$f(X_t^i) = \int_0^t \mathscr{L} f(X_s^i)\,\mathrm{d}s + \int_0^t \mathsf{u}_s \cdot f(X_s^i)\,\mathrm{d}s + \int_0^t V_1 \cdot f(X_s^i)\,\mathrm{d}B_s^i + \int_0^t \mathsf{K}_s \cdot f(X_s^i)\,\mathrm{d}Z_s + \frac{1}{2}\int_0^t \mathsf{K}_s \cdot (\mathsf{K}_s \cdot f)(X_s^i)\,\mathrm{d}s$$

The result is obtained by taking conditional expectation on both sides, interchanging expectation and integration (see Lemma 5.4 in [9]) and noting the fact that B_t^i is a Wiener process. □

Appendix B
Proof of Theorem 1

Using (2) and (5), it suffices to show the two identities

$$\pi_s(\mathsf{K}_s \cdot f) = \big(\pi_s(fh) - \pi_s(h)\,\pi_s(f)\big)$$

$$\pi_s(\mathsf{u}_s \cdot f) + \frac{1}{2}\pi_s\big(\mathsf{K}_s \cdot (\mathsf{K}_s \cdot f)\big) = -\big(\pi_s(fh) - \pi_s(h)\,\pi_s(f)\big)\pi_s(h)$$

for all $0 \le s \le t$ and all $f \in C_c^\infty(G)$.

The first identity is obtained by taking $\psi = f$ in (6). The second identity is obtained as follows: Using the expression (8) for the control function and noting that $\hat{h}_s = \pi_s(h)$

$$\begin{aligned}\pi_s(\mathsf{u}_s \cdot f) &= -\frac{1}{2}\pi_s\big(\big(h - \pi_s(h)\big)\,\mathsf{K}_s \cdot f\big) - \pi_s(h)\,\pi_s\big(\mathsf{K}_s \cdot f\big)\\ &= -\frac{1}{2}\pi_s\big(\mathsf{K}_s \cdot (\mathsf{K}_s \cdot f)\big) - \pi_s(h)\,\pi_s\big(\big(h - \pi_s(h)\big) f\big)\end{aligned}$$

where the second equality follows by using (6) with $\psi = \mathsf{K}_s \cdot f$. □

Appendix C
Justification of the Kernel Gain Approximation

A justification of the fixed-point problem (14) is as follows: The solution ϕ of the Poisson equation (6) is equivalently expressed as, for any fixed $\varepsilon > 0$:

$$\phi = e^{\varepsilon \Delta_\rho}\phi + \int_0^\varepsilon e^{s\Delta_\rho}(h - \hat{h})\,\mathrm{d}s \tag{44}$$

where $e^{\varepsilon\Delta_\rho}$ denotes the semigroup of the weighted Laplacian Δ_ρ. In the harmonic analysis literature, it is shown that in the limit as $\varepsilon \downarrow 0$ and $N \to \infty$, $\mathsf{T}^{(\varepsilon,N)}$ (defined in (13)) is a finite-dimensional approximation of the semigroup $e^{\varepsilon\Delta_\rho}$ (see [73, Proposition 3] and [58, Sec. 3.3]). The fixed-point equation (14) is a finite-dimensional approximation of (44) obtained by approximating $e^{\varepsilon\Delta_\rho}$ by $\mathsf{T}^{(\varepsilon,N)}$ for small ε. The formula (16) for the gain is obtained by evaluating the gradient of the two sides of (14).

Additional details on the kernel algorithm including the error bounds in the Euclidean setting appear in [59]. The extension of these bounds to the general Riemannian manifolds is a subject of ongoing research.

