CONTROLLED PARTICLE SYSTEMS FOR NONLINEAR FILTERING AND GLOBAL OPTIMIZATION

BY

CHI ZHANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Associate Professor Prashant G. Mehta, Chair and Director of Research
Professor Venugopal V. Veeravalli
Professor Harry Dankowicz
Associate Professor Matthew West
Assistant Professor Mohamed-Ali Belabbas

Abstract

This thesis is concerned with the development and applications of controlled interacting particle systems for nonlinear filtering and global optimization problems. These problems are important in a number of engineering domains. In nonlinear filtering, there is a growing interest to develop geometric approaches for systems that evolve on matrix Lie groups. Examples include the problem of attitude estimation and motion tracking in aerospace engineering, robotics and computer vision. In global optimization, the challenges typically arise from the presence of a large number of local minimizers as well as the computational scalability of the solution. Gradient-free algorithms are attractive because in many practical situations, evaluating the gradient of the objective function may be computationally prohibitive.

The thesis comprises two parts that are devoted to theory and applications, respectively. The theoretical part consists of three chapters that describe methods and algorithms for nonlinear filtering, global optimization, and numerical solutions of the Poisson equation that arise in both filtering and optimization.

For the nonlinear filtering problem, the main contribution is to extend the feedback particle filter (FPF) algorithm to connected matrix Lie groups. In its general form, the FPF is shown to provide an intrinsic coordinate-free description of the filter that automatically satisfies the manifold constraint. The properties of the original (Euclidean) FPF, especially the gain-times-error feedback structure, are preserved in the generalization.

For the global optimization problem, a controlled particle filter algorithm is introduced to numerically approximate a solution of the global optimization problem. The theoretical significance of this work comes from its variational aspects: (i) the proposed particle filter is a controlled interacting particle system where the control input represents the solution of a mean-field type optimal control problem; and (ii) the associated density transport is shown to be a gradient flow (steepest descent) for the optimal value function, with respect to the Kullback–Leibler divergence.

For both the nonlinear filtering and optimization problems, the numerical implementation of the pro-

posed algorithms require a solution of a Poisson equation. Two numerical algorithms are described for this purpose. In the Galerkin scheme, the gain function is approximated using a set of pre-defined basis functions; In the kernel-based scheme, a numerical solution is obtained by solving a certain fixed-point equation. Well-posedness results for the Poisson equation are also discussed.

The second part of the thesis contains applications of the proposed algorithms to specific nonlinear filtering and optimization problems. The FPF is applied to the problem of attitude estimation – a nonlinear filtering problem on the Lie group $SO(3)$. The formulae of the filter are described using both the rotation matrix and the quaternion coordinates. A comparison is provided between FPF and the several popular attitude filters including the multiplicative EKF, the invariant EKF, the unscented Kalman filter, the invariant ensemble Kalman filter and the bootstrap particle filter. Numerical simulations are presented to illustrate the comparison.

As a practical application, experimental results for a motion tracking problem are presented. The objective is to estimate the attitude of a wrist-worn motion sensor based on the motion of the arm. In the presence of motion, considered here as the swinging motion of the arm, the observability of the sensor attitude is shown to improve. The estimation problem is mathematically formulated as a nonlinear filtering problem on the product Lie group $SO(3) \times SO(2)$, and experimental results are described using data from the gyroscope and the accelerometer installed on the sensor.

For the global optimization problem, the proposed controlled particle filter is compared with several model-based algorithms that also employ probabilistic models to inform the search of the global minimizer. Examples of the model-based algorithms include the model reference adaptive search, the cross entropy, the model-based evolutionary optimization, and two algorithms based on bootstrap particle filtering. Performance comparisons are provided between the control-based and the sampling-based implementation. Results of Monte-Carlo simulations are described for several benchmark optimization problems.

To my parents.

Acknowledgements

Above all I would like to express sincere gratitude to my advisor, Professor Prashant G. Mehta, for his consistent support, enthusiasm, encouragement and guidance through my doctoral study at the University of Illinois at Urbana-Champaign. I also feel honored to have Professor Venugopal V. Veeravalli, Professor Richard S. Laugesen, Professor Harry Dankowicz, Professor Matthew West, and Professor Mohamed-Ali Belabbas for serving on the committee for my preliminary and final examinations.

I would like to thank the Coordinated Science Laboratory (CSL) for providing me with the working space and research facilities. The CSL is an exceptional place with an interdisciplinary and dynamic research environment. The staff and students of CSL are friendly and inspiring. I especially wish to thank Angie Ellis for her help with logistic issues for conference travels. I am also grateful to the Department of Mechanical Science and Engineering for admitting me to the PhD program. In particular, I truly appreciate Kathy Smith for answering all my questions related to the graduate program. My research was funded by several institutions. I am deeply grateful to the National Science Foundation, and the Air Force Office of Scientific Research for their support of my entire doctoral program.

In addition, my friends have always been supportive and filled my life with joy. I would thank Amirhossein Taghvaei for co-authoring several of my research papers. I have enjoyed our collaboration on solving hard mathematical problems and writing the papers. I would like to thank Tao Yang for his valuable help especially during the early stage of my Ph.D. life. I would like to thank Rohan Arora for being a great lab mate who has helped me with research, course projects, and all sorts of daily trivia such as software installation and program debugging. I would like to thank Kun Deng, Yu Sun, Yunwen Xu, Pengzhi Wang for their useful suggestions on my research and career development. I would like to thank Adam Tilton, Shane Ghiotto, Jin-Won Kim, Mayank Baranwal and Sahand Hariri who have been cheerful lab mates. I am fortunate to have worked with all of you in Room 348 of CSL. I would also like to thank all the professors and teachers with whom I enriched my knowledge and widened my scope.

Finally, I have indescribable gratitude to my parents, Qingnian Zhang and Peihong Liu. Though separated by lands and oceans, they are forever an unconditional source of love, encouragement and wisdom especially when obstacles arise in my path. It is them that helped shape my personality and taught me life lessons over the past twenty-eight years. This thesis is thus a reward to their love and efforts.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis is concerned with particle-based methods and algorithms for the problem of nonlinear filtering and global optimization. Application of particle-based algorithms for these problems has a rich history. For the nonlinear filtering problem, the particle filter has been a popular solution approach since its debut in the mid-1990s [63, 3, 46]. It has been applied to estimation problems that arise in engineering disciplines such as inertial navigation [67], simultaneous localization and mapping [135], computer vision [84], wireless sensor networks [83], and fault diagnosis [182]. For the global optimization problem, particle-based algorithms include the genetic algorithm [62], the particle swarm optimization [91, 196], the ant colony optimization [45], a class of interacting particle systems [133, 136], and a class of model-based algorithms [75, 79].

Nonlinear filtering and global optimization are closely-related problems with similar models and particle-based implementations. For example, several recent works cast the global optimization problem as a particle filtering problem whereby a Bayesian model is used to sequentially guide the search of the optimal solution [215, 165]. Also, the methods for global optimization have influenced the development of particle filter algorithms, e.g., by incorporating the annealing property that is fundamental for many optimization methods [59, 136]. For both filtering and optimization, classical particle-based algorithms typically require certain types of particle interaction mechanisms such as selection, computation of the so-called importance weight, and re-generation of particles.

The aim of this thesis is to develop a class of *controlled particle system* algorithms for the nonlinear filtering and global optimization problems. In the control-based approach advocated in this thesis, each particle evolves according to a control law that depends on both the particle as well as the entire population. There are no additional steps associated with resampling, regeneration or removal of particles. The control-oriented construction is motivated by the feedback particle filter (FPF) [195, 194] that was originally developed for nonlinear filtering problems in an Euclidean setting. Additional motivation comes from the recent development in mean-field control [81, 197] and the optimal transportation theory [183, 31].

The contributions of this thesis are as follows: For the nonlinear filtering problem, the FPF algorithm – originally constructed in the Euclidean space – is extended to matrix Lie groups. Applications of FPF to attitude estimation and motion tracking are presented along with several numerical studies. For the global optimization problem, a controlled particle filter algorithm – a gradient-free approach closely related to FPF – is developed and numerically assessed for several benchmark optimization problems. For both the filtering and optimization problems, the proposed algorithms are compared with the state-of-the-art algorithms in the literature.

The remainder of this chapter provides an expanded summary of these contributions. Sec. 1.1 provides a self-contained summary of background on the basic FPF theory, the mathematical problem statement involving its extension to matrix Lie groups, a literature review, and an overview of our contributions to this problem. Sec. 1.2 provides a similar overview for the global optimization problem. The organization of the thesis appears in Sec. 1.3.

1.1 Nonlinear Filtering on Matrix Lie Groups

1.1.1 Background: Review of the basic feedback particle filter theory

The feedback particle filter was originally developed for the following continuous-time nonlinear filtering problem in the Euclidean space [194, 193]:

$$\mathrm{d}X_t = a(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}B_t, \tag{1.1a}$$

$$\mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \tag{1.1b}$$

where $X_t \in \mathbb{R}^d$ is the hidden state (signal), $Z_t \in \mathbb{R}^m$ is the observation, $a(\cdot)$, $h(\cdot)$, $\sigma(\cdot)$ are C^1 functions of appropriate dimension, and B_t, W_t are mutually independent standard Wiener processes in $\mathbb{R}^d$ and $\mathbb{R}^m$, respectively. The j-th coordinate of h is denoted as h_j, i.e., $h = (h_1, ..., h_m)$. The stochastic differential equation (sde) (1.1a) is expressed in its Itô form. The filtering objective is to numerically approximate the conditional distribution of X_t given the time-history (filtration) of the observations $\mathscr{Z}_t = \sigma(Z_s : s \leq t)$.

The Euclidean feedback particle filter consists of N controlled stochastic processes $\{X_t^i\}_{i=1}^N$. The state

of the i-th particle $X_t^i \in \mathbb{R}^d$ evolves according to,

$$\mathrm{d}X_t^i = a(X_t^i)\,\mathrm{d}t + \sigma(X_t^i)\,\mathrm{d}B_t^i + \mathsf{K}(X_t^i,t) \circ \Big(\mathrm{d}Z_t - \frac{h(X_t^i)+\hat{h}_t}{2}\,\mathrm{d}t\Big), \tag{1.2}$$

where B_t^i for $i = 1,...,N$ are mutually independent standard Wiener processes in $\mathbb{R}^d$, $\mathsf{K}(x,t) \in \mathbb{R}^{d\times m}$ is a *gain function* to be solved, and $\hat{h}_t := \mathsf{E}[h(X_t^i)|\mathscr{Z}_t]$. The $\circ$ after K indicates that the particle sde (1.2) is expressed in its Stratonovich form. We refer the reader to [60] for a composition of Itô and Stratonovich sdes.

The gain function $\mathsf{K}(x,t)$ is obtained by solving a Poisson equation: For $j = 1,...,m$, the function ϕ_j is a solution to the Poisson equation,

$$\begin{aligned} -\nabla\cdot(\rho(x)\nabla\phi_j(x)) &= (h_j(x)-\hat{h}_j)\rho(x), \quad x\in\mathbb{R}^d, \\ \int \phi_j(x)\rho(x)\,\mathrm{d}x &= 0, \end{aligned} \tag{1.3}$$

where ρ denotes the conditional density of X_t^i given the filtration $\mathscr{Z}_t$. The gain function is then given by,

$$[\mathsf{K}]_{lj} = \frac{\partial\phi_j}{\partial x_l}.$$

The gain function needs to be obtained for each value of time t.

The FPF is an *exact* algorithm. That is, the posterior distribution (conditioned on the filtration $\mathscr{Z}_t$) of X_t^i exactly matches the posterior distribution of X_t, provided that the two distributions are initialized according to the same prior [194].

Numerical studies and comparisons with FPF have been reported in several papers [194, 175, 17, 164, 205, 168] where FPF was applied to various nonlinear estimation problems. In many of these studies, the FPF exhibited certain advantages over the conventional bootstrap particle filter (BPF) [3]. For example, the FPF was shown to avoid the particle degeneracy and sample impoverishment issue encountered in BPF, and FPF also reduced the simulation variance. The robustness of FPF has been attributed to the error-correction feedback structure in the particle dynamics (1.2). Indeed, extending the feedback structure to more general spaces such as matrix Lie groups is a major goal of our work.

1.1.2 Problem statement: Filtering problem on matrix Lie groups

Throughout the thesis, a matrix Lie group is denoted as G, and its Lie algebra is denoted as $\mathscr{G}$. The focus of this thesis is on the following continuous-time filtering problem on G,

$$\mathrm{d}X_t = X_t V_0(X_t)\,\mathrm{d}t + X_t V_\alpha(X_t) \circ \mathrm{d}B_t^\alpha, \tag{1.4a}$$

$$\mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \tag{1.4b}$$

where $X_t \in G$ is the hidden state (signal) at time t, $Z_t \in \mathbb{R}^m$ is the observation vector; $V_0 : G \to \mathscr{G}$ and $V_\alpha : G \to \mathscr{G}$ for $\alpha = 1,...,r$ are Lie algebra-valued functions; B_t^α and W_t are mutually independent standard Wiener processes in $\mathbb{R}$ and $\mathbb{R}^m$, respectively, and they are also assumed to be independent of the initial state X_0; $h : G \to \mathbb{R}^m$ is a given vector-valued nonlinear function. The j-th coordinate of Z_t and h are denoted as Z_t^j and h_j, respectively (i.e. $Z_t = (Z_t^1,...,Z_t^m)$ and $h = (h_1,...,h_m)$). The sde (1.4a) is expressed in its Stratonovich form, and the Einstein summation convention for the free index α is used.

The signal model (1.4a) differs from its Euclidean counterpart (1.1a) due to the multiplicative form of the drift and noise terms. By left multiplication, the Lie algebra is used to generate a vector field that models state evolution on a Lie group [104]. This construction, expressed in its Stratonovich form, provides a coordinate-free description of the sde on a Lie group [74, 60, 140].

As in the Euclidean case, the objective of the filtering problem is to numerically approximate the conditional distribution of X_t given the time-history of observations $\mathscr{Z}_t = \sigma(Z_s : s \le t)$. The conditional distribution, denoted as π_t^*, acts on a function $f : G \to \mathbb{R}$ according to

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

1.1.3 Overview of contributions

The main contribution of the thesis is to generalize the feedback particle filter to matrix Lie groups. The FPF algorithm for the filtering problem (1.4a)-(1.4b) consists of N particles $\{X_t^i\}_{i=1}^N$ evolving on the Lie group G,

$$\mathrm{d}X_t^i = \underbrace{X_t^i V_0(X_t^i)\,\mathrm{d}t + X_t^i V_\alpha \circ \mathrm{d}B_t^{\alpha,i}}_{\text{propagation}} + \underbrace{X_t^i\, \mathsf{K}_j(X_t^i,t) \circ \mathrm{dI}_t^{j,i}}_{\text{observation update}}, \tag{1.5}$$

where $B_t^{\alpha,i}$ for $\alpha = 1,...,r$ and $i = 1,...,N$ are mutually independent standard Wiener processes in $\mathbb{R}$, $\mathsf{K}_j \in \mathscr{G}$ for $j = 1,...,m$ are the gain functions obtained by solving a Poisson equation on G, and the error $\mathrm{d}\mathrm{I}_t^{j,i} \in \mathbb{R}$ is a modified form of the innovation process:

$$\mathrm{d}\mathrm{I}_t^{j,i} = \mathrm{d}Z_t^j - \frac{h_j(X_t^i) + \hat{h}_j}{2}\,\mathrm{d}t,$$

for each $j = 1,...,m$, and $\hat{h}_j := \mathsf{E}[h_j(X_t^i)|\mathscr{Z}_t]$.

The particle dynamics (1.5), expressed in its Stratonovich form, are shown to provide a coordinate-free description of the filter that automatically satisfies the geometric constraints of the manifold. The filter is also shown to be exact. That is, in the limit of large number of particles, the empirical distribution of the particles exactly matches the posterior distribution. In a numerical implementation of FPF, however, the source of estimation error include: i) the use of only finitely many particles, ii) the discretization of the time interval, and iii) the error due to approximating the solution of the Poisson equation. Quantification and analysis of these errors call for future research.

The following is a summary of contributions apart from the theoretical foundation of the FPF on matrix Lie groups:

• **Poisson equation on matrix Lie groups.** The FPF algorithm requires numerical approximation of the gain functions $\mathsf{K}_j(x,t)$ as a solution to a linear Poisson equation on the Lie group. The Poisson equation for this case represents an extension of (1.3). An existence-uniqueness result for the solution is described in the Lie group setting. Two numerical methods are proposed to approximate the solution: i) in the Galerkin method, the gain function is approximated using a set of pre-defined basis functions; ii) in the kernel-based method, a numerical solution is obtained by solving a certain fixed-point equation. The two numerical procedures are described for both general Lie groups as well as special cases such as $SO(3)$ and the Euclidean space. For the Galerkin method, an error bound of the approximation is obtained for the case when the basis functions are taken as the eigenfunctions of the weighted Laplacian operator in the Poisson equation.

• **Feedback particle filter for attitude estimation.** The attitude estimation problem represents the important special case where the Lie group is $SO(3)$. For this important special case, the explicit form of FPF is described with respect to both the rotation matrix and the quaternion coordinates, with the latter being demonstrated for computational purposes. Furthermore, in the case where the posterior distribution is con-

centrated, a certain closed-form approximation, referred to as the *constant gain approximation*, of the gain function is obtained. With this approximation, evolution equations for the particle mean and the covariance are also derived and shown to be closely related to the left invariance EKF algorithm proposed in [20].

• **Numerical studies.** For the attitude estimation problem, the performance of FPF with different gain function approximation schemes are investigated and compared with several filters in the literature, including the extended Kalman filters, an unscented Kalman filter, an ensemble Kalman filter and a bootstrap particle filter. The performance of the filters are assessed by varying the prior distribution and the noise parameters. The particle filters are also assessed by varying the number of particles used in the filter.

• **Feedback particle filter for motion tracking.** The application involves tracking the motion of human arm with a wrist-worn motion sensor equipped with a gyroscope and an accelerometer. The tracking problem is cast as a filtering problem on the product Lie group $SO(3) \times SO(2)$. For this problem, experimental results with FPF are presented.

1.1.4 Literature review

In recent years, there has been a growing interest in the nonlinear filtering community to develop geometric approaches for handling constrained systems. In many cases, the constraints are described by smooth Riemannian manifolds, in particular the matrix Lie groups. Engineering applications of filtering on matrix Lie groups include: i) attitude estimation of aircrafts [80, 13], ii) localization of mobile robots [9, 72], and iii) visual tracking of humans and objects [96, 116]. In these applications, the matrix Lie groups of interest include the special orthogonal group $SO(3)$, the special Euclidean group $SE(3)$, and the special linear group $SL(3)$.

The mathematical problem of nonlinear filtering of stochastic processes in non-Euclidean spaces has a rich history; c.f., [51, 139, 146]. More recently, the focus has been on computational approaches to numerically approximate the conditional distribution. Such approaches have been developed, e.g., by extending the classical extended Kalman filter (EKF) to Lie groups. These extensions have appeared in discrete-time [22, 203], continuous-time [20, 58], and continuous-discrete-time settings [13, 21]. In particular, a number of EKF-based filters have been proposed and applied for attitude estimation, e.g., the additive EKF [8] and the multiplicative EKF [105]. The EKF-based attitude filters require a linearized model of the estimation error, typically derived using one of the many three-dimensional attitude representations, e.g. the Euler

angle [7], the rotation vector [145], or the modified Rodrigues parameter [124].

Apart from the EKF, unscented and particle filters for matrix Lie groups and Riemannian manifolds have also been an active area of research [38, 95, 39, 122, 160, 70]. 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 [42] and the bootstrap particle filter [32, 141] have been developed, using one of the attitude representations. Other non-parametric approaches include the point-mass filter [174] and filters based on certain variational formulations on the Lie groups [202, 152, 87].

Based on geometric group-theoretic methods for Lie groups, deterministic nonlinear observers have also been investigated [118, 98, 181]. A class of symmetry-preserving observers have been proposed to exploit certain invariance properties [19], leading to the invariant EKF [20, 13], the invariant unscented Kalman filter [41], the invariant ensemble Kalman filter [13], and the invariant particle filter [12] 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 [37, 186, 191]. More comprehensive surveys appear in [43, 200].

1.2 Global Optimization

The global optimization problem is

$$\min_{x \in \mathbb{R}^d} h(x), \tag{1.6}$$

where $h : \mathbb{R}^d \to \mathbb{R}$ is a real-valued function.

The objective is to develop a particle-based *gradient-free* algorithm to obtain the global minimizer of h. Gradient-free algorithms are sought because in many practical situations, evaluating the gradient of the objective function may be computationally prohibitive.

1.2.1 Background and overview of contributions

The main contribution is to develop the controlled particle filter (CPF) algorithm for the global optimization problem (1.6).

The CPF is developed based on the following Bayesian approach: Given an everywhere positive initial

density (prior) p_0^*, define the (posterior) density at a positive time t by

$$p^*(x,t) := \frac{p_0^*(x)\exp(-\beta h(x)t)}{\int p_0^*(y)\exp(-\beta h(y)t)\,\mathrm{d}y}, \tag{1.7}$$

where β is a positive constant parameter. Under certain additional technical assumptions on h and p_0^*, the density $p^*(x,t)$ weakly converges to the Dirac delta measure at the global minimizer as time $t \to \infty$. The Bayesian approach is attractive because it can be implemented recursively: Given a finite time interval $[0,T]$, define a discrete-time sequence $\{t_0, t_1, t_2, \ldots, t_{\bar{N}}\}$ of sampling instants with $0 = t_0 < t_1 < \ldots < t_{\bar{N}} = T$ and increments given by $\Delta t_n := t_n - t_{n-1}, n = 1, \ldots, \bar{N}$. The posterior distribution is expressed recursively as:

$$\begin{aligned} \text{Initialization:} \quad & \rho_0(x) = p_0^*(x), \\ \text{Update:} \quad & \rho_n(x) = \frac{\rho_{n-1}(x)\exp(-\beta h(x)\Delta t_n)}{\int \rho_{n-1}(y)\exp(-\beta h(y)\Delta t_n)\,\mathrm{d}y}, \quad n \geq 1. \end{aligned} \tag{1.8}$$

Note that at time t_n, $\rho_n(x) = p^*(x,t_n)$ by construction.

The Bayesian model (1.7) is not new: Taking time-derivative on both sides, one arrives at the replicator model used in the model-based evolutionary optimization algorithm [187]. The discretized version (1.8) is also a closely-related variant of the model used in the model reference adaptive search algorithm [76] and a particle filtering algorithm for optimization [215].

A conventional particle filter based on sequential importance sampling and resampling (SISR) may be used to sample from ρ_n. A particle filter is comprised of N stochastic processes $\{X_n^i : 1 \leq i \leq N\}$, where $X_n^i \in \mathbb{R}^d$ is the state for the i^{th} particle at iteration n. The SISR particle filter implements the following recursive steps:

$$\begin{aligned} \text{Initialization:} \quad & X_0^i \overset{\text{i.i.d.}}{\sim} p_0^*, \\ \text{Update:} \quad & X_n^i \overset{\text{i.i.d.}}{\sim} \sum_{i=1}^{N} w_n^i \delta_{X_{n-1}^i}, \quad n \geq 1, \end{aligned}$$

where $w_n^i \propto \exp(-\beta h(X_{n-1}^i)\Delta t_n)$ are referred to as the importance weights and δ_z denotes the Dirac-delta at $z \in \mathbb{R}^d$. In practice, the importance weights w_n^i can potentially suffer from large variance. To address this problem, several extensions have been described in literature based on consideration of suitable sampling (proposal) distributions and efficient resampling schemes; cf., [46, 47].

We present an alternative control-based approach to the construction and simulation of the particle

system for global optimization. In particular, the CPF consists of a controlled interacting particle system where the dynamics of the i^{th} particle evolves according to

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = u(X_t^i, t), \quad X_0^i \sim p_0^*, \tag{1.9}$$

where the *control function* $u(x,t)$ is obtained by solving the Poisson equation given in (1.3). In terms of the solution $\phi(x)$ of (1.3) (the subscript j in (1.3) is omitted since h is scalar-valued in this case), the control function at time t is given by

$$u(x,t) = -\beta \nabla \phi(x).$$

The inspiration for controlling a single particle – via the control input $u(X_t^i, t)$ in (1.9) – comes from the mean-field type control formalisms [81, 16, 24, 197], control methods for optimal transportation [183, 30, 31], and the feedback particle filter (FPF) algorithm for nonlinear filtering [195, 194]. One interpretation of the control input $u(X_t^i, t)$ is that it implements the "Bayesian update step" to steer the ensemble $\{X_t^i : 1 \le i \le N\}$ towards the global minimizer. Structurally, the control-based approach is a significant departure from the importance sampling based implementation of the Bayes rule in conventional particle filters. It is noted that there are no additional steps, e.g., associated with resampling, reproduction, death, or birth of particles. In the language of importance sampling, the particle flow is designed so that the particles automatically have identical importance weights for all time.

Based on the general framework as described above, specific contributions regarding the CPF algorithm are as follows:

• **Theory.** The variational formulation of CPF is presented, including the gradient flow construction of the Bayesian model (1.7) and the associated mean-field type optimal control problem from which the control function $u(x,t)$ is obtained. To the best of our knowledge, it is the first derivation/interpretation of a (Bayesian) particle filter as a solution to an optimal control problem. Numerical algorithms to approximate the control function are provided. The special case with a quadratic objective function is also considered where the control function admits a closed-form expression which is affine in the state.

• **Numerical studies.** The proposed CPF algorithm, together with several model-based methods, are simulated for benchmark optimization problems. Performance comparisons are provided between the control-based and the sampling-based implementation of the Bayesian model. The numerical results are obtained

via Monte-Carlo simulations.

1.2.2 Related work

There are two broad categories of global optimization algorithms: (i) Instance-based algorithms and (ii) Model-based algorithms; cf., [217]. The instance-based algorithms include simulated annealing [93, 149], genetic algorithms [62], differential evolution [166], nested partitions methods [158], and various types of random search [199] and particle swarm [91, 196] algorithms. The optimization is cast as an iterative search where one seeks to balance the exploration of the state-space with the optimization objective. In [150], such algorithms are referred to as "local search heuristics", presumably because they depend upon the local topological structure of the state-space.

In recent years, the focus has been on model-based algorithms where a reference probabilistic model – sequence of recursively-defined distributions of the decision variables – informs the search of the global optimizer. Examples include (i) parametric approaches such as the cross-entropy (CE) [150], the model reference adaptive search (MRAS) [76], and the model-based evolutionary optimization (MEO) [187]; and (ii) non-parametric approaches such as estimation of distribution algorithm [99], sequential Monte Carlo simulated annealing (SMC-SA) [213], and the particle filtering for optimization (PFO) [215]. While non-parametric approaches typically implement importance sampling and resampling to approximate the reference model, parametric version of model-based methods approximate the reference model by a prescribed class of parametric distributions. The latter is also referred to as density projection [214] in particle filtering. Related filtering models for global optimization appear in [114, 113, 130, 165, 59]. Bayesian filtering approaches to the particle swarm optimization also appear in [89, 134].

Multiple reference models have appeared in model-based methods; see recent surveys [75, 77, 79, 212, 114]. Apart from the Bayesian model used in PFO, other models include the optimal importance sampling density used in CE, the proportional selection density used in MRAS, the replicator model used in MEO, and the Boltzmann distribution used in SMC-SA. Many of these models are closely related to the model used in our algorithm.

System and control approaches, though not model-based, have been employed for global optimization. The algorithm based on a continuous-time consensus model appears in [128]. The use of stochastic differential equations with decaying diffusion is presented in [33, 2]. An optimal control problem is constructed

in [133] to improve parameter tuning of the algorithm therein.

1.3 Outline of the Thesis

The remainder of the thesis is comprised of two parts.

1) Part I covers the theory for both nonlinear filtering on matrix Lie groups and global optimization. Specifically, Chapter 2 contains the basic theory for extending the FPF to matrix Lie groups. The theoretical foundation of the proposed optimization algorithm is presented in Chapter 3. The major computation step in these algorithms – approximating the solution of a Poisson equation – is described in Chapter 4.

2) Part II contains applications of the proposed algorithms. The FPF for attitude estimation is developed in Chapter 3. Experimental results of FPF for the motion tracking problem are recorded in Chapter 6. Chapter 7 includes numerical studies for the global optimization problem.

   All the proofs appear in the appendix.

Part I

Theory

Chapter 2

Feedback Particle Filter on Matrix Lie Groups *

2.1 Introduction

In this chapter, we present an extension of the feedback particle filter (FPF) to matrix Lie groups. Throughout the chapter, a matrix Lie group is denoted as G, and its Lie algebra is denoted as $\mathscr{G}$. An overview of matrix Lie groups and related notation is contained in Sec. 2.2.

The mathematical problem involves the following sdes on matrix Lie groups:

$$\mathrm{d}X_t = X_t V_0(X_t)\,\mathrm{d}t + X_t V_\alpha(X_t) \circ \mathrm{d}B_t^\alpha, \tag{2.1a}$$

$$\mathrm{d}Z_t = h(X_t)\,\mathrm{d}t + \mathrm{d}W_t, \tag{2.1b}$$

where $X_t \in G$ is the state at time t, $Z_t \in \mathbb{R}^m$ is the observation vector; $V_0 : G \to \mathscr{G}$ and $V_\alpha : G \to \mathscr{G}$ for $\alpha = 1,...,r$ are Lie algebra-valued functions on G; B_t^α and W_t are mutually independent standard Wiener processes in $\mathbb{R}$ and $\mathbb{R}^m$, respectively, and they are also assumed to be independent of the initial state X_0; $h : G \to \mathbb{R}^m$ is a given vector-valued function. The j-th coordinate of Z_t and h are denoted as Z_t^j and h_j, respectively (i.e. $Z_t = (Z_t^1,...,Z_t^m)$ and $h = (h_1,...,h_m)$). The sde (2.1a) is expressed in its Stratonovich form, and the Einstein summation convention for the index α is used.

The objective is to numerically approximate the conditional distribution of X_t given the time-history (filtration) of observations $\mathscr{Z}_t = \sigma(Z_s : s \le t)$. The conditional distribution, denoted as π_t^*, acts on a function $f : G \to \mathbb{R}$ according to

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

whose time-evolution is described by the Kushner-Stratonovich filtering equation (see Theorem 5.7 in

*The content of this chapter is related to the publication [208, 210, 209].

[192]),

$$\pi_t^*(f) = \pi_0^*(f) + \int_0^t \pi_s^*(\mathscr{L}^* f)\,\mathrm{d}s + \int_0^t \left(\pi_s^*(fh) - \pi_s^*(h)\pi_s^*(f)\right)^T \left(\mathrm{d}Z_s - \pi_s^*(h)\,\mathrm{d}s\right), \tag{2.2}$$

for all smooth functions f with compact support, where $\mathscr{L}^* f := V_0 \cdot f + \frac{1}{2}\sum_{\alpha=1}^r V_\alpha \cdot (V_\alpha \cdot f)$. The operations $V_0 \cdot f$ and $V_\alpha \cdot f$ are defined in Sec. 2.2.

In this chapter, an algorithmic solution to this problem is presented. The algorithm is based on the FPF originally proposed in [193, 194, 195] in the Euclidean setting. The main result is to show that the gain-times-error update formula in the original Euclidean setting carries over to the manifold setting.

The primary challenge in implementing the FPF algorithm arises due to the gain function approximation. Apart from the numerical schemes that will be presented in Chapter 4, this chapter also considers a certain special case, namely where the posterior distribution is *concentrated*. In this case, a closed-form approximation, referred to as the *constant gain approximation*, is obtained. With this approximation, evolution equations for the mean and the covariance are also derived and shown to be closely related to the left invariance EKF algorithm proposed in [20]. Although the derivations for the concentrated distributions are restricted to the Lie group $SO(3)$, extending the results to general matrix Lie groups is straightforward.

The remainder of this chapter is organized as follows: A review of the relevant Lie group preliminaries is contained in Sec. 2.2. In Sec. 2.3, the generalization of the FPF algorithm to matrix Lie groups is presented, including both theory and numerical algorithms.

2.2 Mathematical Preliminaries

2.2.1 Geometry of Matrix Lie Groups

This section includes a brief review of matrix Lie groups based on [104, 68, 34]. The intent is to fix the notation used in subsequent sections.

The vector space of $n \times n$ real matrices is denoted as $M(n;\mathbb{R})$. The *general linear group*, denoted as $GL(n;\mathbb{R})$, is the group of $n \times n$ real invertible matrices, where the group operations are the matrix multiplication and matrix inversion. The identity element is the identity matrix, denoted as I. A *matrix Lie group*, denoted as G, is a closed subgroup of $GL(n;\mathbb{R})$. G is assumed to be connected. The dimension of G is

denoted as d.

For any $x \in G$, the tangent space at x is denoted as T_xG, which is a d-dimensional subspace of $M(n;\mathbb{R})$. The *left translation* map associated with x, denoted as $L_x : G \to G$, is a smooth map defined as

$$L_x(y) := xy$$

for all $y \in G$. The differential of L_x evaluated at y is a function denoted as $\mathrm{d}(L_x) : T_xG \to T_{xy}G$. For any $V \in T_yG$, the differential is evaluated according to,

$$\mathrm{d}(L_x)_y(V) = \frac{\mathrm{d}}{\mathrm{d}\tau}\Big|_{\tau=0} L_x\big(\gamma(\tau)\big), \tag{2.3}$$

where $\gamma : \mathbb{R} \to G$ is a smooth curve such that $\gamma(0) = y$ and $\gamma'(0) = V$.

A *vector field* on G, denoted as $\mathscr{V}$, is a function of x such that $\mathscr{V}(x) \in T_xG$. $\mathscr{V}$ is further called a *left-invariant vector field* if

$$\mathrm{d}(L_x)_y(\mathscr{V}(y)) = \mathscr{V}(xy)$$

for all $x, y \in G$.

The *Lie algebra* of G, denoted as $\mathscr{G}$, is the Lie algebra of all smooth left-invariant vector fields on G with the Lie bracket defined for vector fields (see Chapter 8 of [104]). $\mathscr{G}$ is identified with the tangent space T_IG via the following construction: For every left-invariant vector field $\mathscr{V} \in \mathscr{G}$, there is a unique $V \in T_IG$ such that

$$\mathscr{V}(x) = \mathrm{d}(L_x)_I(V) \tag{2.4}$$

for all $x \in G$. With this construction, T_IG is considered as a Lie algebra where the Lie bracket is given by the commutator,

$$[V,W] := VW - WV$$

for all $V, W \in T_IG$. From now on, we will use the notation $\mathscr{G}$ and T_IG interchangeably.

The *exponential map* of G is denoted as $\exp : \mathscr{G} \to G$. For matrix Lie groups, the exponential map is

given by the matrix exponential:

$$\exp(V) := \sum_{k=0}^{\infty} \frac{1}{k!} V^k = I + V + \frac{1}{2} V^2 + \cdots, \tag{2.5}$$

for $V \in \mathscr{G}$. The function $\gamma(\tau) := \exp(\tau V)$ defines a smooth curve on G such that $\gamma(0) = I$ and $\gamma'(0) = V$. Consequently, a left-invariant vector field $\mathscr{V}$, defined in (2.4), is calculated as,

$$\mathscr{V}(x) \overset{(2.3)}{=} \frac{\mathrm{d}}{\mathrm{d}\tau}\Big|_{\tau=0} L_x\big(\exp(\tau V)\big) \overset{(2.5)}{=} xV. \tag{2.6}$$

The vector space $\mathscr{G}$ is equipped with an inner product, denoted as $\langle \cdot, \cdot \rangle_{\mathscr{G}}$, and an orthonormal basis $\{E_1, ..., E_d\}$ such that $\langle E_i, E_j \rangle_{\mathscr{G}} = \delta_{ij}$. Define a set of left-invariant vector fields $\{\mathscr{E}_1, ..., \mathscr{E}_d\}$ according to,

$$\mathscr{E}_n(x) \overset{(2.4)}{:=} \mathrm{d}(L_x)_I(E_n) \overset{(2.6)}{:=} xE_n$$

for $n \in \{1, ..., d\}$. Then, $\{\mathscr{E}_1, ..., \mathscr{E}_d\}$ form a global frame of G, i.e., the set $\{xE_1, ..., xE_d\}$ form a basis of T_xG for all $x \in G$. Using these basis of T_xG, a vector field $\mathscr{V}$ is expressed as,

$$\mathscr{V}(x) = v_1(x)\, xE_1 + \cdots + v_d(x)\, xE_d$$

with $v_n(x) : G \to \mathbb{R}$ for $n = 1, ..., d$. We write $\mathscr{V}(x) = xV(x)$, where $V(x) := v_1(x)\, E_1 + \cdots + v_d(x)\, E_d$ is an element of $\mathscr{G}$ for each $x \in G$. The functions $\big(v_1(x), ..., v_d(x)\big)$ are referred to as the *coordinates* of the vector field $\mathscr{V}$. The construction of vector fields on G is illustrated in Figure 2.1. With a slight abuse of notation, the *action* of the vector field $\mathscr{V}$ on a function $f : G \to \mathbb{R}$ is denoted as,

$$V \cdot f(x) := \frac{\mathrm{d}}{\mathrm{d}\tau}\Big|_{\tau=0} f\big(x \exp(\tau V(x))\big). \tag{2.7}$$

A choice of the inner product for the tangent space T_xG is defined as follows: For $\overline{V}, \overline{W} \in T_xG$, the inner product of $\overline{V}$ and $\overline{W}$ is then given by,

$$\langle \overline{V}, \overline{W} \rangle := \langle \mathrm{d}(L_{x^{-1}})_x(\overline{V}), \mathrm{d}(L_{x^{-1}})_x(\overline{W}) \rangle_{\mathscr{G}} = \langle x^{-1}\overline{V}, x^{-1}\overline{W} \rangle_{\mathscr{G}}.$$

If $\overline{V} = \sum_{n=1}^{d} v_n xE_n$ and $\overline{W} = \sum_{n=1}^{d} w_n xE_n$ for an orthonormal basis $\{E_1, ..., E_d\}$ of $\mathscr{G}$, we then have $\langle \overline{V}, \overline{W} \rangle =$

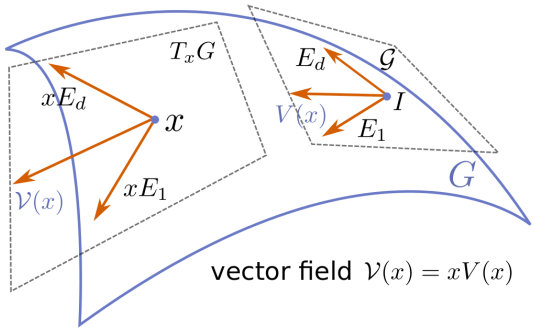


Figure 2.1: Construction of vector fields on G.

$\sum_{n=1}^{d} v_n w_n$.

For two vector fields $\mathscr{V}$ and $\mathscr{W}$, we define a function $\langle \mathscr{V}, \mathscr{W} \rangle : G \to \mathbb{R}$ as

$$\langle \mathscr{V}, \mathscr{W} \rangle(x) := \langle \mathscr{V}(x), \mathscr{W}(x) \rangle = \sum_{n=1}^{d} v_n(x) w_n(x). \tag{2.8}$$

For a vector field $\mathscr{V}$ such that $\mathscr{V}(x) = xV(x)$ and $V(x) \in \mathscr{G}$, we define the norms,

$$|V|_{\mathscr{G}}(x) := \sqrt{\langle V(x), V(x) \rangle_{\mathscr{G}}}, \quad \text{and} \quad |\mathscr{V}|_G(x) := \sqrt{\langle \mathscr{V}, \mathscr{V} \rangle(x)}. \tag{2.9}$$

The vector field, $\operatorname{grad}(f)$, for a differentiable function $f : G \to \mathbb{R}$ is defined such that

$$\langle \operatorname{grad}(f), \mathscr{V} \rangle(x) = V \cdot f(x) \tag{2.10}$$

holds for all smooth vector fields $\mathscr{V}$. By taking $\mathscr{V}$ as $\mathscr{E}_1, \ldots, \mathscr{E}_d$, we obtain the coordinates of $\operatorname{grad}(\phi)$ as $\big(E_1 \cdot f(x), \ldots, E_d \cdot f(x)\big)$, where similar as (2.7),

$$E_n \cdot f(x) = \frac{\mathrm{d}}{\mathrm{d}\tau}\Big|_{\tau=0} f\big(x \exp(\tau E_n)\big)$$

for $n = 1, \ldots, d$.