Appendix D
Proof of Proposition 2

Since the time t is fixed, the subscript t is dropped. Consider the parameterization $R = \mu\exp(\varepsilon[\chi]_\times)$ where $\chi \in \mathbb{R}^3$ and the parameter ε is small. For a test function $\psi \in C^\infty(SO(3))$

$$(RE_n)\cdot\psi(R) = \frac{1}{\varepsilon}\frac{\partial\psi}{\partial\chi_n}\big(\mu\exp(\varepsilon[\chi]_\times)\big) + O(\varepsilon) \tag{45}$$

This is because

$$\begin{aligned}(RE_n)\cdot\psi(R) &= \frac{\mathrm{d}}{\mathrm{d}t}\psi\big(R\exp(tE_n)\big)\Big|_{t=0}\\ &= \frac{\mathrm{d}}{\mathrm{d}t}\psi\big(\mu\exp(\varepsilon[\chi]_\times)\exp(t[e_n]_\times)\big)\Big|_{t=0}\\ &= \frac{\mathrm{d}}{\mathrm{d}t}\psi\big(\mu\exp([\varepsilon\chi + te_n + O(t^2) + tO(\varepsilon)]_\times)\big)\Big|_{t=0}\\ &= \frac{\mathrm{d}}{\mathrm{d}t}\psi\big(\mu\exp([\varepsilon\chi + te_n]_\times)\big)\Big|_{t=0} + O(\varepsilon)\end{aligned}$$

where the Baker–Campbell–Hausdorff formula [20, Chapter 3] is used to obtain the third equality.

Consider the Poisson equation (6) on $SO(3)$ with $\mathsf{K}(R) = R[l(R)]_\times = \sum_{n=1}^3 (l(R))_n RE_n$ and test function $\psi = \psi_\beta(R) = \psi_\beta(\mu\exp(\varepsilon[\chi]_\times)) = \varepsilon e_\beta^\top\chi$ for $\beta = 1,2,3$. Then

$$\sum_{n=1}^3 \pi\big((l(R))_n\,(RE_n)\cdot\psi_\beta\big) = \pi\big((h-\hat{h})\psi_\beta\big) \tag{46}$$

Using the formula (45), $(RE_n)\cdot\psi_\beta(R) = e_\beta^\top e_n + O(\varepsilon)$. Substituting this expression into (46) yields

$$\pi\big((l(R))_\beta\big) = \pi\big((h-\hat{h})\psi_\beta\big) + \text{h.o.t.}$$

for $\beta = 1,2,3$. Therefore

$$\pi(l) = \varepsilon\pi\big((h-\hat{h})\chi\big) + \text{h.o.t.}$$

which is the formula (30).

Proof of the constant gain formula (31): The expression for right-hand side of (30) is further simplified as

$$\pi\big((h-\hat{h})\psi_\beta\big) \;= \varepsilon e_\beta^\top\,\pi\big((h-\hat{h})\chi\big) = \varepsilon e_\beta^\top\,\pi\big(h\chi\big) \tag{47}$$

where $\pi(\hat{h}\chi) = 0$ is used since $\pi(\chi) = 0$ by definition and $\hat{h}$ is a constant. Substitute h with h^j where h^j is the j-th column of $h(R) = (-R^\top r^g, R^\top r^b) \in \mathbb{R}^6$ (see (22)-(24)). For $j = 1,2,3$, the Taylor expansion of h^j is given by

$$\begin{aligned}h^j(R) = h^j\big(\mu\exp(\varepsilon[\chi]_\times)\big) &= -e_j^\top\exp(-\varepsilon[\chi]_\times)\,\mu^\top r^g\\ &= -e_j^\top\mu^\top r^g + \varepsilon e_j^\top[\chi]_\times\mu^\top r^g + O(\varepsilon^2)\\ &= -e_j^\top\mu^\top r^g - \varepsilon\chi^\top[\mu^\top r^g]_\times e_j + O(\varepsilon^2)\end{aligned}$$

Therefore, $\pi(h^j\chi) = -\varepsilon\overline{\Sigma}[\mu^\top r^g]_\times e_j + O(\varepsilon^2)$, where we used the definition $\overline{\Sigma} = \pi(\chi\chi^\top)$. By similar calculations, $\pi(h^j\chi) = \varepsilon\overline{\Sigma}[\mu^\top r^b]_\times e_{j-3} + O(\varepsilon^2)$ for $j = 4,5,6$. Substituting these results into (47) we obtain (31). □

APPENDIX E
PROOF OF THEOREM 3

The $SO(3)$ FPF with the constant gain approximation (32) is expressed as

$$\mathrm{d}R_t^i = R_t^i\,[\,\omega_t\,]_\times\,\mathrm{d}t + R_t^i \circ [\,\mathrm{d}B_t^i\,]_\times + R_t^i\,[\,\bar{l}_t \bar{Q}_W^{-1} \circ \mathrm{d}\mathrm{I}_t^i\,]_\times \tag{48}$$

where $\bar{l}_t := \bar{\Sigma}_t \mathsf{H}_t$ is a 3×6 matrix and $\mathrm{d}\mathrm{I}_t^i = \mathrm{d}Z_t - \frac{1}{2}(h(R_t^i) + \hat{h}_t)\,\mathrm{d}t$ is a 6×1 vector.