A measure μ on G is called a (left) *Haar measure* if $\mu(xA) = \mu(A)$ for all $x \in G$ and all Borel set $A \subset G$, where $xA := \{xa : a \in A\}$ (see Chapter 5 of [57]). Given a vector field $\mathscr{V}$, we define a function $\operatorname{div}\mathscr{V} : G \to \mathbb{R}$ that satisfies

$$-\int_G \operatorname{div}(\mathscr{V})(x) f(x) \mu(\mathrm{d}x) = \int_G \langle \mathscr{V}, \operatorname{grad}(f) \rangle(x) \mu(\mathrm{d}x) \tag{2.11}$$

for all smooth functions $f : G \to \mathbb{R}$, where μ is the Haar measure. Given an orthonormal basis $\{E_1, ..., E_d\}$ of $\mathscr{G}$, $\text{div}\mathscr{V}(x)$ is calculated as,

$$\text{div}\mathscr{V}(x) = E_1 \cdot v_1(x) + \cdots + E_d \cdot v_d(x), \tag{2.12}$$

where $(v_1(x), ..., v_d(x))$ are the coordinates of $\mathscr{V}$.

Example 2.1 (Geometry of $SO(3)$) *The special orthogonal group $SO(3)$ is the group of 3×3 matrices R such that $RR^T = I$ and $\det(R) = 1$. The Lie algebra $so(3)$ is identified with the 3-dimensional vector space of skew-symmetric matrices. An inner product is $\langle \Omega_1, \Omega_2 \rangle_{so(3)} = \frac{1}{2} Tr(\Omega_1^T \Omega_2)$ for $\Omega_1, \Omega_2 \in so(3)$, and an orthonormal basis $\{E_1, E_2, E_3\}$ of $so(3)$ are,*

$$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad E_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.13}$$

These matrices have the physical interpretation of generating rotations about the three canonical axes in $\mathbb{R}^3$. Here, $\det(\cdot)$ and $Tr(\cdot)$ denote the determinant and trace of a matrix, respectively. Given the basis in (2.13), a vector $\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ is uniquely mapped to an element in $so(3)$, denoted as

$$[\omega]_\times := \omega_1 E_1 + \omega_2 E_2 + \omega_3 E_3. \tag{2.14}$$

We consider the following function spaces: The vector space of smooth real-valued functions $f : G \to \mathbb{R}$ with compact support is denoted as $C_c^\infty(G)$. For a probability distribution π on G, $L^2(G;\pi)$ denotes the Hilbert space of functions on G that satisfy $\pi(|f|^2) < \infty$ (here $\pi(|f|^2) := \int_G |f|^2 \, \mathrm{d}\pi(x)$, where $\mathrm{d}\pi(x) = \rho(x)\mu(\mathrm{d}x)$, ρ is the density associated with π, and μ denotes the Haar measure on the Lie group); $H^1(G;\pi)$ denotes the Hilbert space of functions f such that f and $E_n \cdot f$ (defined in the weak sense) are all in $L^2(G;\pi)$; and $H_0^1(G;\pi) := \{\phi \in H^1(G;\pi) \,|\, \pi(\phi) = 0\}$.

2.2.2 Quaternions

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

$$q = (q_0,\ q_1,\ q_2,\ q_3) = \Big(\cos(\frac{\theta}{2}),\ \omega_1 \sin(\frac{\theta}{2}),\ \omega_2 \sin(\frac{\theta}{2}),\ \omega_3 \sin(\frac{\theta}{2})\Big), \tag{2.15}$$

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

$$p \otimes q := \begin{bmatrix} p_0 q_0 - p_V \cdot q_V \\ p_0 q_V + q_0 p_V + p_V \times q_V \end{bmatrix},$$

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 between two vectors.

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

$$R = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & 2q_0^2 + 2q_2^2 - 1 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}. \tag{2.16}$$

For more comprehensive introduction of quaternions, we refer the reader to [178].

2.2.3 Sdes on matrix Lie groups

The sde (2.1a) is expressed in its Stratonovich form which provides a coordinate-free description of the sde. The difference between the Stratonovich form and the Itô form is illustrated next with an example of an sde defined on the unit circle.

Example 2.2 (sde on unit circle, c.f., Example 5.1.4 in [140]) *The unit circle S^1 is a smooth manifold embedded in $\mathbb{R}^2$. It is also identified with the Lie group $SO(2)$. Consider the following Itô sde expressed using*

the Cartesian coordinates,

$$\begin{bmatrix} \mathrm{d}X_1(t) \\ \mathrm{d}X_2(t) \end{bmatrix} = -\frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix}\mathrm{d}t + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix}\mathrm{d}B_t, \tag{2.17}$$

where B_t is the standard Wiener process in $\mathbb{R}$. the solution of (2.17) is $(\cos(B_t),\ \sin(B_t))$, which represents the Brownian motion on S^1. Denote $X_t = (X_1(t), X_2(t))^T$ and define $f(X_t) = |X_t|^2$. The fact that the solution X_t of (2.17) satisfies the constraint $f(X_t) \equiv 1$ for all $t \geq 0$ can be verified directly: Rewrite the sde (2.17) in a compact form,

$$\mathrm{d}X_t = AX_t\,\mathrm{d}t + MX_t\,\mathrm{d}B_t, \tag{2.18}$$

where according to (2.17), $A = -(1/2)I$ and the matrix M is shown to satisfy $M^TM = I$ and $X_t^TMX_t = 0$. Then, applying the Itô's formula (see Section 4.3.3 in [60]) and the Itô's rule (i.e., $\mathrm{d}B_t\,\mathrm{d}t = 0$, and $(\mathrm{d}B_t)^2 = \mathrm{d}t$),

$$\begin{aligned} \mathrm{d}f(X_t) &= (\nabla f(X_t))^T\,\mathrm{d}X_t + \frac{1}{2}\,\mathrm{d}X_t^T\,\Delta f(X_t)\,\mathrm{d}X_t \\ &= 2X_t^T\,\mathrm{d}X_t + \mathrm{d}X_t^T\,\mathrm{d}X_t \\ &= 2X_t^T(A\,X_t\,\mathrm{d}t + M\,X_t\,\mathrm{d}B_t) + X_t^TM^TM\,X_t\,\mathrm{d}t \\ &= -X_t^TX_t\,\mathrm{d}t + 0 + X_t^TX_t\,\mathrm{d}t = 0, \end{aligned}$$

where ∇f and Δf denote the gradient vector and the Hessian matrix of f, respectively. Hence, $f(X_t) \equiv 1$ for all $t > 0$ if $f(X_0) = 1$.

The Stratonovich form of (2.17) is obtained as (see Section 4.3.6 in [60] for the conversion formulas between Itô and Stratonovich sdes),

$$\begin{bmatrix} \mathrm{d}X_1(t) \\ \mathrm{d}X_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} X_1(t) \\ X_2(t) \end{bmatrix} \circ \mathrm{d}B_t. \tag{2.19}$$

For a Stratonovich sde, the change of variable formula is identical to the ordinary chain rule. Using this fact, one verifies that $\mathrm{d}f(X_t) = 0$ for the Stratonovich sde (2.19).

Although (2.17) and (2.19) are equivalent (i.e., has the same solution), they have different interpreta-

tions. The form of the Itô sde depends upon the choice of the coordinate system, e.g., the additional drift term $AX_t\,\mathrm{d}t$ *is needed in* (2.18) *for the solution to satisfy the manifold constraint* $|X_t|^2 = 1$. *In the Stratonovich sde* (2.19)*, the right-hand-side represents a tangent vector in* S^1. *In general, the definition of a tangent vector is coordinate-free, and the solution of the resulting sde automatically satisfies the manifold constraint. The coordinate-free representation of* (2.19) *is given by*

$$\mathrm{d}X_t = V(X_t) \circ \mathrm{d}B_t,$$

where $X_t \in S^1$, *and* $V(x) \in T_x S^1$.

Analogous examples of Itô sde whose solution is a Brownian motion in the Lie group $SO(3)$ are provided in Section 20.4 of [34] and also in [143]. Similar as in (2.17), a coordinate-dependent drift term is needed to satisfy the manifold constraint. The corresponding Stratonovich sde is coordinate-free and admits a similar form as (2.1a). Itô sdes constructed using the local coordinates of a Riemannian manifold appear in [161]. A comparison of Itô and Stratonovich sdes for manifolds is also presented in [119].

2.2.4 Numerical Integration of sdes on matrix Lie groups

This section describes a numerical integration method for the sde (2.1a). Consider a finite time interval $[0,T]$ with an associated discrete-time sequence $\{t_0, t_1, t_2, \ldots, t_{\bar{N}}\}$ of sampling instants with $0 = t_0 < t_1 < \cdots < t_{\bar{N}} = T$ and uniform increments given by $\Delta t := t_n - t_{n-1}$. A numerical simulation algorithm is tabulated in Algorithm 1. This algorithm is a direct extension of the Euler-Maruyama approximation scheme [94] to Lie groups, based on the explicit representation of the matrix exponential. Closely related algorithms also appear in [119, 144] as well as in [122] particularly for implementing a particle filter algorithm.

Higher-order numerical algorithms have recently been developed for matrix Lie groups [120] and the Stiefel manifold [121]. The algorithm described in Algorithm 1 represents the simplest form of these higher-order algorithms.

2.3 Feedback Particle Filter on Matrix Lie Groups

This section extends the FPF algorithm to matrix Lie groups, with necessary modifications to the original framework to account for the manifold structure.

Algorithm 1 Numerical integration of sde (2.1a)

1: **Input:** Initial condition $X_0 \in G$

2: Assign $t = t_0$

3: **Iteration:** from t to $t + \Delta t$

4: Generate sample ΔB_t^α from $\mathcal{N}(0, \Delta t)$ for $\alpha = 1, ..., r$

5: Calculate $\Delta V_t = V_0(X_t)\,\Delta t + V_\alpha(X_t)\,\Delta B_t^\alpha$

6: Propagate the state $X_{t+\Delta t} = X_t \exp\big(\Delta V_t\big)$

7: Assign $t = t + \Delta t$

2.3.1 Particle Dynamics and Control Architecture

The FPF on a matrix Lie group G is a controlled system comprising of N stochastic processes $\{X_t^i\}_{i=1}^N$ with $X_t^i \in G$. The particles are modeled by the Stratonovich sde,

$$\mathrm{d}X_t^i = X_t^i\big(V_0(X_t^i) + \mathsf{u}(X_t^i, t)\big)\,\mathrm{d}t + X_t^i V_\alpha(X_t^i) \circ \mathrm{d}B_t^{\alpha,i} + X_t^i\,\mathsf{K}_j(X_t^i, t) \circ \mathrm{d}Z_t^j, \tag{2.20}$$

where $B_t^{\alpha,i}$ for $\alpha = 1, ..., r$ and $i = 1, ..., N$ are mutually independent standard Wiener processes in $\mathbb{R}$, and the Einstein summation convention is used for the free indices α and j. The functions $\mathsf{u}(x,t)$, $\mathsf{K}_j(x,t) : G \times [0,T] \to \mathscr{G}$ are referred to as the *control* and *gain* function, respectively, whose coordinates are denoted as $(\mathsf{u}_1, ..., \mathsf{u}_d)$ and $(\mathsf{k}_{1,j}, ..., \mathsf{k}_{d,j})$, for $j = 1, ..., m$. These functions need to be chosen. The following admissibility requirement is imposed on u and K_j:

Definition 2.1 *(Admissible input): The functions $\mathsf{u}(x,t)$ and $\mathsf{K}_j(x,t)$ are admissible if they are $\mathscr{Z}_t-$measurable and $\mathsf{E}[\big(\sum_n |\mathsf{u}_n(X_t^i, t)|\big)] < \infty$, $\mathsf{E}[\sum_n |\mathsf{k}_{n,j}(X_t^i, t)|^2] < \infty$ for each $j = 1, ..., m$ and for all t.*

The conditional distribution of the particle X_t^i given $\mathscr{Z}_t$ is denoted by π_t, which acts on a function f according to

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

The evolution equation for π_t is given by the proposition below. The proof appears in Appendix A.1.

Proposition 2.1 *Consider the particle X_t^i with dynamics described by (2.20). The forward evolution equa-*

tion of the conditional distribution π_t is given by,

$$\pi_t(f) = \pi_0(f) + \int_0^t \pi_s(\mathscr{L} f)\,\mathrm{d}s + \int_0^t \pi_s(\mathsf{K}_j \cdot f)\,\mathrm{d}Z_s^j, \tag{2.21}$$

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

$$\mathscr{L} f := (V_0 + \mathsf{u}) \cdot f + \frac{1}{2}\sum_{\alpha=1}^{r} V_\alpha \cdot (V_\alpha \cdot f) + \frac{1}{2}\sum_{j=1}^{m} \mathsf{K}_j \cdot (\mathsf{K}_j \cdot f).$$

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 functions $\mathsf{u}(x,t)$, $\mathsf{K}_j(x,t)$ are said to be *exact* if $\pi_t = \pi_t^*$ for all $t \in [0,T]$. Thus, the objective is to choose u and K_j such that, given $\pi_0 = \pi_0^*$, the evolution of the two conditional distributions are identical (see (2.2) and (2.21)).

Solution: The FPF represents the following choice of the gain function K and the control function u:

1) Gain function: The gain function is obtained as follows: For $j = 1,\ldots,m$, let $\phi_j \in H^1(G;\pi_t)$ be the solution of a linear Poisson equation:

$$\begin{aligned} &\pi_t\big(\langle \mathrm{grad}(\phi_j), \mathrm{grad}(\psi)\rangle\big) = \pi_t\big((h_j - \hat{h}_j)\psi\big), \\ &\pi_t(\phi_j) = 0 \quad (\text{normalization}), \end{aligned} \tag{2.22}$$

for all $\psi \in H^1(G;\pi_t)$, where $\hat{h}_j := \pi_t(h_j)$. The gain function $\mathsf{K}_j \in \mathscr{G}$ is then chosen as,

$$x\mathsf{K}_j(x,t) = \mathrm{grad}(\phi_j)(x). \tag{2.23}$$

Given a basis $\{E_n\}_{n=1}^d$ of the Lie algebra $\mathscr{G}$, and noting that (see (2.10))

$$\mathrm{grad}(\phi_j)(x) = E_1 \cdot \phi_j(x)\, E_1^x + \cdots + E_d \cdot \phi_j(x)\, E_d^x$$

where $E_n^x = xE_n$, the coordinates of K_j is given by

$$\mathsf{k}_{n,j}(x,t) = E_n \cdot \phi_j(x) \ , \quad \text{for } n = 1,...,d. \tag{2.24}$$

2) Control function: The function u is chosen as,

$$\mathsf{u}(x,t) = -\frac{1}{2}\sum_{j=1}^{m} \mathsf{K}_j(x,t)\left(h_j(x) + \hat{h}_j\right). \tag{2.25}$$

Feedback particle filter: Using these choice of u and K, the i-th particle in the FPF has the following representation:

$$\mathrm{d}X_t^i = \underbrace{X_t^i V_0(X_t^i)\,\mathrm{d}t + X_t^i V_\alpha \circ \mathrm{d}B_t^{\alpha,i}}_{\text{propagation}} + \underbrace{X_t^i\, \mathsf{K}_j(X_t^i,t) \circ \mathrm{dI}_t^{j,i}}_{\text{observation update}}, \tag{2.26}$$

where the error $\mathrm{dI}_t^{j,i} \in \mathbb{R}$ is a modified form of the innovation process:

$$\mathrm{dI}_t^{j,i} = \mathrm{d}Z_t^j - \frac{h_j(X_t^i) + \hat{h}_j}{2}\,\mathrm{d}t, \tag{2.27}$$

for each $j = 1,...,m$. 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 (2.22) must be solved for each $j = 1,...,m$, and for each time t.

The exactness is asserted in the following theorem. The proof is contained in appendix A.2.

Theorem 2.1 *Let π_t^* and π_t satisfy the forward evolution equations (2.2) and (2.21), respectively. Suppose that the gain functions K_j, $j = 1,...,m$, are obtained using (2.22)-(2.23), and the control function u is obtained using (2.25). Suppose also that these functions are admissible. Then, assuming $\pi_0 = \pi_0^*$, we have,*

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

for all $t \in [0,T]$ and all function $f \in C_c^\infty(G)$.

Remark 2.1 *In the original Euclidean setting, the FPF has the prettiest – gain times error – representation*

of the update step in the Stratonovich form of the filter (see Remark 1 in [194]). In the Itô form, the filter includes an additional Wong-Zakai correction term. For sdes on a manifold, it has been shown in Sec. 2.2.3 that the Stratonovich form is invariant to coordinate transformations while the Itô form is not [140]. So, for the gain times error form of the update step to have an intrinsic coordinate independent form, the multiplication must necessarily be in the Stratonovich form.

Remark 2.2 *The equation (2.22) is the weak form of a Poisson equation. Suppose π_t admits an everywhere positive density, denoted as ρ_t. Then the strong form of (2.22) is given by the Poisson equation,*

$$\Delta_{\rho_t}\phi_j = -(h_j - \hat{h}_j), \tag{2.28}$$

where $\Delta_{\rho_t}\phi_j := \frac{1}{\rho_t} div\big(\rho_t\, grad(\phi_j)\big)$ is the weighted Laplacian on the manifold [64], and $div(\cdot)$ is defined according to (2.11). Multiplying both sides of (2.28) by $\psi(x)\rho_t(x)$ and integrating by parts, one arrives at the weak form (2.22).

In the Euclidean case, the gain function was obtained as the gradient of the solution of a Poisson equation [194]. Remark 2.2 shows that the Euclidean gain function is a special case of the more general Lie group formula (2.22)-(2.23). For the latter, the definition of divergence and gradient ensures that the Poisson equation has a coordinate-free representation. The gain function, expressed as gradient of the solution of the Poisson equation, is an element of the Lie algebra. This is consistent with the use of Lie algebra to define vector fields for dynamics evolving on the Lie group.

2.3.2 FPF Algorithm Summary

The numerical algorithm of the FPF on a matrix Lie group is summarized in Algorithm 2. The algorithm simulates N particles, $\{X_t^i\}_{i=1}^N$, according to the sde (2.26) with the initial conditions $\{X_0^i\}_{i=1}^N$ sampled i.i.d. from a given prior distribution π_0^*. Numerical approximation of the gain function K_j is the subject of Chapter 4, where the gain function is approximated using either the Galerkin scheme (see Sec. 4.3 and Algorithm 4) or the kernel-based scheme (see Sec. 4.4 and Algorithm 5).

Algorithm 2 Feedback Particle Filter on a matrix Lie group

Initialization: Samples $\{X_0^i\}_{i=1}^N$ i.i.d. from π_0^*

Assign $t = 0$

Iteration: from t to $t + \Delta t$

Calculate $\hat{h}_j^{(N)} = \frac{1}{N} \sum_{i=1}^N h_j(X_t^i)$ for $j = 1, 2, ..., m$

for $i = 1$ to N **do**

  Generate samples $\Delta B_t^{\alpha,i}$ from $N(0, \Delta t)$ for $\alpha = 1, ..., r$

  Assign $\Delta U_t^i = 0$

  for $j = 1$ to m **do**

    Calculate the error $\Delta \mathrm{I}_{j,t}^i := \Delta Z_{j,t} - \frac{1}{2}\left(h_j(X_t^i) + \hat{h}_j^{(N)}\right) \Delta t$

    Calculate gain function $\mathsf{K}_j(X_t^i, t)$ (see Chapter 4)

    Assign $\Delta U_t^i = \Delta U_t^i + \mathsf{K}_j(X_t^i, t)\, \Delta \mathrm{I}_{j,t}^i$

  end for

  Calculate $\Delta V_t^i = V_0(X_t^i)\, \Delta t + V_\alpha(X_t^i)\, \Delta B_t^{\alpha,i} + \Delta U_t^i$

  Propagate the particle $X_{t+\Delta t}^i = X_t^i \exp\left(\Delta V_t^i\right)$

end for

Return: empirical mean of $\{X_{t+\Delta t}^i\}_{i=1}^N$

Assign $t = t + \Delta t$

2.4 Feedback Particle Filter with Concentrated Distributions

In its original Euclidean setting [194], the FPF algorithm is shown to represent a generalization of the Kalman filter in the following sense: Suppose that the signal and the observation models are linear and that the prior distribution is Gaussian. Then, it is shown that:

1. The gain K_t is a constant for each t whose value equals the Kalman gain;

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

For the general nonlinear non-Gaussian case, the gain function K_t is no longer a constant and must be numerically approximated. However, the conditional expectation of the gain function, $\mathsf{E}[\mathsf{K}_t \,|\, \mathscr{Z}_t]$, 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 [194, 17, 164].

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 [186]), 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 matrix Lie groups:

1. A closed-form formula for the constant gain approximation is derived and shown to admit the form of the Kalman gain;

2. The equation for the mean and covariance are derived and shown to be closely related to the continuous-time left invariant EKF algorithm in [20].

In this section, we restrict our attention to the following filtering problem on the Lie group $SO(3)$,

$$\mathrm{d}R_t = R_t\,\Omega_t\,\mathrm{d}t + R_t \circ [\,\sigma_B\,\mathrm{d}B_t\,]_\times, \tag{2.29a}$$

$$\mathrm{d}Z_t = h(R_t)\,\mathrm{d}t + \sigma_W\,\mathrm{d}W_t, \tag{2.29b}$$

where $R_t \in SO(3)$ is the hidden state, $\Omega_t \in so(3)$, $h : SO(3) \to \mathbb{R}^m$ is a given function, B_t and W_t are mutually independent standard Wiener processes in $\mathbb{R}^3$ and $\mathbb{R}^m$, respectively, and σ_B and σ_W are positive scalar parameters. The definition of $[\,\cdot\,]_\times$ (see (2.14) in Sec. 2.2.1) indicates that $[\,\sigma_B\,\mathrm{d}B_t\,]_\times \in so(3)$. The problem (2.29a)-(2.29b) is also used to model the attitude estimation problem in Chapter 5 (see (5.5a)-(5.5b)).

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 (These are described in Sec. 5.5 where numerical results for the attitude estimation problem are presented).

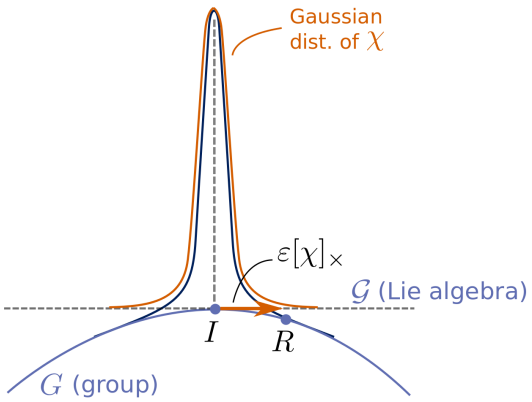


Figure 2.2: Concentrated distribution on G with mean at the identity, where $R = \exp(\varepsilon\,[\,\chi\,]_\times)$ is close to the mean. The random variable χ has a Gaussian distribution in $so(3) \cong \mathbb{R}^3$. A concentrated distribution with non-identity mean is defined using left translation [21].

2.4.1 Constant Gain Approximation of FPF

For the filtering problem (2.29a)-(2.29b) on $SO(3)$, the (strong form) Poisson equation in the feedback particle filter is given by

$$\Delta_\rho \phi_j = -\frac{1}{\sigma_W^2}(h_j - \hat{h}_j) \tag{2.30}$$

for $j = 1, \ldots, m$, where $\Delta_\rho \phi_j = \frac{1}{\rho}\,\mathrm{div}\big(\rho\,\mathrm{grad}(\phi_j)\big)$, and ρ denotes the probability density function associated with the particle distribution π. The dependence on the time t is suppressed in this section (i.e., we express R_t as R, π_t as π, ρ_t as ρ etc.).

Consider *concentrated distribution* whereby the random variable R on $SO(3)$ is parametrized as,

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

where $\chi \in so(3) \cong \mathbb{R}^3$ is a Gaussian random variable with mean 0 and covariance $\overline{\Sigma}$, and ε is a small parameter. 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 A.3.

Proposition 2.2 *Consider the Poisson equation* (2.30) *where the random variable* $R = \mu\,\exp(\varepsilon\,[\,\chi\,]_\times)$, *and* $\chi \in \mathbb{R}^3$ *is a Gaussian random variable with mean* 0 *and covariance* $\overline{\Sigma}$. *Suppose* $\sigma_W = \varepsilon\,\overline{\sigma}_W$. *Let* $[\mathsf{K}_j]$:

$SO(3) \to \mathbb{R}^3$ denote the coordinates of the gain function K_j. Then, in the asymptotic limit as $\varepsilon \to 0$,

$$[\mathsf{K}_j] = \frac{1}{\overline{\sigma}_W^2} \overline{\Sigma} H_j + O(\varepsilon),$$

where $H_j := \big(E_1 \cdot h_j(\mu), E_2 \cdot h_j(\mu), E_3 \cdot h_j(\mu)\big) \in \mathbb{R}^3$ for $j = 1, ..., m$.

In the attitude estimation problem described in Chapter 5, the observation model $h(R) = R^T r \in \mathbb{R}^3$ (i.e., $m = 3$), where $r \in \mathbb{R}^3$ is a known vector (see the accelerometer and magnetometer model (5.3) and (5.4)). In this special case, the constant gain approximation is analogous to the Kalman gain.

Corollary 2.1 *Consider the Poisson equation (2.30) where the random variable $R = \mu \exp(\varepsilon [\chi]_\times)$, where $\chi \in \mathbb{R}^3$ is a Gaussian random variable with mean 0 and covariance $\overline{\Sigma}$. Let $h_j(R) = e_j^T R^T r$ for $j = 1, 2, 3$. Let $\mathsf{K} \in \mathbb{R}^{3\times 3}$ whose columns contain the coordinates of the gain functions K_1, K_2, K_3. Then, in the asymptotic limit as $\varepsilon \to 0$,*

$$\mathsf{K} = \frac{1}{\overline{\sigma}_W^2} \overline{\Sigma} H^T + O(\varepsilon), \tag{2.31}$$

where $H := [\mu^T r]_\times$.

In a numerical implementation, the constant gain approximation is calculated as,

$$\mathsf{K}^c = \frac{1}{\sigma_W^2} \hat{\Sigma} [\hat{\mu}^T r]_\times^T,$$

where $\hat{\mu}$ and $\hat{\Sigma}$ denote the empirical mean and covariance of the particles $\{R^i\}_{i=1}^N$, respectively. The geometric definition of $\hat{\mu}$ and related numerical algorithms appear in [132, 155]. An algorithm using the quaternion coordinates of $SO(3)$ is described in Sec. 5.3. The formula of $\hat{\Sigma}$ is given by,

$$\hat{\Sigma} = \frac{1}{N-1} \sum_{i=1}^{N} [\log(\hat{\mu}^T R^i)]^\vee [\log(\hat{\mu}^T R^i)]^{\vee T},$$

where $\log(\cdot) : SO(3) \to so(3)$ denotes the matrix logarithm, and for $\Omega = \omega_1 E_1 + \omega_2 E_2 + \omega_3 E_3 \in so(3)$, $[\Omega]^\vee := (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ (i.e., $[\cdot]^\vee$ is the inverse of $[\cdot]_\times$).

2.4.2 FPF with Concentrated Distributions

Consider the attitude estimation problem,

$$\begin{aligned} \mathrm{d}R_t &= R_t\,\Omega_t\,\mathrm{d}t + R_t \circ [\,\varepsilon\,\overline{\sigma}_B\,\mathrm{d}B_t\,]_\times, \\ \mathrm{d}Z_t &= R_t^T\,r\,\mathrm{d}t + \varepsilon\,\overline{\sigma}_W\,\mathrm{d}W_t, \end{aligned}$$

with initial condition $R_0 = \mu_0 \exp(\varepsilon\,[\,\chi_0\,]_\times)$, where μ_0 is exactly known and $\chi_0 \sim \mathcal{N}(0, \overline{\Sigma}_0)$.

Using the constant gain approximation (2.31), the FPF for this problem is given by,

$$\mathrm{d}R_t^i = R_t^i\,\Omega_t\,\mathrm{d}t + R_t^i \circ [\,\varepsilon\,\overline{\sigma}_B\,\mathrm{d}B_t^i\,]_\times + R_t^i\,[\,\mathsf{K}_t \circ \mathrm{dI}_t^i\,]_\times, \tag{2.32}$$

for small ε and small time $t \in [0, \varepsilon T]$, where $\mathsf{K}_t := \frac{1}{\overline{\sigma}_W^2}\,\overline{\Sigma}_t\,H_t^T$ is the constant gain, $H_t = [\,\mu_t^T\,r\,]_\times$, and the error

$$\mathrm{dI}_t^i = \frac{h(R_t^i) + \hat{h}}{2}\,\mathrm{d}t$$

where $h(R) = R^T r$. The general form of FPF on $SO(3)$ will be presented in Sec. 5.3.

In the following theorem, it is shown that μ_t and $\overline{\Sigma}_t$ evolve according to the equations that are closely related to the left invariant EKF. The proof is contained in Appendix A.4.

Theorem 2.2 *Consider the FPF (2.32) where K_t is given by the constant gain approximation. Suppose that over a time horizon $[0, \varepsilon T]$, $R_t^i = \mu_t \exp(\varepsilon\,[\,\chi_t^i\,]_\times)$ where $\chi_t^i \sim \mathcal{N}(0, \overline{\Sigma}_t)$. Then, in the asymptotic limit as $\varepsilon \to 0$, μ_t and $\overline{\Sigma}_t$ evolve according to the respective sdes,*

$$\mathrm{d}\mu_t = \mu_t\,\Omega_t\,\mathrm{d}t + \mu_t\,[\,\mathsf{K}_t \circ \mathrm{dI}_t\,]_\times, \tag{2.33}$$

$$\mathrm{d}\overline{\Sigma}_t = (A_t\,\mathrm{d}t - [\,\mathsf{K}_t\,\mathrm{dI}_t\,]_\times)\,\overline{\Sigma}_t + \overline{\Sigma}_t\,(A_t\,\mathrm{d}t - [\,\mathsf{K}_t\,\mathrm{dI}_t\,]_\times)^T + \overline{\sigma}_B^2\,I\,\mathrm{d}t - \frac{1}{\overline{\sigma}_W^2}\,\overline{\Sigma}_t H_t^T H_t \overline{\Sigma}_t\,\mathrm{d}t, \tag{2.34}$$

where $A_t = -\Omega_t$ and $\mathrm{dI}_t = \mathrm{d}Z_t - \mu_t^T\,r\,\mathrm{d}t$.

The equation for the mean (2.33) is identical to the left invariant EKF [20]. The equation of the covariance (2.34) 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 [21], where these

terms are induced by the so-called re-parametrization step in the observation update. Other applications of concentrated Gaussian distributions for filtering on matrix Lie groups have appeared in [26, 27]. Related results on error propagation and Bayesian fusion on matrix Lie groups also appear in [186, 35, 191, 10].

2.5 Conclusions

In this chapter, the feedback particle filter was constructed for nonlinear filtering problems on matrix Lie groups. It was shown that FPF is an intrinsic algorithm that has a coordinate-free representation. The FPF update formula (2.26) not only provides for a generalization of the Kalman filter to the nonlinear non-Gaussian case but also that the generalization carries over to nonlinear spaces such as the matrix Lie groups.

For the Lie group $SO(3)$, The special case with concentrated Gaussian posterior distributions was also considered. A constant gain approximation of the Poisson equation was derived. The resulting filter equations, represented by the evolution of the conditional mean and covariance, are closely related to the left invariant EKF. This result is analogous to the fact that the original (Euclidean) FPF reduces to the classical Kalman filter for a linear Gaussian filtering problem.

Future research directions related to filtering on manifolds include the following:

1) Extension of FPF when the observation Z_t also evolves on a manifold. A geometric framework has recently been developed in [153, 139] that transform the problem to the Euclidean space. An EKF algorithm with discrete-time group-valued observations also appears in [21].

2) Finite-dimensional filters on matrix Lie groups and the simplification of the gain function in this case. This is equivalent as seeking a Lie group analogue of the well-known Beneš filter [15].

3) Extension of FPF to filtering problems defined in other matrix manifolds, such as the Stiefel manifold and the Grassmann manifold [54]. These manifolds have important applications in signal processing problems, e.g., subspace estimation of signals [162, 163].

Chapter 3

A Controlled Particle Filter for Global Optimization

*

3.1 Introduction

We consider the global optimization problem:

$$\min_{x\in\mathbb{R}^d} h(x),$$

where $h:\mathbb{R}^d\to\mathbb{R}$ is a real-valued function. This chapter is concerned with gradient-free simulation-based algorithms to obtain the global minimizer, denoted as

$$\bar{x} = \arg\min_{x\in\mathbb{R}^d} h(x).$$

It is assumed that such a minimizer exists and is unique.

As described in Sec. 1.2.1, a Bayesian approach to solve the problem is as follows: Given an everywhere positive initial density (prior) p_0^*, define the (posterior) density at a positive time t by

$$p^*(x,t) := \frac{p_0^*(x)\exp(-\beta h(x)t)}{\int p_0^*(y)\exp(-\beta h(y)t)\,\mathrm{d}y}, \tag{3.1}$$

where β is a positive constant parameter. Under certain additional technical assumptions on h and p_0^*, the density $p^*(x,t)$ weakly converges to the Dirac delta measure at $\bar{x}$ as time $t\to\infty$ (See Appendix B.4). The Bayesian approach is attractive because it can be implemented recursively: Consider a finite time interval $[0,T]$ with an associated discrete-time sequence $\{t_0,t_1,t_2,\ldots,t_{\bar{N}}\}$ of sampling instants, with $0=t_0<t_1<\ldots<t_{\bar{N}}=T$, and increments given by $\Delta t_n := t_n - t_{n-1}, n=1,\ldots,\bar{N}$. The posterior distribution is expressed

*The content of this chapter is related to the publication [207, 204].

recursively as:

$$\begin{aligned} \text{Initialization:} \quad & \rho_0(x) = p_0^*(x), \\ \text{Update:} \quad & \rho_n(x) = \frac{\rho_{n-1}(x)\exp(-\beta h(x)\Delta t_n)}{\int \rho_{n-1}(y)\exp(-\beta h(y)\Delta t_n)\,\mathrm{d}y}, \quad n \geq 1, \end{aligned} \tag{3.2}$$

such that, at time t_n, $\rho_n(x) = p^*(x,t_n)$ by construction.

A conventional particle filter based on sequential importance sampling and resampling (SISR) may be used to sample from ρ_n. A particle filter is comprised of N stochastic processes $\{X_n^i : 1 \leq i \leq N\}$, where $X_n^i \in \mathbb{R}^d$ is the state for the i^{th} particle at iteration n. The SISR particle filter implements the following recursive steps:

$$\begin{aligned} \text{Initialization:} \quad & X_0^i \overset{\text{i.i.d.}}{\sim} p_0^*, \\ \text{Update:} \quad & X_n^i \overset{\text{i.i.d.}}{\sim} \sum_{i=1}^{N} w_n^i \delta_{X_{n-1}^i}, \quad n \geq 1, \end{aligned} \tag{3.3}$$

where $w_n^i \propto \exp(-\beta h(X_{n-1}^i)\Delta t_n)$ are referred to as the importance weights and δ_z denotes the Dirac-delta at $z \in \mathbb{R}^d$. In practice, the importance weights w_n^i can potentially suffer from large variance. To address this problem, several extensions have been described in literature based on consideration of suitable sampling (proposal) distributions and efficient resampling schemes; cf., [46, 47].

The use of probabilistic models to derive recursive sampling algorithms is by now a standard solution approach to the global optimization problem: The model (3.1) appears in [187], and its recursive form (3.2) can be regarded as a closely related variant of the models in [76, 215]. Importance sampling type schemes, of the form (3.3), based on these and more general (stochastic) models appear in [215, 213, 113, 114, 165].

In this chapter, we present an alternate control-based approach – referred to as the controlled particle filter – to the construction and simulation of the particle filter for global optimization. In our approach, the particle filter is a controlled interacting particle system where the dynamics of the i^{th} particle evolve according to

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = u(X_t^i, t), \quad X_0^i \sim p_0^*, \tag{3.4}$$

where the *control function* $u(x,t)$ is obtained by solving a weighted Poisson equation:

$$\begin{aligned} -\nabla\cdot(\rho(x)\nabla\phi(x)) &= (h(x) - \hat{h})\rho(x), \quad x \in \mathbb{R}^d, \\ \int \phi(x)\rho(x)\,\mathrm{d}x &= 0, \end{aligned} \tag{3.5}$$

where $\hat{h} := \int h(x)\rho(x)\,\mathrm{d}x$, ∇ and $\nabla\cdot$ denote the gradient and the divergence operators, respectively, and at time t, $\rho(x) = p(x,t)$ denotes the density of $X_t^{i\dagger}$. In terms of the solution $\phi(x)$ of (3.5), the control function at time t is given by

$$u(x,t) = -\beta\nabla\phi(x). \tag{3.6}$$

Note that the control function u is vector-valued (with dimension $d \times 1$) and it needs to be obtained for each value of time t. The basic results on existence and uniqueness of $\nabla\phi$ will be discussed in Sec. 4.2. These results require additional assumptions on the prior p_0^* and the function h. These assumptions appear at the end of this section. The Poisson equation (3.5) also appears in the feedback particle filter [195] and other related algorithms [148, 44] for nonlinear filtering in the Euclidean space.

The contributions of this chapter are as follows:

• **Variational formulation:** A time-stepping procedure is introduced consisting of successive minimization problems in the space of probability densities. The construction shows the density transport (3.1) may be regarded as a gradient flow, or a steepest descent, for the expected value of the function h, with respect to the Kullback–Leibler divergence. More significantly, the construction is used to motivate a mean-field type optimal control problem. The control law (3.4)-(3.6) for the proposed particle filter represents the solution to this problem. The Poisson equation (3.5) is derived from the first-order analysis of the Bellman's optimality principle. For a discussion on the importance of the variational aspects of nonlinear filtering, see [131] and [100].

• **Quadratic Gaussian cases:** For a quadratic objective function h and a Gaussian prior p_0^*, the solution of the Bayes' model (3.1) is Gaussian with explicit expressions for its mean and covariance. The partial differential equation (pde) (3.5) admits a closed-form solution, and the resulting control law is shown to be affine in the decision variables. This affine control law is also used as an approximation of the control function for the general (non-quadratic) case.

• **Parametric cases:** The quadratic Gaussian problem is an example of the more general parametric case

[†] Although this chapter is limited to $\mathbb{R}^d$, the proposed algorithm is applicable to global optimization problems on differential manifolds, e.g., matrix Lie groups (For an intrinsic form of the Poisson equation, see Chapter 4). For domains with boundary, the pde (1.3) is accompanied by a Neumann boundary condition:

$$\nabla\phi(x)\cdot n(x) = 0$$

for all x on the boundary of the domain where $n(x)$ is a unit normal vector at the boundary point x.

where the density is of a (known) parametrized form. The gradient flow in the parameter space is derived from an optimal control problem that is analogous to the general non-parametric case. The parametric models in this chapter are related to the stochastic approximation type model-based algorithms [78] and the natural gradient algorithm [90].

The two sets of theoretical results in this chapter – the non-parametric results in Sec. 3.2.1 and the parametric results in Sec. 3.2.3 – represent the control counterparts of the non-parametric and the parametric model-based algorithms (see Sec. 1.2.2 for a brief survey of model-based algorithms and Sec. 7.2 for their numerical procedures). The variational analysis serves to provide the connection between these as well as suggest systematic approaches for approximation of the optimal control law (3.6).

The outline of the remainder of this chapter is as follows: The variational aspects of the filter – including the non-parametric and parametric cases – appears in Sec. 3.2. All the proofs are contained in the Appendix.

Notation: The Euclidean space $\mathbb{R}^d$ is equipped with the Borel σ-algebra denoted as $\mathscr{B}(\mathbb{R}^d)$. The space of Borel probability measures on $\mathbb{R}^d$ with finite second moment is denoted as $\mathscr{P}$:

$$\mathscr{P} \doteq \left\{ \rho : \mathbb{R}^d \to [0,\infty) \text{ meas. density} \,\middle|\, \int |x|^2 \rho(x)\,\mathrm{d}x < \infty \right\}.$$

The density for a Gaussian random variable with mean m and variance Σ is denoted as $\mathscr{N}(m,\Sigma)$. For vectors $x, y \in \mathbb{R}^d$, the dot product is denoted as $x \cdot y$ and $|x| := \sqrt{x \cdot x}$; x^T denotes the transpose of the vector. Similarly, for a matrix K, K^T denotes the matrix transpose, and $\mathsf{K} \succ 0$ denotes positive-definiteness. For $l, k \in \mathbb{Z}^+$ (natural numbers), the tensor notation δ_{lk} is used to denote the identity matrix ($\delta_{lk} = 1$ if $l = k$ and 0 otherwise). C^k is used to denote the space of k-times continuously differentiable functions on $\mathbb{R}^d$. For a function f, $\nabla f = \frac{\partial f}{\partial x_i}$ is used to denote the gradient vector, and $D^2 f = \frac{\partial^2 f}{\partial x_i \partial x_j}$ is used to denote the Hessian matrix. L^∞ denotes the space of bounded functions on $\mathbb{R}^d$ with associated norm denoted as $\|\cdot\|_\infty$. $L^2(\mathbb{R}^d;\rho)$ is the Hilbert space of square integrable functions on $\mathbb{R}^d$ equipped with the inner-product, $\langle \phi, \psi \rangle := \int \phi(x)\psi(x)\rho(x)\,\mathrm{d}x$. The associated norm is denoted as $\|\phi\|_2^2 := \langle \phi, \phi \rangle$. The space $H^1(\mathbb{R}^d;\rho)$ is the space of square integrable functions ϕ whose derivative (defined in the weak sense) is in $L^2(\mathbb{R}^d;\rho)$. For a function $\phi \in L^2(\mathbb{R}^d;\rho)$, $\hat{\phi} := \int \phi(x)\rho(x)\,\mathrm{d}x$ denotes the mean. L^2_0 and H^1_0 denote the co-dimension 1 subspaces of functions whose mean is zero.

Assumptions: The following assumptions are made throughout the chapter:

Assumption 3.1 *The prior probability density function $p_0^* \in \mathscr{P}$ and is of the form $p_0^*(x) = e^{-V_0(x)}$ where $V_0 \in C^2$, $D^2V_0 \in L^\infty$, and*

$$\liminf_{|x|\to\infty} \nabla V_0(x) \cdot \frac{x}{|x|} = \infty.$$

Assumption 3.2 *The function $h \in C^2 \cap L^2(\mathbb{R}^d; p_0^*)$ with $D^2h \in L^\infty$ and*

$$\liminf_{|x|\to\infty} \nabla h(x) \cdot \frac{x}{|x|} > -\infty.$$

Assumption 3.3 *The function h has a unique minimizer $\bar{x} \in \mathbb{R}^d$ with minimum value $h(\bar{x}) =: \bar{h}$. Outside some compact set $D \subset \mathbb{R}^d$, $\exists\, r > 0$ such that*

$$h(x) > \bar{h} + r \qquad \forall\, x \in \mathbb{R}^d \setminus D.$$

Remark 3.1 *Assumptions 3.1 and 3.2 are important to prove existence, uniqueness and regularity of the solutions of the Poisson equation (see [100]). Assumptions 3.1 holds for density with Gaussian tails. Assumption 3.3 is used to obtain weak convergence of $p^*(x,t)$ to Dirac delta at $\bar{x}$. The uniqueness of the minimizer $\bar{x}$ can be relaxed to obtain weaker conclusions on convergence (See Appendix B.4).*

3.2 Variational Formulation

3.2.1 Non-parametric case

A variational formulation of the Bayes recursion (3.2) is the following *time-stepping procedure*: For the discrete-time sequence $\{t_0, t_1, t_2, \ldots, t_{\bar{N}}\}$ with increments $\Delta t_n := t_n - t_{n-1}$ (see Sec. 3.1), set $\rho_0 = p_0^* \in \mathscr{P}$ and recursively define $\{\rho_n\}_{n=1}^{\bar{N}} \subset \mathscr{P}$ by taking $\rho_n \in \mathscr{P}$ to minimize the functional

$$\mathsf{I}(\rho|\rho_{n-1}) := \frac{1}{\Delta t_n}\mathsf{D}(\rho \mid \rho_{n-1}) + \beta \int h(x)\rho(x)\,\mathrm{d}x, \tag{3.7}$$

where D denotes the relative entropy or Kullback–Leibler divergence,

$$\mathsf{D}(\rho \mid \rho_{n-1}) := \int \rho(x) \ln\Big(\frac{\rho(x)}{\rho_{n-1}(x)}\Big)\,\mathrm{d}x.$$

The proof that ρ_n, as defined in (3.2), is in fact the minimizer is straightforward: By Jensen's formula, $\mathsf{I}(\rho|\rho_{n-1}) \geq -\ln(\int \rho_{n-1}(y)\exp(-h(y)\Delta t_n)\,\mathrm{d}y)$ with equality if and only if $\rho = \rho_n$. Although the optimizer is known, a careful look at the first order optimality equations associated with ρ_n leads to i) the replicator dynamics pde for the gradient flow (in Theorem 3.1), and ii) the proposed particle filter algorithm for approximation of the posterior (in Theorems 3.2 and 3.3).

The sequence of minimizers $\{\rho_n\}_{n=0}^{\bar{N}}$ is used to construct, via interpolation, a density function $\rho^{(\bar{N})}(x,t)$ for $t \in [0,T]$: Define $\rho^{(\bar{N})}(x,t)$ by setting

$$\rho^{(\bar{N})}(x,t) := \rho_n(x), \quad \text{for } t \in [t_n, t_{n+1})$$

for $n = 0,1,2,\ldots,\bar{N}-1$. The proof of the following theorem appears in Appendix B.1.

Theorem 3.1 (Gradient flow) *In the limit as $\bar{N} \to \infty$ the density $\rho^{(\bar{N})}(x,t)$ converges pointwise to the density $\rho(x,t)$ which is a weak solution of of the following replicator dynamics pde:*

$$\frac{\partial \rho}{\partial t}(x,t) = -\beta\,(h(x) - \hat{h}_t)\,\rho(x,t), \quad \rho(x,0) = p_0^*(x). \tag{3.8}$$

To construct the particle filter, the key idea is to view the gradient flow time-stepping procedure as a dynamic programming (DP) recursion from time $t_{n-1} \to t_n$:

$$\rho_n = \operatorname*{argmin}_{\rho^{(u)} \in \mathscr{P}} \underbrace{\frac{1}{\Delta t_n}\mathsf{D}(\rho^{(u)}|\rho_{n-1})}_{\text{control cost}} + V(\rho^{(u)}),$$

where $V(\rho^{(u)}) := \beta \int \rho^{(u)}(x)h(x)\,\mathrm{d}x$ is the cost-to-go. The notation $\rho^{(u)}$ for density corresponds to the following construction: Consider the differential equation

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = u(X_t^i, t)$$

and denote the associated flow from $t_{n-1} \to t_n$ as $x \mapsto s_n(x)$. Under suitable assumptions on u (Lipschitz in x and continuous in t), the flow map s_n is a well-defined diffeomorphism on $\mathbb{R}^d$ and $\rho^{(u)} := s_n^{\#}(\rho_{n-1})$, where

$s_n^{\#}$ denotes the push-forward operator. The push-forward of a probability density ρ by a smooth map s is defined through the change-of-variables formula

$$\int f(x)[s^{\#}(\rho)](x)\,\mathrm{d}x = \int f(s(x))\rho(x)\,\mathrm{d}x$$

for all continuous and bounded test functions f.

Via a formal but straightforward calculation, in the asymptotic limit as $\Delta t_n \to 0$, the control cost is expressed in terms of the control u as

$$\frac{1}{\Delta t_n}\mathsf{D}(\rho^{(u)}|\rho_{n-1}) = \frac{\Delta t_n}{2}\int \left|\frac{1}{\rho_{n-1}}\nabla\cdot(\rho_{n-1}u)\right|^2 \rho_{n-1}\,\mathrm{d}x + o(\Delta t_n). \tag{3.9}$$

These considerations help motivate the following optimal control problem:

$$\begin{aligned} \underset{u}{\text{Minimize:}} \quad & J(u) = \int_0^T \mathsf{L}(\rho_t, u_t)\,\mathrm{d}t + \beta\int h(x)\rho_T(x)\,\mathrm{d}x \\ \text{Constraint:} \quad & \frac{\partial \rho_t}{\partial t} + \nabla\cdot(\rho_t u_t) = 0, \quad \rho_0(x) = p_0^*(x), \end{aligned} \tag{3.10}$$

where the Lagrangian is defined as

$$\mathsf{L}(\rho,u) := \frac{1}{2}\int_{\mathbb{R}^d}\left|\frac{1}{\rho(x)}\nabla\cdot(\rho(x)u(x))\right|^2 \rho(x)\,\mathrm{d}x + \frac{\beta^2}{2}\int_{\mathbb{R}^d}|h(x)-\hat{h}|^2\,\rho(x)\,\mathrm{d}x,$$

where $\hat{h} := \int h(x)\rho(x)\,\mathrm{d}x$.

The Hamiltonian is defined as

$$\mathsf{H}(\rho,\mathsf{q},u) := \mathsf{L}(\rho,u) - \int \mathsf{q}(x)\nabla\cdot(\rho(x)u(x))\,\mathrm{d}x \tag{3.11}$$

where q is referred to as the momentum (or co-state).

Suppose $\rho_t \in \mathscr{P}$ is the density at time t. The value function is defined as

$$V(\rho,t) := \inf_u \left[\int_t^T \mathsf{L}(\rho_s,u_s)\,\mathrm{d}s + \beta\int h\rho_T\,\mathrm{d}x\right]. \tag{3.12}$$

The value function is a functional on the space of densities. For a fixed $\rho \in \mathscr{P}$ and time $t \in [0,T)$, the

(Gâteaux) derivative of V is a function on $\mathbb{R}^d$, and an element of the function space $L^2(\mathbb{R}^d;\rho)$. This function is denoted as $\frac{\partial V}{\partial \rho}(\rho,t)(x)$ for $x \in \mathbb{R}^d$. Additional details appear in the Appendix B.2 where the following Theorem is proved.

Theorem 3.2 (Finite-horizon optimal control) *Consider the optimal control problem* (3.10) *with the value function defined in* (3.12)*. Then V solves the following DP equation:*

$$\frac{\partial V}{\partial t}(\rho,t) + \inf_{u \in L^2} \mathsf{H}(\rho, \frac{\partial V}{\partial \rho}(\rho,t), u) = 0, \quad t \in [0,T),$$
$$V(\rho,T) = \beta \int h(x)\rho(x)\,\mathrm{d}x.$$

The solution of the DP equation is given by

$$V(\rho,t) = \beta \int_{\mathbb{R}^d} h(x)\rho(x)\,\mathrm{d}x,$$

and the associated optimal control is a solution of the following pde:

$$\frac{1}{\rho(x)} \nabla \cdot (\rho(x)u(x)) = \beta(h(x) - \hat{h}), \quad \forall\, x \in \mathbb{R}^d. \tag{3.13}$$

It is also useful to consider the following infinite-horizon version of the optimal control problem:

$$\begin{aligned} \text{Minimize:} \quad & J(u) = \int_0^\infty \mathsf{L}(\rho_t, u_t)\,\mathrm{d}t \\ \text{Constraints:} \quad & \begin{cases} \frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t u_t) = 0, \quad \rho_0(x) = p_0^*(x), \\ \lim_{t\to\infty} \int h(x)\rho_t(x) = h(\bar{x}). \end{cases} \end{aligned} \tag{3.14}$$

For this problem, the value function is defined as

$$V(\rho) = \inf_u J(u). \tag{3.15}$$

The solution is given by the following Theorem whose proof appears in Appendix B.2:

Theorem 3.3 (Infinite-horizon optimal control) *Consider the infinite horizon optimal control problem*

(3.14) *with the value function defined in* (3.15)*. The value function is given by*

$$V(\rho) = \beta \int_{\mathbb{R}^d} h(x)\rho(x)\,\mathrm{d}x - \beta h(\bar{x})$$

and the associated optimal control law is a solution of the pde (3.13)*.*

The particle filter algorithm (3.4)-(3.6) in Sec. 3.1 is obtained by *additionally* requiring the solution u of (3.13) to be of the gradient form. One of the advantages of doing so is that the optimizing control law, obtained instead as solution of (3.5), is uniquely defined (See Theorem 4.1 in Sec. 4.2). In part, this choice is guided by the L^2 optimality of the gradient form solution (The proof appears in the Appendix B.2):

Lemma 3.1 (L^2 optimality) *Consider the pde* (3.13) *where* ρ *and* h *satisfy Assumptions (A1)-(A2). The general solution is given by*

$$u = -\beta\nabla\phi + v,$$

where ϕ *is the solution of* (3.5)*,* v *solves* $\nabla\cdot(\rho v) = 0$*, and*

$$\|u\|_2^2 = \beta^2\|\nabla\phi\|_2^2 + \|v\|_2^2.$$

That is, $u = -\beta\nabla\phi$ *is the minimum* L^2*-norm solution of* (3.13)*.*

Remark 3.2 *In Appendix B.3, the Pontryagin's minimum principle of optimal control is used to express the particle filter* (3.4)-(3.6) *in its Hamilton's form:*

$$\boxed{\begin{aligned} \frac{\mathrm{d}X_t^i}{\mathrm{d}t} &= u(X_t^i,t), \quad X_0^i \sim p_0^* \\ 0 \equiv \mathsf{H}(p(\cdot,t),\beta h,u(\cdot,t)) &= \min_{v\in L^2} \mathsf{H}(p(\cdot,t),\beta h,v) \end{aligned}}$$

The Poisson equation (3.5) *is simply the first order optimality condition to obtain a minimizing control. Under this optimal control, the density* $p(x,t)$ *is the optimal trajectory. The associated optimal trajectory for the momentum (co-state) is a constant equal to its terminal value* $\beta h(x)$*.*

The following theorem shows that the particle filter implements the Bayes' transport of the density, and

establishes the asymptotic convergence for the density (The proof appears in the Appendix (B.4)). We recall the notation for the two types of density in our analysis:

1. $p(x,t)$: Defines the density of X_t^i.

2. $p^*(x,t)$: The Bayes' density given by (3.1).

Theorem 3.4 (Bayes' exactness and convergence) *Consider the particle filter* (3.4)-(3.6). *If* $p(\cdot,0) = p^*(\cdot,0)$, *we have for all* $t \geq 0$,

$$p(\cdot,t) = p^*(\cdot,t).$$

As $t \to \infty$, $\int h(x)p(x,t)\,\mathrm{d}x$ *decreases monotonically to* $h(\bar{x})$ *and* $X_t^i \to \bar{x}$ *in probability.*

The hard part of implementing the controlled particle filter is solving the Poisson equation (3.5). For the quadratic Gaussian case – where the objective function h is quadratic and the prior p_0^* is Gaussian – the solution can be obtained in an explicit form. This is the subject of the Sec. 3.2.2. In the quadratic Gaussian case, the infinite-dimensional particle filter can be replaced by a finite-dimensional filter involving only the mean and the variance of the Gaussian density. The simplification arises because the density admits a parameterized form. A more general version of this result – finite-dimensional filters for general class of parametrized densities – is the subject of Sec. 3.2.3. For the general case where a parametric form of density is not available, numerical algorithms for approximating the control function solution appear in Chapter 4.

Remark 3.3 *In the construction of the time-stepping procedure* (3.7), *we considered a gradient flow with respect to the divergence metric. In the optimal transportation literature, the Wasserstein metric is widely used. In the conference version of this chapter [204], it is shown that the limiting density with the Wasserstein metric evolves according to the Liouville equation [24]:*

$$\frac{\partial \rho}{\partial t}(x,t) = \nabla \cdot (\rho(x,t)\nabla h(x)).$$

The particle filter is the gradient descent algorithm:

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = -\nabla h(X_t^i).$$

The divergence metric is chosen here because of the Bayesian nature of the resulting solution.

3.2.2 Quadratic Gaussian case

For the quadratic Gaussian problem, the solution of the Poisson equation can be obtained in an explicit form as described in the following Lemma. The proof appears in the Appendix B.5.

Lemma 3.2 *Consider the Poisson equation (3.5). Suppose the objective function h is a quadratic function such that $h(x) \to \infty$ as $|x| \to \infty$ and the density ρ is a Gaussian with mean m and variance Σ. Then the control function*

$$u(x) = -\beta \nabla \phi(x) = -\beta \mathsf{K}(x-m) - \beta b, \tag{3.16}$$

where the affine constant vector

$$b = \int x(h(x) - \hat{h})\rho(x)\,\mathrm{d}x, \tag{3.17}$$

and the gain matrix $\mathsf{K} = \mathsf{K}^T \succ 0$ is the solution of the Lyapunov equation:

$$\Sigma \mathsf{K} + \mathsf{K}\Sigma = \int (x-m)(x-m)^T (h(x) - \hat{h})\rho(x)\,\mathrm{d}x. \tag{3.18}$$

Using an affine control law (3.16), it is straightforward to verify that $p(x,t) = p^*(x,t)$ is a Gaussian whose mean $m_t \to \bar{x}$ and variance $\Sigma_t \to 0$. The proofs of the following Proposition and the Corollary appear in the Appendix B.5:

Proposition 3.1 *Consider the particle filter (3.4) with the affine control law (3.16). Suppose the objective function h is a quadratic function such that $h(x) \to \infty$ as $|x| \to \infty$ and the prior density p_0^* is a Gaussian with mean m_0 and variance Σ_0. Then the posterior density p is a Gaussian whose mean m_t and variance Σ_t evolve according to*

$$\begin{aligned} \frac{\mathrm{d}m_t}{\mathrm{d}t} &= -\beta \mathsf{E}\left[X_t^i (h(X_t^i) - \hat{h}_t)\right], \\ \frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} &= -\beta \mathsf{E}\left[(X_t^i - m_t)(X_t^i - m_t)^T (h(X_t^i) - \hat{h}_t)\right], \end{aligned} \tag{3.19}$$

where $\hat{h}_t := \mathsf{E}[h(X_t^i)]$.

Corollary 3.1 *Under the hypothesis of Proposition 3.1, with an explicit form for quadratic objective function* $h(x) = \frac{1}{2}(x-\bar{x})^T H(x-\bar{x}) + c$ *where* $H = H^T \succ 0$*, the expectations on the righthand-side of* (3.19) *are computed in closed-form and the resulting evolution is given by*

$$\frac{\mathrm{d}m_t}{\mathrm{d}t} = \beta \Sigma_t H(\bar{x} - m_t), \tag{3.20a}$$

$$\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} = -\beta \Sigma_t H \Sigma_t, \tag{3.20b}$$

whose explicit solution is given by

$$\begin{aligned} m_t &= m_0 + \Sigma_0 S_t^{-1}(\bar{x} - m_0), \\ \Sigma_t &= \Sigma_0 - \Sigma_0 S_t^{-1} \Sigma_0, \end{aligned} \tag{3.21}$$

where $S_t := \frac{1}{\beta t} H^{-1} + \Sigma_0$ *for* $t > 0$*. In particular,* $m_t \to \bar{x}$ *and* $\Sigma_t \to 0$*.*

In practice, the affine control law (3.16) is implemented as:

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = -\beta \mathsf{K}_t^{(N)}(X_t^i - m_t^{(N)}) - \beta b_t^{(N)} =: u_t^i, \tag{3.22}$$

where the terms are approximated empirically from the particle ensemble $\{X_t^i\}_{i=1}^N$. The algorithm appears in Algorithm 3 (the dependence on time t is suppressed). An alternative construction of the affine control law is presented in Example 4.3 in Sec. 4.3.1.

As $N \to \infty$, the approximations become exact and (3.16) represents the mean-field limit of the finite-N control in (3.22). Consequently, the empirical distribution of the ensemble approximates the posterior distribution (density) $p^*(x,t)$.

Remark 3.4 *The finite-dimensional system* (3.19) *is the optimization counterpart of the Kalman filter. Likewise the particle filter* (3.22) *is the counterpart of the ensemble Kalman filter [56]. While the affine control law* (3.16) *is optimal for the quadratic Gaussian case, it can be implemented for more general non-quadratic non-Gaussian settings - as long as the various approximations in Algorithm 3 can be obtained at each step. The situation is analogous to the filtering setup where the Kalman filter is often used as an approximate algorithm even in nonlinear non-Gaussian settings.*

Algorithm 3 Affine approximation of the control function

1: **Input:** $\{X^i\}_{i=1}^N$, parameter β

2: Calculate $m^{(N)} := \frac{1}{N}\sum_{i=1}^N X^i$,

3: Calculate $\Sigma^{(N)} := \frac{1}{N}\sum_{i=1}^N \left(X^i - m^{(N)}\right)\left(X^i - m^{(N)}\right)^T$

4: Calculate $\hat{h}^{(N)} := \frac{1}{N}\sum_{i=1}^N h(X^i)$

5: Calculate $b^{(N)} := \frac{1}{N}\sum_{i=1}^N X^i \left(h(X^i) - \hat{h}^{(N)}\right)$

6: Calculate

$$C^{(N)} := \frac{1}{N}\sum_{i=1}^N \left(X^i - m^{(N)}\right)\left(X^i - m^{(N)}\right)^T \left(h(X^i) - \hat{h}^{(N)}\right)$$

7: Calculate $\mathsf{K}^{(N)}$ by solving $\Sigma^{(N)}\mathsf{K}^{(N)} + \mathsf{K}^{(N)}\Sigma^{(N)} = C^{(N)}$

8: Calculate $u^i = -\beta\mathsf{K}^{(N)}(X^i - m^{(N)}) - \beta b^{(N)}$

9: **Output:** $\{u^i\}_{i=1}^N$

3.2.3 Parametric case

Consider next the case where the density has a known parametric form,

$$p(x,t) = \varrho\left(x;\theta_t\right), \tag{3.23}$$

where $\theta_t \in \mathbb{R}^M$ is the parameter vector. For example, in the quadratic Gaussian problem, ϱ is a Gaussian with parameters m_t and Σ_t.

For the parametric density $\varrho\left(x;\vartheta\right)$, $\frac{\partial}{\partial\vartheta}\left(\log\varrho\left(x;\vartheta\right)\right)$ is a $M\times 1$ column vector whose k^{th} entry,

$$\left[\frac{\partial}{\partial\vartheta}\left(\log\varrho\left(x;\vartheta\right)\right)\right]_k = \frac{\partial}{\partial\vartheta_k}\left(\log\varrho\left(x;\vartheta\right)\right),$$

for $k = 1,\ldots,M$.

The Fisher information matrix is a $M\times M$ matrix:

$$G_{(\vartheta)} := \int \frac{\partial}{\partial\vartheta}\left(\log\varrho\left(x;\vartheta\right)\right)\left[\frac{\partial}{\partial\vartheta}\left(\log\varrho\left(x;\vartheta\right)\right)\right]^T \varrho\left(x;\vartheta\right)\mathrm{d}x. \tag{3.24}$$

By construction, $G_{(\vartheta)}$ is symmetric and positive semidefinite. In the following, it is furthermore assumed

that $G_{(\vartheta)}$ is strictly positive definite, and thus invertible, for all $\vartheta \in \mathbb{R}^M$.

In terms of the parameter,

$$e(\vartheta) := \int h(x)\, \varrho\,(x;\vartheta)\, \mathrm{d}x,$$

and its gradient is a $M \times 1$ column vector:

$$\nabla e(\vartheta) = \int h(x) \frac{\partial}{\partial \vartheta} \left(\log \varrho\,(x;\vartheta)\right) \varrho\,(x;\vartheta)\, \mathrm{d}x. \tag{3.25}$$

We are now prepared to describe the induced evolution for the parameter vector θ_t. The proof of the following proposition appears in the Appendix B.6.

Proposition 3.2 *Consider the particle filter (3.4)-(3.6). Suppose the density admits the parametric form (3.23) whose Fisher information matrix, defined in (3.24), is assumed to be invertible. Then the parameter vector θ_t is a solution of the following ordinary differential equation,*

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\beta G_{(\theta_t)}^{-1} \nabla e(\theta_t). \tag{3.26}$$

Remark 3.5 *The filter (3.26) is referred to as the natural gradient; cf., [90]. There are several variational interpretations:*

(i) *The filter can be obtained via a time stepping procedure, analogous to (3.7). The sequence $\{\theta_n\}_{n=1}^N$ is inductively defined as a minimizer of the function,*

$$\mathsf{I}(\theta|\theta_{n-1}) := \left[\frac{1}{\Delta t_n} \mathsf{D}(\varrho\,(\cdot;\theta)|\, \varrho\,(\cdot;\theta_{n-1})) + \beta e(\theta)\right].$$

On taking the limit as $\Delta t_n \to 0$, one arrives at the filter (3.26).

(ii) *The optimal control interpretation of (3.26) is based on the Pontryagin's minimum principle (see also Remark 3.2). For the finite-dimensional problem, the Hamiltonian*

$$\mathsf{H}(\theta, \mathsf{q}, u) = \mathsf{L}(\theta, u) + \mathsf{q} \cdot u,$$

where $\mathsf{q} \in \mathbb{R}^M$ *is the momentum. With* $\dot{\theta} = u$*, the counterpart of* (3.9) *is*

$$\frac{1}{\Delta t_n} \mathsf{D}(\varrho^{(u)}(\cdot;\theta) | \, \varrho(\cdot;\theta_{n-1})) = \frac{1}{2} u^T G_{(\theta)} u + o(\Delta t_n).$$

With $\frac{1}{2} u^T G_{(\theta)} u$ *as the control cost component in the Lagrangian, the first order optimality condition gives*

$$G_{(\theta)} u = -\mathsf{q} = -\beta \nabla e(\theta),$$

where we have used the fact that $\beta e(\theta)$ *is the value function. Note that it was not necessary to write the explicit form of the Lagrangian to obtain the optimal control.*

(iii) *Finally, the filter* (3.26) *represents the gradient flow (in* $\mathbb{R}^M$*) for the objective function* $e(\theta)$ *with respect to the Riemannian metric* $\langle v, w \rangle_\theta = v^T G_{(\theta)} w$ *for all* $v, w \in \mathbb{R}^M$.

Example 3.1 *In the quadratic Gaussian case, the natural gradient algorithm* (3.26) *with parameters* m_t *and* Σ_t *reduces to* (3.19).

Remark 3.6 *While the systems* (3.26) *and* (3.19) *are finite-dimensional, the righthand-sides will still need to be approximated empirically. The convergence properties of a class of related algorithms is studied using a stochastic approximation framework in [78].*

The stochastic approximation is not necessary if the problem admits a certain affine structure in the parameters:

Example 3.2 *Suppose the density is of the following exponential parametric form:*

$$\varrho(x;\vartheta) = \frac{\exp(\vartheta \cdot \psi(x))}{\int \exp(\vartheta \cdot \psi(y)) \, \mathrm{d}y},$$

where $\vartheta \in \mathbb{R}^M$*, and* $\psi(x) := (\psi_1(x), \psi_2(x), \ldots, \psi_M(x))$ *is a given set of linearly independent (basis) functions, expressed here as a vector. Furthermore, suppose* h *is expressed as a linear combination of these functions:*

$$h(x) = \alpha \cdot \psi(x),$$

where $\alpha \in \mathbb{R}^M$.

The elements of the Fisher information matrix (3.24) *and the gradient* (3.25) *are given by the respective formulae:*

$$[G]_{lk}(\theta) = \int (\psi_l(x) - \hat{\psi}_l)(\psi_k(x) - \hat{\psi}_k)\, \varrho\,(x;\theta)\, \mathrm{d}x,$$
$$[\nabla e]_k(\theta) = \int (\alpha \cdot \psi(x))\, (\psi_k(x) - \hat{\psi}_k)\, \varrho\,(x;\theta)\, \mathrm{d}x,$$

where $\hat{\psi}_k := \int \psi_k(x)\, \varrho\,(x;\theta)\, \mathrm{d}x$. *The ode* (3.26) *simplifies to*

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\beta\,\alpha.$$

Although interesting, there do not appear to be any non-trivial examples where the affine structure applies.

3.3 Conclusions

In this chapter, a controlled particle filter was introduced as an algorithmic solution to the global optimization problem. The main conclusions are as follows:

1) Two types of variational interpretations were provided for the filter: (i) the density transport was shown to be a gradient flow for the expected value of the objective function; (ii) the control law was shown to be a solution of a mean-field type optimal control problem.

2) For the special case of quadratic objective function and Gaussian prior distribution, closed-form formulae for the optimal control law were obtained and shown to be closely related to the ensemble Kalman filter for the linear Gaussian filtering problem.

3) Comparison with other types of non-parametric and parametric algorithms were discussed.

There are a number of possible directions for future research:

1) Characteristic conditions on the objective function h (e.g., convexity) such that the affine control law yields the optimal solution.

2) Convergence analysis of the finite-N system (3.22) for the quadratic Gaussian problem.

Chapter 4

Poisson equation on Matrix Lie groups *

4.1 Introduction

Certain Poisson equations appear in both the feedback particle filter on matrix Lie groups (equation (2.22) and (2.28) in Chapter 2) and the controlled particle filter in the Euclidean space (equation (3.5) in Chapter 3). This chapter concerns the well-posedness of the Poisson equation and numerical methods to approximate its solution. The majority of the numerical algorithms in this chapter are described for matrix Lie groups. Specialization of these algorithms to the Euclidean space is straightforward and will be discussed as needed.

Recall from Sec. 2.3.1 that the (weak form) Poisson equation in the feedback particle filter on a matrix Lie group is given by,

$$\begin{aligned} &\pi\big(\langle \mathrm{grad}(\phi), \mathrm{grad}(\psi)\rangle\big) = \pi\big((h-\hat{h})\psi\big), \\ &\pi(\phi) = 0 \quad \text{(normalization)}, \end{aligned} \tag{4.1}$$

for all $\psi \in H^1(G;\pi)$, where the solution $\phi \in H^1_0(G;\pi)$, $h : G \to \mathbb{R}$ is a given function, and π denotes the particle distribution. The strong form of this Poisson equation is given by,

$$\Delta_\rho \phi = -(h-\hat{h}), \tag{4.2}$$

where $\Delta_\rho := \frac{1}{\rho}\mathrm{div}\big(\rho\,\mathrm{grad}(\phi)\big)$ denotes the weighted Laplacian operator, and ρ denotes the density function associated with π. Compared with its general form (2.22), the explicit dependence on j and t is suppressed in the Poisson equation (4.1) since the equations for each $j = 1,\ldots,m$ are uncoupled and the time t is fixed.

The numerical problem of gain function approximation is as follows: Given samples $\{X^1,\ldots,X^i,\ldots,X^N\}$ drawn i.i.d. from the distribution π, approximate the solution $\{\mathsf{K}(X^1),\ldots,\mathsf{K}(X^i),\ldots,\mathsf{K}(X^N)\}$, where accord-

*The content of this chapter is related to the publication [208, 209].

ing to (2.23),

$$X^i \mathsf{K}(X^i) = \text{grad}(\phi)(X^i)$$

for $i = 1, ..., N$. The density function of ρ is not explicitly known.

Two numerical schemes for approximating the solution of (4.2) are described in this chapter:

1) **Galerkin algorithm:**

The Galerkin solution is the best least square approximation of the solution ϕ of (4.1) in a finite-dimensional subspace of the function space $H_0^1(G;\pi)$. The subspace is defined as a span of a prescribed set of basis functions. Numerically, the algorithm entails solving a matrix equation where the entries of the matrices are approximated empirically using the particles.

2) **Kernel-based algorithm:**

In the kernel-based algorithm, the solution ϕ of (4.2) is equivalently expressed as a solution of a fixed-point problem defined in terms of the semigroup $e^{t\Delta_\rho}$ of the weighted Laplacian Δ_ρ. In a numerical implementation, the semigroup is approximated as a Markov matrix defined on the discrete graph associated with the particles $\{X^i\}_{i=1}^N$. The entries of the Markov matrix are approximated empirically using the particles.

The remainder of this chapter is organized as follows: In Sec. 4.2, the well-posedness results of the Poisson equation are presented. In Sec. 4.3, the Galerkin scheme is described, together with formulae for basis functions for $SO(3)$ and the Euclidean space. Sec. 4.4 contains the numerical procedure for the kernel-based gain function approximation. Related approaches are discussed in Sec. 4.5.

4.2 Well-posedness and Admissibility of the Gain

The admissibility of the gain function solution, i.e., $\mathsf{E}[\left(\sum_n |\mathsf{u}_n(X^i)|\right)] < \infty$ and $\mathsf{E}[\sum_n |\mathsf{k}_{n,j}(X^i)|^2] < \infty$, requires a well-posedness analysis of the Poisson equation. Similar as in the original Euclidean setting [194], we make the following assumptions:

Assumption 4.1 *The function $h \in L^2(G;\pi)$.*

Assumption 4.2 *The distribution π admits a uniform spectral gap (or Poincaré inequality) with constant $\bar{\lambda}$ (Sec. 4.2 in [6]): That is, for a function $\phi \in H_0^1(G;\pi)$,*

$$\pi\big(|\phi|^2\big) \leq \frac{1}{\bar{\lambda}}\,\pi\big(|grad(\phi)|_G^2\big). \tag{4.3}$$

The proof of the following well-posedness theorem appears in Appendix C.1. The proof is nearly identical to the proof presented in [194] for the Euclidean case.

Theorem 4.1 *Under Assumption 4.1 and Assumption 4.2, the Poisson equation (2.22) possesses a unique solution $\phi \in H_0^1(G;\pi)$, satisfying*

$$\pi\big(|grad(\phi)|_G^2\big) \leq \frac{1}{\bar{\lambda}}\,\pi\big(|h-\hat{h}|^2\big). \tag{4.4}$$

For this solution, one has the following bounds,

$$\pi\big(|\mathsf{K}|_{\mathscr{G}}^2\big) \leq \frac{1}{\bar{\lambda}}\,\pi\big(|h-\hat{h}|^2\big), \tag{4.5}$$

$$\pi\Big(\sum_{n=1}^{d}|\mathsf{u}_n|\Big) \leq C\,\pi\big(|h|^2\big), \tag{4.6}$$

where the constant C depends on $\bar{\lambda}$. That is, the resulting gain and control functions are admissible according to Definition 2.1.

The norms $|\cdot|_G$ and $|\cdot|_{\mathscr{G}}$ are defined in (2.9) in Sec. 2.2.

Remark 4.1 (Remark on Assumptions A1-A2) *Suppose the Lie group G is compact, e.g., $SO(3)$. In this case, if π has an everywhere positive density ρ, then Assumption 4.1 and 4.2 automatically hold. For non compact manifolds, e.g., $SE(3)$, the assumptions hold if the density ρ has a Gaussian tail (see Remark 2 in [194]).*

The main challenge to implement the FPF algorithm is to approximate the gain function solution. Since the problem (2.22) is linear, the approximation involves constructing a matrix problem to obtain the approximate solution. In the following two sections, two numerical schemes for the approximation are presented.

4.3 Galerkin Gain Function Approximation

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

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

where $\{\psi_l\}_{l=1}^{L} \subset H_0^1(G;\pi)$ is a given (assumed) set of *basis functions* on G. Denote $S := \text{span}\{\psi_1, ..., \psi_L\}$. The finite-dimensional approximation of the Poisson equation (2.22) is to choose coefficients $\{\kappa_l\}_{l=1}^{L}$ such that,

$$\pi\big(\langle \text{grad}(\phi^{(L)}), \text{grad}(\psi)\rangle\big) = \pi\big((h-\hat{h})\psi\big), \tag{4.7}$$

for all $\psi \in H^1(G;\pi)$. On taking $\psi = \psi_1, ..., \psi_L$, (4.7) is compactly written as a linear matrix equation,

$$A\kappa = b, \tag{4.8}$$

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

$$A_{kl} = \pi\big(\langle \text{grad}(\psi_l), \text{grad}(\psi_k)\rangle\big),$$
$$b_k = \pi\big((h-\hat{h})\psi_k\big),$$

In numerical implementations with a finite set of particles $\{X^i\}_{i=1}^{N}$ sampled from the distribution π, the empirical approximation of (4.8) is denoted as

$$A^{(N)}\kappa^{(N)} = b^{(N)},$$

where the entries of $A^{(N)}$ and $b^{(N)}$ are given by,

$$A_{kl}^{(N)} = \frac{1}{N}\sum_{i=1}^{N} \langle \text{grad}(\psi_l), \text{grad}(\psi_k)\rangle(X^i) \overset{(2.8)}{=} \frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{d} (E_n \cdot \psi_l)(X^i)\,(E_n \cdot \psi_k)(X^i),$$
$$b_k^{(N)} = \frac{1}{N}\sum_{i=1}^{N} (h(X^i) - \hat{h}^{(N)})\psi_k(X^i),$$

Algorithm 4 Galerkin gain function approximation

Input: Particles $\{X^i\}_{i=1}^N$ sampled i.i.d. from π, basis functions $\{\psi_l\}_{l=1}^L$

Calculate $\hat{h}^{(N)} = \frac{1}{N}\sum_{i=1}^N h(X^i)$

for $k = 1$ to L **do**

  Calculate $b_k^{(N)} = \frac{1}{N}\sum_{i=1}^N \left(h(X^i) - \hat{h}^{(N)}\right)\psi_k(X^i)$

  for $l = 1$ to L **do**

    Calculate $A_{kl}^{(N)} = \frac{1}{N}\sum_{i=1}^N \sum_{n=1}^d (E_n \cdot \psi_l)(X^i)\,(E_n \cdot \psi_k)(X^i)$

  end for

end for

Solve the matrix equation $A^{(N)}\kappa^{(N)} = b^{(N)}$, with $A^{(N)} = [A_{kl}^{(N)}]$, $b^{(N)} = [b_k^{(N)}]$

Calculate $\mathsf{k}_n^{(L,N)}(X^i) = \sum_{l=1}^L \kappa_l^{(N)} E_n \cdot \psi_l(X^i)$, for $n = 1, ..., d$

Return: Coordinates $\left\{\left(\mathsf{k}_n^{(L,N)}(X^1), ..., \mathsf{k}_n^{(L,N)}(X^N)\right)\right\}_{n=1}^d$

where $\hat{h}^{(N)} := \frac{1}{N}\sum_{i=1}^N h(X^i)$. The solution ϕ is then empirically approximated by the function

$$\phi^{(L,N)} := \sum_{l=1}^L \kappa_l^{(N)} \psi_l,$$

and the gain function K is empirically approximated by the function $\mathsf{K}^{(L,N)} \in \mathscr{G}$ whose coordinates with respect to a basis $\{E_n\}_{n=1}^d$ of $\mathscr{G}$ are given by

$$\mathsf{k}_n^{(L,N)} = \sum_{l=1}^L \kappa_l^{(N)} E_n \cdot \psi_l, \quad n = 1, ..., d.$$

The numerical procedure of the Galerkin gain function approximation is tabulated in Algorithm 4.

On a compact Lie group where the density ρ is everywhere positive, the following proposition provides error bounds of the Galerkin approximation for the special case where the basis functions are chosen to be the eigenfunctions of the weighted Laplacian Δ_ρ. The proof appears in Appendix C.2. In the proposition statement and proof, the following notation is used: For two functions $\phi, \psi : G \to \mathbb{R}$, their inner product in $L^2(G;\pi)$ is defined as $\langle \phi, \psi \rangle := \pi(\phi\psi) = \int_G \phi(x)\,\psi(x)\,\rho(x)\,\mathrm{d}x$, and the induced norm $\|\phi\|_2 := \sqrt{\langle \phi, \phi \rangle}$. Let the gradient vector field $\mathrm{grad}(\phi)(x) = x\mathsf{K}(x)$ where $\mathsf{K}(x) \in \mathscr{G}$. Define $\|\mathrm{grad}(\phi)\|_G := \sqrt{\pi(|\mathrm{grad}(\phi)|_G^2)}$.

Define also $\|\mathsf{K}\|_{\mathscr{G}} := \sqrt{\pi(|\mathsf{K}|^2_{\mathscr{G}})}$. The norms $|\cdot|_G$ and $|\cdot|_{\mathscr{G}}$ are defined in Sec. 2.2.1, where the definition (2.8) also implies $\|\mathsf{K}\|_{\mathscr{G}} = \|\mathrm{grad}(\phi)\|_G$ due to the correspondence between $\mathrm{grad}(\phi)$ and K.

Proposition 4.1 *Consider the empirical Galerkin approximation of the Poisson equation (4.2) on the space $S := span\{e_1, e_2, \ldots, e_L\}$, where the basis functions of S consist of the first L eigenfunctions of Δ_ρ. Fix $L < \infty$. Suppose the Lie group G is compact and the density ρ is everywhere positive. Then there exists a unique solution for the matrix equation (4.8), and there is a sequence of random variables $\{\varepsilon_N\}$ such that*

$$\|\mathsf{K} - \mathsf{K}^{(L,N)}\|_{\mathscr{G}} \leq \frac{1}{\sqrt{\lambda_L}} \|\tilde{h} - \Pi_S \tilde{h}\|_2 + \varepsilon_N, \tag{4.9}$$

where $\varepsilon_N \to 0$ as $N \to \infty$ a.s, $\tilde{h} := h - \hat{h}$, and $\Pi_S \tilde{h} := \sum_{l=1}^{L} \langle e_l, \tilde{h} \rangle e_l$ is the projection of $\tilde{h}$ onto S.

Remark 4.2 (Variational interpretation) *Suppose ϕ is the exact solution of the weak form of the Poisson equation (2.22). The Galerkin solution $\phi^{(L)}$ is the optimal least-square approximation of ϕ in $S \subset H^1_0(G;\rho)$, i.e,*

$$\phi^{(L)} = \arg\min_{\psi \in S} \ \|grad(\phi) - grad(\psi)\|_G,$$

where $grad(\phi) - grad(\psi)$ is understood as a vector field defined as

$$(grad(\phi) - grad(\psi))(x) := grad(\phi)(x) - grad(\psi)(x) \in T_x G$$

for all $x \in G$. The Galerkin approximation (4.7) is simply the statement of the projection theorem (see Theorem 0.3.3 in [23]).