For small parameter ε, using the concentrated distribution ansatz (see Definition 2)

$$R_t^i = \mu_t \exp(\varepsilon\,[\,\chi_t^i\,]_\times) = \mu_t + \varepsilon\,\mu_t\,[\,\chi_t^i\,]_\times + O(\varepsilon^2) \tag{49}$$

$$\mathrm{d}\mathrm{I}_t^i = \underbrace{\mathrm{d}Z_t - h(\mu_t)\,\mathrm{d}t}_{\mathrm{d}\mathrm{I}_t} - \frac{1}{2}\,\varepsilon\,\mathsf{H}_t^\top \chi_t^i\,\mathrm{d}t + O(\varepsilon^2) \tag{50}$$

The evolution equation for the mean μ_t and covariance $\bar{\Sigma}_t$ are derived using a regular perturbation approach: On substituting (49) and (50) into the FPF (48) and matching terms, the $O(1)$ balance gives

$$\mathrm{d}\mu_t = \mu_t\,[\,\omega_t\,]_\times\,\mathrm{d}t + \mu_t[\,\bar{l}_t \bar{Q}_W^{-1} \circ \mathrm{d}\mathrm{I}_t\,]_\times \tag{51}$$

The $O(\varepsilon)$ balance gives

$$\begin{aligned}\mathrm{d}(\mu_t\,[\,\chi_t^i\,]_\times) = {}& \mu_t\,[\,\chi_t^i\,]_\times\,[\,\omega_t\,]_\times\,\mathrm{d}t + \frac{1}{\varepsilon}\,\mu_t\,[\,\mathrm{d}B_t^i\,]_\times \\ & + \mu_t\,[\,\chi_t^i\,]_\times[\,\bar{l}_t \bar{Q}_W^{-1} \circ \mathrm{d}\mathrm{I}_t\,]_\times - \frac{1}{2}\,\mu_t\,[\,\bar{l}_t \bar{Q}_W^{-1}\,\mathsf{H}_t^\top\,\chi_t^i\,]_\times\,\mathrm{d}t\end{aligned}$$

Using the formula (51), this is simplified to obtain the following equation of χ_t^i, expressed in its Itô form as

$$\begin{aligned}\mathrm{d}\chi_t^i = {}& -[\,\omega_t\,]_\times \chi_t^i\,\mathrm{d}t + \frac{1}{\varepsilon}\,\mathrm{d}B_t^i - \frac{1}{2}\,\bar{l}_t\,\bar{Q}_W^{-1}\,\mathsf{H}_t^\top\,\chi_t^i\,\mathrm{d}t \\ & - [\,\bar{l}_t \bar{Q}_W^{-1}\,\mathrm{d}\mathrm{I}_t\,]_\times\,\chi_t^i + O(\varepsilon^2)\end{aligned}$$