Note that both the Poisson equation (4.2) as well as its Galerkin finite-dimensional approximation (4.8) are coordinate-free representations. Particle-based approximation of the solution (4.8) can be carried out for any choice of coordinates. Certain coordinates may offer computational advantages, e.g., quaternions for $SO(3)$.

The non-trivial step in the Galerkin approximation is the choice of the basis function. In general, this choice is problem dependent. For matrix Lie groups, one choice is to use the Fourier basis. This is illustrated by two examples below.

Example 4.1 (Basis functions on $SO(2)$) *The Lie group $SO(2)$ is identified with the unit circle S^1. Using the angle coordinate $\theta \in S^1$, the simplest choice of the basis functions are the Fourier basis, e.g.,*

$$\psi_1(\theta) = \sin(\theta), \quad \psi_2(\theta) = \cos(\theta). \tag{4.10}$$

Note that these two basis functions are the eigenfunctions of the Laplacian on $SO(2)$, associated with its smallest non-zero eigenvalue.

Example 4.2 (Basis functions on $SO(3)$) *For the rotation group $SO(3)$, the Fourier basis are the eigenfunctions of the Laplace-Beltrami operator on the manifold defined as $\Delta\phi := div(grad(\phi))$ [97]. These functions are also associated with the irreducible unitary representation of $SO(3)$ [36]. Given its importance in applications, the eigenfunctions associated with the smallest eigenvalue for $SO(3)$ are tabulated in Appendix C.3. Also included is a brief overview of the derivation of these basis functions.*

4.3.1 Basis functions in the Euclidean space

Recall that the Poisson equation for the global optimization algorithm in the Euclidean space is given by, (see equation (3.5)),

$$\begin{aligned} -\nabla \cdot (\rho(x)\nabla\phi(x)) &= (h(x) - \hat{h})\rho(x), \quad x \in \mathbb{R}^d, \\ \int \phi(x)\rho(x)\,\mathrm{d}x &= 0, \end{aligned}$$

whose weak form is obtained as,

$$\int \nabla\phi(x) \cdot \nabla\psi(x)\rho(x)\,\mathrm{d}x = \int (h(x) - \hat{h})\psi(x)\rho(x)\,\mathrm{d}x, \tag{4.11}$$

for all $\psi \in H^1(\mathbb{R}^d;\rho)$. The control function is given by $u(x) = -\nabla\phi(x)$.

The following examples provide certain choice of basis functions in the Euclidean space. These may be used for approximating the control function for solving the global optimization problem presented in Chapter 3:

Example 4.3 *Two types of approximations follow from consideration of first order and second order polynomials as basis functions:*

1) The constant approximation is obtained by taking basis functions as $\psi_l(x) = x_l$ for $l = 1,\dots,d$. With this choice, A is the identity matrix and the control function is a constant vector:

$$u(x) = -\beta b = -\beta \int x\,(h(x) - \hat{h})\,\rho(x)\,\mathrm{d}x,$$

where A and b are defined as in (4.8).

2) The affine approximation is obtained by taking the basis functions as quadratic polynomials, $\psi_l(x) = x_l$ for $l = 1,\dots,d$ and $\psi_{lk}(x) = (x_l - m_l)\,(x_k - m_k)$ for $1 \le l \le k \le d$, where $m := \int x\rho(x)\,\mathrm{d}x$ is the mean. In this case,

$$u(x) = -\beta \mathsf{K}(x - m) - \beta b,$$

and direct calculations show that the matrix equation (4.8) leads to the same equations for K and b as given by (3.18) and (3.17). Hence, the Galerkin method provides an alternative way to calculate the affine control law. Note that the Galerkin derivation of the affine control law does not require that the density be Gaussian or the objective function be quadratic. This is consistent with the observations in Remark 3.4.

Example 4.4 *One choice for the basis functions is the Fourier basis functions, e.g.,*

$$\left\{ \sin\left(\frac{2\pi x_k}{T}\right),\ \cos\left(\frac{2\pi x_k}{T}\right),\quad 1 \le k \le d \right\}.$$

Example 4.5 *With a single basis function $\psi(x) = h(x)$, the approximate Galerkin solution is*

$$\phi(x) = \frac{\int (h(x) - \hat{h})^2 \rho(x)\,\mathrm{d}x}{\int |\nabla h(x)|^2 \rho(x)\,\mathrm{d}x}\, h(x).$$

Using an empirical approximation, the finite-N system (3.4) is the gradient-descent algorithm:

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = -\beta \frac{\sum_{i=1}^N (h(X_t^i) - \hat{h}^{(N)})^2}{\sum_{i=1}^N |\nabla h(X_t^i)|^2}\, \nabla h(X_t^i).$$

Remark 4.3 *There is no general guideline for choosing a proper set of basis functions. The basis can be defined either globally or locally. The spectral Galerkin method [156] uses the Fourier basis or the*

various orthogonal polynomials (e.g., Hermite, Chebyshev, Laguerre polynomials) that are defined over the entire domain. A class of meshless Galerkin methods [109], in contrast, employ the moving least square construction [108] where a basis function typically has only a compact support centered at a nodal point. Such local approximation has given rise to the reproducing kernel particle method [115], the partition of unity method [49], the meshless local Petrov-Galerkin method [4], and the meshless Galerkin method using radial basis functions [190, 48]. Some meshless methods have recently been applied to numerically solving pdes on 3D surfaces that arise in certain computer vision applications [110, 107]. Galerkin approximation methods on spheres with localized kernel basis function appear in [101, 138].

4.4 Kernel-based Gain Function Approximation

In this section, we present the kernel-based method whose attractive feature is that it does not involve selection of basis functions. In the numerical results presented in Chapter 5 and 7, this approach is also shown to be very effective. The original development of the kernel-based method in the Euclidean space is presented in [170, 169], The implementation specifically for the global optimization problem appears in [207].

In a kernel-based method, the unknown function $\phi(x)$ – solution of the strong form Poisson equation (4.2) – is approximated by its values at the particles $\{X^i\}_{i=1}^N$:

$$\Phi := \big(\phi(X^1), \phi(X^2), \ldots, \phi(X^N)\big) \in \mathbb{R}^N.$$

In terms of Φ, the Poisson equation (4.2) is approximated as a finite-dimensional fixed-point problem,

$$\Phi = T^{(\varepsilon,N)}\Phi + \varepsilon H^{(N)}, \tag{4.12}$$

on the co-dimension 1 subspace of normalized (i.e., mean zero) vectors, where ε is a small positive parameter, $H^{(N)} := \big(h(X^1) - \hat{h}^{(N)}, h(X^2) - \hat{h}^{(N)}, \ldots, h(X^N) - \hat{h}^{(N)}\big) \in \mathbb{R}^N$, and $T^{(\varepsilon,N)} \in \mathbb{R}^{N\times N}$ is a Markov matrix that is assembled from the ensemble $\{X^i\}_{i=1}^N$. It is shown in [169] that:

1) The Markov matrix $T^{(\varepsilon,N)}$ is a strict contraction on the subspace, and thus

2) the finite-dimensional problem (4.12) admits a unique normalized solution Φ,

3) this solution can be obtained by successive approximations, and

4) Φ approximates the true solution ϕ as $\varepsilon \to 0$ and $N \to \infty$.

For the manifold, the $(i,j)^{\text{th}}$ element of the $N \times N$ matrix $T^{(\varepsilon,N)}$ is constructed as,

$$T_{ij}^{(\varepsilon,N)} = \frac{\tilde{k}^{(\varepsilon)}(X^i, X^j)}{\sum_{l=1}^{N} \tilde{k}^{(\varepsilon)}(X^i, X^l)}, \tag{4.13}$$

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

$$\tilde{k}^{(\varepsilon)}(X^i, X^j) = \frac{k^{(\varepsilon)}(X^i, X^j)}{\sqrt{\frac{1}{N}\sum_{l=1}^{N} k^{(\varepsilon)}(X^i, X^l)}\sqrt{\frac{1}{N}\sum_{l=1}^{N} k^{(\varepsilon)}(X^j, X^l)}}, \tag{4.14}$$

and $k^{(\varepsilon)}$ is the Gaussian kernel,

$$k^{(\varepsilon)}(X^i, X^j) := \frac{1}{(4\pi\varepsilon)^{d/2}} \exp\Big(-\frac{\zeta^2(X^i, X^j)}{4\varepsilon}\Big), \tag{4.15}$$

where d is the dimension of G, and $\zeta : G \times G \to \mathbb{R}$ denotes a distance metric on G induced from the Euclidean space in which G is smoothly embedded (see Assumption 19 in [71]).

Remark 4.4 *The justification of the fixed-point problem* (4.12) *is as follows: The weighted Laplacian is the infinitesimal generator of a Markov semigroup, denoted as* $e^{\varepsilon\Delta_\rho}$*; c.f., [6]. In terms of this semigroup, the Poisson equation* (4.2) *is equivalently expressed as the following fixed-point problem,*

$$\phi = e^{\varepsilon\Delta_\rho}\phi + \int_0^\varepsilon e^{s\Delta_\rho}(h - \hat{h})\,\mathrm{d}s, \tag{4.16}$$

for fixed $\varepsilon > 0$*. If the distribution* π *admits a spectral gap (i.e.,* (4.3) *holds for some* $\bar{\lambda} > 0$*), then* $e^{\varepsilon\Delta_\rho}$ *is a contraction on* $L_0^2(G;\pi)$ *and a unique solution exists by the contracting mapping theorem.*

In the limit as $\varepsilon \to 0$ *and* $N \to \infty$*,* $e^{\varepsilon\Delta_\rho}$ *is approximated by an operator* $T^{(\varepsilon)}$ *defined as:*

$$T^{(\varepsilon)} f(x) = \frac{1}{n^{(\varepsilon)}(x)} \int_G \tilde{k}^{(\varepsilon)}(x,y) f(y) \rho(y)\,\mathrm{d}y,$$

where $n^{(\varepsilon)}(x) = \int_G \tilde{k}^{(\varepsilon)}(x,y)\rho(y)\,\mathrm{d}y$ *is a normalization factor chosen such that* $T^{(\varepsilon)}1 = 1$ *(see Proposition 3*

in [40]). The fixed-point problem (4.16) is thus approximated as,

$$\phi^{(\varepsilon)} = T^{(\varepsilon)}\phi^{(\varepsilon)} + \varepsilon(h - \hat{h}), \tag{4.17}$$

where $\int_0^{\varepsilon} e^{s\Delta_\rho}(h-\hat{h})\,\mathrm{d}s \approx \varepsilon(h-\hat{h})$ *for small* $\varepsilon > 0$*. In the limit as* $N \to \infty$*, the fixed-point problem (4.12) represents a finite-dimensional approximation of (4.17), with* $T^{(\varepsilon,N)}$ *representing a finite-dimensional approximation of the semigroup* $e^{\varepsilon\Delta_\rho}$*.*

The coordinates of the gain function $\mathsf{k}_n = E_n \cdot \phi$ for $n = 1, ..., d$ are obtained by taking an explicit derivative of (4.17). The numerical procedure to obtain k_n is summarized below:

1) Define the vector

$$\widetilde{H}_n := \big(E_n \cdot h(X^1), E_n \cdot h(X^2), ..., E_n \cdot h(X^N)\big),$$

and define the $N \times N$ matrix $\widetilde{Z}_n$ whose elements are given by,

$$(\widetilde{Z}_n)_{ij} := E_n \cdot \zeta^2(X^i, X^j),$$

where $E_n \cdot \zeta^2(x,y) := \frac{\mathrm{d}}{\mathrm{d}\tau}\big|_{\tau=0}\zeta^2\big(x\exp(\tau E_n), y\big)$ for $x, y \in G$.

2) Define the $N \times N$ matrix,

$$S_n := T^{(\varepsilon,N)} * \widetilde{Z}_n,$$

where $*$ denotes the Hadamard (element-wise) product of two matrices, i.e., $(S_n)_{ij} = (T^{(\varepsilon,N)})_{ij}\,(\widetilde{Z}_n)_{ij}$.

3) Define $\Upsilon_n := \big(\mathsf{k}_n(X^1), \mathsf{k}_n(X^2), ..., \mathsf{k}_n(X^N)\big) \in \mathbb{R}^N$. Then,

$$\Upsilon_n = \varepsilon\widetilde{H}_n - \frac{1}{4\varepsilon}\big[S_n\Phi - (S_n\mathbf{1}) * (T^{(\varepsilon,N)}\Phi)\big], \tag{4.18}$$

where $\mathbf{1} = (1, 1, ..., 1) \in \mathbb{R}^N$, and $*$ denotes the element-wise product of two vectors.

The numerical procedure of the kernel-based gain function approximation is tabulated in Algorithm 5.

The choice of distance metric $\zeta^2(x,y)$ depends on the manifold. Two examples are given below:

Algorithm 5 Kernel-based gain function approximation

Input: Particles $\{X^i\}_{i=1}^N$ sampled i.i.d. from π, parameters ε, K

Calculate $\hat{h}^{(N)} = \frac{1}{N}\sum_{i=1}^N h(X^i)$

Calculate $H_i^{(N)} = h(X^i) - \hat{h}^{(N)}$ for $i = 1,...,N$

Calculate $k^{(\varepsilon)}(X^i,X^j)$, $\tilde{k}^{(\varepsilon)}(X^i,X^j)$ by (4.15), (4.14) for all i,j

Calculate $T_{ij}^{(\varepsilon,N)}$ according to (4.13) for all i,j

Assign initial condition Φ_0

for $k = 0$ to $K-1$ **do**

  Calculate $\Phi_{k+1} = T^{(\varepsilon,N)}\Phi_k + \varepsilon H^{(N)}$, with $T^{(\varepsilon,N)} = [T_{ij}^{(\varepsilon,N)}]$

  Assign $\Phi_{k+1} = \Phi_{k+1} - \frac{1}{N}\sum_{i=1}^N (\Phi_{k+1})_i$

end for

Calculate $\Upsilon_n = \big(\mathsf{k}_n(X^1),...,\mathsf{k}_n(X^N)\big)$ for $n = 1,...,d$ according to (4.18) with $\Phi = \Phi_K$

Return: Coordinates $\big\{\big(\mathsf{k}_n(X^1),...,\mathsf{k}_n(X^N)\big)\big\}_{n=1}^d$

Example 4.6 (Distance metric in $SO(3)$) *On the Lie group $SO(3)$, $d = 3$, and the distance metric is given by (see [82]),*

$$\zeta^2(R_1,R_2) = |R_1 - R_2|_F^2,$$

for $R_1,R_2 \in SO(3)$, where $|\cdot|_F$ is the Frobenius norm of a matrix. This metric is induced from the Euclidean space $\mathbb{R}^9$, where the smooth embedding $i : SO(3) \to \mathbb{R}^9$ is defined as $i(R) = (R_{11},R_{12},...,R_{33})$. Using the basis of $so(3)$ given by (2.13), we have

$$E_n \cdot (\zeta^2)(R^i,R^j) = -2\,Tr(R^i E_n R^j), \qquad n = 1,2,3.$$

Example 4.7 (Distance metric in $\mathbb{R}^d$) *In the Euclidean space $\mathbb{R}^d$, the distance metric is the conventional Euclidean distance,*

$$\zeta^2(X^i,X^j) = |X^i - X^j|^2$$

for $X^i,X^j \in \mathbb{R}^d$. This is used in the kernel-based control function approximation for the controlled particle filter algorithm presented in Chapter 3.

An example of the distance metric on the product Lie group $SO(3) \times SO(2)$ is given in Sec. 6.4.3 for the motion tracking application with FPF.

4.5 Related Approximation Methods

The gain function approximation is a hard problem. The Galerkin algorithm represents a straightforward solution where the hard part is to select the basis functions. A number of papers have considered related approaches: i) the use of proper orthogonal decomposition (POD) to select basis functions in [18]; ii) a continuation scheme in [129]; and iii) certain probabilistic approaches involving dynamic programming in [147]. We expect that many of these approaches can be also generalized to the manifold setting.

Kernel-based approaches have received considerable attention in machine learning applications including dimensionality reduction [40, 14], spectral clustering [184], and other unsupervised learning models [177]. The semigroup associated with the weighted Laplacian operator has also been intensively studied [65, 28].

Apart from the Galerkin and the kernel-based numerical schemes, the meshless collocation method has been widely used for numerical approximations [211, 73, 185] and recently also extended to certain manifold [102]. However, the collocation method typically requires an explicit expression of the differential operator in the (strong form) pde, hence not suitable for the Poisson equation (4.2) where the density function ρ is unknown in general. It is noted that both the Galerkin and the kernel-based method described in this chapter are completely adapted to data. That is, no explicit computation of the density is ever required. Instead, one only needs to either evaluate a given set of basis functions at the particles, or construct a weighted graph using the particles.

4.6 Conclusions

In this chapter, we discussed algorithms for numerical solutions of the Poisson equation in the feedback particle filter. In the Lie group setting, the well-posedness results were presented, and two algorithms were described to numerically approximate the gain function:

1) In the Galerkin gain function approximation, the solution is approximated via a projection onto the subspace spanned by a prescribed set of basis functions. Numerically, the algorithm entails solving a

finite-dimensional linear matrix equation.

2) In the kernel-based gain function approximation, the solution is obtained via a Markov approximation of the semigroup.

Both the Galerkin and the kernel-based scheme are shown to respect the intrinsic geometry of the manifold, and are completely adapted to data. Numerical procedures to obtain the approximations were provided, including examples on important Lie groups such as $SO(3)$ as well as the Euclidean space $\mathbb{R}^d$.

Part II

Applications

Chapter 5

Attitude Estimation with Feedback Particle Filter *

5.1 Introduction

This chapter is concerned with the problem of attitude estimation, modeled here as a continuous-time nonlinear filtering problem on the Lie group $SO(3)$. For this important special case, the explicit form of the feedback particle filter is described with respect to both the rotation matrix and the quaternion coordinates, with the latter being demonstrated for computational purposes.

Apart from theory, a comparison is provided between FPF and the several popular attitude filters including the multiplicative EKF, the invariant EKF, the unscented Kalman filter, the invariant ensemble Kalman filter and the bootstrap particle filter (BPF). Numerical algorithms of these filters are described in a self-contained manner. The comparison is illustrated by two numerical studies: (i) an attitude estimation problem on $SO(3)$, and (ii) a filtering problem with a bimodal prior distribution supported on the subgroup $SO(2)$.

For the attitude estimation problem, the filters are assessed with respect to their averaged estimation error and simulation variance across multiple Monte-Carlo runs. Their performance is also investigated by varying the prior distribution, the process noise, the observation noise, as well as the number of particles used in particle filters. The particle filters are also compared in terms of their computational complexity.

For the filtering problem with support on $SO(2)$, a static process model is considered such that the posterior density admits a closed-form Bayes' formula. The filters are simulated with a bimodal initial distribution. FPF and BPF are compared regarding their capability of handling non-Gaussian posterior distributions. The kernel-based gain function approximation in FPF is also investigated.

Numerical studies that contain comparisons of state-of-the-art attitude filters and attitude observers also appear in several Ph.D. theses [200, 85, 11, 117] as well as research papers [86, 13, 25].

The remainder of this chapter is organized as follows: The mathematical formulation of the attitude

*The content of this chapter is related to the publication [208, 209].

estimation problem is presented in Sec. 5.2. The feedback particle filter algorithm for attitude estimation is described in Sec. 5.3. Sec. 5.4 provides a self-contained description of the attitude filters that are considered in the numerical studies. Simulation results for the attitude estimation problem on $SO(3)$ and the problem on the subgroup $SO(2)$ appear in Sec. 5.5 and Sec. 5.6, respectively.

5.2 Problem Formulation

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

$$\mathrm{d}R_t = R_t\,\Omega_t\,\mathrm{d}t + R_t \circ [\,\sigma_B\,\mathrm{d}B_t\,]_\times, \tag{5.1}$$

where $R_t \in SO(3)$ is the orientation of the rigid body at time t, expressed with respect to an inertial frame; $\Omega_t = [\,\omega_t\,]_\times$ where $\omega_t \in \mathbb{R}^3$ represents the angular velocity expressed in the body frame; B_t is a standard Wiener process in $\mathbb{R}^3$, and σ_B is a positive scalar. Both Ω_t and $[\,\sigma_B\,\mathrm{d}B_t\,]_\times$ are elements of the Lie algebra $so(3)$ (see (2.14) for the definition of $[\,\cdot\,]_\times$).

Using the quaternion coordinates, (5.1) is written as,

$$\mathrm{d}q_t = \frac{1}{2}\,q_t \otimes (\omega_t\,\mathrm{d}t + \sigma_B\,\mathrm{d}B_t), \tag{5.2}$$

where, by a slight abuse of notation, $\omega_t \in \mathbb{R}^3$ is interpreted as a quaternion $(0, \omega_t)$, and $\mathrm{d}B_t$ is interpreted similarly. The sde (5.2) is also interpreted in the Stratonovich sense.

Accelerometer: In the absence of translational motion, the accelerometer is modeled as (see [118]),

$$\mathrm{d}Z_t^g = -R_t^T\,r^g\,\mathrm{d}t + \sigma_W\,\mathrm{d}W_t^g, \tag{5.3}$$

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

Magnetometer: The model of the magnetometer is of a similar form (see [118]),

$$\mathrm{d}Z_t^b = R_t^T\,r^b\,\mathrm{d}t + \sigma_W\,\mathrm{d}W_t^b, \tag{5.4}$$

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

In terms of the process and observation models (5.1)-(5.4), the nonlinear filtering problem for attitude estimation is succinctly expressed as,

$$\mathrm{d}R_t = R_t\,\Omega_t\,\mathrm{d}t + R_t \circ [\,\sigma_B\,\mathrm{d}B_t\,]_\times, \tag{5.5a}$$

$$\mathrm{d}Z_t = h(R_t)\,\mathrm{d}t + \sigma_W\,\mathrm{d}W_t, \tag{5.5b}$$

where $h : SO(3) \to \mathbb{R}^6$ is a given function whose j-th coordinate is denoted as h_j, and W_t is a standard Wiener process in $\mathbb{R}^6$. Note that (5.5b) encapsulates the sensor models given in (5.3) and (5.4) within a single equation. It is assumed that B_t and W_t are mutually independent, and both are independent of the initial condition R_0.

Remark 5.1 *There are a number of simplifying assumptions implicit in the model defined in* (5.5a)-(5.5b). *In practice, ω_t needs to be estimated from noisy gyroscope measurements and there is translational motion as well. This requires additional models which are easily incorporated within the proposed filtering framework. The purpose here is to elucidate the geometric aspects of the FPF in the simplest possible setting of $SO(3)$. More practical FPF-based filters that also incorporate models for translational motion, measurements of ω_t from gyroscope, effects of translational motion on accelerometer, and effects of sensor bias are subject of separate publication.*

5.3 FPF for Attitude Estimation

Following the general framework of FPF described in Sec. 2.3, the dynamics of the i-th particle is defined by,

$$\mathrm{d}R_t^i = R_t^i\,\Omega_t\,\mathrm{d}t + R_t^i \circ [\,\sigma_B\,\mathrm{d}B_t^i\,]_\times + R_t^i\,[\,\mathsf{K}(R_t^i,t) \circ \mathrm{d}\mathrm{I}_t^i\,]_\times, \tag{5.6}$$

where B_t^i for $i = 1,\ldots,N$ are mutually independent standard Wiener processes in $\mathbb{R}^3$. The error $\mathrm{d}\mathrm{I}_t^i \in \mathbb{R}^6$ is given by,

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

Algorithm 6 Feedback Particle Filter for attitude estimation

Initialization: Samples $\{q_0^i\}_{i=1}^N$ i.i.d. from π_0^*

Assign $t = 0$

Iteration: from t to $t + \Delta t$

Calculate $\hat{h}^{(N)} = \frac{1}{N}\sum_{i=1}^N h(q_t^i)$

for $i = 1$ to N **do**

  Generate a sample, ΔB_t^i, from $N\big(0, (\Delta t)I\big)$

  Calculate the error $\Delta \mathrm{I}_t^i := \Delta Z_t - \frac{1}{2}\left(h(q_t^i) + \hat{h}^{(N)}\right)\Delta t$

  Calculate gain function $\mathsf{K}(q_t^i, t)$ using Galerkin or kernel-based scheme (see Chapter 4)

  Calculate $\Delta v_t^i = \omega_t\,\Delta t + \sigma_B\,\Delta B_t^i + \mathsf{K}(q_t^i, t)\,\Delta \mathrm{I}_t^i$

  Propagate the particle q_t^i according to (see [178], and $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^3$)

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

end for

Define matrix $Q = \frac{1}{N}\sum_{i=1}^N q_{t+\Delta t}^i\, q_{t+\Delta t}^{i\,T}$

Return: empirical mean of $\{q_{t+\Delta t}^i\}_{i=1}^N$, i.e., the unit eigenvector of Q associated with its largest eigenvalue

Assign $t = t + \Delta t$

The gain function K is a 3×6 matrix whose entries are obtained as follows: For $j = 1, 2, ..., 6$, the j-th column of K contains the coordinates of the vector-field $\mathrm{grad}(\phi_j)$, where the function $\phi_j \in H^1(SO(3); \pi_t)$ is a solution to the Poisson equation,

$$\begin{aligned} &\pi_t\big(\langle \mathrm{grad}(\phi_j), \mathrm{grad}(\psi)\rangle\big) = \frac{1}{\sigma_W^2}\,\pi_t\big((h_j - \hat{h}_j)\psi\big), \\ &\pi_t(\phi_j) = 0 \quad \text{(normalization)}, \end{aligned} \tag{5.7}$$

for all $\psi \in H^1(SO(3); \pi)$.

For numerical purposes, it is convenient to express the FPF with respect to the quaternion coordinates. In this coordinate representation, the dynamics of the i-th particle is given by,

$$\mathrm{d}q_t^i = \frac{1}{2}\, q_t^i \otimes \mathrm{d}v_t^i, \tag{5.8}$$

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

$$\mathrm{d}v_t^i = \omega_t\, \mathrm{d}t + \mathrm{d}B_t^i + \mathsf{K}(q_t^i, t) \circ \Big(\mathrm{d}Z_t - \frac{h(q_t^i) + \hat{h}}{2}\, \mathrm{d}t \Big), \tag{5.9}$$

where $\mathsf{K}(q,t) = \mathsf{K}(R(q),t)$ and $h(q) = h(R(q))$, with $R = R(q)$ given by the formula (2.16).

The FPF algorithm is numerically implemented using the quaternion coordinates, and is described in Algorithm 6. The algorithm simulates N particles, $\{q_t^i\}_{i=1}^N$, according to the sde's (5.8) and (5.9), with the initial conditions $\{q_0^i\}_{i=1}^N$ sampled i.i.d. from a given prior distribution π_0^*. The gain function is approximated using either the Galerkin scheme (see Sec. 4.3) with the basis functions given in Appendix C.3, or the kernel-based scheme (see Sec. 4.4).

Given a particle set $\{q_t^i\}_{i=1}^N$, its empirical mean is obtained as the eigenvector (with norm 1) of the 4×4 matrix $Q = \frac{1}{N}\sum_{i=1}^N q_t^i q_t^{i\,T}$ corresponding to its largest eigenvalue [125].

5.4 Review of Some Attitude Filters

This section presents detailed algorithms for the attitude filters that are simulated for the comparison in Sec. 5.5. The following attitude estimation problem is considered,

$$\mathrm{d}q_t = \frac{1}{2}\, q_t \otimes \big(\omega_t\, \mathrm{d}t + \sigma_B\, \mathrm{d}B_t \big), \tag{5.10a}$$

$$\mathrm{d}Z_t = R(q_t)^T r\, \mathrm{d}t + \sigma_W\, \mathrm{d}W_t, \tag{5.10b}$$

where $r \in \mathbb{R}^3$ denotes a generic reference vector. The simplified observation model (5.10b) avoids undue notations in the description of the filters. A more realistic model with both gravity and magnetic field observed is used in Sec. 5.5 for numerical simulations.

Most of the filters presented in this section are discrete-time filters. They require a discrete-time filtering

model that is chosen to be consistent with the continuous-time model (5.10a)-(5.10b). For the discrete-time filters, the sampled observations, denoted as $\{Y_n\}$, are made at discrete times $\{t_n\}$, whose model is formally expressed as $Y_n := \frac{\Delta Z_{t_n}}{\Delta t} = R(q_{t_n})^T r + W_n^\Delta$ where $\{W_n^\Delta\}$ are i.i.d. with the distribution $\mathcal{N}(0, \frac{\sigma_W^2}{\Delta t} I)$, and Δt denotes the time step. Such a model leads to the correct scaling between the continuous and the discrete-time filter implementations.

The multiplicative EKF, unscented quaternion estimator and bootstrap particle filter use one of the three-dimensional parametrization of $SO(3)$, e.g., the Euler angle, the rotation vector, and the Rodrigues parameter, to represent the attitude estimation error. The filter is then implemented as an EKF on a subset of $\mathbb{R}^3$. The subset is referred to as the *parameter space*. In our numerical studies, the modified Rodrigues parameter (MRP) is used. Conversion between an error MRP x and an error quaternion δq is given by [124],

$$\delta q(x) = \frac{1}{16 + |x|^2} \begin{bmatrix} 16 - |x|^2 \\ 8x \end{bmatrix}, \qquad x(\delta q) = \frac{4}{1 + \delta q_0} \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \end{bmatrix}, \tag{5.11}$$

where notationally $\delta q = (\delta q_0, \delta q_1, \delta q_2, \delta q_3)$.

All the filters are implemented using the quaternion coordinates. For discrete-time filters, the posterior filter estimate at time t_n is denoted as $\hat{q}_n$. The filter estimate after the propagation step between t_{n-1} and t_n is denoted as $\hat{q}'_n$. For Kalman-type filters, the associated covariance matrices are denoted as Σ_n and Σ'_n, respectively. Each filter is described for one iteration that maps $\hat{q}_{n-1}$ to $\hat{q}_n$ and Σ_{n-1} to Σ_n.

5.4.1 Multiplicative EKF

The MEKF algorithm is described in [124, 178]. The linearized model of the estimation error, represented using the MRP, is given by,

$$\mathrm{d}x_t = -[\,\omega_t\,]_\times x_t \,\mathrm{d}t - \sigma_B \,\mathrm{d}B_t, \tag{5.12}$$

where x_t denotes the error MRP at time t. Such a linearization is reasonable when $|x_t|$ and σ_B are both sufficiently small. The MEKF then follows the classical EKF based on the linearized model (5.12) defined in $\mathbb{R}^3$. The detailed algorithm is presented below.

Input: Current quaternion estimate $\hat{q}_{n-1}$ and Σ_{n-1}. The estimate of error MRP is $\hat{x}_{n-1} = 0$.

Propagation:

$$\hat{q}'_n = \hat{q}_{n-1} \otimes \exp(\omega_{n-1}\Delta t), \tag{5.13}$$

$$\Sigma'_n = \Phi\Sigma_{n-1}\Phi^T + Q, \tag{5.14}$$

where by slight abuse of notation,

$$\exp(\omega) := \begin{bmatrix} \cos\left(|\omega|/2\right) \\ \frac{\omega}{|\omega|}\sin\left(|\omega|/2\right) \end{bmatrix}$$

for $\omega \in \mathbb{R}^3$, and $\omega_n := \omega_{t_n}$. The matrix Φ and Q are given by

$$\Phi = I - [\,\omega_{n-1}\,]_\times \Delta t\ , \quad Q = (\sigma_B^2 \Delta t)\, I.$$

Update: The observation update is first carried out for the error MRP:

$$\begin{aligned} \hat{x}_n &= \mathsf{K}_n \mathrm{I}_n, \\ \mathsf{K}_n &= \Sigma'_n H_n^T S_n^{-1}, \\ S_n &= H_n \Sigma'_n H_n^T + \mathrm{R}, \\ \widetilde{\Sigma}_n &= (I - \mathsf{K}_n H_n)\,\Sigma'_n, \end{aligned}$$

where $I_n = Y_n - R(\hat{q}'_n)^T r$, $H_n = [R(\hat{q}'_n)^T r]_\times$, and $\mathrm{R} = (\sigma_W^2/\Delta t)\, I$.

Reset: The update of error MRP is then incorporated in the quaternion estimate in a multiplicative way:

$$\hat{q}_n = \hat{q}'_n \otimes \delta q(\hat{x}_n), \tag{5.15}$$

$$\Sigma_n = G_n \widetilde{\Sigma}_n G_n^T, \tag{5.16}$$

where $G_n = I - \frac{1}{2}[\,\mathsf{K}_n \mathrm{I}_n\,]_\times$, and $\delta q(\hat{x}_n)$ is calculated by (5.11). The multiplicative nature of the formula (5.15) is the reason for calling this filter the multiplicative EKF.

The reset step is included such that $\hat{x}_n$ is reset to zero at each time step. This is analogous to the "re-linearization" step in a classical EKF [88] and also coincides with the "re-parametrization" step in the EKF algorithm proposed in [21]. While the reset step is implicit and trivial in a Euclidean EKF, it explicitly

modifies the updated covariance (see (5.16)) in a Lie group setting due to the multiplicative nature of the update formula (5.15).

Remark 5.2 *The formula* (5.14) *for covariance propagation is obtained by discretizing the continuous-time error model* (5.12). *$O((\Delta t)^2)$ terms are neglected in both Φ and Q. For higher-order terms in these matrices, c.f., [178]. For the sake of consistency, all the filters discussed in this section use the same order of numerical approximation.* □

5.4.2 Unscented Quaternion Estimator

The unscented quaternion estimator (USQUE) is described in [42]. The estimation error is parameterized using the MRP, and sigma points are employed to represent the error distribution. A conventional unscented Kalman filter is then implemented in the parameter space. The number of sigma points is $L = 2d+1$ where $d = 3$ is the dimensional of the MRP. The detailed algorithm is presented below.

Input: Current quaternion estimate $\hat{q}_{n-1}$ and Σ_{n-1}. The error estimate in MRP is $\hat{x}_{n-1} = 0$.

Sigma points generation:

i) Generate L sigma points of error MRP $\{x_{n-1}(l)\}_{l=0}^{L-1} \in \mathbb{R}^3$ from Σ_{n-1},

$$x_{n-1}(0) = \hat{x}_{n-1} = 0, \quad \{x_{n-1}(l)\}_{l=1}^{L-1} = 2d \text{ columns of } \pm\sqrt{(d+\lambda)\Sigma_{n-1}},$$

where λ is a tuning parameter. It is recommended in [42] that $\lambda = 1$. In practice, the square root of a positive definite matrix is computed using the Cholesky decomposition.

ii) Convert error MRP $x_{n-1}(l)$ to error quaternion $\delta q_{n-1}(l)$,

$$\delta q_{n-1}(0) = q_I, \quad \delta q_{n-1}(l) = \delta q(x_{n-1}(l)) \text{ using (5.11) for } l = 1,...,L-1.$$

where q_I denotes the identity quaternion, i.e., $q_I = (1,0,0,0)$.

iii) Generate quaternion sigma points $\{q_{n-1}(l)\}_{l=0}^{L-1}$,

$$q_{n-1}(0) = \hat{q}_{n-1}, \quad q_{n-1}(l) = \hat{q}_{n-1} \otimes \delta q_{n-1}(l).$$

Propagation:

i) Propagate quaternion sigma points according to the process model,

$$q'_n(l) = q_{n-1}(l) \otimes \exp(\omega_{n-1}\,\Delta t), \quad \forall\, l.$$

ii) Calculate propagated error quaternion,

$$\delta q'_n(0) = q_I, \quad \delta q'_n(l) = \left(q'_n(0)\right)^{-1} \otimes q'_n(l).$$

iii) Calculate propagated error MRP,

$$x'_n(0) = 0, \quad x'_n(l) = x(\delta q'_n(l)) \quad \text{(see (5.11))}.$$

iv) Calculate propagated mean $\hat{x}'_n$ and covariance Σ'_n from the sigma points $\{x'_n(l)\}_{l=0}^{L-1}$ using (5.17) and (5.18) below. Additionally, process noise matrix $Q = \frac{1}{2}(\sigma_B^2 \Delta t)I$ is added to Σ'_n.

Given a set of sigma points $\{x(l)\}_{l=0}^{L-1} \in \mathbb{R}^d$, their mean (weighted average) and covariance are calculated as,

$$\hat{x} = \frac{1}{d+\lambda}\left(\lambda\, x(0) + \frac{1}{2}\sum_{l=1}^{L-1} x(l)\right), \tag{5.17}$$

$$\Sigma = \frac{1}{d+\lambda}\left(\lambda\,(x(0)-\hat{x})(x(0)-\hat{x})^T + \frac{1}{2}\sum_{l=1}^{L-1}(x(l)-\hat{x})(x(l)-\hat{x})^T\right). \tag{5.18}$$

Update:

i) Calculate predicted observation of each sigma point,

$$y_n(l) = R(q'_n(l))^T r, \quad \forall\, l,$$

whose covariance, denoted as $\widetilde{\Sigma}_n^{yy}$, is calculated using (5.18).

ii) Calculate the cross-correlation matrix Σ_n^{xy} from the two sets of sigma points $\{\hat{x}'_n\}_{l=0}^{L-1}$ and $\{y_n\}_{l=0}^{L-1}$. The formula is analogous to (5.18).

iii) Calculate the gain matrix

$$\mathsf{K}_n = \Sigma_n^{xy} \Sigma_n^{yy-1},$$

where $\Sigma_n^{yy} = \widetilde{\Sigma}_n^{yy} + (\sigma_W^2/\Delta t) I$.

iv) Calculate the innovation error

$$\mathrm{I}_n = Y_n - y_n(0).$$

v) Update the error MRP estimate,

$$\hat{x}_n = \mathsf{K}_n \mathrm{I}_n.$$

vi) Update the covariance matrix,

$$\Sigma_n = \Sigma_{n-1} - \mathsf{K}_n \Sigma_n^{yy} \mathsf{K}_n^T.$$

vii) Update quaternion estimate,

$$\hat{q}_n = \hat{q}'_n \otimes \delta q(\hat{x}_n).$$

Reset: Reset $\hat{x}_n = 0$. The covariance matrix is unchanged after reset in USQUE.

5.4.3 Bootstrap Particle Filter

The bootstrap particle filter (BPF) is described in [32]. The estimation error is parameterized using the MRP, and N particles are employed to represent the error distribution. A conventional particle filter is then implemented in the parameter space based on importance sampling and resampling . The algorithm is presented below.

Input: Current quaternion estimate $\hat{q}_{n-1}$ and quaternion particles $\{q_{n-1}^i\}_{i=1}^N$.

Propagation: Propagate the quaternion estimate and particles according to the process model,

$$\hat{q}'_n = \hat{q}_{n-1} \otimes \exp(\omega_{n-1}\Delta t), \quad q_n'^i = q_{n-1}^i \otimes \exp(\omega_{n-1}\Delta t + B_n^i) \quad \forall\, i,$$

where the process noise $\{B_n^i\}_{i=1}^N$ are i.i.d. samples from $\mathcal{N}(0, (\sigma_B^2 \Delta t) I)$.

Update:

i) Calculate error quaternion particles,

$$\delta\tilde{q}_n^i = \left(\hat{q}_n'\right)^{-1} \otimes q_n'^i.$$

ii) Calculate error MRP particles $\{\tilde{x}_n^i\}_{i=1}^N$ from $\{\delta\tilde{q}_n^i\}_{i=1}^N$ using formula (5.11).

iii) Calculate weights,

$$w_n^i \propto \exp\Big\{-\frac{1}{2}(Y_n - R(q_n'^i)^T r)^T R^{-1}(Y_n - R(q_n'^i)^T r)\Big\},$$

where $R = \left(\sigma_W^2/\Delta t\right) I$.

iv) Calculate the mean error MRP $\hat{x}_n = \sum_i w_n^i \tilde{x}_n^i$

v) Calculate the updated quaternion estimate,

$$\hat{q}_n = \hat{q}_n' \otimes \delta q(\hat{x}_n).$$

vi) Resample to obtain uniformly weighted error MRP particles $\{x_n^i\}_{i=1}^N$ from $\{\tilde{x}_n^i\}_{i=1}^N$. The resampling procedure follows Algorithm 2 presented in [3].

vii) Perturb the new error MRP particles,

$$x_n^i \leftarrow x_n^i + \nu_n^i, \quad \nu_n^i \sim \mathcal{N}(0, \sigma^2 \Sigma_n),$$

where $\Sigma_n = \frac{1}{N-1}\sum_i x_n^i x_n^{iT}$, and σ is a small tuning parameter.

viii) Update the quaternion particles,

$$q_n^i = q_n'^i \otimes \delta q(x_n^i).$$

For problems with large initialization error and concentrated likelihood functions, only a few particles have significant weights w_n^i after a few steps. Since resampling eliminates the majority of particles with very small weight, the particle filter quickly loses particle diversity. The perturbation in Step vii) above may alleviate this issue. Additionally, the *progressive correction* technique is applied in [32], which splits the likelihood function as a product of several "wider" likelihood functions, and then sequentially implements

the update (Step i) to viii) above) due to each split likelihood. Detailed procedure for doing this is provided in [32] and Chapter 12 of [46].

5.4.4 Left invariant EKF

The left invariant EKF (LIEKF) is a continuous-time filter on $SO(3)$ whose equation is given by,

$$\mathrm{d}\hat{q}_t = \frac{1}{2}\hat{q}_t \otimes (\omega_t\,\mathrm{d}t + \mathsf{K}_t\,\mathrm{dI}_t),$$

where $\hat{q}_t$ denotes the quaternion estimate at time t, $\mathrm{dI}_t = \mathrm{d}Z_t - R(\hat{q}_t)^T r$, and $\mathsf{K}_t = \frac{1}{\sigma_W^2}\Sigma_t H_t^T$ with $H_t = [R(\hat{q}_t)^T r]_\times$.

The equation of the covariance is given by the Riccati equation,

$$\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} = A_t\Sigma_t + \Sigma_t A_t^T + \sigma_B^2 I - \frac{1}{\sigma_W^2}\Sigma_t H_t^T H_t \Sigma_t.$$

For additional details, c.f., [20].

The connection between LIEKF and the feedback particle filter with concentrate distributions was discussed in Sec. 2.4.2.

5.4.5 Invariant EKF

The invariant EKF (IEKF) is a discrete-time filter proposed in [13]. The IEKF is originally developed to solve a slightly different attitude estimation problem,

$$\mathrm{d}q_t = \frac{1}{2}q_t \otimes \omega_t\,\mathrm{d}t + \frac{1}{2}(\sigma_B\,\mathrm{d}B_t)\otimes q_t, \tag{5.19a}$$

$$\mathrm{d}Z_t = R(q_t)^T(r + \sigma_W\,\mathrm{d}W_t), \tag{5.19b}$$

where both the process and observation noise are defined in the inertial frame instead of the body frame. However, the noise defined in two frames are statistically equivalent if the noise is *isotropic* (see [13] for a definition of isotropic noise). The noise in the filtering problem (5.10a)-(5.10b) is isotropic and rotation-invariant since the components of the noise are mutually independent and have equal variance.

The detailed algorithm is presented below:

Input: Current quaternion estimate $\hat{q}_{n-1}$ and Σ_{n-1}. The error estimate $\hat{e}_{n-1} = 0$.

Propagation:

$$\hat{q}'_n = \hat{q}_{n-1} \otimes \exp(\omega_{n-1}\Delta t),$$
$$\Sigma'_n = \Sigma_{n-1} + \left(\sigma_B^2 \Delta t\right) I,$$

The propagation step is identical to MEKF.

Update: The observation update is carried out for the Lie-algebraic error as follows,

$$\hat{e}_n = \mathsf{K}_n \mathrm{I}_n,$$
$$\mathsf{K}_n = \Sigma'_n H^T S_n^{-1},$$
$$S_n = H \Sigma'_n H^T + (\sigma_W^2/\Delta t) I,$$

where $H = [r]_\times$, and the innovation $\mathrm{I}_n = R(\hat{q}'_n) Y_n - r$ is modeled in the inertial frame. The update in the error estimate is then incorporated in the quaternion estimate in a multiplicative way:

$$\hat{q}_n = \delta q(\hat{e}_n) \otimes \hat{q}'_n, \tag{5.20}$$
$$\Sigma_n = (I - \mathsf{K}_n H)\Sigma'_n,$$

where $\delta q(\hat{e}_n) = \exp(\hat{e}_n)$. The Lie-algebraic error is used in IEKF since it leads to an intrinsic construction of the filter, and the error dynamics exhibits certain invariance property. This is a key difference between IEKF and MEKF.

5.4.6 Invariant Ensemble Kalman Filter

Whereas the MEKF and IEKF algorithms compute the gain by linearizing the error model, the invariant ensemble Kalman filter (IEnKF) employs an ensemble of random samples whose empirical covariance is used for gain computation.

The detailed algorithm is presented below:

Input: Current quaternion estimate $\hat{q}_{n-1}$ and error samples $\{\eta_{n-1}^i\}_{i=1}^N$.

Gain computation:

i) Propagate the error samples,

$$\eta_n'^i = \exp(B_n^i) \otimes \eta_{n-1}^i,$$

where $\{B_n^i\}$ are i.i.d. and drawn from $\mathcal{N}(0, (\sigma_B^2 \Delta t) I)$.

ii) Calculate predicted observation for each sample,

$$y_n^i = R(\eta_n'^i)^T r + W_n^i,$$

where $\{W_n^i\}$ are i.i.d. and drawn from $\mathcal{N}(0, (\sigma_W^2 / \Delta t) I)$.

iii) Calculate empirical covariance,

$$\Sigma_n = \frac{1}{N} \sum_{i=1}^{N} \log(\eta_n'^i) \log(\eta_n'^i)^T,$$

where the calculation of $\log(\cdot)$ is remarked at the end of the algorithm.

iv) Calculate the empirical covariance of the observation noise,

$$\mathrm{R}_n = \frac{1}{N} \sum_{i=1}^{N} y_n^i y_n^{iT}.$$

v) Calculate the gain,

$$\mathsf{K}_n = \Sigma_n H^T S_n^{-1},$$

where $S_n = H \Sigma_n H^T + \mathrm{R}_n$ and $H = [r]_\times$.

vi) Update the error samples (with the predicted observations $\{y_n^i\}$),

$$\eta_n^i = \eta_n'^i \otimes \exp\big(-\mathsf{K}_n (y_n^i - r)\big).$$

Filter estimate:

$$\text{(Propagation)} \quad \hat{q}'_n = \hat{q}_{n-1} \otimes \exp(\omega_{n-1}\,\Delta t),$$
$$\text{(Observation update)} \quad \hat{q}_n = \exp\left(\mathsf{K}_n(R(\hat{q}'_n)\,Y_n - r)\right) \otimes \hat{q}'_n.$$

Remark 5.3 *The function* $\log : S^3 \to \mathbb{R}^3$ *converts a quaternion to the corresponding rotation vector: Let* $q = (q_0, q_1, q_2, q_3)$*, and define* $\theta = 2\arccos(|q_0|) \in [0, \pi]$*. Then,*

$$\log(q) = \frac{\theta}{\sin(\frac{\theta}{2})}\,(q_1, q_2, q_3),$$

when $\theta \neq 0$*. Clearly,* $\log(q) = (0,0,0)$ *when* $\theta = 0$*.*

Remark 5.4 (Invariance properties) *The invariant filters exploit certain invariance or symmetry properties of the Lie group. A process on the Lie group is called invariant if it is unchanged when a chosen group action is applied to the state and the control input (e.g., the angular velocity* ω_t *in* (5.10a)*). Similarly, an observation model is called equivariant if it is unchanged under a chosen group action applied to the state, the external input (e.g., the reference vector* r *in* (5.10b)*) and the output variables. In particular, the left- and right-invariance are distinguished by the left and right multiplication when the group action is applied. The estimation error is also defined differently, with* $\eta_t := \hat{q}_t^{-1} \otimes q_t$ *for the left-invariant filter, and* $\eta_t := q_t \otimes \hat{q}_t^{-1}$ *for the right-invariant filter. More detailed explanation of the left and right-invariance and the resulting properties of the filters is contained in [20, 13].*

A prominent feature of the right-invariant filters is that the resulting error dynamics do not depend on the actual trajectory of the filter estimate nor the input angular velocity. Consequently, right-invariant filters may converge on a wider range of trajectories. Computationally, the gain can be conveniently computed off-line. In the subsequent numerical studies, only right-invariant filters, i.e., IEKF and IEnKF, are included since they were found to slightly outperform the left invariant filter.

5.5 Numerical Results for Attitude Estimation

For the attitude estimation problem, consider the following model with observations from both the accelerometer and magnetometer (see (5.3) and (5.4)),

$$\mathrm{d}q_t = \frac{1}{2} q_t \otimes \big(\omega_t \,\mathrm{d}t + \sigma_B \,\mathrm{d}B_t\big), \tag{5.21a}$$

$$\mathrm{d}Z_t = \begin{bmatrix} -R(q_t)^T & 0 \\ 0 & R(q_t)^T \end{bmatrix} \begin{bmatrix} r^g \\ r^b \end{bmatrix} \mathrm{d}t + \sigma_W \,\mathrm{d}W_t, \tag{5.21b}$$

where the model for angular velocity is taken from [200],

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

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.

This section presents various simulation results that compare the attitude filters in Sec. 5.4 with the proposed feedback particle filters. These filters are listed as below:

1) MEKF: the multiplicative EKF algorithm described in [124, 178] using the modified Rodrigues parameter (MRP).

2) USQUE: the unscented quaternion estimator described in [42] also using the MRP.

3) BPF: the bootstrap particle filter described in [32] also using the MRP. The progressive correction technique is applied with 20 splits of the likelihood function during the initial transient phase of the filter. The perturbation parameter $\sigma = 0.1$ (see Step vii) in the update step of BPF).

4) IEKF: the invariant EKF algorithm described in [13].

5) IEnKF: the invariant ensemble Kalman filter described in [13].

6) FPF-G: the FPF using the Galerkin gain function approximation: Algorithm 4 in Sec. 4.3 with the nine basis functions in Table C.1 in Appendix C.3.

7) FPF-K: the FPF using the kernel-based gain function approximation: Algorithm 5 in Sec. 4.4 with the parameter $\varepsilon = 1$.

8) FPF-C: the FPF using the constant gain approximation described in Sec. 2.4.

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 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 $\delta\varphi_t := 2\arccos(|\delta q_t^0|) \in [0, \pi]$, where δq_t^0 denotes the first component of δq_t.

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

$$\langle \delta\varphi^j \rangle_T := \frac{1}{T} \int_0^T \delta\varphi_t^j \, \mathrm{d}t, \tag{5.22}$$

and the time-averaged error of the M runs is defined as,

$$\langle \widehat{\delta\varphi} \rangle_T := \frac{1}{M} \sum_{j=1}^{M} \langle \delta\varphi_t^j \rangle_T. \tag{5.23}$$

The average error of the M Monte Carlo runs as a function of time is defined according to,

$$\widehat{\delta\varphi}_t := \frac{1}{M} \sum_{j=1}^{M} \delta\varphi_t^j, \tag{5.24}$$

The simulation parameters are as follows: The simulations are carried out over a finite time-horizon $t \in [0, T]$ with fixed time step Δt. The filters are all initialized with a Gaussian distribution, denoted as $\mathcal{N}(q_0, \Sigma_0)$, with mean q_0 and $\Sigma_0 = \sigma_0^2 I$ is a diagonal matrix representing the variance in each axis of the Lie algebra. For the BPF and FPF implementation, the initial set of particles are sampled from this distribution as follows: First, $\{v^i\}_{i=1}^N$ are sampled i.i.d. from the Gaussian distribution $\mathcal{N}(0, \Sigma_0)$ in $\mathbb{R}^3$. Next, the particles $\{q_0^i\}_{i=1}^N$ are obtained by,

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

The IEnKF also uses the same set of initial particles as the BPF and 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. The instability in FPF-G is exacerbated by possible ill-conditioning of the matrix A in constructing the Galerkin approximation (see Algorithm 4). 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 6) 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 sub-dividing technique has the same effect of "widening" the likelihood function as obtained by the progressive correction in BPF (see Sec. 5.4.3).

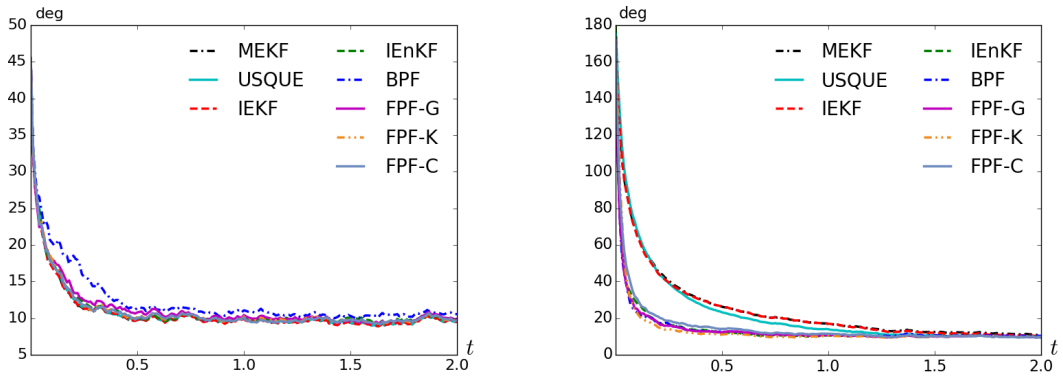
To provide a fair comparison, the same set of observations are used for all the continuous-time and the discrete-time algorithms. The particle-based filters use the same set of initial particles.

The nominal parameter values are chosen as: $T = 2$, $\Delta t = 0.01$, $N = 100$, $M = 100$, $T_f = 0.2$, $N_f = 100$. 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:

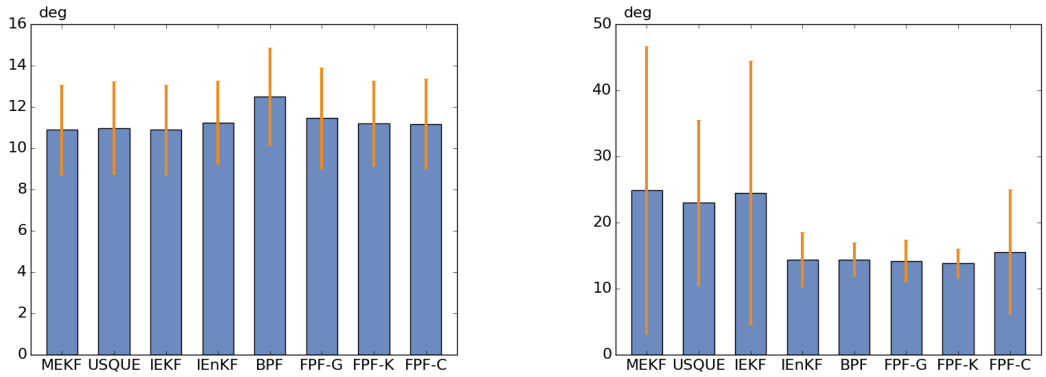
1. **The average error $\widehat{\delta\varphi}_t$ as a function of the initial uncertainty:** Figure 5.1 depicts the average error $\widehat{\delta\varphi}_t$ (see (5.24)) of the filters over $M = 100$ simulation runs, with two choices of initial variance: (a) $\Sigma_0 = 0.5236^2 I$ and (b) $\Sigma_0 = 1.0472^2 I$. The two cases correspond to a standard deviation of 30° and 60°, respectively. For the two priors, the 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 Figure 5.1 show that the performance is nearly identical across filters for case (a) when the initial uncertainty is small, except that the BPF has slightly higher error due to the additional diffusion after the resampling step. For case (b) when the initial uncertainty is large, the particle-based filters (IEnKF, BPF and FPF) exhibit superior performance compared to the EKFs and the unscented filter. 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 EKFs and the unscented filter.



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

Figure 5.1: Comparison of the average error $\widehat{\delta\varphi}_t$ for two prior distributions.



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

Figure 5.2: Statistical analysis of filter performance: The bars indicate the mean and the lines indicate the ± 1 standard deviation of $\{\langle \delta\varphi^j \rangle_T\}_{j=1}^M$ across $M = 100$ Monte-Carlo runs

As the results in Figure 5.1 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 Figure 5.2, which depicts the mean and standard deviation of $\{\langle \delta\varphi^j \rangle_T\}_{j=1}^M$ (see (5.22)). Apart from poorer performance on average, the EKFs and the unscented filter also exhibit a greater variability in performance across the Monte-Carlo runs. For some trajectories, these filters exhibit slow convergence because the gain becomes very small. The particle-based filters all exhibit superior error performance, except that the FPF with constant gain approximation exhibit larger simulation variance as the concentrated distribution assumption is violated in case (b).

2. **The time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$ as a function of the process noise:** In this simulation, the process noise $\sigma_B \in \{0.05, 0.2, 0.5, 1.0\}$ for fixed $\sigma_W = 0.05236$ and prior distribution according to case (b) in Figure 5.1. Figure 5.3 (a) depicts the time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$ (see (5.23)) across filters as the

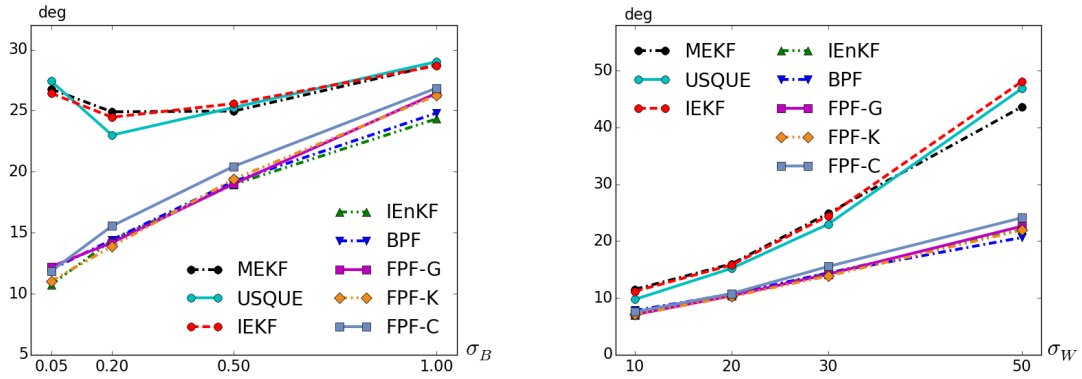
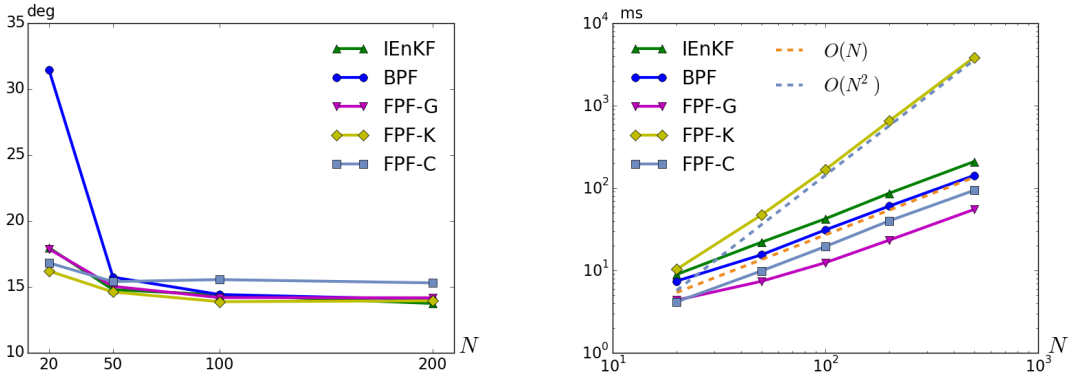


Figure 5.3: Time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$ of filters as a function of σ_B and σ_W. The value of σ_W is converted to the standard deviation (in degree) of the corresponding discrete-time observation model.

process noise parameter is varied. One would have expected the error to decrease monotonically with the σ_B value reduced. The fact that such is not the case for the EKFs and the unscented filter indicates that the relatively poor performance of these filters for small values of process noise is an artifact of the linearization assumption that lead to overly small gains. These small gains adversely affect the filter performance during the initial transients.

3. **The time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$ as a function of the observation noise:** In this simulation, the observation noise parameter $\sigma_W \in \{0.01745, 0.03491, 0.05236, 0.08727\}$ for fixed $\sigma_B = 0.2$ and prior distribution according to case (b) in Figure 5.1. The σ_W parameter values correspond to the choice of the standard deviation of 10°, 20°, 30° and 50° in the discrete-time model. Figure 5.3 (b) depicts the time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$. As expected, the error deteriorates as the observation noise increases. The particle filters not only continue to exhibit better performance but also the performance deterioration is more graceful for larger values of σ_W.

4. **The time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$ 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 Figure 5.1. Figure 5.4 (a) depicts the time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$. 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 issues in the gain computation.



(a) (b)

Figure 5.4: (a): Time-averaged error $\langle \widehat{\delta\varphi} \rangle_T$, and (b): mean computational time of a single propagation-update step, both as a function of the number of particles N.

5. **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 100 Monte Carlo runs) is depicted as a function of N in Figure 5.4 (b). The $O(N)$ and $O(N^2)$ lines are included to aid the comparison. The computational cost of particle 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 [13]. The experiments were conducted on a platform with an Intel i3-2120 3.3GHz CPU.

5.6 Filtering with a Bimodal Distribution

In this section, we consider the following static model:

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

where $\omega_t = (0,0,0)$. The prior distribution is assumed to be supported on the subgroup $SO(2)$, parametrized by the angle $\theta \in [-\pi, \pi)$. Its density is denoted as $\rho_0^*(\theta)$. An arbitrary element in $SO(2)$ is represented as $q = \big(\cos(\frac{\theta}{2}), 0, 0, \sin(\frac{\theta}{2})\big)$.

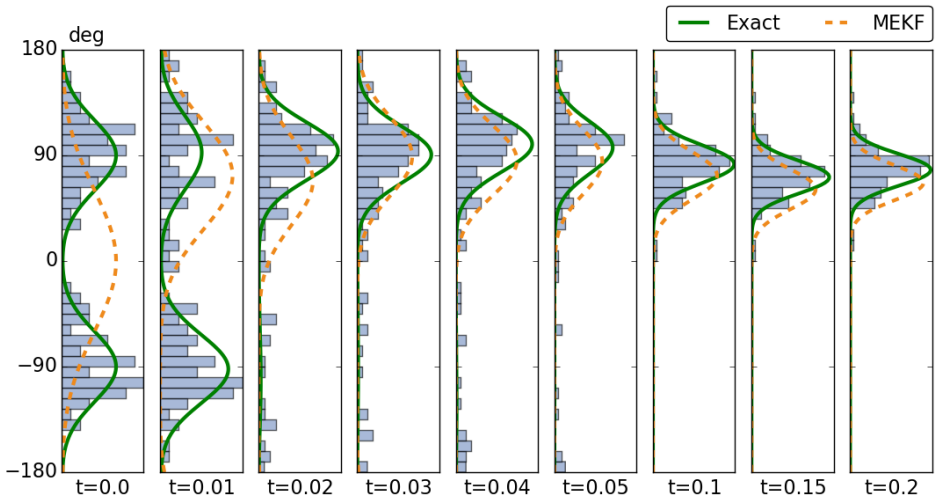


Figure 5.5: Density evolution of FPF-K on $SO(2)$ with bimodal distribution. The particle distributions are represented by histograms and are also compared with the exact posterior (5.25) (solid line) and the MEKF solution (dashed line). The FPF-K can handle general non-Gaussian posteriors.

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 a standard Wiener process in $\mathbb{R}^2$.

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

$$\rho^*(\theta,t) = (\text{const.})\,\exp\Big(\frac{1}{\sigma_W^2}h^T(\theta)Z_t - \frac{1}{2\sigma_W^2}|h(\theta)|^2 t\Big)\,\rho_0^*(\theta). \tag{5.25}$$

For the numerical results described next, the FPF is simulated according to (5.8) and (5.9):

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

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 a fixed time step $\Delta t = 0.01$. The following filters are simulated for a comparison:

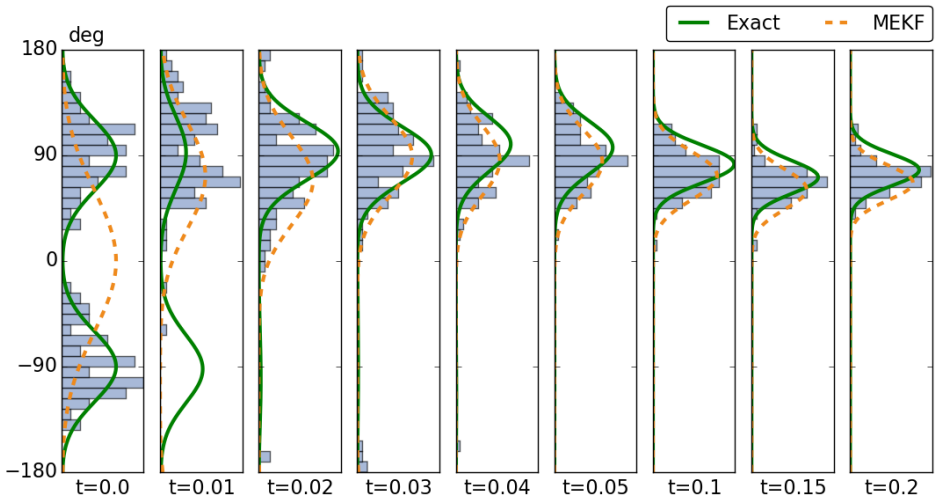


Figure 5.6: Density evolution of BPF on $SO(2)$ with bimodal initial distribution. The particle distributions are represented by histograms and are also compared with the exact posterior (5.25) (solid line) and the MEKF solution (dashed line). The BPF fails to capture the bimodal posterior at $t = 0.01$.

1) MEKF with a Gaussian prior $\mathcal{N}(0^\circ, 60^\circ)$.

2) BPF with $N = 100$ and $\sigma = 0.1$.

3) FPF-K with $N = 100$ and $\varepsilon = 0.2$.

The modified Rodrigues parameter in MEKF and BPF is replaced by the angle parameter θ. The same initial particles are used in BPF and FPF-K.

Figure 5.5 depicts the results of one simulation run that include the exact posterior (see (5.25)), the histogram of the particles of FPF-K, and the MEKF solution. Figure 5.6 depicts the corresponding simulation results produced by BPF. This example shows that the FPF-K algorithm can easily handle a general class of non-Gaussian distributions. In particular, at $t = 0.01$, the BPF fails to capture the bimodal posterior due to the importance sampling and resampling procedure that immediately eliminates particles with low importance weight.

A further investigation of the gain computation is provided in Figure 5.7 and 5.8, where the kernel-based gain approximation in FPF-K is plotted at selected time instants. The two gains depicted in the respective figures correspond to the two observations $h_1(\theta) = \cos(\theta)$ and $h_2(\theta) = -\sin(\theta)$. The FPF-K is seen to reasonably approximate the exact gain function.

The exact gain functions are obtained analytically as follows: Using the angle parametrization, the

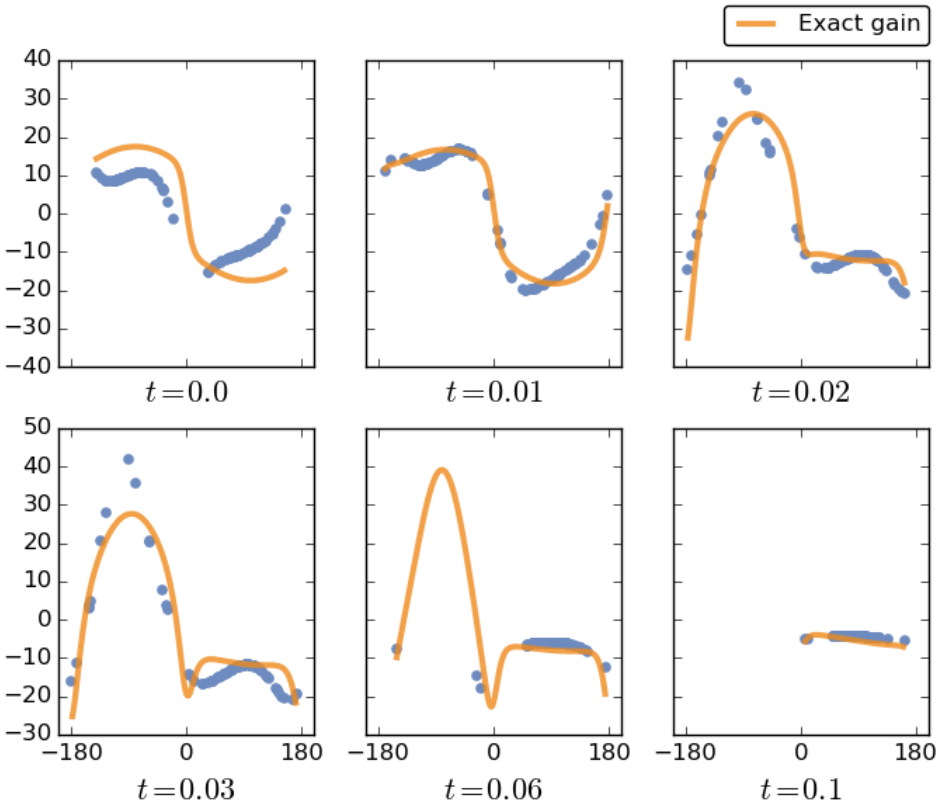


Figure 5.7: Kernel-based gain approximation on $SO(2)$, corresponding to the observation $h_1(\theta) = \cos(\theta)$. The blue dots depict the gain function approximation evaluated at the particles.

Poisson equation (2.28) restricted to $SO(2)$ is given by,

$$\frac{\partial}{\partial\theta}\Big(\rho(\theta)\,\frac{\partial\phi_j}{\partial\theta}(\theta)\Big) = -\big(h_j(\theta)-\hat{h}_j\big)\,\rho(\theta), \quad j=1,2,$$

where $\hat{h}_j := \int_{-\pi}^{\pi} h_j(\theta)\,\mathrm{d}\theta$, and the gain function $\mathsf{K}_j(\theta) = \frac{\partial\phi_j}{\partial\theta}(\theta)$. The explicit formulas for K_j is given by,

$$\mathsf{K}_j(\theta) = \frac{1}{\sigma_W^2}\,\frac{1}{\rho(\theta)}\Big(C - \int_{-\pi}^{\theta}\rho(\varphi)\big(h_j(\varphi)-\hat{h}_j\big)\,\mathrm{d}\varphi\Big), \tag{5.26}$$

where the constant C is determined by the periodicity condition $\phi(-\pi) = \phi(\pi)$ and is given by

$$C = \frac{\int_{-\pi}^{\pi}\frac{1}{\rho(\alpha)}\int_{-\pi}^{\alpha}\rho(\beta)\big(h_j(\beta)-\hat{h}_j\big)\,\mathrm{d}\beta\,\mathrm{d}\alpha}{\int_{-\pi}^{\pi}\frac{1}{\rho(\alpha)}\,\mathrm{d}\alpha}.$$

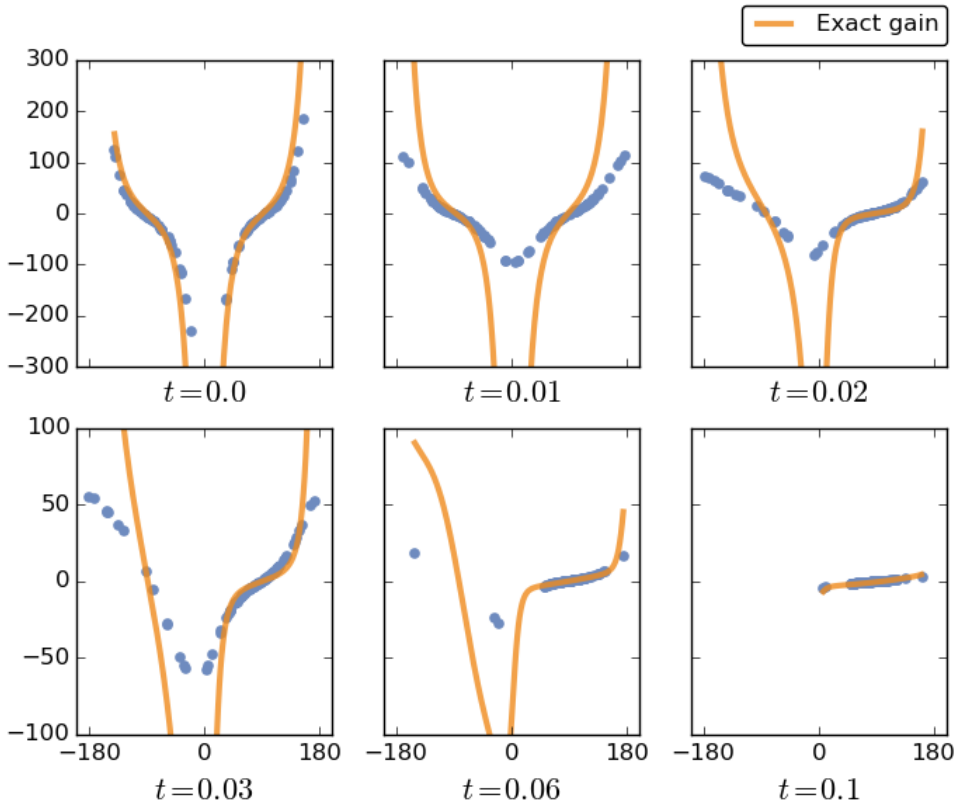


Figure 5.8: Kernel-based gain approximation on $SO(2)$, corresponding to the observation $h_2(\theta) = -\sin(\theta)$. The blue dots depict the gain function approximation evaluated at the particles.

5.7 Conclusions

In this chapter, the feedback particle filter was applied to the problem of attitude estimation – modeled as a continuous-time filtering problem on the Lie group $SO(3)$. The FPF was described using both the rotation matrix and the quaternion coordinates.

Numerical studies of the FPF and several state-of-the-art attitude filters were carried out for an attitude estimation problem on $SO(3)$, as well as a filtering problem on the subgroup $SO(2)$. For the attitude estimation problem, it was shown that:

1) The particle-based filters, including the IEnKF, BPF and FPF, exhibited superior estimation accuracy and simulation variance compared to the EKFs and the unscented filter. This was particularly true in the presence of large uncertainty in the prior distribution. In this case, the linearization assumption in the EKF no longer holds, which can result in a large estimation error.

2) Among the particle filters, the FPF with kernel-based gain function approximation (FPF-K) exhibited

superior accuracy when the number of particles is small, whereas the performance of the BPF severely deteriorated as the number of particles is reduced. The computational complexity of the FPF -K is approximately quadratic in the number of particles N, while the complexity of the other particle-based filters scales linearly with N.

For the filtering problem on $SO(2)$ with a bimodal initial distribution, it was shown that:

1) The feedback particle filter is better suited to capture general non-Gaussian posterior compared to the MEKF and the BPF, and

2) the kernel-based scheme provided a good approximation of the gain function, leading to accurate posterior computation.

It is the subject of ongoing and future work to compare the FPF with more attitude filters in the literature, such as the Kalman filter with norm constraints on $SO(3)$ [203, 58], the invariant UKF [41], and the filters based on a variational construction [87, 202], just to name a few. Particularly for the filtering problems on $SO(2)$, the PHD filter [126], the wrapped Kalman filter [127], and the minimum-energy filter [201] have appeared recently. In addition, the feedback particle filter may also be tested with filtering problems where translational motion and sensor bias are involved.

Chapter 6

Attitude Estimation of a Wearable Motion Sensor *

6.1 Introduction

This chapter is concerned with the problem of human motion tracking using body-worn sensors where the attitude (orientation) of the sensor must also be jointly estimated. The motivation for the problem comes from applications where sensors are embedded in apparel worn by the user, e.g., an activity tracker worn on the wrist. A body-worn sensor has a nominal location and orientation based on the apparel it is embedded in. However, there is often some degree of uncertainty, e.g., a wrist worn sensor may be worn interchangeably on the two wrists, or the piece of clothing may slip. Since the measurements are often recorded in the sensor body coordinate, changes in sensor orientation can cause the signals to transform via a rotation group.

The specific problem considered in this chapter involves the arm motion with wrist-worn inertial sensors that contain a 3-axis gyroscope and a 3-axis accelerometer. In the absence of motion, with the arm in its natural resting state, the attitude of the motion sensor is unobservable because the accelerometer's measurement of gravity vector alone can not distinguish between configurations of the sensor obtained by rotating the sensor around the wrist. In the presence of motion, considered here to be the swinging motion of the arm, the observability is shown to improve. The problem is mathematically formulated as a continuous-time filtering problem on the product Lie group $SO(3) \times SO(2)$, where $SO(3)$ captures the attitude of the motion sensor, and $SO(2)$ captures the phase of the periodic swing motion. The feedback particle filter algorithm is explicitly constructed in this setting, including a kernel-based gain function approximation scheme. Experimental results with real sensor data are provided to illustrate the tracking performance of the proposed filter.

Tracking of human motion using body-worn motion sensors (e.g., accelerometer and gyroscope) is a problem of substantial current interest due to its importance to a wide range of applications, e.g., activity

*The content of this chapter is related to the publication [206].

tracking [157, 173], medical rehabilitation [216, 92], virtual reality [189, 180] etc. A number of avant garde motion tracking products are reviewed in the March '16 issue of the IEEE Spectrum magazine. These products "*[embedded with] minuscule chips help to count steps, track calories burned, and monitor heart-rate* [111]." Companies such as Under Armour are developing 'smart clothing' products that they expect to launch in a few years [61]. Smart Health is widely regarded as a key driver of these products. According to the Economist magazine, "*[These products] may even provide a glimpse of the future of health care, in which a greater emphasis is placed on monitoring, using a variety of gizmos, to prevent disease, prolong lives and reduce medical costs* [53]."

In almost all the cases of practical interest, the mathematical models of human motion involve matrix Lie groups such as the special orthogonal group $SO(3)$ and the special Euclidean group $SE(3)$. The associated estimation/filtering problem therefore require consideration of the geometry of the non-Euclidean state space involving nonlinear dynamic and measurement models and non-Gaussian probability distributions. In many ways, these problems have an old and rich history specifically with applications in aerospace and target tracking [43]. In other ways, these problems are new: In contrast to an aircraft or a missile, humans exhibit far richer dynamic sets of behaviors which are computationally prohibitive to model a priori. This necessitates new paradigms and architectures for modeling and inference.

The proposed solution has three steps:

• **Modeling:** Mathematically, the motion dynamics are modeled on the product Lie group $SO(3) \times SO(2)$. This leads to coupled dynamics whereby the attitude of the wrist-worn sensor is coupled to the swinging motion of the arm. The measurement models for the gyroscope and the accelerometer are related to both the attitude and the (assumed) kinematic model of the motion.

• **Feedback particle filter:** The problem is to jointly estimate the motion of the arm and the attitude of the wrist-worn motion sensor. This problem is formulated as a continuous-time filtering problem on $SO(3) \times SO(2)$. The explicit form of the filter is described with respect to both the rotation matrix and the quaternion coordinates. A kernel-based approximation scheme is used to solve for the gain function on this product manifold.

• **Experiments.** The modeling and the inference algorithms are demonstrated in an experimental environment using a wrist-worn motion sensor. The tracking performance is assessed in terms of the attitude of the sensor and the relative sensor rotation on the wrist. An observability issue is briefly discussed for the arm

swing motion. Experimental results help illustrate the performance of the filter with real sensor data.

A number of authors have investigated the attitude estimation problem in the context of motion: Algorithmic approaches to solving this problem include the extended Kalman filter [167, 172, 198], the unscented Kalman filter [5, 55], and the bootstrap particle filter [171]. These filters are typically formulated in a discrete-time setting, and the intrinsic geometry of the underlying matrix Lie group is often not explicitly considered. For a more comprehensive review of attitude filters, c.f., [43].

An important theme in motion tracking applications pertains to the modeling of inertial sensors. Typically, the gyroscope measurements are incorporated in the filter propagation step [112, 198]. The accelerometer measurements are used to provide an estimate of the gravity vector typically for static or slow motion [118]. In problems involving dynamic and fast motion, the external acceleration needs to be distinguished from the gravity. A vector selection method that compares the true measurements with the gravity appears in [151]. In [112, 103], the external acceleration is incorporated in the system state with an assumed temporal model. A method closely related to our approach appears in [137], where the external acceleration is explicitly calculated using a kinematic model of the motion.

The remainder of this chapter is organized as follows: A brief overview of the product Lie group $SO(3) \times SO(2)$ is included in Sec. 6.2. The modeling framework and the estimation algorithms appears in Sec. 6.3 and Sec. 6.4, respectively. The experiment results are contained in Sec. 6.5.

6.2 Preliminaries: Geometry of $SO(3) \times SO(2)$

An element X of $G := SO(3) \times SO(2)$ is written as $X = (R, S)$, where $R \in SO(3)$ and $S \in SO(2)$. The projection maps are defined as,

$$P_1 : G \to SO(3), \quad P_1(X) = R,$$
$$P_2 : G \to SO(2), \quad P_2(X) = S.$$

The Lie algebra of G, denoted as $\mathscr{G}$, is a four-dimensional vector space that is identified with $so(3) \oplus$

$so(2)$ (see Proposition 3.14 in [104]). A basis $\{\widetilde{E}_1, \widetilde{E}_2, \widetilde{E}_3, \widetilde{E}_4\}$ of $\mathscr{G}$ is defined such that,

$$\begin{aligned} E_n &= \mathrm{d}P_1(\widetilde{E}_n) \quad \text{for } n = 1,2,3, \\ E &= \mathrm{d}P_2(\widetilde{E}_4), \end{aligned}$$

where $\{E_1, E_2, E_3\}$ is the orthonormal basis of $so(3)$ (see (2.13)), E is the basis of $so(2)$, and $\mathrm{d}P_1$, $\mathrm{d}P_2$ denote the differential of the projection maps at the identity element of G.