Define $\Gamma_t^i := \chi_t^i\,\chi_t^{i\top}$. Using the Itô's lemma

$$\begin{aligned}\mathrm{d}\mathrm{I}_t^i = {}& \mathrm{d}\chi_t^i\,\chi_t^{i\top} + \chi_t^i\,\mathrm{d}(\chi_t^{i\top}) + \mathrm{d}\chi_t^i\,\mathrm{d}(\chi_t^{i\top}) \\ = {}& -([\,\omega_t\,]_\times\,\mathrm{d}t)\Gamma_t^i - \Gamma_t^i([\,\omega_t\,]_\times\,\mathrm{d}t)^\top + \bar{Q}_B\,\mathrm{d}t \\ & - [\,\bar{l}_t \bar{Q}_W^{-1}\,\mathrm{d}\mathrm{I}_t\,]_\times \Gamma_t^i - \Gamma_t^i[\,\bar{l}_t \bar{Q}_W^{-1}\,\mathrm{d}\mathrm{I}_t\,]_\times^\top \\ & + \frac{1}{\varepsilon}\left(\mathrm{d}B_t^i\,\chi_t^{i\top} + \chi_t^i\,\mathrm{d}B_t^{i\top}\right) \\ & - \frac{1}{2}\left(\bar{l}_t\,\bar{Q}_W^{-1}\,\mathsf{H}_t^\top\,\Gamma_t^i + \Gamma_t^i\,\mathsf{H}_t\,\bar{Q}_W^{-1}\,\bar{l}_t^\top\right)\mathrm{d}t + O(\varepsilon^2)\end{aligned}$$

By definition, $\bar{\Sigma}_t = \pi_t(\Gamma_t^i)$. Taking the conditional expectation on both sides and using the formula of K_t

$$\begin{aligned}\mathrm{d}\bar{\Sigma}_t = {}& -([\,\omega_t\,]_\times\,\mathrm{d}t)\bar{\Sigma}_t - \bar{\Sigma}_t([\,\omega_t\,]_\times\,\mathrm{d}t)^\top + \bar{Q}_B\,\mathrm{d}t \\ & - [\,\bar{l}_t \bar{Q}_W^{-1}\,\mathrm{d}\mathrm{I}_t\,]_\times \bar{\Sigma}_t - \bar{\Sigma}_t[\,\bar{l}_t \bar{Q}_W^{-1}\,\mathrm{d}\mathrm{I}_t\,]_\times^\top - \bar{\Sigma}_t\mathsf{H}_t\bar{Q}_W^{-1}\mathsf{H}_t^\top\bar{\Sigma}_t\,\mathrm{d}t\end{aligned}$$

where $O(\varepsilon^2)$ terms have been ignored. □

REFERENCES

[1] A. Srivastava and E. Klassen, "Bayesian and geometric subspace tracking," *Adv. Appl. Prob.*, vol. 36, no. 1, pp. 43–56, Mar. 2004.

[2] M. Hua, G. Ducard, T. Hamel, R. Mahony, and K. Rudin, "Implementation of a nonlinear attitude estimator for aerial robotic vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 1, pp. 201–213, Jan. 2014.

[3] A. Barrau and S. Bonnabel, "Intrinsic filtering on Lie groups with applications to attitude estimation," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 436–449, Feb. 2015.

[4] M. Barczyk, S. Bonnabel, J. Deschaud, and F. Goulette, "Invariant EKF design for scan matching-aided localization," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2440–2448, 2015.

[5] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency improvement," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 182–201, Jan. 2014.

[6] J. Kwon, H. S. Lee, F. C. Park, and K. M. Lee, "A geometric particle filter for template-based visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 625–643, Apr. 2014.

[7] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the special Euclidean group: A combined approach of keypoint and edge features," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 498–519, Apr. 2012.

[8] E. P. Hsu, *Stochastic Analysis on Manifolds*. Providence, RI, USA: Amer. Math. Soc., 2002.

[9] J. Xiong, *An Introduction to Stochastic Filtering Theory*. Oxford, UK: Oxford Univ. Press, 2008.

[10] T. E. Duncan, "Some filtering results in Riemann manifolds," *Inform. Control*, vol. 35, no. 3, pp. 182–195, Nov. 1977.

[11] S. K. Ng and P. E. Caines, "Nonlinear filtering in Riemannian manifolds," *IMA J. Math. Control Inform.*, vol. 2, no. 1, pp. 25–36, Jan. 1985.

[12] V. Solo, "On nonlinear state estimation in a Riemannian manifold," in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, 2009, pp. 8500–8505.

[13] F. Tompkins and P. J. Wolfe, "Bayesian filtering on the Stiefel manifold," in *2nd IEEE CAMPSAP*, St. Thomas, VI, USA, 2007, pp. 261–264.

[14] H. Snoussi, "Particle filtering on Riemannian manifolds. Application to covariance matrices tracking," in *Matrix Information Geometry*. Springer, 2013, pp. 427–449.