For a function $f : G \to \mathbb{R}$, the coordinates of $\mathrm{grad}(f)$, evaluated at (R,S), are obtained as,

$$\begin{aligned} \widetilde{E}_n \cdot f(R,S) &= E_n \cdot f(R,S) \quad \text{for } n = 1,2,3, \\ \widetilde{E}_4 \cdot f(R,S) &= E \cdot f(R,S), \end{aligned} \tag{6.1}$$

where $E_n \cdot f(R,S) := \frac{\mathrm{d}}{\mathrm{d}\tau}\big|_{\tau=0} f\big(R\exp(\tau E_n),\ S\big)$ for $n = 1,2,3$, and $E \cdot f(R,S) := \frac{\mathrm{d}}{\mathrm{d}\tau}\big|_{\tau=0} f\big(R,\ S\exp(\tau E)\big)$.

6.3 Modeling

This section describes the modeling paradigm for the motion tracking problem, whereby the motion is considered to be the arm swing. The objective is to formulate the motion tracking as a continuous-time filtering problem on the product $SO(3) \times SO(2)$.

6.3.1 Kinematics Model of Arm Swing

The inertial frame, represented as the $x-y-z$ coordinate system, is fixed at the shoulder. The arm swings in the $x-y$ plane in a quasi-periodic manner. The motion is modeled via the time-evolution of and angle θ_t (see Figure 6.1). The motion sensor is worn on the wrist, and $x'-y'-z'$ are the axes of the body frame attached to the sensor. It is assumed that the arm swings in a plane, remains straight, and that the length of the arm, denoted as L, is known. Therefore, the position of the motion sensor, expressed in the inertial frame, is uniquely described by the angle θ_t.

The model for the arm angle θ_t is parametrized in terms of a phase variable,

$$\theta_t = f(\varphi_t; \lambda), \tag{6.2}$$

where $\varphi_t \in S^1$ is referred to as the *phase* of the swing motion, and λ denotes model parameters.

Example: The sinusoidal model of the arm swing is of the form,

$$f(\varphi_t; \theta_m, \theta_0) = \theta_m \sin(\varphi_t) + \theta_0, \tag{6.3}$$

where the parameter θ_m determines the amplitude of the arm swing, and the parameter θ_0 models the asymmetry in the arm swing with respect to the inertial frame: The maximum swing angle is typically larger during the forward portion of the arm swing. □

The phase φ_t is modeled as a noisy oscillator,

$$\mathrm{d}\varphi_t = \omega \, \mathrm{d}t + \sigma_B^s \, \mathrm{d}B_t^s, \qquad \mod 2\pi, \tag{6.4}$$

where the constant ω represents the frequency of the arm motion, B_t^s is a standard Wiener process in $\mathbb{R}$, and σ_B^s is a positive scalar. The process noise is used to model the random variability in the motion. The oscillator model has been employed for modeling periodic motions, e.g., gaits [176].

For the purpose of formulating and solving the filtering problem in Sec. 6.4, it is convenient to identify S^1 with the Lie group $SO(2)$. The equivalent process model of (6.4) in $SO(2)$ is given by,

$$\mathrm{d}S_t = S_t(\omega E)\, \mathrm{d}t + S_t E \circ (\sigma_B^s \, \mathrm{d}B_t^s), \tag{6.5}$$

where $S_t \in SO(2)$ is represented using φ_t as,

$$S_t = \begin{bmatrix} \cos(\varphi_t) & -\sin(\varphi_t) \\ \sin(\varphi_t) & \cos(\varphi_t) \end{bmatrix}, \tag{6.6}$$

and

$$E = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

denotes a basis of the Lie algebra $so(2)$.

The parameters in the swing model are listed in Table 6.1.

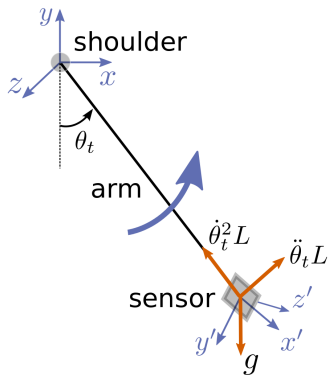


Figure 6.1: Illustration of arm swing

Table 6.1: Model parameters

θ_t	angle of swing at time t	φ_t	phase variable at time t
θ_m	model parameter	θ_0	model parameter
ω	frequency of swing	L	arm length

6.3.2 Sensor Model

Kinematic model of the attitude: The angular velocity of the motion sensor, expressed in the inertial frame, is denoted as $\Omega_t \in \mathbb{R}^3$. For the planar arm swing model described in Sec. 6.3.1, $\Omega_t = (0, 0, \dot{\theta}_t)$. The kinematic model of the sensor is given by,

$$\mathrm{d}R_t = [\,\Omega_t\,]_\times R_t \,\mathrm{d}t + R_t \circ [\,\sigma_B^r \,\mathrm{d}B_t^r\,]_\times, \tag{6.7}$$

where $R_t \in SO(3)$ denotes the attitude of the sensor at time t, with respect to the inertial frame; B_t^r is a standard Wiener process in $\mathbb{R}^3$, accounting for random effects that cause small variation in sensor attitude.

The motion sensor is equipped with a gyroscope and an accelerometer, whose models are described as follows:

Gyroscope: A gyroscope measures the angular velocity of the sensor in the sensor body frame. It is modeled as,

$$\mathrm{d}Z_t^g = R_t^T \Omega_t \,\mathrm{d}t + \sigma_W \,\mathrm{d}W_t^g, \tag{6.8}$$

where Z_t^g denotes the gyroscope observation at time t, W_t^g is a standard Wiener process in $\mathbb{R}^3$, and σ_W is a positive scalar.

Accelerometer: An accelerometer measures the *specific acceleration* (that is, the external acceleration

minus the gravity) in the body frame of the sensor [106]. For a prescribed θ_t, the specific acceleration is easily obtained, expressed with respect to the $x-y-z$ inertial frame as,

$$\begin{aligned} A_t = \big(&\ddot{\theta}_t L \cos(\theta_t) - \dot{\theta}_t^2 L \sin(\theta_t), \\ &\ddot{\theta}_t L \sin(\theta_t) + \dot{\theta}_t^2 L \cos(\theta_t) + g,\ 0\big), \end{aligned} \tag{6.9}$$

where $\ddot{\theta}_t L$ and $\dot{\theta}_t^2 L$ represent the magnitude of the tangential and the centripedal acceleration, respectively (see Figure 6.1).

The model for the accelerometer is simply obtained by transforming the vector A_t to the sensor body frame,

$$\mathrm{d}Z_t^a = R_t^T A_t \,\mathrm{d}t + \sigma_W \,\mathrm{d}W_t^a. \tag{6.10}$$

One difficulty arises because θ_t is not directly prescribed but must be estimated from the measurements. The difficulty is circumvented by assuming the following models for $\dot{\theta}_t$ and $\ddot{\theta}_t$ in terms of the phase variable φ_t:

$$\dot{\theta}_t := \omega\, \theta_m \cos(\varphi_t), \tag{6.11}$$

$$\ddot{\theta}_t := -\omega^2\, \theta_m \sin(\varphi_t). \tag{6.12}$$

These models are used in defining the acceleration vector A_t in (6.9), which is subsequently used for the accelerometer model (6.10). Finally, W_t^a in (6.10) is the observation noise modeled here as a standard Wiener process in $\mathbb{R}^3$, which is furthermore assumed to be independent of the process noise.

Remark 6.1 *The gyroscope model* (6.8) *and the accelerometer model* (6.10) *have the general form,*

$$\mathrm{d}Z_t^{(\cdot)} = R_t^T r_t \,\mathrm{d}t + \sigma_W \,\mathrm{d}W_t^{(\cdot)},$$

where $r_t \in \mathbb{R}^3$ is a time-varying reference vector in the inertial frame. Such a model is somewhat different from the standard model in attitude estimation, where the reference vector is a known time-independent constant vector (e.g., the gravity and the (local) magnetic field [118]). Attitude observers that handles time-varying reference vectors appear in [66, 179].

In a typical attitude estimation problem, at least two reference vectors are required to ensure observability and used in the filter update [123]. In our approach, the gyroscope is modeled such that it measures a reference vector, namely the angular velocity vector Ω_t. This new usage of gyroscope is different from a typical attitude filter where the gyroscope measurements are incorporated in the propagation step. □

6.4 Estimation

6.4.1 Filtering Problem

The filtering problem is to jointly estimate the attitude of the motion sensor and the phase of the arm swing. The state space G is a four-dimensional product Lie group $SO(3) \times SO(2)$. Using (6.7), (6.5) and (6.8), (6.10), the signal and the observation models are,

$$\text{Signal: } \begin{cases} \mathrm{d}R_t = [\,\Omega_t\,]_\times R_t\,\mathrm{d}t + R_t \circ [\,\sigma_B^r\,\mathrm{d}B_t^r\,]_\times, \\ \mathrm{d}S_t = S_t(\omega E)\,\mathrm{d}t + S_t E \circ (\sigma_B^s\,\mathrm{d}B_t^s), \end{cases}$$

$$\text{Observation: } \mathrm{d}Z_t = h(R_t, S_t)\,\mathrm{d}t + \sigma_W\,\mathrm{d}W_t,$$

where $h : G \to \mathbb{R}^6$ encapsulates the sensor models given in (6.8) and (6.10) within a single vector-valued function, and B_t^r, B_t^s, W_t are mutually independent standard Wiener processes of appropriate dimension.

The filtering objective is to numerically approximate the conditional distribution of (R_t, S_t) given the observation history $\mathscr{Z}_t = \sigma(Z_s : s \le t)$.

6.4.2 FPF Algorithm

Analogous to its construction for the attitude estimation problem on $SO(3)$ (see Sec. 5.3), the FPF is comprised of N stochastic processes $\{(R_t^i, S_t^i)\}_{i=1}^N$ that evolve on G. The particles evolve according to the sdes,

$$\mathrm{d}R_t^i = [\,\Omega_t^i\,]_\times R_t^i\,\mathrm{d}t + R_t^i \circ [\,\sigma_B^r\,\mathrm{d}B_t^{r,i}\,]_\times + R_t^i\,[\,\mathsf{K}^r(R_t^i, S_t^i, t) \circ \mathrm{d}\mathrm{I}_t^i\,]_\times, \tag{6.13a}$$

$$\mathrm{d}S_t^i = S_t^i(\omega^i E)\,\mathrm{d}t + S_t^i E \circ (\sigma_B^s\,\mathrm{d}B_t^{s,i}) + S_t^i E\big(\mathsf{K}^s(R_t^i, S_t^i, t) \circ \mathrm{d}\mathrm{I}_t^i\big). \tag{6.13b}$$

For the sake of self-contained exposition, models for each of the sdes is summarized next:

1) ω^i are sampled i.i.d. from a uniform distribution on the interval $[\omega_0 - 2\pi\gamma, \omega_0 + 2\pi\gamma]$ for certain ω_0 and γ. The variability of ω^i is used to capture the uncertainty in the actual frequency of the motion.

2) $\Omega_t^i = (0, 0, \dot{\theta}_t^i)$, where $\dot{\theta}_t^i = \omega^i \theta_m \cos(\varphi_t^i)$ according to (6.11). Here, φ_t^i denotes the phase of the i-th particle, and it is related to S_t^i by (6.6).

3) $B_t^{r,i}$ and $B_t^{s,i}$ for $i = 1, ..., N$ are mutually independent standard Wiener processes in $\mathbb{R}^3$ and $\mathbb{R}$, respectively.

4) $\mathsf{K}^r : G \times [0,T] \to \mathbb{R}^{3\times 6}$ and $\mathsf{K}^s : G \times [0,T] \to \mathbb{R}^{1\times 6}$ denote the gain functions. By slight abuse of notation, their vertical concatenation is denoted as $\mathsf{K} := (\mathsf{K}^r, \mathsf{K}^s) \in \mathbb{R}^{4\times 6}$.

5) The innovation process $\mathrm{I}_t^i \in \mathbb{R}^6$ is defined as,

$$\mathrm{d}\mathrm{I}_t^i = \mathrm{d}Z_t - \frac{h(R_t^i, S_t^i) + \hat{h}_t}{2}\,\mathrm{d}t,$$

where $\hat{h}_t := \pi_t(h)$, and π_t denotes the conditional distribution of $X_t^i := (R_t^i, S_t^i)$. In a numerical implementation, $\hat{h}_t \approx \frac{1}{N}\sum_{i=1}^N h(R_t^i, S_t^i) =: \hat{h}_t^{(N)}$.

The gain functions K is obtained as follows: For $j = 1, 2, ..., 6$, the j-th column of K contains the coordinates of the vector-field $\mathrm{grad}(\phi_j)$, where the function ϕ_j is a solution to the Poisson equation,

$$\begin{aligned} &\pi_t\big(\langle \mathrm{grad}(\phi_j), \mathrm{grad}(\psi)\rangle\big) = \frac{1}{\sigma_W^2}\,\pi_t\big((h_j - \hat{h}_j)\psi\big), \\ &\pi_t(\phi_j) = 0 \quad \text{(normalization)}, \end{aligned} \tag{6.14}$$

for all test functions ψ.

6.4.3 Numerical implementation

The FPF algorithm is numerically implemented using the quaternion coordinates. In these coordinates, the sdes (6.13a)-(6.13b) are expressed as,

$$\mathrm{d}q_t^{r,i} = \frac{1}{2}\, q_t^{r,i} \otimes \mathrm{d}v_t^{r,i}, \tag{6.15a}$$

$$\mathrm{d}q_t^{s,i} = \frac{1}{2}\, q_t^{s,i} \otimes \mathrm{d}v_t^{s,i}, \tag{6.15b}$$

where $(q_t^{r,i}, q_t^{s,i})$ is the quaternion representation of the i-th particle, and $v_t^{r,i}$, $v_t^{s,i} \in \mathbb{R}^3$ evolve according to,

$$\begin{aligned} \mathrm{d}v^{r,i} &= R(q_t^{r,i})^T \Omega_t \, \mathrm{d}t + \sigma_B^r \, \mathrm{d}B_t^{r,i} + \mathsf{K}^r(q_t^{r,i}, q_t^{s,i}, t) \circ \mathrm{dI}_t^i, \\ \mathrm{d}v^{s,i} &= [\, \omega^i \, \mathrm{d}t + \sigma_B^s \, \mathrm{d}B_t^{s,i} + \mathsf{K}^s(q_t^{r,i}, q_t^{s,i}, t) \circ \mathrm{dI}_t^i \,]^\vee, \end{aligned}$$

where $[\,\omega\,]^\vee := (0, 0, \omega) \in \mathbb{R}^3$ for $\omega \in \mathbb{R}$, and $R(q)$ converts a quaternion to its associated rotation matrix (c.f., formula (2.16)).

Algorithm 7 Feedback particle filter for motion tracking

1: **initialization:** Samples $\{(q_0^{r,i}, q_0^{s,i})\}_{i=1}^N$ i.i.d. from π_0^*

2: Assign $t = 0$

3: **iteration:** from t to $t + \Delta t$

4: Calculate $\hat{h}_t^{(N)} = \frac{1}{N} \sum_{i=1}^N h(q_t^{r,i}, q_t^{s,i})$

5: **for** $i = 1$ to N **do**

6: Generate a sample, $\Delta B_t^{r,i}$, from $N\big(0, (\Delta t) I\big)$

7: Generate a sample, $\Delta B_t^{s,i}$, from $N\big(0, \Delta t\big)$

8: Calculate the error $\Delta \mathrm{I}_t^i := \Delta Z_t - \frac{1}{2}\left(h(q_t^{r,i}, q_t^{s,i})) + \hat{h}_t^{(N)}\right) \Delta t$

9: Calculate gain function $\mathsf{K}(q_t^{r,i}, q_t^{s,i})$ using the kernel-based scheme (see Sec. 6.4.3)

10: Calculate $\Delta v_t^{r,i} = \Omega_t \, \Delta t + \sigma_B^r \, \Delta B_t^{r,i} + \mathsf{K}^r(q_t^{r,i}, q_t^{s,i}) \, \Delta \mathrm{I}_t^i$

11: Calculate $\Delta v_t^{s,i} = [\, \omega^i \, \Delta t + \sigma_B^s \, \Delta B_t^{s,i} + \mathsf{K}^s(q_t^{r,i}, q_t^{s,i}) \, \Delta \mathrm{I}_t^i \,]^\vee$

12: Propagate the particle $q_t^{r,i}$ according to (see [178], and $|\cdot|$ denotes the Euclidean norm in $\mathbb{R}^3$)

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

13: Propagate the particle $q_t^{s,i}$ similarly

14: **end for**

15: **return:** empirical mean of $\{(q_{t+\Delta t}^{r,i}, q_{t+\Delta t}^{s,i})\}_{i=1}^N$ (see Line 13 in Algorithm 6)

16: Assign $t = t + \Delta t$

The FPF algorithm using the quaternion coordinates is described in Algorithm 7. The algorithm sim-

ulates N particles, $\{(q_t^{r,i}, q_t^{s,i})\}_{i=1}^N$, according to the sde's (6.15a) and (6.15b), with the initial conditions $\{(q_0^{r,i}, q_0^{s,i})\}_{i=1}^N$ sampled i.i.d. from a given prior distribution π_0^* supported on $SO(3) \times SO(2)$.

Gain function approximation: In the experimental results to be presented in Sec. 6.5, the FPF is implemented using the kernel-based gain function approximation whose numerical procedure is described in Sec. 4.4. The kernel function is defined as,

$$k^{(\varepsilon)}(X^i, X^j) := \frac{1}{(4\pi\varepsilon)^{d/2}} \exp\Big(-\frac{\zeta^2(X^i, X^j)}{4\varepsilon}\Big), \tag{6.17}$$

where $d = 4$ for the product Lie group $G = SO(3) \times SO(2)$, and $\zeta : G \times G \to \mathbb{R}$ denotes a distance metric on G. One choice of the distance metric is,

$$\zeta^2(X^i, X^j) = |R^i - R^j|_F^2 + \frac{1}{2}|S^i - S^j|_F^2, \tag{6.18}$$

where $|\cdot|_F$ is the Frobenius norm of a matrix. This metric is induced from the Euclidean distance metric defined in $\mathbb{R}^9 \times \mathbb{R}^2$ in which G is smoothly embedded. Note that the time index t is omitted.

The gain function is denoted as $\mathsf{K} = (\mathsf{k}_1, ..., \mathsf{k}_4)$ with coordinates $\mathsf{k}_n = \widetilde{E}_n \cdot \phi$, where $\{\widetilde{E}_1, \widetilde{E}_2, \widetilde{E}_3, \widetilde{E}_4\}$ is the basis of the Lie algebra of G. The formulae of k_n are obtained similarly as the formula (4.18) in Sec. 4.4,

$$\widetilde{E}_n \cdot \phi(X^i) = \varepsilon\, \widetilde{E}_n \cdot h(X^i) - \frac{1}{4\varepsilon}\big[\big(M_n \Phi\big)_i - \big(M_n \mathbf{1}\big)_i \big(T^{(\varepsilon,N)} \Phi\big)_i\big],$$

for $n = 1, ..., 4$, where $\mathbf{1} = (1, 1, ..., 1) \in \mathbb{R}^N$, and the entries of the $N \times N$ matrix M_n are calculated as,

$$(M_n)_{ij} = T_{ij}^{(\varepsilon,N)}\, \widetilde{E}_n \cdot \zeta^2(X^i, X^j),$$

and the elements of the Markov matrix $T^{(\varepsilon,N)}$ are given by (4.13). The derivative of ζ^2 is taken with respect to its first argument. For the metric defined by (6.18), the explicit expressions of $\widetilde{E}_n \cdot \zeta^2$ are obtained following the formula (6.1) in Sec. 6.2:

$$\widetilde{E}_n \cdot \zeta^2(X^i, X^j) = -2\,\mathrm{Tr}(R^i E_n R^j), \quad \text{for } n = 1, 2, 3,$$
$$\widetilde{E}_4 \cdot \zeta^2(X^i, X^j) = -\mathrm{Tr}(S^i E S^j).$$

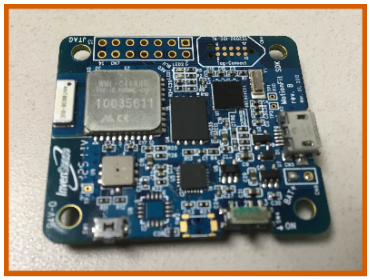
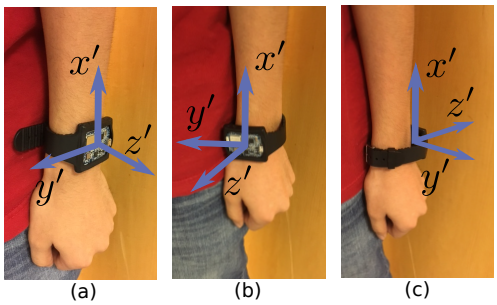


Figure 6.2: The InvenSense chip.



(a) (b) (c)

Figure 6.3: Sensor orientation on the wrist: (a) Nominal orientation; (b) Rotation of 90° clockwise about the wrist ($-90°$ around x'-axis); (c) Rotation of 90° counter-clockwise about the wrist (90° around x'-axis).

6.5 Experiments

6.5.1 Experimental Testbed

The experiments were carried out with the InvenSense MP MPU-9150 chip (see Figure 6.2). The InvenSense hardware includes a 3-axis gyroscope and a 3-axis accelerometer and also provides a USB and Bluetooth connectivity back to a host computer. For the experiments, the InvenSense chip is used as the wearable device, worn on the user's wrist. The *nominal* orientation of the sensor faces outwards but the sensor orientation may change, typically rotate around the wrist. Figure 6.3 depicts the nominal sensor orientation as well as some variations.

One of the motivations of the present study is to use the dynamic motion of arm swing to better estimate the sensor orientation. Note that, in the absence of motion, the accelerometer only measures the (downward) gravity vector. Any rotation of the sensor around the wrist will yield the same accelerometer reading. Hence, in the absence of motion, the problem is not observable. The experimental results presented next demonstrate that the observability is improved by using dynamic information from the sensor while the user is performing arm swing.

Table 6.2: Experiment parameters

Model	θ_m	θ_0	L	g	σ_B^r	σ_B^s	σ_W
	30°	10°	0.6	9.8	0.2	0.2	0.4
FPF	N	ω_0	γ	ε	Δt	Initial dist.	
	50	1.8π	0.5	0.5	0.02	Uniform	

6.5.2 Filter Parameters

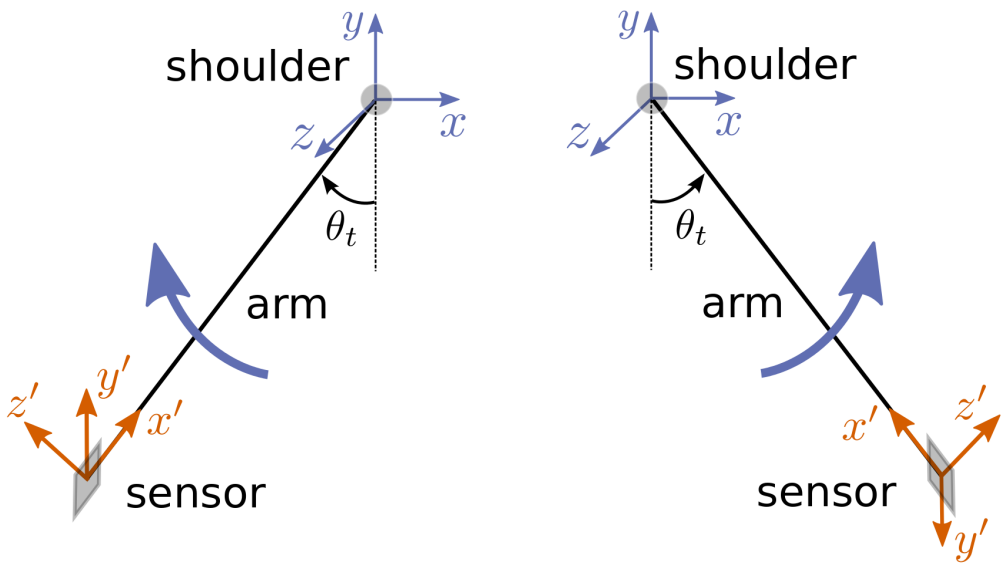
While the arm swing is performed, the measurements from the 3-axis gyroscope and 3-axis accelerometer were sampled at a fixed sampling rate of 50 Hz. Figure 6.5 depicts the raw sensor measurements in the six sensor axes from a single experiment run.

The sensor measurements were processed by the FPF algorithm. Table 6.2 tabulates the filter parameters used: The model parameters θ_m, θ_0 and ω_0 were chosen based on an offline assessment of the natural arm swing. For example, the frequency ω_0 was chosen to be close to the natural frequency of a rigid rod of length L with uniform mass distribution. The uncertainty in these model parameters is modeled by choosing relatively large values of process noise parameters σ_B^r, σ_B^s and the frequency de-tuning parameter γ. The observation noise parameter σ_W is chosen based on assessment of various sources of the sensor noise. This parameter affects the convergence speed of the filter. In numerical implementation of the filter, too small a choice of the parameter σ_W can lead to numerical instabilities on account of large gain values during the initial transients. Finally, ε is a parameter required by the by the kernel-based gain function approximation.

Once the numerical parameter values have been selected, the FPF sde is simulated using the numerical algorithm described in Algorithm 7. The particles in FPF are initially sampled i.i.d. from the uniform distribution on the compact Lie group $SO(3) \times SO(2)$. In the quaternion coordinates, the uniform sampling scheme is taken from [159]. A fixed discrete time-step equal to the sampling time-step $\Delta t = \frac{1}{50}$ seconds is used for numerical integration.

6.5.3 Performance Metric

During each experiment run, the user wears the motion sensor in a fixed orientation. Denote q_t and $\hat{q}_t$ as the ground-truth and the estimated attitude, respectively. Denote $\delta q_t := \hat{q}_t^{-1} \otimes q_t$. The performance metric is defined as $\delta\alpha_t = 2\arccos(|\delta q_t^0|) \in [0^\circ, 180^\circ]$, where δq_t^0 is the first component of δq_t. Physically, the metric $\delta\alpha_t$ represents rotation angle between the ground truth and the estimated attitude.



(a) "fictitious" motion (swinging backward) (b) actual motion (swinging forward)

Figure 6.4: In the case of a symmetric model of the arm swing, the two sensor orientations and swing patterns, as depicted in (a) and (b) respectively, yield the same sensor measurements. The sensor orientation differs by 180° for the two motion patterns.

6.5.4 Experimental Results

In an experimental run, referred to as Experiment-1, the user performed arm swing with the sensor orientation chosen to be a rotation of (approximately) 90° clockwise from the nominal orientation (see Figure 6.3 (b)). The tracking results for the FPF are illustrated in Figure 6.5. The experimentally obtained sensor measurements from the gyroscope and the accelerometer are depicted, together with the prediction $\hat{h}_t^{(N)}$ computed using the FPF (see Line 4 in Algorithm 7). It is seen that the predictions closely follow the measurements, indicating that the filter estimates of attitude and phase both converge.

6.5.5 Observability Issue

In the case of a symmetric model of the arm swing, i.e., $\theta_0 = 0$ in the model (6.3), the following two sensor orientations and swing patterns yield the same sensor measurements:

1) The forward swing motion (see Figure 6.4 (b)) with the sensor orientation depicted in Figure 6.3(b);

2) The backward swing motion (see Figure 6.4 (a)) with the sensor orientation depicted in Figure 6.3 (c). The sensor orientation differs from case 1) by 180°.

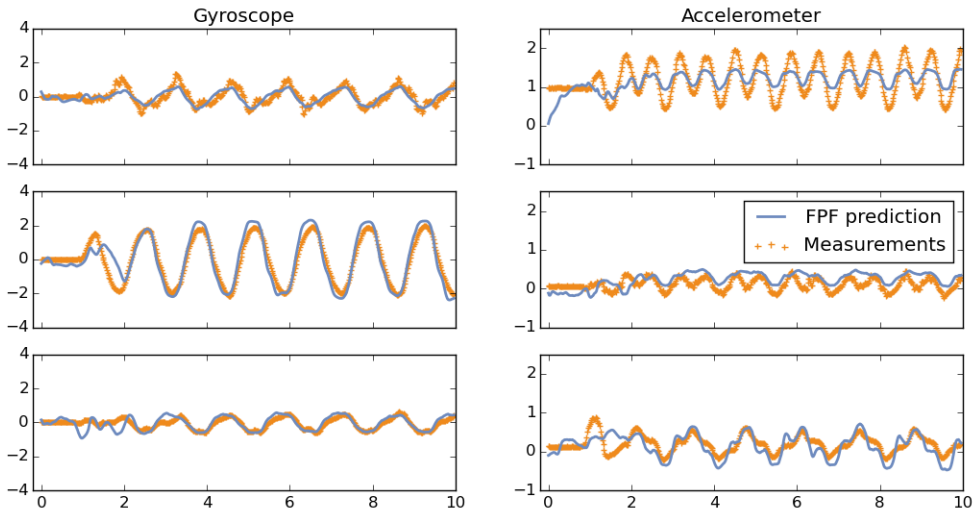


Figure 6.5: Sensor measurements and FPF prediction $\hat{h}_t^{(N)}$ in Experiment-1. $\hat{h}_t^{(N)}$ closely follows the measurements.

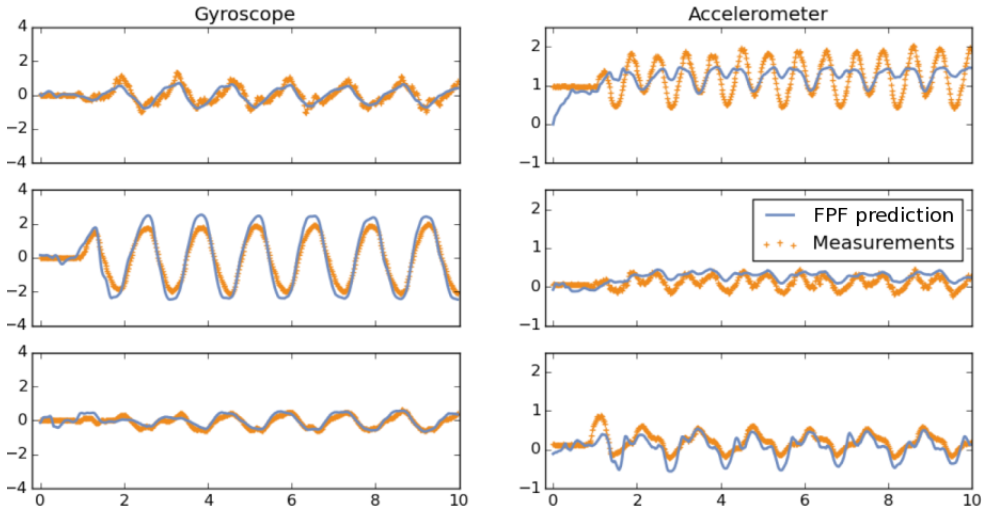
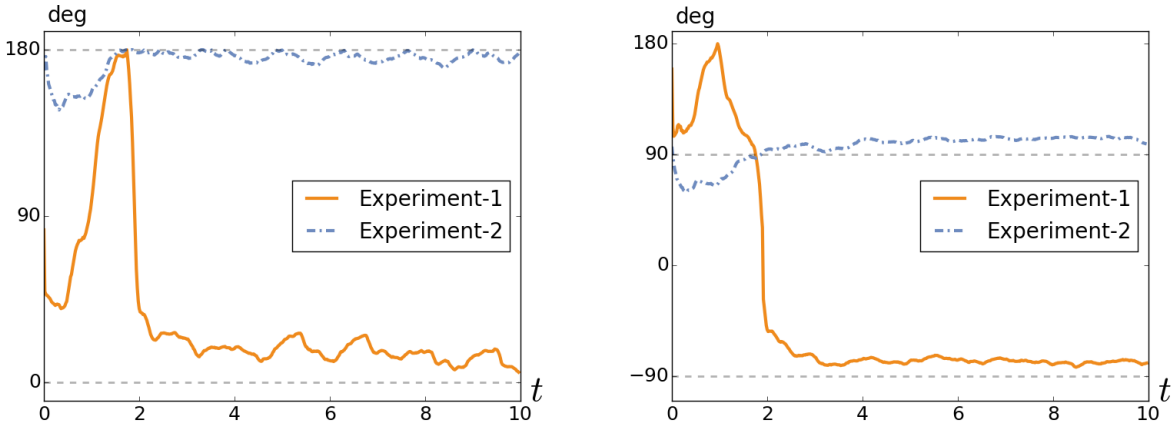


Figure 6.6: Sensor measurements and FPF prediction $\hat{h}_t^{(N)}$ in Experiment-2. The results are nearly identical with the case depicted in Figure 6.5.



(a) Estimation error of the sensor attitude

(b) Estimated sensor rotation on the wrist

Figure 6.7: FPF may converge to two equilibria. The FPF in Experiment-1 (orange lines) converges to the true sensor attitude (see (a)), which is a rotation of -90° from the nominal orientation around the wrist (see (b)); The FPF in Experiment-2 (blue dashed lines) converges to an attitude with 180° error.

In our experiments, the asymmetry was found to be small, $\theta_0 = 10^\circ$. As a result, the filter sometimes converged to the out-of-phase equilibria whereby the error is 180°. Figure 6.6 depicts the results of another independent experimental run, referred to as Experiment-2. Figure 6.5 and Figure 6.6 show that both the sensor measurements and FPF predictions are nearly identical in the two experiments. However, as depicted in Figure 6.7 (a), the FPF in Experiment-1 converges to the true sensor attitude, whereas the FPF in Experiment-2 converges to an attitude with error of 180°. Equivalently, as shown in Figure 6.7 (b), the two FPFs converge to sensor orientations that correspond to rotation by -90° and 90°, respectively, from the nominal orientation around the wrist. Detailed analysis of the loss of observability and strategies to mitigate it is a subject of continuing work.

6.6 Conclusions

In this chapter, we presented experimental results for the problem of jointly estimating the motion of a swinging arm and the attitude of the motion sensor. The conclusions are as follows:

1) The presence of dynamic arm motion serves to improve the overall observability of the sensor attitude. However, for a symmetric motion, two configurations (where the sensor is rotated by 180°) cannot be distinguished.

2) A new modeling paradigm was proposed. The accelerometer was modeled without simplifications and approximations that have appeared in the literature. A phase variable was introduced to capture periodicity of the motion.

3) The feedback particle filter was shown to easily handle the highly nonlinear aspects of dynamics and sensing that arise in motion tracking applications even as simple as arm swing. Using a kernel-based gain function approximation, the FPF achieved satisfactory tracking performance with moderate computation load and a degree of robustness against model uncertainties.

There are several directions for future work on this problem:

1) In a realistic setting, it will be important to include additional degrees of freedom, e.g., due to bending and twisting of the arm. It will be useful to investigate the influence of these types of motion modalities.

2) A comparison of the FPF with other filters, e.g., the EKF and the bootstrap particle filter, will be useful to provide insights as for why the feedback structure in the FPF may be advantageous in the presence of severe nonlinearities.

3) For real-time implementations, it will be important to develop more computationally efficient algorithms for the gain function approximation.

Chapter 7

Numerical Results for the Global Optimization Problem *

7.1 Introduction

In this chapter, results of numerical experiments are described for some benchmark global optimization problems. For these problems, performance comparison results between the controlled particle filter and other state-of-the-art model-based algorithms are also presented.

The remainder of this chapter is organized as follows: Sec. 7.2 provides a summary of several model-based algorithms for global optimization. The intent is to describe, in a self-contained manner, both the model as well as the particle update rules for each of these algorithms. Sec. 7.3 contains results of numerical experiments for the quadratic function $h(x) = \frac{1}{2}|x|^2$. The quadratic function is considered because the optimal solution is known in closed-form (see Sec. 3.2.2). Performance comparisons are described, using Monte-Carlo simulations as a function of the problem dimension and the number of particles. Sec. 7.4 contains simulation results for a one-dimensional double-well potential which represents the simplest non-convex function with multiple (two in this case) local minimizers. Sec. 7.5 includes a comparison for other benchmark problems, including the Ackley function, the Griewank function, the Pintér's function, the Salomon function, the Rastrigin function, and the Trigonometric function.

7.2 Review of Model-based Algorithms

In a model-based optimization algorithm, the search of the global minimizer is guided by a (prescribed) *reference model* – a sequence of probability densities that asymptotically assigns all of its probability mass to the global minimizer. The reference model is denoted as $\{p_n^*(x)\}$ where $n = 1, 2, ...$ are the discrete-time indices. The optimal solution is obtained by sampling from p_n^*. However, since the objective function h is

*The content of this chapter is related to the publication [207, 204].

Table 7.1: Comparison of some model-based optimization algorithms

Algorithm	Reference model	Model update	Parametric
MRAS	$p^*_{n+1}(x) \propto p^*_n(x)\,\phi(h(x))\,\mathbf{1}_{\{h(x)\leq\gamma_n\}}$	Importance sampling + Density projection	yes
CE	$p^*_{n+1}(x) \propto q(x,\theta_n)\,\phi(h(x))\,\mathbf{1}_{\{h(x)\leq\gamma_n\}}$	Importance sampling + Density projection	yes
MEO	Replicator: $\frac{\mathrm{d}p^*_t}{\mathrm{d}t}(x) = -(h(x)-\hat{h}_t)\,p^*_t(x)$	Importance sampling + Density projection	yes
PFO	$p^*_{n+1}(x) \propto p^*_n(x)\,\phi(y_n - h(x))$	Importance sampling + Resampling	no
SMC-SA	Boltzmann: $p^*_n(x) \propto \exp(-h(x)/T_n)$	Importance sampling + Resampling + SA	no
SISR	$p^*_{n+1}(x) \propto p^*_n(x)\,\exp(-\beta h(x)\Delta t_n)$	Importance sampling + Resampling	no
CPF	$p^*_t(x) \propto p^*_0(x)\,\exp(-\beta h(x)t))$ (same as replicator if $\beta = 1$)	Optimal control without resampling	no

used in defining p^*_n, the problem of sampling directly from the reference model is not straightforward.

This section reviews several model-based algorithms (see Table 7.1). Broadly, there are two types of model-based algorithms: (i) parametric algorithms and (ii) non-parametric algorithms. Parametric algorithms include the cross-entropy (CE) algorithm, the model reference adaptive search (MRAS) algorithm and the model-based evolutionary optimization (MEO) algorithm. Non-parametric algorithms include the particle filter optimization (PFO) algorithm, the sequential Monte-Carlo simulated annealing (SMC-SA), and the sequential importance sampling and resampling (SISR) algorithm.

The non-parametric algorithms employ the importance sampling and resampling procedure to sample from the reference model. The numerical procedure mirrors the Bayes' update step in the particle filter [3]. In a parametric algorithm, the difficulty of sampling from an arbitrary distribution is circumvented by sampling from a *surrogate density model* that best approximates p^*_n for each n. The surrogate model is typically chosen from a family of *parametric* densities, denoted as $q(x;\theta_n)$, where $\{\theta_n\}_{n\in\mathbb{N}} \subset \Theta$ denotes the sequence of parameters in a given parameter space $\Theta \subset \mathbb{R}^d$. A popular choice for $q(x;\theta_n)$ is the Gaussian density function.

The primary difference among the algorithms is the choice of the reference model which leads to different importance sampling formulae for the non-parametric algorithms and parameter update formulae for the parametric algorithms. The various reference models are tabulated in Table 7.1.

7.2.1 Model Reference Adaptive Search (MRAS)

MRAS is a parametric model-based algorithm. The reference model is of the following general form:

$$p^*_{n+1}(x) := \frac{p^*_n(x)\,\phi(h(x))\,\mathbf{1}_{\{h(x)\le\gamma_n\}}}{\int p^*_n(y)\,\phi(h(y))\,\mathbf{1}_{\{h(y)\le\gamma_n\}}\,\mathrm{dy}} \tag{7.1}$$

with a given prior density $p^*_0(x)$. The surrogate density $q(x;\theta_n)$ is Gaussian with parameters $\theta_n = (\mu_n, \Sigma_n)$ as the mean and covariance:

$$q(x;\theta_n) = \frac{1}{(2\pi)^{d/2}|\Sigma_n|}\exp\big(-\frac{1}{2}(x-\mu_n)^T\Sigma_n^{-1}(x-\mu_n)\big). \tag{7.2}$$

The terms in the model (7.1) are explained next:

1) The function $\phi(\cdot)$ is chosen such that the reference model is biased towards region with lower function values. Typically, ϕ is taken to be some non-increasing non-negative function of its argument, e.g., $\phi(z) = 1$. In the numerical studies reported in this thesis, an exponential function is used for ϕ,

$$\phi(h(x)) = e^{-rh(x)},$$

where $r > 0$ is a tuning parameter.

2) The indicator $\mathbf{1}_{\{h(x)\le\gamma_n\}}$ is used to further restrict the search within the region where the function value is below a threshold. The sequence of threshold $\{\gamma_n\}$ is specified according to the sequence of surrogate model $\{q(\cdot;\theta_n)\}$, e.g.,

$$\gamma_n := \inf\,\{\gamma : P_{\theta_n}(h(X)\le\gamma) \ge \zeta\}, \tag{7.3}$$

where $\zeta \in (0,1]$ is a prescribed parameter, and $P_{\theta_n}(\cdot)$ denotes the probability with respect to the density $q(\cdot,\theta_n)$.

The explicit solution of (7.1) is given by,

$$p^*_{n+1}(x) \propto p^*_0(x)\,\phi(h(x))^{n+1}\,\mathbf{1}_{\{h(x)\le\gamma_n\}}, \tag{7.4}$$

where $\{\gamma_n\}$ is a non-increasing sequence.

The MRAS algorithm is described next. At the end of the $(n-1)$-th time step, one has a collection of particles $\{X_n^i\}_{i=1}^N$ sampled i.i.d. from the density $q(\cdot;\theta_n)$ with uniform weight. For the n-th time step, the objective is to estimate a new parameter, θ_{n+1}, and generate new samples from the density $q(\cdot;\theta_{n+1})$. This is carried out by iterating the following two steps:

1) *Calculation of importance weights*: The importance sampling procedure is used to approximate the reference model p_{n+1}^* given the samples $\{X_n^i\}_{i=1}^N$ from the surrogate model $q(\cdot;\theta_n)$ which is viewed as the *proposal* density, and the importance weight of each particle is calculated as,

$$w_n^i \propto \frac{p_{n+1}^*(X_n^i)}{q(X_n^i;\theta_n)} \overset{(7.4)}{\propto} \frac{p_0^*(X_n^i)}{q(X_n^i;\theta_n)}\,\phi(h(X_n^i))^{n+1}\,\mathbf{1}_{\{h(X_n^i)\le\gamma_n\}}, \qquad \sum_{i=1}^N w_n^i = 1. \tag{7.5}$$

The empirical approximation of the reference model p_{n+1}^* is represented as

$$\hat{p}_{n+1}^*(x) := \sum_{i=1}^N w_n^i \delta(x - X_n^i). \tag{7.6}$$

2) *Parameter estimation*: A popular algorithm for estimating the new parameter θ_{n+1} is via density projection, i.e., minimizing the Kullback-Leibler (K-L) divergence between $\hat{p}_{n+1}^*$ and $q(\cdot;\theta_{n+1})$. By the definition of the K-L divergence,

$$\begin{aligned}
\theta_{n+1} &= \arg\min_{\theta\in\Theta} \mathsf{D}\big(\hat{p}_{n+1}^* | q(x;\theta)\big) \\
&= \arg\min_{\theta\in\Theta} \int \hat{p}_{n+1}^*(x)\,\log\Big(\frac{\hat{p}_{n+1}^*(x)}{q(x;\theta)}\Big)\,\mathrm{d}x \\
&= \arg\max_{\theta\in\Theta} \int \hat{p}_{n+1}^*(x)\,\log q(x;\theta)\,\mathrm{d}x \\
&\overset{(7.6)}{=} \arg\max_{\theta\in\Theta} \sum_{i=1}^N w_n^i \log q(X_n^i;\theta)
\end{aligned} \tag{7.7}$$

Note that only the samples satisfying $h(X_n^i) \le \gamma_n$ have non-zero weights and contribute to the parameter update.

In the Gaussian case, the solution of (7.7) can be obtained in closed-form. For example, when $q(\cdot;\theta)$ is

Gaussian as given in (7.2), the solution is given by,

$$\mu_{n+1} = \sum_{i=1}^{N} w_n^i X_n^i,$$
$$\Sigma_{n+1} = \sum_{i=1}^{N} w_n^i (X_n^i - \mu_{n+1})(X_n^i - \mu_{n+1})^T.$$

The derivation appears in Appendix D.1. These formulae also appear in [76].

The numerical procedure of the MRAS algorithm is tabulated in Algorithm 8; see also [76]. Note that the sequence $\{\gamma_n\}_{n\geq 0}$ is guaranteed to be non-increasing, and a smoothing parameter ν is introduced to "dampen" the parameter update (see Line 12). A modified version of MRAS is also described in [76] where the parameter ζ and the sample size are both adaptive.

Remark 7.1 *If $\phi(h(x)) = e^{-\beta\, h(x)\,\Delta t_n}$ is used, where Δt_n is the time step, the reference model of MRAS becomes,*

$$p_{n+1}^*(x) = \frac{p_n^*(x)\, \exp(-\beta\, h(x)\, \Delta t_n) \mathbf{1}_{\{h(X)\leq\gamma_n\}}}{\int p_n^*(y)\, \exp(-\beta\, h(y)\, \Delta t_n)\, \mathbf{1}_{\{h(X)\leq\gamma_n\}}\, \mathrm{d}y}, \tag{7.8}$$

with a given prior p_0^*. Without the factor $\mathbf{1}_{\{h(X)\leq\gamma_n\}}$, this model is identical to the Bayes' model used in the controlled particle filter algorithm (see (3.2)).

7.2.2 Cross-entropy (CE)

The reference model in the CE algorithm is closely related to the model in the MRAS algorithm:

$$p_{n+1}^*(x) := \frac{q(x;\theta_n)\, \phi(h(x))\, \mathbf{1}_{\{h(x)\leq\gamma_n\}}}{\int q(y;\theta_n)\, \phi(h(y))\, \mathbf{1}_{\{h(y)\leq\gamma_n\}}\, \mathrm{d}y}, \tag{7.9}$$

where the definitions of $q(\cdot;\theta_n)$, $\phi(\cdot)$ and $\mathbf{1}_{\{h(x)\leq\gamma_n\}}$ are identical to the case with the MRAS algorithm.

The CE algorithm follows the same numerical procedure as MRAS at each time step. However, compared with the model (7.1) used in MRAS, the density p_n^* is replaced with the surrogate model $q(\cdot;\theta_n)$ in (7.9). Correspondingly, the importance weights in the CE algorithm becomes (see (7.5)),

$$w_n^i \propto \frac{p_{n+1}^*(X_n^i)}{q(X_n^i;\theta_n)} \overset{(7.9)}{\propto} \phi(h(X_n^i))\, \mathbf{1}_{\{h(X_n^i)\leq\gamma_n\}}, \qquad \sum_{i=1}^{N} w_n^i = 1.$$

Algorithm 8 MRAS numerical algorithm

Input: Initial distribution $q(x;\theta_0)$, parameters $\zeta \in (0,1]$, $\bar{\varepsilon} \in (0,1)$, $\nu \in [0,1]$

Iteration n **(**$n \geq 0$**):**

Sample $\{X_n^i\}_{i=1}^N$ i.i.d. from $q(x;\theta_n)$

Calculate $\tilde{\gamma}_n := \mathbf{h}_{(\lceil \zeta N \rceil)}$, where $\mathbf{h}_{(k)}$ is the k-th smallest element in $\mathbf{h} := \big(h(X_n^i), ..., h(X_n^N)\big)$, and $\lceil a \rceil$ is the smallest integer greater than a

if $n = 0$ or $\tilde{\gamma}_n \leq \gamma_{n-1} - \bar{\varepsilon}$ **then**

  Assign $\gamma_n = \tilde{\gamma}_n$

else

  Assign $\gamma_n = \gamma_{n-1}$

end if

Calculate importance weights,

$$w_n^i \propto \frac{p_0^*(X_n^i)}{q(X_n^i;\theta_n)}\, \phi(h(X_n^i))^{n+1}\, \mathbf{1}_{\{h(X_n^i) \leq \gamma_n\}}, \qquad \sum_{i=1}^N w_n^i = 1$$

Update the model parameter,

$$\tilde{\theta}_{n+1} := \arg\max_{\theta \in \Theta} \sum_{i=1}^N w_n^i \log q(X_n^i;\theta)$$

Assign $\theta_{n+1} = \nu\, \tilde{\theta}_{n+1} + (1-\nu)\, \theta_n$

Assign $n = n+1$ if the stopping rule is not satisfied; otherwise terminate

The numerical procedure for the CE algorithm is tabulated in Algorithm 9; see also [150, 76].

7.2.3 Model-based Evolutionary Optimization (MEO)

The reference model in MEO evolves according to the replicator dynamics,

$$\frac{\mathrm{d}p_t^*}{\mathrm{d}t}(x) = -(h(x) - \hat{h}_t)\, p_t^*(x), \tag{7.10}$$

Algorithm 9 CE numerical algorithm

1: **Input:** Initial distribution $q(x;\theta_0)$, parameter $\zeta \in (0,1]$

2: **Iteration** n **(**$n \geq 0$**):**

3: Sample $\{X_n^i\}_{i=1}^N$ i.i.d. from $q(x;\theta_n)$

4: Calculate $\gamma_n := \mathbf{h}_{(\lceil \zeta N \rceil)}$, where $\mathbf{h}_{(k)}$ is the k-th smallest element in $\mathbf{h} := \big(h(X_n^i),...,h(X_n^N)\big)$, and $\lceil a \rceil$ denotes the smallest integer greater than a

5: Calculate importance weights,

$$w_n^i \propto \phi(h(X_n^i))\mathbf{1}_{\{h(X_n^i)\leq\gamma_n\}}, \qquad \sum_{i=1}^N w_n^i = 1$$

6: Update the model parameter,

$$\theta_{n+1} := \arg\max_{\theta\in\Theta} \sum_{i=1}^N w_n^i \log q(X_n^i;\theta)$$

7: Assign $n = n+1$ if the stopping rule is not satisfied; otherwise terminate

where $\hat{h}_t := \int h(x)\, p_t^*(x)\, \mathrm{d}x$. In a sampling-based implementation, the density is approximated as $p_t^*(x) \approx \sum_{i=1}^N w_t^i \delta(x - X_t^i)$, and the corresponding odes for the weights are obtained as,

$$\frac{\mathrm{d}w_t^i}{\mathrm{d}t} = -\big(h(X_t^i) - \hat{h}_t^{(N)}\big)\, w_t^i, \qquad i = 1,...,N, \tag{7.11}$$

where $\hat{h}_t^{(N)} := \sum_{i=1}^N w_t^i h(X_t^i)$. Denoting Δt_n as the step size, the discrete-time system for the weight update in MEO is given by,

$$w_{n+1}^i = w_n^i - \big(h(X_n^i) - \hat{h}_n^{(N)}\big)\, w_n^i \Delta t_n,$$

The normalization $\sum_{i=1}^N w_{n+1}^i = 1$ automatically holds. The reference model p_{n+1}^* is then approximated as

$$\hat{p}_{n+1}^* \approx \sum_{i=1}^N w_{n+1}^i \delta(x - X_n^i).$$

The parameter estimation is obtained by density projection as in the MRAS and CE algorithms. The formula was given by (7.7).

The numerical procedure of the MEO algorithm is tabulated in Algorithm 10; see also [187].

Algorithm 10 MEO numerical algorithm

1: **Input:** Initial distribution $q(x;\theta_0)$, parameter $\zeta \in (0,1]$

2: **Iteration** n **(**$n \geq 0$**):**

3: Sample $\{X_n^i\}_{i=1}^N$ i.i.d. from $q(x,\theta_n)$ with *equal* weights $w_n^i = 1/N,\ \forall i$

4: Calculate $\hat{h}_n^{(N)} = \frac{1}{N}\sum_{i=1}^N h(X_n^i)$

5: Calculate importance weights,

$$w_{n+1}^i = w_n^i - \left(h(X_n^i) - \hat{h}_n^{(N)}\right) w_n^i \Delta t_n,$$

6: Update the model parameter,

$$\theta_{n+1} := \arg\max_{\theta\in\Theta} \sum_{i=1}^N w_{n+1}^i \log q(X_n^i;\theta)$$

7: Assign $n = n+1$ if the stopping rule is not satisfied; otherwise terminate

7.2.4 Particle Filtering for Optimization (PFO)

A particle filtering framework for global optimization was introduced in [215]. The PFO explicitly casts the global optimization problem as a filtering problem with the following state-space mode:

$$\begin{aligned} X_{n+1} &= X_n + B_n, \\ Y_{n+1} &= h(X_{n+1}) + W_{n+1}, \end{aligned}$$

for $n \in \mathbb{N}$, where $\{B_n\}_{n\geq 1}$ and $\{W_n\}_{n\geq 1}$ are mutually independent sequence of random variables that are also independent of X_0. The hidden state $\{X_n\}_{n\in\mathbb{N}}$ can be viewed as a perturbed random process which converges to the global minimizer $\bar{x}$ as the perturbation B_n gradually reduces to zero [215].

The reference model is given by the Bayes' rule,

$$\begin{aligned} \text{(Prediction)} \quad p_{n+1|n}^*(x) &= \int K_n(x|x_n)\, p_n^*(x_n)\, \mathrm{d}x_n, \\ \text{(Update)} \qquad p_{n+1}^*(x) &= \frac{p_{n+1|n}^*(x)\, \phi(Y_{n+1} - h(x))}{\int p_{n+1|n}^*(z)\, \phi(Y_{n+1} - h(z))\, \mathrm{d}z}, \end{aligned} \tag{7.12}$$

where the transition kernel $K_n(\cdot|\cdot)$ and the function $\phi(\cdot)$ specify the distribution of the random processes B_n and W_n, respectively.

Algorithm 11 PFO framework

1: **Input:** Initial samples $\{X_0^i\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} p_0^*(x)$, kernel functions $\{K_n(\cdot|\cdot)\}_{n\in\mathbb{N}}$, function $\phi(\cdot)$

2: **Iteration** n ($n \geq 0$)**:**

3: Sample $\widetilde{X}_{n+1}^i \sim K_n(\cdot|X_n^i)$ for $i = 1,...,N$

4: *Observation generation*: Take Y_{n+1} to be a sample function value according to certain rule. If $n \geq 1$ and $Y_{n+1} > Y_n$, then set $Y_{n+1} = Y_n$

5: *Bayes' update*: Calculate importance weights,

$$w_{n+1}^i \propto \phi(Y_{n+1} - h(\widetilde{X}_{n+1}^i)), \quad \sum_{i=1}^N w_{n+1}^i = 1$$

6: *Resampling*: Generate samples $\{X_{n+1}^i\}_{i=1}^N$ from $\{\widetilde{X}_{n+1}^i\}_{i=1}^N$ using resampling with replacement; c.f., [3]

7: Assign $n = n+1$ if the stopping rule is not satisfied; otherwise terminate

The implementation of the PFO algorithm is identical to a bootstrap particle filter [3], and a non-increasing sequence of observations $\{Y_n\}_{n\geq 0}$ need to be generated using the samples. The numerical procedure of PFO is tabulated in Algorithm 11; see also [215].

Remark 7.2 *If the observation* $Y_{n+1} := min\,(h(\widetilde{X}_{n+1}^1),...,h(\widetilde{X}_{n+1}^N))$, *the function* $\phi(z) = e^{\beta\, z\, \Delta t_n}$, *and* $B_n = 0, \forall n \in \mathbb{N}$, *the update formula in the Bayes' model of PFO becomes,*

$$p_{n+1}^*(x) = \frac{p_n^*(x)\, \exp(-\beta\,(h(x) - Y_{n+1})\,\Delta t_n)}{\int p_n^*(z)\, \exp(-\beta\,(h(z) - Y_{n+1})\,\Delta t_n)\, \mathrm{d}z}, \tag{7.13}$$

which is closely related to the reference model (7.8) *used in MRAS. The difference here is the presence of a fictitious observation* Y_{n+1}. *Whereas the MRAS, CE and MEO algorithms generate samples from a parametric distribution, PFO is non-parametric and uses the resampling procedure to obtain new samples instead of the density projection.*

Algorithm 12 SMC-SA numerical algorithm

Input: Initial samples $\{X_0^i\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} p_0^*(x)$, cooling schedule $\{T_n\}_{n\in\mathbb{N}}$, kernel functions $\{K_n(\cdot|\cdot)\}_{n\in\mathbb{N}}$

Iteration n ($n \geq 0$)**:**

Importance update: Calculate importance weights,

$$w_0^i \propto \exp(-h(X_0^i)/T_0)/p_0^*(X_0^i),$$

$$w_n^i \propto \exp\left(h(X_{n-1}^i)(1/T_{n-1} - 1/T_n)\right), \quad n \geq 1, \qquad \sum_{i=1}^N w_n^i = 1$$

Resampling: Generate samples $\{\widetilde{X}_n^i\}_{i=1}^N$ from $\{X_{n-1}^i\}_{i=1}^N$ using resampling with replacement

for $i = 1, \ldots, N$ **do**

Generate $Y_n^i \sim K_n(y|\widetilde{X}_n^i)$

Calculate acceptance probability

$$\zeta_n^i = \min\left\{\exp\left((h(\widetilde{X}_n^i) - h(Y_n^i))/T_n\right),\ 1\right\}$$

Accept/Reject

$$X_n^i = \begin{cases} Y_n^i, \text{ w.p. } \zeta_n^i \\ \widetilde{X}_n^i, \text{ w.p. } 1 - \zeta_n^i \end{cases}$$

end for

Assign $n = n+1$ if the stopping rule is not satisfied; otherwise terminate

7.2.5 Sequential Monte-Carlo Simulated Annealing (SMC-SA)

The SMC-SA algorithm, proposed recently in [213], combines importance sampling with the simulated annealing (SA) ideas [93]. The reference model is the Boltzmann distribution,

$$p_n^*(x) = \frac{\exp(-h(x)/T_n)}{\int \exp(-h(y)/T_n)\, \mathrm{d}y}, \tag{7.14}$$

where the non-increasing real-valued sequence $\{T_n\}_{n\geq 0}$ is a prescribed cooling schedule. The SMC-SA algorithm approximates the Boltzmann model (7.14) using an importance sampling procedure followed by a classical SA procedure for each sample independently. The numerical procedure of the SMC-SA algorithm

Algorithm 13 SISR numerical algorithm

1: **Input:** Initial samples $\{X_0^i\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} p_0^*(x)$, kernel functions $\{K_n(\cdot|\cdot)\}_{n\in\mathbb{N}}$

2: **Iteration** n ($n \geq 0$)**:**

3: *Bayes' update:* Calculate the weights,

$$w_n^i \propto \exp\big(-\beta\, h(X_n^i)\,\Delta t_n\big), \quad \sum_{i=1}^N w_n^i = 1$$

4: *Resampling*: Generate samples $\{\widetilde{X}_n^i\}_{i=1}^N$ from $\{X_n^i\}_{i=1}^N$ using resampling with replacement

5: *Diffusion*: Sample $X_{n+1}^i \sim K_n(\cdot|\widetilde{X}_n^i)$ for $i = 1,...,N$

6: Assign $n = n+1$ if the stopping rule is not satisfied; otherwise terminate

is tabulated in Algorithm 12; see also [213].

The following cooling schedule is used in [213],

$$T_n = \frac{|h_n^*|}{\log(n+1)},$$

where h_n^* denotes the best function value found by the algorithm up to the n-th time step.

7.2.6 Sequential Importance Sampling and Resampling (SISR)

SISR is a non-parametric algorithm based on importance sampling and resampling. The reference model is the Bayes' model (3.2),

$$p_{n+1}^*(x) = \frac{p_n^*(x)\,\exp(-\beta\, h(x)\,\Delta t_n)}{\int p_n^*(y)\,\exp(-\beta\, h(y)\,\Delta t_n)\,\mathrm{d}y}, \tag{7.15}$$

where $\Delta t_n = t_{n+1} - t_n$ is the time step. The numerical procedure of the SISR algorithm is described in Sec. 3.1 and also tabulated in Algorithm 13.

7.3 Simulation Results – Quadratic Function

In this section, we describe simulation results for the quadratic function,

$$h(x) = \frac{1}{2}|x|^2.$$

Results of numerical experiments with the following algorithms is described except for CPF-G and CPF-K. These two algorithms will be studied in Sec. 7.4 and Sec. 7.5 for non-quadratic problems.

1) MRAS: the model reference adaptive search algorithm described in [76] and Algorithm 8.

2) CE: the cross-entropy algorithm described in [150] and Algorithm 9.

3) MEO: the model-based evolutionary optimization algorithm described in [187] and Algorithm 10.

4) PFO: the particle filtering optimization algorithm described in [215] and Algorithm 11.

5) SISR: the sequential importance sampling and resampling algorithm described in Algorithm 13.

6) CPF-A: the controlled particle filter algorithm with the affine control law described in Table 3 in Sec. 3.2.2.

7) CPF-G: the controlled particle filter algorithm with the Galerkin control law described in Table 4 in Sec. 4.3.

8) CPF-K: the controlled particle filter algorithm with the kernel-based control law described in Table 5 in Sec. 4.4.

The simulation parameters are as follows: The simulations are carried out over a finite time-horizon $[0,T]$ with $T=1$, a fixed time step $\Delta t = 0.001$, and the parameter $\beta = 1$. The algorithms are initialized with the same samples drawn i.i.d. from the Gaussian distribution $\mathcal{N}(m_0,\Sigma_0)$, where $m_0 = (10,...,10) \in \mathbb{R}^d$ and $\Sigma_0 = \text{diag}(5^2,...,5^2)$. The parameters used in these algorithms are as follows:

1) **Reference models:** The reference models (7.8), (7.9), (7.10), (7.13), and (7.15) are used for the MRAS, CE, MEO, PFO, and SISR algorithms, respectively. The reference models used in these algorithms are closely related.

2) **Diffusion kernel:** The diffusion kernel $\{K_n(\cdot|\cdot)\}_{n\in\mathbb{N}}$ used in PFO and SISR is chosen as Gaussians. That is, at the n-th time step,

$$K_n(x|x') \propto \exp\big(-\frac{1}{2}(x-x')^T\widetilde{\Sigma}_n^{-1}(x-x')\big), \tag{7.16}$$

where $\widetilde{\Sigma}_n = \text{diag}(\widetilde{\sigma}_n^2,...,\widetilde{\sigma}_n^2)$, and $\{\widetilde{\sigma}_n\}_{n\in\mathbb{N}}$ is a decaying sequence $\widetilde{\sigma}_n = \widetilde{\sigma}_0\kappa^n$ [212]. $\kappa = 0.95$ is used in all the simulations. The choice of $\widetilde{\sigma}_0$ may depend on the dimension of the problem as well as the

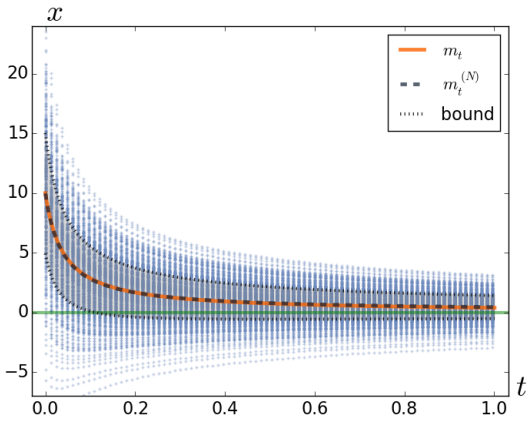


Figure 7.1: Simulation results of CPF-A with $h(x) = \frac{1}{2}x^2$. Trajectories of $N = 500$ particles is depicted as dots in the background. The solid line is the mean m_t obtained using the exact formula (3.21) and the dashed line is its empirical estimate obtained using the particles. The shaded region depicts the ± 1 standard deviation bound.

initial condition. For the quadratic function considered in this section, $\widetilde{\sigma}_0 = 0.1$ for $d < 5$, and $\widetilde{\sigma}_0 = 0.5$ otherwise. Large value of $\widetilde{\sigma}_0$ may prevent undesirable premature convergence.

3) The MRAS, CE and MEO algorithms are simulated **without** the factor $\mathbf{1}_{\{h(x)\leq\gamma_n\}}$ in their reference models. Even thought it was observed that including this term may expedite the search and lead to superior convergence speed, the intent here is to compare the parametric and non-parametric algorithms with consistent reference models.

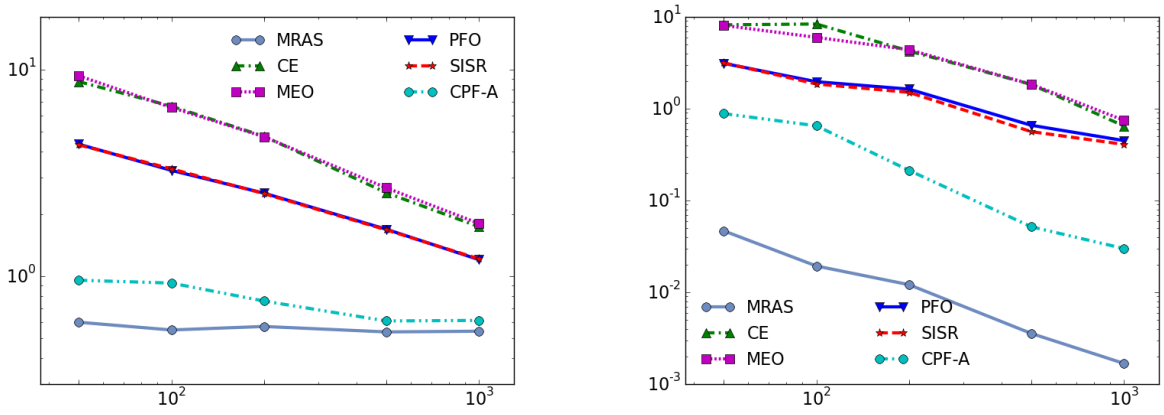
4) For the CPF-A algorithm, an Euler discretization is used to numerically integrate the ode (3.4) with $\beta = 1$.

Figure 7.1 depicts a typical simulation result of CPF-A for $d = 1$ and $N = 500$. The empirical mean $m_t^{(N)}$ is seen to closely match its mean-field limit m_t obtained using the exact formula (3.21).

The performance of the algorithms is compared based on $J = 100$ independent Monte-Carlo (M.C.) runs. The performance metrics are the M.C. average and variance of the empirical mean at the terminal time T. The M.C. average and variance of the terminal empirical mean are defined as,

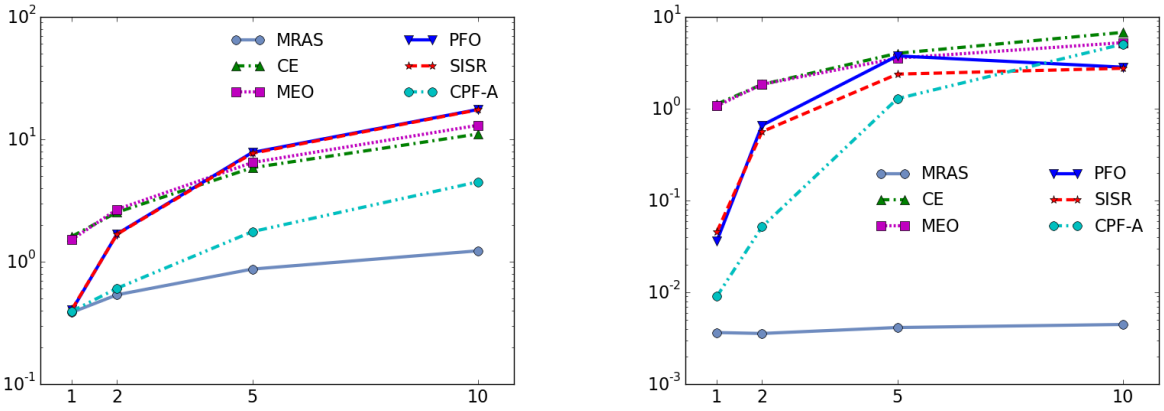
$$\text{M.C. average:} \quad \widehat{m}_T^{(N)} := \frac{1}{J}\sum_{j=1}^{J} m_{T,j}^{(N)},$$

$$\text{M.C. variance:} \quad \mathrm{Var}(m_T^{(N)}) := \frac{1}{J}\sum_{j=1}^{J} |m_{T,j}^{(N)} - \widehat{m}_T^{(N)}|^2,$$



(a) $\widehat{m}_T^{(N)}$ v.s. N (b) $\mathrm{Var}(m_T^{(N)})$ v.s. N

Figure 7.2: $\widehat{m}_T^{(N)}$ and $\mathrm{Var}(m_T^{(N)})$ as a function of the number of particles N.



(a) $\widehat{m}_T^{(N)}$ v.s. d (b) $\mathrm{Var}(m_T^{(N)})$ v.s. d

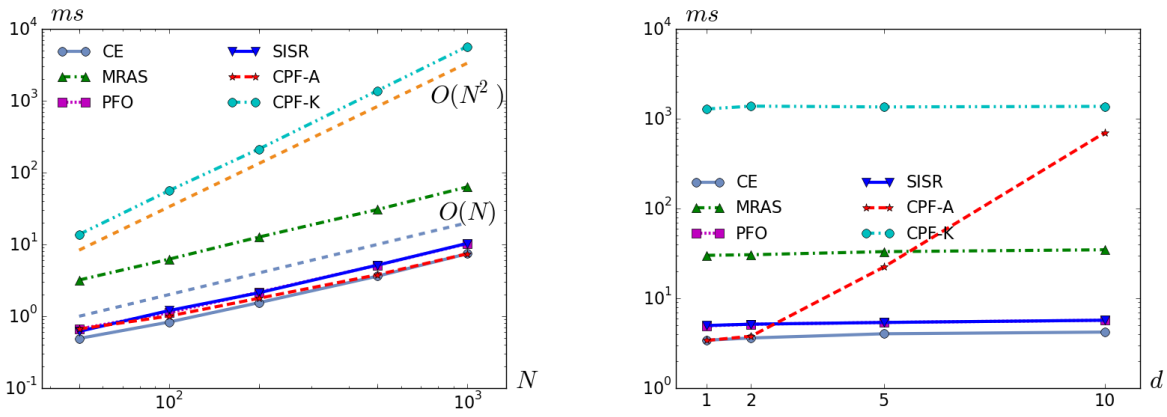
Figure 7.3: $\widehat{m}_T^{(N)}$ and $\mathrm{Var}(m_T^{(N)})$ as a function of dimension d.

where $m_{T,j}^{(N)}$ denotes the empirical mean of the particles at terminal time for the j-th M.C. run.

The simulation results for the quadratic function is presented next:

1) $\widehat{m}_T^{(N)}$ **and** $\mathbf{Var}(m_T^{(N)})$ **as a function of** N**:** In this simulation, the number of particles varies in the range $N \in \{50, 100, 200, 500, 1000\}$, for a fixed dimension $d = 2$. Figure 7.2 depicts the M.C. average and variance. As expected, both $\widehat{m}_T^{(N)}$ and $\mathrm{Var}(m_T^{(N)})$ decrease with N for all the algorithms.

2) $\widehat{m}_T^{(N)}$ **and** $\mathbf{Var}(m_T^{(N)})$ **as a function of** d**:** In this simulation, the dimension $d \in \{1, 2, 5, 10\}$, for a fixed number of particles $N = 500$. Figure 7.3 depicts the M.C. average and variance. Whereas the parametric algorithms (CE, MRAS, MEO) continue to exhibit robust performance, non-parametric algorithms (PFO, SISR, CPF-A) are strongly susceptible to the increase in dimension. Such a drastic performance



(a) Computational time v.s. N

(b) Computational time v.s. d

Figure 7.4: Computational time per iteration as a function of the number of particles N (part (a)) and dimension d (part (b)).

deterioration suggests that more particles are required for high-dimensional problems.

In both Figure 7.2 and Figure 7.3, the MRAS consistently performs the best and its simulation variance is the smallest for all choice of d simulated here. For all values of N and moderate values of d, the control-based algorithm CPF-A exhibits less error and simulation variance compared with the sampling-based non-parametric algorithms.

3) **Computational time as a function of N:** The mean computational time (per iteration of the algorithm, averaged over 500 Monte Carlo runs) is depicted in Figure 7.4 (a) as a function of N. The CPF-K is also included to assess its computational complexity. In addition, the $O(N)$ and $O(N^2)$ lines are plotted to aid the comparison. The plot shows that the computational time scales linearly with N for all the algorithms except CPF-K which scales quadratically. The MRAS has relatively higher computational cost due to the more intricate importance weight calculation (see (7.5)) than the other model-based algorithms.

4) **Computational time as a function of d:** The mean computational time (per iteration of the algorithm, averaged over 500 Monte Carlo runs) is depicted in Figure 7.4 (b) as a function of d. For all the algorithms except CPF-A, the computational time is nearly independent of d. The exceptionally high computational cost of CPF-A for high dimensions is due to the affine control scheme in which the number of second-order polynomial basis functions scales quadratically with d.

The experiments in 3) and 4) were conducted on a platform with an Intel i3-2120 3.3GHz CPU.

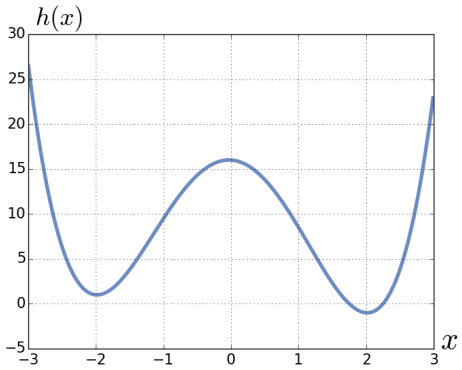


Figure 7.5: Double-well potential.

7.4 Simulation Results – Double-well Potential

In this section, we describe simulation results for the double-well potential

$$h(x) = (x-2)^2(x+2)^2 - \frac{x}{2}.$$

The function is depicted in Figure 7.5.

The simulations are conducted over a finite time-horizon $[0,T]$ with $T = 10$ and a fixed time step $\Delta t = 0.01$. For each of the simulations, $N = 500$ particles are used. The initial particles X_0^i are sampled i.i.d. from a mixture of two Gaussians, $\mathcal{N}(-2, 0.6^2)$ and $\mathcal{N}(2, 0.6^2)$, with equal weights. The Gaussian diffusion kernel given in (7.16) is used in PFO and SISR where $\widetilde{\sigma}_n = \widetilde{\sigma}_0 r^n$ with $\widetilde{\sigma}_0 = 0.1$ and $\kappa = 0.95$. An Euler discretization is used for numerical integration in the CPF algorithms with $\beta = 1$. For the Galerkin approximation, the basis functions are $\left\{x,\ \cos\left(\frac{2\pi}{10}x\right),\ \sin\left(\frac{2\pi}{10}x\right)\right\}$. For the kernel approximation, the parameter $\varepsilon = 0.5$.

The CPF algorithm with the three types of control laws (i.e., the affine, Galerkin and kernel-based control laws) are depicted in Figure 7.6 (g)-(i). The kernel-based control law leads to a more graceful transient behavior than the other two control laws. The affine control law is unable to merge the two mode in the posterior distribution, and the Galerkin control law may suffer from numerical instability on account of ill-conditioning of the matrix A (see Table 4). This can lead to relatively large values of control requiring small time-steps for numerical integration.

Figure 7.7 depicts a comparison of $\hat{h}_t^{(N)} = \frac{1}{N}\sum_i h(X_t^i)$ with the three types of control laws. With the optimal control, Theorem 3.4 shows that $\hat{h}_t$ decreases monotonically as a function of time. This was indeed

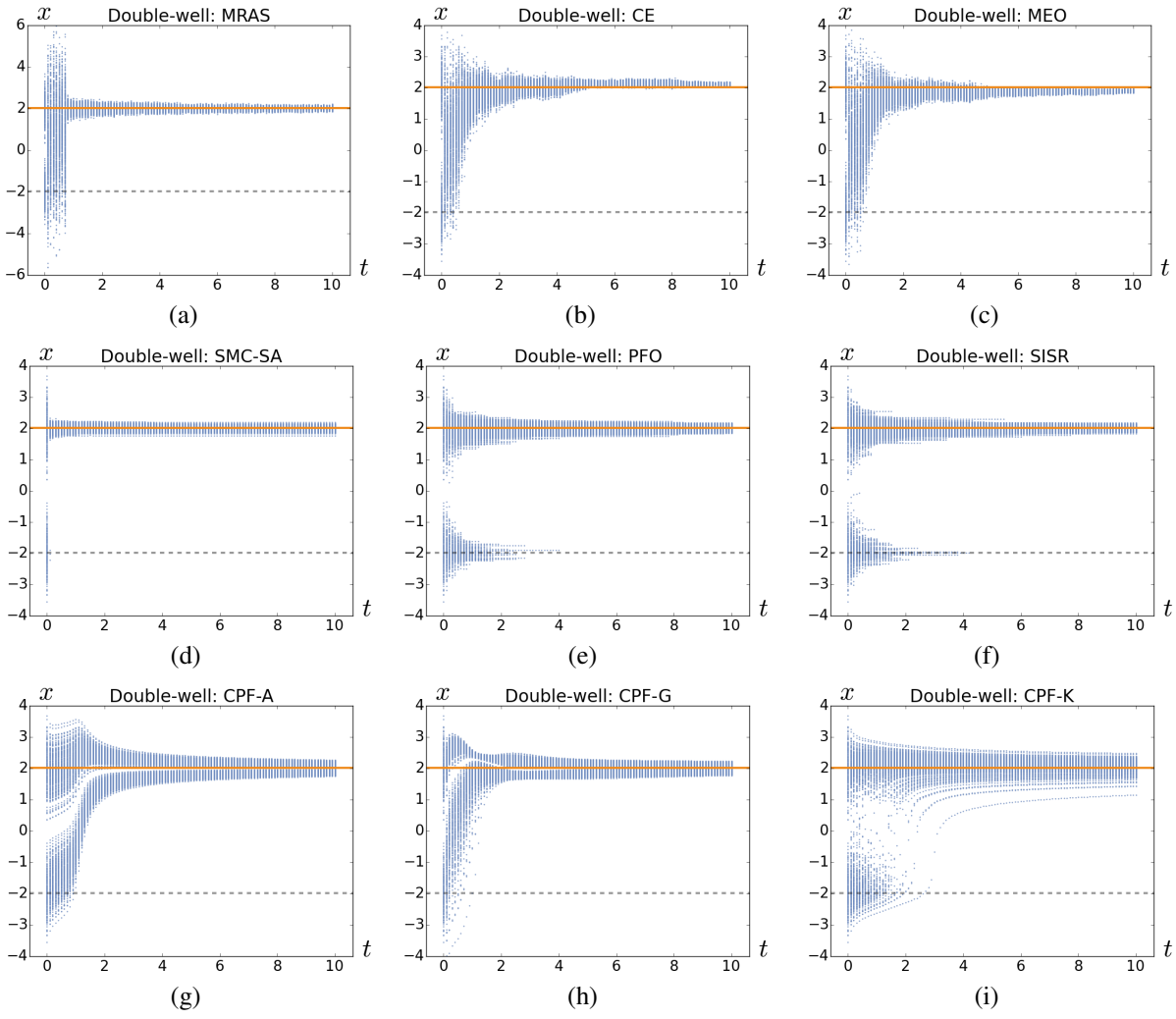


Figure 7.6: Particle trajectories of the algorithms for the Double-well potential.

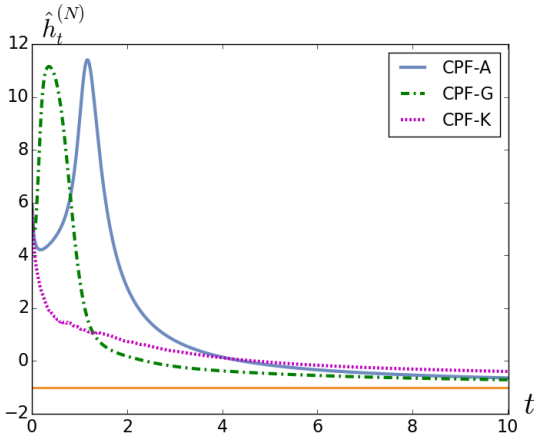


Figure 7.7: Comparison of $\hat{h}_t^{(N)}$ with the three types of approximate control laws. The global minimum value $\bar{h}$ is also plotted at the bottom.

Table 7.2: M.C. average and std. dev. of empirical error at terminal time, M.C. std. dev. in the parenthesis.