[15] S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented Kalman filtering on Riemannian manifolds," *J. Math. Imaging Vis.*, vol. 46, no. 1, pp. 103–120, May 2013.

[16] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, "Discrete extended Kalman filter on Lie groups," in *Proc. 21st Eur. Signal Process. Conf.*, Marrakech, Morocco, 2013, pp. 1–5.

[17] R. Zanetti, M. Majji, R. H. Bishop, and D. Mortari, "Norm-constrained Kalman filtering," *J. Guid. Control Dynam.*, vol. 32, no. 5, pp. 1458–1465, Sept.-Oct. 2009.

[18] S. Bonnabel, P. Martin, and E. Salaün, "Invariant extended Kalman filter: theory and application to a velocity-aided attitude estimation problem," in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, 2009, pp. 1297–1304.

[19] J. R. Forbes, A. H. de Ruiter, and D. E. Zlotnik, "Continuous-time norm-constrained Kalman filtering," *Automatica*, vol. 50, no. 10, pp. 2546–2554, Oct. 2014.

[20] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus, "Continuous-discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions," *J. Math. Imaging Vis.*, vol. 51, no. 1, pp. 209–228, Jan. 2015.

[21] I. Y. Bar-Itzhack and Y. Oshman, "Attitude determination from vector observations: Quaternion estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, no. 1, pp. 128–136, Jan. 1985.

[22] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid. Control Dynam.*, vol. 5, no. 5, pp. 417–429, Sept.-Oct. 1982.

[23] I. Y. Bar-Itzhack and M. Idan, "Recursive attitude determination from vector observations: Euler angle estimation," *J. Guid. Control Dynam.*, vol. 10, no. 2, pp. 152–157, Mar.-Apr. 1987.

[24] M. E. Pittelkau, "Rotation vector in attitude estimation," *J. Guid. Control Dynam.*, vol. 26, no. 6, pp. 855–860, Nov.-Dec. 2003.

[25] F. L. Markley, "Attitude error representations for Kalman filtering," *J. Guid. Control Dynam.*, vol. 26, no. 2, pp. 311–317, Mar.-Apr. 2003.

[26] A. Chiuso and S. Soatto, "Monte Carlo filtering on Lie groups," in *Proc. 39th IEEE Conf. Decision Control*, Sydney, AU, 2000, pp. 304–309.

[27] J. Kwon, M. Choi, F. C. Park, and C. Chun, "Particle filtering on the Euclidean group: framework and applications," *Robotica*, vol. 25, no. 06, pp. 725–737, Nov. 2007.

[28] G. Marjanovic and V. Solo, "An engineer's guide to particle filtering on matrix Lie groups," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Shanghai, China, 2016, pp. 3969–3973.

[29] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *J. Guid. Control Dynam.*, vol. 26, no. 4, pp. 536–542, July-Aug. 2003.

[30] Y. Cheng and J. L. Crassidis, "Particle filtering for attitude estimation using a minimal local-error representation," *J. Guid. Control Dynam.*, vol. 33, no. 4, pp. 1305–1310, July-Aug. 2010.

[31] Y. Oshman and A. Carmi, "Attitude estimation from vector observations using a genetic-algorithm-embedded quaternion particle filter," *J. Guid. Control Dynam.*, vol. 29, no. 4, pp. 879–891, July-Aug. 2006.

[32] M. Zamani, J. Trumpf, and R. Mahony, "Minimum-energy filtering for attitude estimation," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2917–2921, Nov. 2013.

[33] M. Izadi and A. K. Sanyal, "Rigid body attitude estimation based on the Lagrange-d'Alembert principle," *Automatica*, vol. 50, no. 10, pp. 2570–2577, Oct. 2014.

[34] R. Mahony, T. Hamel, and J. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1203–1218, May 2008.

[35] C. Lageman, J. Trumpf, and R. Mahony, "Gradient-like observers for invariant dynamics on a Lie group," *IEEE Trans. Autom. Control*, vol. 55, no. 2, pp. 367–377, Feb. 2010.

[36] J. F. Vasconcelos, R. Cunha, C. Silvestre, and P. Oliveira, "A nonlinear position and attitude observer on SE(3) using landmark measurements," *Syst. Control Lett.*, vol. 59, no. 3-4, pp. 155–166, Mar.-Apr. 2010.

[37] S. Bonnabel, P. Martin, and P. Rouchon, "Non-linear symmetry-preserving observers on Lie groups," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1709–1713, July 2009.

[38] J. P. Condomines, C. Seren, and G. Hattenberger, "Nonlinear state estimation using an invariant unscented Kalman filter," in *AIAA Guid. Nav. Control Conf.*, Boston, MA, USA, 2013, pp. 1–15.

[39] A. Barrau and S. Bonnabel, "Invariant particle filtering with application to localization," in *Proc. 53rd IEEE Conf. Decision Control*, Los Angeles, CA, USA, 2014, pp. 5599–5605.

[40] G. S. Chirikjian and A. B. Kyatkin, *Harmonic Analysis for Engineers and Applied Scientists: Updated and Expanded Edition*. Mineola, NY, USA: Dover Publications, 2016.

[41] Y. Wang and G. S. Chirikjian, "Error propagation on the Euclidean group with applications to manipulator kinematics," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 591–602, Aug. 2006.

[42] K. Wolfe, M. Mashner, and G. S. Chirikjian, "Bayesian fusion on Lie groups," *J. Algebr. Stat.*, vol. 2, no. 1, pp. 75–97, Apr. 2011.

[43] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *J. Guid. Control Dynam.*, vol. 30, no. 1, pp. 12–28, Jan.-Feb. 2007.

[44] M. Zamani, "Deterministic attitude and pose filtering, an embedded Lie groups approach," Ph.D. dissertation, Austr. Nat. Univ., Canberra, AU, May 2013.

[45] T. Yang, "Feedback particle filter and its applications," Ph.D. dissertation, Univ. of Illinois Urbana-Champaign, Urbana, IL, USA, Sept. 2014.

[46] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn, "Multivariable feedback particle filter," *Automatica*, vol. 71, no. 9, pp. 10–23, Sept. 2016.

[47] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback particle filter," *IEEE Trans. Autom. Control*, vol. 58, no. 10, pp. 2465–2480, Oct. 2013.

[48] C. Zhang, A. Taghvaei, and P. G. Mehta, "Feedback particle filter on matrix Lie groups," in *Proc. Amer. Control Conf.*, Boston, MA, USA, 2016, pp. 2723–2728.

[49] ——, "Attitude estimation with feedback particle filter," in *IEEE 55th Conf. Decision Control*, Las Vegas, NV, USA, 2016, pp. 5440–5445.

[50] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, 6th ed. Springer, 2007.

[51] A. Grigor'yan, *Heat Kernel and Analysis on Manifolds*. Providence, RI, USA: Amer. Math. Soc., 2009.

[52] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed. Springer, 2012.

[53] M. P. do Carmo, *Riemannian geometry*. Birkhäuser Basel, 1992.

[54] D. Bakry, I. Gentil, and M. Ledoux, *Analysis and geometry of Markov diffusion operators*. Springer, 2013.

[55] Y. Matsuura, R. Ohata, K. Nakakuki, and R. Hirokawa, "Suboptimal gain functions of feedback particle filter derived from continuation method," in *AIAA Guid. Nav. Control Conf.*, San Diego, CA, USA, 2016, p. 1620.

[56] A. Radhakrishnan, A. M. Devraj, and S. P. Meyn, "Learning techniques for feedback particle filter design," in *Proc. 55th IEEE Conf. Decision Control*, Las Vegas, NV, USA, 2016, pp. 5452–5459.

[57] K. Berntorp and P. Grover, "Data-driven gain computation in the feedback particle filter," in *Proc. Amer. Control Conf.*, Boston, MA, USA, 2016, pp. 2711–2716.

[58] M. Hein, J.-Y. Audibert, and U. von Luxburg, "Graph Laplacians and their convergence on random neighborhood graphs," *J. Mach. Learn. Res.*, vol. 8, pp. 1325–1368, 2006.

[59] A. Taghvaei, P. G. Mehta, and S. P. Meyn, "Error estimates for the kernel gain function approximation in the feedback particle filter," in *Proc. Amer. Control Conf.*, Seattle, WA, USA, 2017, pp. 4576–4582.

[60] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer, 2015.

[61] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Birkhäuser Basel, 2011.

[62] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," Univ. of Minnesota, Dept. of Comp. Sci. and Eng., Minneapolis, MN, USA, Tech. Rep. 2005-002, 2005.

[63] M. J. Piggott and V. Solo, "Geometric Euler-Maruyama schemes for stochastic differential equations in SO(n) and SE(n)," *SIAM J. Numer. Anal.*, vol. 54, no. 4, pp. 2490–2516, Aug. 2016.

[64] J. H. Manton, "A primer on stochastic differential geometry for signal processing," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 4, pp. 681–699, Aug. 2013.

[65] G. Marjanovic, M. J. Piggott, and V. Solo, "A simple approach to numerical methods for stochastic differential equations in Lie groups," in *Proc. 54th IEEE Conf. Decision Control*, Osaka, Japan, 2015, pp. 7143–7150.

[66] ——, "Numerical methods for stochastic differential equations in the Stiefel manifold made simple," in *Proc. 55th IEEE Conf. Decision Control*, Las Vegas, NV, USA, 2016, pp. 2853–2860.

[67] M. Hua, T. Hamel, R. Mahony, and J. Trumpf, "Gradient-like observer design on the special Euclidean group SE(3) with system outputs on the real projective space," in *54th IEEE Conf. Decision Control*, Osaka, Japan, 2015, pp. 2139–2145.

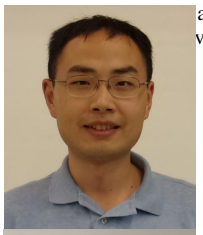
[68] K. Berntorp, "Feedback particle filter: Application and evaluation," in *Proc. 18th Int. Conf. Inform. Fusion*, Washington DC, USA, 2015, pp. 1633–1640.

[69] P. M. Stano, A. K. Tilton, and R. Babuška, "Estimation of the soil-dependent time-varying parameters of the hopper sedimentation model: The FPF versus the BPF," *Control Eng. Pract.*, vol. 24, pp. 67–78, Mar. 2014.

[70] G. S. Chirikjian and M. Kobilarov, "Gaussian approximation of non-linear measurement models on Lie groups," in *Proc. 53rd IEEE Conf. Decision Control*, Los Angeles, CA, USA, 2014, pp. 6401–6406.

[71] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *J. Guid. Control Dynam.*, vol. 30, no. 4, pp. 1193–1197, Aug. 2007.

[72] S. Watanabe and N. Ikeda, *Stochastic Differential Equations and Diffusion Processes*. North Holland, 1981.



Chi Zhang received the B.Eng. degree in Automotive Engineering from Tsinghua University, Beijing, China, in 2011, the M.S. degree in Mathematics from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2014, and the Ph.D. degree in Me- chanical Engineering from UIUC in 2017. He is currently a Data Scientist at Experian DataLabs, San Diego, CA. His research interests include particle-based nonlinear filtering and stochastic optimization, and their extension to Riemannian manifolds.

Amirhossein Taghvaei graduated from Sharif University of Technology, Tehran, Iran, in 2013, receiving two B.S degrees; in Mechanical Engineering and in Physics. He is currently a PhD student in the Department of Mechanical Science and Engineering at the University of Illinois Urbana-Champaign. He is working in the area of Decision and Control in the Coordinated Science Laboratory. His research interests is the control and filtering algorithms

Prashant G. Mehta (M'06) received the Ph.D. degree in Applied Mathematics from Cornell University, Ithaca, NY, in 2004. He is an Associate Professor in the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign. Prior to joining Illinois, he was a Research Engineer at the United Technologies Research Center (UTRC). His current research interests are in nonlinear filtering and mean-field games. He received the Outstanding Achievement Award at UTRC for his contributions to the modeling and control of combustion instabilities in jet-engines. His students received the Best Student Paper Awards at the IEEE Conference on Decision and Control 2007 and 2009 and were finalists for these awards in 2010 and 2012. He has served on the editorial boards of the *ASME Journal of Dynamic Systems, Measurement, and Control* and the *Systems and Control Letters*.