MRAS	CE	MEO
3.15e-3 (2.31e-3)	5.88e-2 (4.16e-2)	1.68e-1 (6.55e-1)
SMC-SA	PFO	SISR
3.61e-3 (2.50e-3)	4.65e-3 (4.10e-3)	4.95e-3 (3.66e-3)
CPF-A	CPF-G	CPF-K
8.67e-3 (2.27e-3)	6.845e-3 (1.89e-2)	1.20e-1 (4.03e-3)

found to be the case with the kernel-based algorithm but not so with the other two. Even though the particles in all three cases eventually converge to the correct equilibrium (see Figure 7.6 (g)-(i)), the approximate nature of the control can lead to a transient growth of $\hat{h}_t^{(N)}$.

Other model-based algorithms are depicted in Figure 7.6 (a) - (f). It is observed that the non-parametric algorithms such as PFO, SISR and CPF-K handle bi-modal distributions better than the Gaussian-based parametric algorithms (MRAS, CE, MEO) during the initial transient phase of the search. Among the parametric algorithms, the MRAS eventually settles at the global minimizer while CE and MEO exhibit fluctuations around the global minimizer. This indicates that the reference model and the associated weight calculation in MRAS are advantageous over those in CE and MEO. The SMC-SA algorithm described in Sec. 7.2.5 is also included in the comparison.

Monte-Carlo simulations are carried out to assess the empirical error of the algorithms at the terminal time T. Let $m_{T,j}^{(N)}$ denotes the empirical mean of the particles at time T for the j-th M.C. run. The M.C.

average and standard deviation (std. dev.) of the terminal empirical error are then defined as,

$$\widehat{e}_T^{(N)} := \frac{1}{J}\sum_{j=1}^{J} e_{T,j}^{(N)}, \tag{7.18a}$$

$$\text{Std. dev.}(e_T^{(N)}) := \sqrt{\frac{1}{J}\sum_{j=1}^{J} \left|e_{T,j}^{(N)} - \widehat{e}_T^{(N)}\right|^2}, \tag{7.18b}$$

where $e_{T,j}^{(N)} := |m_{T,j}^{(N)} - \bar{x}|$. For each algorithm, the two metrics obtained from 100 independent M.C. runs are tabulated in Table 7.2. It is seen that the MEO has the largest score in both metrics. The CPF-K also exhibits larger M.C. average of the error due to a relatively large value of ε used in the simulation which slows down the asymptotic convergence. Adaptively setting the value of ε requires further investigation.

7.5 Simulation Results – Benchmark Problems

In this section, we present performance comparison results of the optimization algorithms for six benchmark optimization problems listed as below. Numerical results for these problems appear in recent publications, e.g., [76, 78, 1].

1) *Ackley function*,

$$h(x) = -20\exp\Big\{-0.02\sqrt{\frac{|x|^2}{d}}\Big\} - \exp\Big\{\frac{1}{d}\sum_{k=1}^{d}\cos(2\pi x_k)\Big\} + 20 + e,$$

where the global minimizer $\bar{x} = (0,...,0)$, and $\bar{h} := h(\bar{x}) = 0$.

2) *Griewank function*,

$$h(x) = \frac{1}{4000}|x|^2 - \prod_{k=1}^{d}\cos\Big(\frac{x_k}{\sqrt{k}}\Big) + 1,$$

where $\bar{x} = (0,...,0)$, and $\bar{h} = 0$.

3) *Pintér's function*,

$$\begin{aligned} h_1(x) = &\sum_{k=1}^{d} k x_n^2 + 20\sum_{k=1}^{d} k\sin^2\left(x_{k-1}\sin(x_k) - x_k + \sin(x_{k+1})\right) \\ &+ \sum_{k=1}^{d} k\log_{10}\left(1 + k(x_{k-1}^2 - 2x_k + 3x_{k+1} - \cos(x_k) + 1)^2\right), \end{aligned}$$

where $\bar{x} = (0, ..., 0)$, and $\bar{h} = 0$.

4) *Rastrigin function*,

$$h_2(x) = |x|^2 - 10 \sum_{k=1}^{d} \cos(2\pi x_k) + 10d,$$

where $\bar{x} = (0, ..., 0)$, and $\bar{h} = 0$.

5) *Salomon function*,

$$h_4(x) = 1 - \cos(2\pi|x|) + 0.1|x|,$$

where $\bar{x} = (0, ..., 0)$, and $\bar{h} = 0$.

6) *Trigonometric function*,

$$h_5(x) = \sum_{k=1}^{d} \left[8\sin^2(7(x_k - 0.9)^2) + 6\sin^2(14(x_k - 0.9)^2) + (x_k - 0.9)^2\right],$$

where $\bar{x} = (0.9, ..., 0.9)$, and $\bar{h} = 0$.

These functions are non-convex and possess multiple local minimizers. Their graphs are depicted in Figure 7.8 for $d = 2$.

Table 7.3: Time horizon and time step for simulating each problem

	Ackley	Griewank	Pintér's	Rastrigin	Salomon	Trigonometric
T	50	100	0.05	0.05	50	0.05
Δt	0.1	0.1	0.0001	0.0001	0.1	0.0001

The simulations parameters are as follows:

1) **Time horizon:** The simulations are carried out over a finite time-horizon $[0, T]$ with a fixed time step Δt. The choice of T and Δt may vary for different problems, as tabulated in Table 7.3. For the Pintér's, Rastrigin, and Trigonometric function, smaller Δt is used to prevent numerical instability due to large function values. The values of T and Δt are determined upon case-by-case based preliminary experiments.

2) **Initial condition:** For all the problems, the algorithms are initialized with the Gaussian distribution $\mathcal{N}(\mu_0, \Sigma_0)$, where the mean $\mu_0 \in \mathbb{R}^d$ is *randomly* selected from the uniform distribution on $[-50, 50]^d$,

and the covariance $\Sigma_0 \in \mathbb{R}^{d\times d}$ is a diagonal matrix with all the diagonal elements chosen as 500. Same initialization scheme and parameters were used in the numerical studies presented in [76, 213].

3) **Algorithm parameters:** The MRAS, CE and MEO algorithms are simulated without the factor $\mathbf{1}_{\{h(x)\leq\gamma_n\}}$ in their reference model. The Gaussian diffusion kernel (see the formula (7.16)) is used for simulating PFO and SISR with $\widetilde{\Sigma}_n = \mathrm{diag}(\widetilde{\sigma}_n^2, ..., \widetilde{\sigma}_n^2)$, and a decaying sequence $\widetilde{\sigma}_n = \widetilde{\sigma}_0 \kappa^n$ with $\widetilde{\sigma}_0 = 5$ and $\kappa = 0.95$. A large value of $\widetilde{\sigma}_0$ is used to increase "coverage" of the search. All the CPF algorithms are simulated with $\beta = 1$. The CPF-G algorithm uses the following basis functions,

$$\left\{x_k,\ \ \cos\left(\frac{2\pi x_k}{T}\right),\ \ \sin\left(\frac{2\pi x_k}{T}\right),\ \ k = 1, ..., d\right\}$$

with $T = 400$. The parameter ε in the CPF-K algorithm is adaptive: At each time step, ε is chosen as the largest empirical covariance of the current particle set among all the directions. It was observed that too small a value of ε easily led to numerical instability. More judicious choice of the basis functions in CPF-G as well as the parameter ε in CPF-K is the subject of future research.

The algorithms are assessed based on $J = 100$ independent Monte-Carlo (M.C.) runs. Figure 7.9 depicts the M.C. average of the *empirical error* as a function of time t for all the problems and all the algorithms. Let $m_{t,j}^{(N)}$ denotes the empirical mean of the particles at time t for the j-th M.C. run. The M.C. average of the empirical error at time t is then defined as,

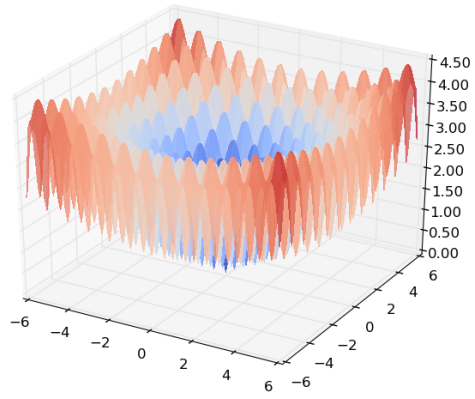
$$\widehat{e}_t^{(N)} := \frac{1}{J}\sum_{j=1}^{J} \left|m_{t,j}^{(N)} - \bar{x}\right|. \tag{7.19}$$

It is seen from Figure 7.9 that:

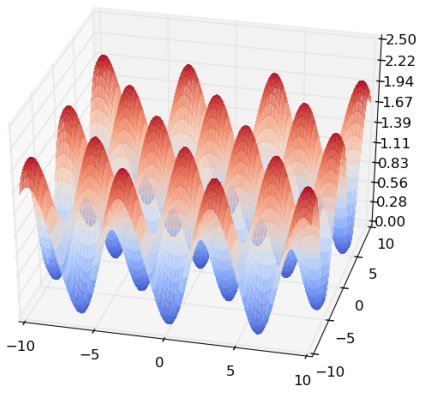
1) Except for the Salomon function, the CPF-K algorithm exhibits the the lowest error within the allocated computation budget. Improvement over the affine and Galerkin control law is also observed.

2) Except for the Salomon function, the CPF algorithms will potentially continue to search for better solutions after the terminal time T, whereas the CE and MEO stop the search very early at local minimizers. The MRAS also suffers from such early stagnation for the Ackley and the Griewank function.

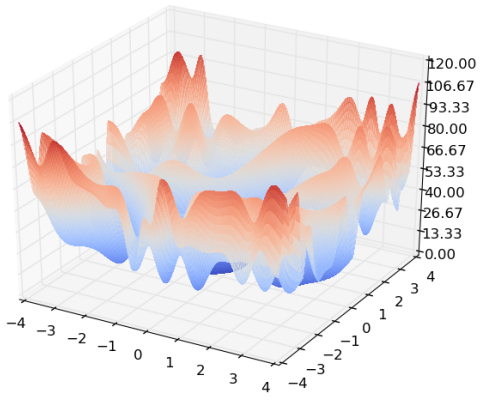
3) The MRAS algorithm may exhibit numerical instability during both the transient phase (e.g., for the



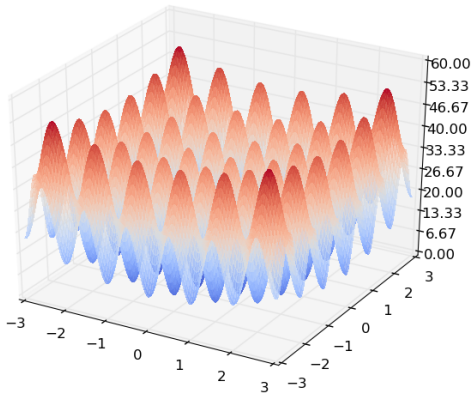
(a) Ackley function



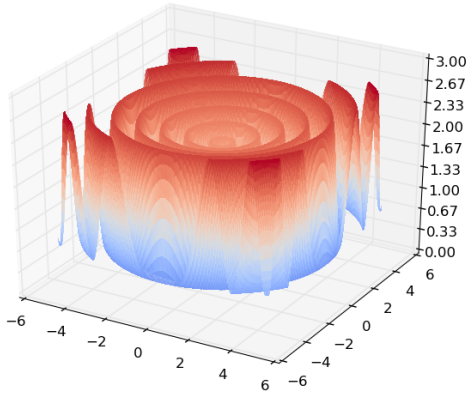
(b) Griewank function



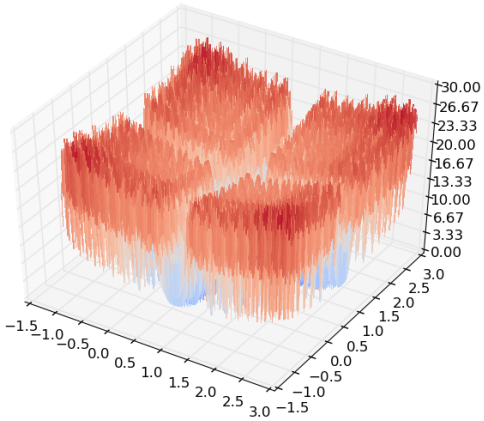
(c) Pintér's function



(d) Rastrigin function

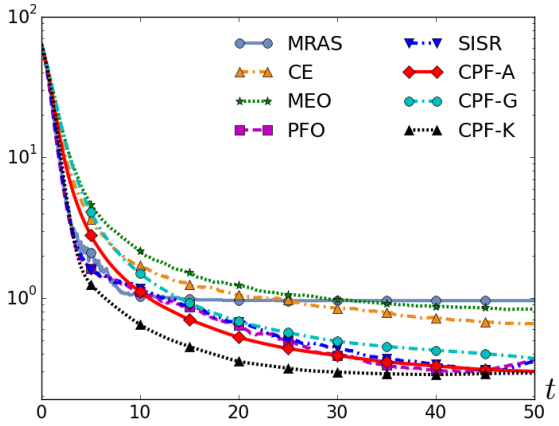


(e) Salomon function

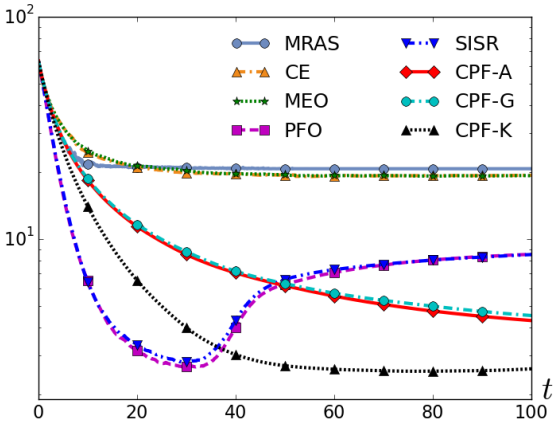


(f) Trigonometric function

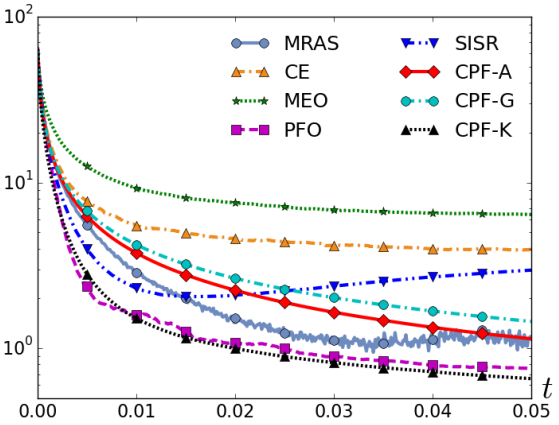
Figure 7.8: Plots of some benchmark objective functions in 2D.



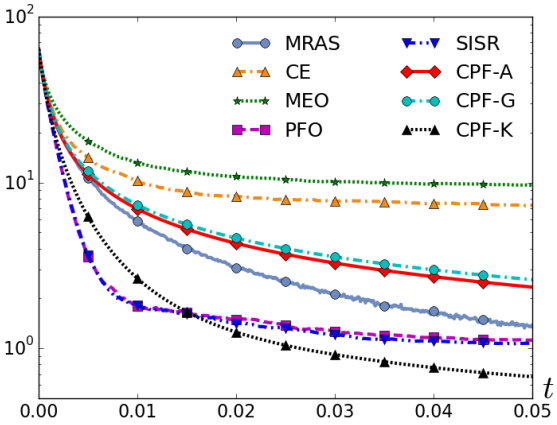
(a) Ackley function



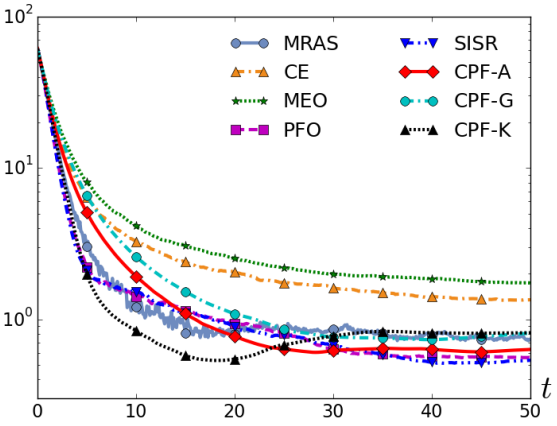
(b) Griewank function



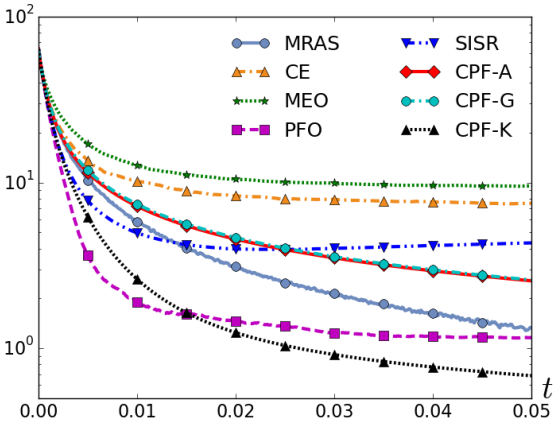
(c) Pintér's function



(d) Rastrigin function



(e) Salomon function



(f) Trigonometric function

Figure 7.9: M.C. average of the empirical error $\hat{e}_t^{(N)}$ as a function of t.

Table 7.4: Monte-Carlo average and std. dev. of $|m_{T,j}^{(N)} - \bar{x}|$ (Part I)

	Ackley	Griewank	Pintér's
MRAS	9.55e-1 (9.80e-1)	2.07e1 (8.30e0)	1.14e0 (8.23e-1)
CE	6.55e-1 (4.77e-1)	1.94e1 (7.14e0)	3.96e0 (1.98e0)
MEO	8.31e-1 (4.70e-1)	1.93e1 (5.87e0)	6.41e0 (3.08e0)
PFO	3.65e-1 (1.18e-1)	8.53e0 (2.91e0)	7.58e-1 (3.60e-1)
SISR	3.53e-1 (1.13e-1)	8.51e0 (2.70e0)	2.96e0 (1.17e0)
CPF-A	3.00e-1 (8.99e-2)	4.29e0 (1.91e0)	1.14e0 (9.27e-1)
CPF-G	3.71e-1 (1.53e-1)	4.52e0 (1.60e0)	1.45e0 (8.26e-1)
CPF-K	2.91e-1 (1.15e-1)	2.60e0 (7.75e-1)	6.57e-1 (2.64e-1)

Table 7.5: Monte-Carlo average and std. dev. of $|m_{T,j}^{(N)} - \bar{x}|$ (Part II)

	Rastrigin	Salomon	Trigonometric
MRAS	1.38e0 (3.35e-1)	7.34e-1 (5.06e-1)	1.33e0 (3.03e-1)
CE	7.29e0 (3.60e0)	1.34e0 (6.38e-1)	7.50e0 (3.20e0)
MEO	9.63e0 (4.24e0)	1.73e0 (1.04e0)	9.54e0 (4.12e0)
PFO	1.12e0 (4.03e-1)	5.58e-1 (1.77e-1)	1.16e0 (3.46e-1)
SISR	1.08e0 (3.57e-1)	5.34e-1 (1.90e-1)	4.32e0 (1.50e0)
CPF-A	2.34e0 (1.37e0)	6.31e-1 (2.21e-1)	2.55e0 (1.72e0)
CPF-G	2.60e0 (9.97-1)	8.09e-1 (4.45e-1)	2.59e0 (9.78e-1)
CPF-K	6.74e-1 (2.39e-1)	8.14e-1 (2.67e-1)	6.84e-1 (2.53e-1)

Ackley and Salomon function) and the asymptotic phase (e.g., for the Pintér's function). This is likely due to the randomness in the density projection step and the associated weigh calculation of the particles (see formula (7.5)).

4) The PFO and SISR algorithms perform identically except for the Pintér's and Trigonometric function. For all the problems, particularly the Griewank function, these two algorithms converge fast initially due to the strong diffusion noise added to expedites the search. However, strong diffusion may also adversely affect the asymptotic convergence rate, as shown in the cases with the Griewank, the Rastrigin and the Trigonometric functions.

5) The Griewank function is challenging due to the fact that the global minimum value is very close to the multiple local minimum values around it. While the parametric and the importance sampling-based algorithms eventually stagnates at a local minimum, the CPF algorithms are capable of continuing the search towards areas closer to the global minimizer.

The M.C. average and standard deviation of the empirical error at the terminal time T is tabulated in Table 7.4 and Table 7.5. These metrics are calculated according to (7.18a)-(7.18b). For most of the problems, the CE and the MEO algorithms exhibit relatively larger error and simulation variance, whereas the CPF-K algorithm performs the best with the lowest error and simulation variance. The importance sampling-based algorithms (PFO and SISR) and the other CPF algorithms (CPF-A and CPF-G) exhibit similar error patterns.

7.6 Conclusions

Numerical studies of the controlled particle filter (CPF) algorithm as well as several model-based algorithms were carried out for multiple global optimization problems, including a quadratic function, a one-dimensional Double-well potential, and six benchmark optimization problems of dimension five.

The conclusions are as follows:

1. For the quadratic Gaussian case, the MRAS algorithm exhibited the best accuracy and robustness with the increase of dimension. Among the other algorithms, the CPF algorithm with the affine control law provided more accurate approximation of the Gaussian posterior. However, the computation complexity of CPF-A scales quadratically with the dimension.

2. For the benchmark optimization problems, the algorithms were assessed based on Monte-Carlo simulations. For most of the test problems, the CPF algorithms exhibited higher potential to prevent premature convergence. The CPF algorithm with the kernel-based control law exhibited the lowest error and simulation variance among all the algorithms.

It is the subject of ongoing and future work to compare the FPF with more global optimization algorithms, e.g., the SMC-SA algorithm (see Sec. 7.2.5), the particle swarm optimization algorithm [91], and the consensus-based algorithm [128]. It is also desirable to pursue deeper theoretical analysis of these algorithms for certain type of problems, e.g., the quadratic problem and the more general convex optimization problems.

Part III

Appendices

Appendix A

Proof for Chapter 2

A.1 Proof of Proposition 2.1

For any function $f \in C_c^\infty(G)$, $f(X_t^i)$ is a continuous semimartingale that satisfies [74],

$$\mathrm{d}f(X_t^i) = (V_0 + \mathsf{u}) \cdot f(X_t^i)\,\mathrm{d}t + V_\alpha \cdot f(X_t^i) \circ \mathrm{d}B_t^{\alpha,i} + \mathsf{K}_j \cdot f(X_t^i) \circ \mathrm{d}Z_t^j. \tag{A.1}$$

For the ease of taking the expectation, we convert (A.1) to its Itô form (see Theorem 1.2 in [188]): For real-valued continuous semi-martingales A, B, C,

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

$$(A \circ \mathrm{d}B)\,\mathrm{d}C = A(\mathrm{d}B\,\mathrm{d}C). \tag{A.3}$$

For the second term on the right hand side of (A.1), taking A in (A.2) to be $V_\alpha \cdot f(X_t^i)$ and B to be $B_t^{\alpha,i}$,

$$V_\alpha \cdot f(X_t^i) \circ \mathrm{d}B_t^{\alpha,i} = V_\alpha \cdot f(X_t^i)\,\mathrm{d}B_t^{\alpha,i} + \frac{1}{2}\mathrm{d}(V_\alpha \cdot f)(X_t^i)\,\mathrm{d}B_t^{\alpha,i}. \tag{A.4}$$

Replacing f by $V_\alpha \cdot f$ in (A.1),

$$\mathrm{d}(V_\alpha \cdot f) = (V_0 + \mathsf{u}) \cdot (V_\alpha \cdot f)\,\mathrm{d}t + V_\beta \cdot (V_\alpha \cdot f) \circ \mathrm{d}B_t^{\beta,i} + \mathsf{K}_j \cdot (V_\alpha \cdot f) \circ \mathrm{d}Z_t^j.$$

Using (A.3) and Itô's rule ($\mathrm{d}B_t^{\alpha,i}\,\mathrm{d}B_t^{\beta,i} = \delta_{\alpha,\beta}\,\mathrm{d}t$, $\mathrm{d}B_t^{\alpha,i}\,\mathrm{d}t = 0$, and $\mathrm{d}B_t^{\alpha,i}\,\mathrm{d}Z_t^j = 0$ for all α, β, j),

$$\mathrm{d}(V_\alpha \cdot f)(X_t^i)\,\mathrm{d}B_t^{\alpha,i} = \sum_{\alpha=1}^{r} V_\alpha \cdot (V_\alpha \cdot f)(X_t^i)\,\mathrm{d}t,$$

which when substituted in (A.4) yields,

$$V_\alpha \cdot f(X_t^i) \circ \mathrm{d}B_t^{\alpha,i} = V_\alpha \cdot f(X_t^i)\,\mathrm{d}B_t^{\alpha,i} + \frac{1}{2}\sum_{\alpha=1}^{r} V_\alpha \cdot (V_\alpha \cdot f)(X_t^i)\,\mathrm{d}t.$$

The third term on the right hand side of (A.1) is similarly converted. The Itô form of (A.1) is then given by,

$$\mathrm{d}f(X_t^i) = \mathscr{L} f(X_t^i)\,\mathrm{d}t + V_\alpha \cdot f(X_t^i)\,\mathrm{d}B_t^{\alpha,i} + \mathsf{K}_j \cdot f(X_t^i,t)\,\mathrm{d}Z_t^j,$$

where the operator $\mathscr{L}$ is defined by,

$$\mathscr{L} f := (V_0 + \mathsf{u}) \cdot f + \frac{1}{2}\sum_{\alpha=1}^{r} V_\alpha \cdot (V_\alpha \cdot f) + \frac{1}{2}\sum_{j=1}^{m} \mathsf{K}_j \cdot (\mathsf{K}_j \cdot f).$$

In its integral form,

$$f(X_t^i) = f(X_0^i) + \int_0^t \mathscr{L} f(X_s^i)\,\mathrm{d}s + \int_0^t V_\alpha \cdot f(X_s^i)\,\mathrm{d}B_s^{\alpha,i} + \int_0^t \mathsf{K}_j \cdot f(X_s^i)\,\mathrm{d}Z_s^j.$$

By taking conditional expectation on both sides, interchanging expectation and integration (see Lemma 5.4 in [192]) and noting the fact that $B_t^{\alpha,i}$ is a Wiener process,

$$\pi_t(f) = \pi_0(f) + \int_0^t \pi_s(\mathscr{L} f)\,\mathrm{d}s + \int_0^t \pi_s(\mathsf{K}_j \cdot f)\,\mathrm{d}Z_s^j,$$

which is the desired formula (2.21). □

A.2 Proof of Theorem 2.1

Using (2.2) and (2.21) and the expressions for the operators $\mathscr{L}^*$ and $\mathscr{L}$, it suffices to show that,

$$\begin{aligned}
&\pi_s(\mathsf{u} \cdot f)\,\mathrm{d}s + \frac{1}{2}\sum_{j=1}^{m} \pi_s\big(\mathsf{K}_j \cdot (\mathsf{K}_j \cdot f)\big)\,\mathrm{d}s + \pi_s(\mathsf{K}_j \cdot f)\,\mathrm{d}Z_s^j \\
&= \sum_{j=1}^{m} \big(\pi_s(fh_j) - \pi_s(h_j)\pi_s(f)\big)\big(\mathrm{d}Z_s^j - \pi_s(h_j)\,\mathrm{d}s\big)
\end{aligned} \tag{A.5}$$

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

On taking $\psi = f$ in (2.22) and using the formula (2.10) with $V = \mathsf{K}_j$ and $\mathscr{V} = \mathrm{grad}(\phi_j)$,

$$\pi_s\big(\mathsf{K}_j \cdot f\big) = \pi_s\big(\langle \mathrm{grad}(\phi_j), \mathrm{grad}(f)\rangle\big) = \pi_s\big((h_j - \pi_s(h_j))f\big). \tag{A.6}$$

Using the expression (2.25) for the control function and noting that $\hat{h}_j = \pi_s(h_j)$,

$$\mathsf{u} \cdot f = -\frac{1}{2}\sum_{j=1}^{m}\big(h_j - \pi_s(h_j)\big)\big(\mathsf{K}_j \cdot f\big) - \sum_{j=1}^{m}\pi_s(h_j)\big(\mathsf{K}_j \cdot f\big).$$

Using (A.6) repeatedly then leads to,

$$\pi_s(\mathsf{u}\cdot f) = -\frac{1}{2}\sum_{j=1}^{m}\pi_s\big(\mathsf{K}_j\cdot(\mathsf{K}_j\cdot f)\big) - \sum_{j=1}^{m}\pi_s(h_j)\pi_s\big((h_j-\pi_s(h_j))f\big). \tag{A.7}$$

The desired equality (A.5) is now verified by substituting in (A.6) and (A.7). □

A.3 Proof of Proposition 2.2

Given a basis $\{E_1, E_2, E_3\}$ of the Lie algebra $so(3)$, the strong form Poisson equation (2.30) is expressed as,

$$\sum_{n=1}^{3} E_n \cdot \big(\rho\, \mathsf{k}_{n,j}\big) = -\frac{1}{\sigma_W^2}(h_j - \hat{h}_j)\,\rho, \tag{A.8}$$

where we used the formula (2.12) of $\mathrm{div}(\cdot)$, and the relation $\mathrm{grad}(\phi_j)(x) = x\mathsf{K}_j(x)$. For $n = 1,2,3$, $\mathsf{k}_{n,j} : SO(3) \to \mathbb{R}$, and $[\mathsf{K}_j] = (\mathsf{k}_{1,j}, \mathsf{k}_{2,j}, \mathsf{k}_{3,j})$.

Since $\chi \sim \mathscr{N}(0, \overline{\Sigma})$, $R = \mu\,\exp(\varepsilon\,[\chi]_\times)$, the density ρ is of the form [10],

$$\rho(R) = C(R)\exp\Big(-\frac{1}{2}\big[\,\log(\mu^T R)\,\big]^{\vee T}\Sigma^{-1}\big[\,\log(\mu^T R)\,\big]^{\vee}\Big),$$

where $C(R) \approx 1/\sqrt{(2\pi)^3|\Sigma|}$ (i.e., a constant) when the distribution is concentrated [186], and $\Sigma := \varepsilon^2\overline{\Sigma}$. By definition,

$$\begin{aligned} E_n \cdot \rho &= \frac{\mathrm{d}}{\mathrm{d}\tau}\Big|_{\tau=0}\rho\big(R\exp(\tau E_n)\big) \\ &= \frac{\mathrm{d}}{\mathrm{d}\tau}\Big|_{\tau=0}\frac{1}{\sqrt{(2\pi)^3|\Sigma|}}\exp\Big(-\frac{1}{2}\big[\,\log(\mu^T R\exp(\tau E_n))\,\big]^{\vee T}\Sigma^{-1}\big[\,\log(\mu^T R\exp(\tau E_n))\,\big]^{\vee}\Big). \end{aligned} \tag{A.9}$$

Furthermore,

$$\begin{aligned}\left[\log(\mu^T R\exp(\tau E_n))\right]^\vee &= \left[\log\left(\exp(\varepsilon[\chi]_\times)\exp([\tau e_n]_\times)\right)\right]^\vee \\ &= \varepsilon\chi+\tau e_n+\frac{\tau}{2}\varepsilon[\chi]_\times e_n+\frac{\tau}{12}\varepsilon^2[\chi]_\times^2 e_n-\frac{\tau}{720}\varepsilon^4[\chi]_\times^4 e_n+O(\tau^2) \\ &= \varepsilon\chi+\tau M e_n+O(\tau^2),\end{aligned}$$

where $M = I+\frac{1}{2}\varepsilon[\chi]_\times+\frac{1}{12}\varepsilon^2[\chi]_\times^2-\frac{1}{720}\varepsilon^4[\chi]_\times^4 = I+O(\varepsilon)$, $\{e_1,\, e_2,\, e_3\}$ denote the canonical basis of $\mathbb{R}^3$, and the second equality above used the Baker-Campbell-Hausdorff (BCH) formula to expand the logarithm of the product of two matrix exponentials (see Chapter 3 of [68]). Substituting this result into (A.9), we obtain,

$$E_n\cdot\rho = -\frac{1}{\varepsilon}\chi^T\overline{\Sigma}^{-1}M e_n\rho. \tag{A.10}$$

Using (A.10), the left-hand-side of (A.8) is expanded as follows,

$$\sum_{n=1}^{3}E_n\cdot\left(\rho\,\mathsf{k}_{n,j}\right) = \sum_{n=1}^{3}\left(E_n\cdot\rho\right)\mathsf{k}_{n,j}+\rho\sum_{n=1}^{3}E_n\cdot\mathsf{k}_{n,j} = -\frac{1}{\varepsilon}\chi^T\overline{\Sigma}^{-1}M[\mathsf{K}_j]\rho+\rho\sum_{n=1}^{3}E_n\cdot\mathsf{k}_{n,j}.$$

The Taylor expansion of h_j is given by,

$$h_j(R) = h_j(\mu)+\varepsilon\chi^T H_j+O(\varepsilon^2),$$

where $H_j := \left(E_1\cdot h_j(\mu), E_2\cdot h_j(\mu), E_3\cdot h_j(\mu)\right)\in\mathbb{R}^3$. Using the fact that $\pi(\chi)=0$, we have $\hat{h}_j = \pi(h_j) = h_j(\mu)+O(\varepsilon^2)$, leading to $h_j-\hat{h}_j = \varepsilon\chi^T H_j+O(\varepsilon^2)$. Hence, (A.8) becomes,

$$-\chi^T\overline{\Sigma}^{-1}M[\mathsf{K}_j]+\varepsilon\sum_{n=1}^{3}E_n\cdot\mathsf{k}_{n,j} = -\frac{1}{\overline{\sigma}_W^2}\chi^T H_j+O(\varepsilon).$$

In the asymptotic limit as $\varepsilon\to 0$, $[\mathsf{K}_j] = \frac{1}{\overline{\sigma}_W^2}\overline{\Sigma}H_j+O(\varepsilon)$. □

A.4 Proof of Theorem 2.2

Under the constant gain approximation, the FPF is given by (see (2.32)),

$$\mathrm{d}R_t^i = R_t^i\,\Omega_t\,\mathrm{d}t + R_t^i \circ [\,\varepsilon\,\overline{\sigma}_B\,\mathrm{d}B_t^i\,]_\times + R_t^i\,[\,\mathsf{K}_t \circ \mathrm{dI}_t^i\,]_\times, \tag{A.11}$$

where $\mathsf{K}_t := \frac{1}{\overline{\sigma}_W^2}\,\overline{\Sigma}_t\,H_t^T$ and $H_t = [\,\mu_t^T r\,]_\times$. A concentrated distribution is assumed, i.e.,

$$R_t^i = \mu_t\,\exp(\varepsilon\,[\,\chi_t^i\,]_\times) = \mu_t + \varepsilon\,\mu_t\,[\,\chi_t^i\,]_\times + O(\varepsilon^2), \tag{A.12}$$

where $\chi_t^i \sim \mathcal{N}(0, \overline{\Sigma}_t)$.

The evolution equation for the mean μ_t and covariance $\overline{\Sigma}_t$ are derived using a perturbation analysis approach. We begin by simplifying the modified form of the innovation error,

$$\begin{aligned}
\mathrm{dI}_t^i &= \mathrm{d}Z_t - \frac{R_t^{iT} r + \pi_t(R_t^{iT} r)}{2}\,\mathrm{d}t = \mathrm{d}Z_t - \frac{R_t^{iT} r + \mu_t^T r}{2}\,\mathrm{d}t + O(\varepsilon^2) \\
&= \mathrm{dI}_t - \frac{R_t^{iT} r - \mu_t^T r}{2}\,\mathrm{d}t + O(\varepsilon^2) = \mathrm{dI}_t - \frac{\exp(-\varepsilon\,[\,\chi_t^i\,]_\times)\,\mu_t^T r - \mu_t^T r}{2}\,\mathrm{d}t + O(\varepsilon^2) \qquad \text{(A.13)} \\
&= \mathrm{dI}_t + \frac{1}{2}\,\varepsilon\,[\,\chi_t^i\,]_\times \mu_t^T r\,\mathrm{d}t + O(\varepsilon^2) = \mathrm{dI}_t - \frac{1}{2}\,\varepsilon\,H_t\,\chi_t^i\,\mathrm{d}t + O(\varepsilon^2), \qquad \text{(A.14)}
\end{aligned}$$

where $\mathrm{dI}_t = \mathrm{d}Z_t - \mu_t^T r\,\mathrm{d}t$, and we used the fact that $\pi_t(R_t^{iT} r) = \mu_t^T r + O(\varepsilon^2)$ due to the representation (A.12).

On substituting (A.12) and (A.14) into the FPF (A.11) and matching terms, the $O(1)$ balance gives,

$$\mathrm{d}\mu_t = \mu_t\,\Omega_t\,\mathrm{d}t + \mu_t[\,\mathsf{K}_t \circ \mathrm{dI}_t\,]_\times. \tag{A.15}$$

The $O(\varepsilon)$ balance gives,

$$\mathrm{d}(\mu_t\,[\,\chi_t^i\,]_\times) = \mu_t\,[\,\chi_t^i\,]_\times\,\Omega_t\,\mathrm{d}t + \overline{\sigma}_B\,\mu_t\,[\,\mathrm{d}B_t^i\,]_\times + \mu_t\,[\,\chi_t^i\,]_\times[\,\mathsf{K}_t \circ \mathrm{dI}_t\,]_\times - \frac{1}{2}\,\mu_t\,[\,\mathsf{K}_t\,H_t\,\chi_t^i\,]_\times\,\mathrm{d}t.$$

Using the formula (A.15), this is simplified to obtain the following equation of χ_t^i, expressed in its Itô form:

$$\mathrm{d}\chi_t^i = A_t\chi_t^i\,\mathrm{d}t + \sigma_B\,\mathrm{d}B_t^i - \frac{1}{2}\,\mathsf{K}_t\,H_t\,\chi_t^i\,\mathrm{d}t - [\,\mathsf{K}_t\,\mathrm{dI}_t\,]_\times\,\chi_t^i + O(\varepsilon^2), \tag{A.16}$$

where $A_t = -\Omega_t$.

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

$$\begin{aligned}
\mathrm{d}\Gamma_t^i =& \ \mathrm{d}\chi_t^i \chi_t^{iT} + \chi_t^i \,\mathrm{d}(\chi_t^{iT}) + \mathrm{d}\chi_t^i \,\mathrm{d}(\chi_t^{iT}) \\
=& \left(A_t \,\mathrm{d}t - [\,\mathsf{K}_t \,\mathrm{dI}_t\,]_\times\right)\Gamma_t^i + \Gamma_t^i\left(A_t \,\mathrm{d}t - [\,\mathsf{K}_t \,\mathrm{dI}_t\,]_\times\right)^T + \overline{\sigma}_B \left(\mathrm{d}B_t^i \chi_t^{iT} + \chi_t^i \,\mathrm{d}B_t^{iT}\right) + \overline{\sigma}_B^2 I \,\mathrm{d}t \\
& - \frac{1}{2}\left(\mathsf{K}_t H_t \Gamma_t^i + \Gamma_t^i H_t^T \mathsf{K}_t^T\right)\mathrm{d}t + O(\varepsilon^2).
\end{aligned}$$

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

$$\mathrm{d}\overline{\Sigma}_t = \left(A_t \,\mathrm{d}t - [\,\mathsf{K}_t \,\mathrm{dI}_t\,]_\times\right)\overline{\Sigma}_t + \overline{\Sigma}_t\left(A_t \,\mathrm{d}t - [\,\mathsf{K}_t \,\mathrm{dI}_t\,]_\times\right)^T + \overline{\sigma}_B^2 I \,\mathrm{d}t - \frac{1}{\overline{\sigma}_W^2}\overline{\Sigma}_t H_t^T H_t \overline{\Sigma}_t \,\mathrm{d}t,$$

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

Appendix B

Proof for Chapter 3

B.1 Gradient Flow

Proof of Theorem 3.1: As $\Delta t_n \downarrow 0$, $\rho^{(\bar{N})}(x,t) \to p^*(x,t)$, the posterior density defined in (3.1). By direct substitution, it is verified that p^* is a solution of the replicator pde (3.8).

In the conference version of Chapter 3 (see [204]), the replicator pde is derived based on variational analysis. The main steps of the variational proof are as follows:

(i) By taking the first variation of the functional (3.7), the minimizer ρ_n is shown to satisfy the E-L equation:

$$\int \frac{\rho_n}{\rho_{n-1}} \nabla \cdot (\rho_{n-1}\, \varsigma)\, \mathrm{d}x - \Delta t_n\, \beta \int \nabla h \cdot \varsigma \rho_n\, \mathrm{d}x = 0, \tag{B.1}$$

for each vector field $\varsigma \in L^2(\mathbb{R}^d; \rho_{n-1})$.

(ii) Given any C^1 smooth and compactly supported (test) function f, let $\xi_n \in L^2(\mathbb{R}^d; \rho_{n-1})$ be the solution of

$$\nabla \cdot (\rho_{n-1} \xi_n) = (f - \hat{f}_{n-1})\rho_{n-1}, \tag{B.2}$$

where $\hat{f}_{n-1} := \int f \rho_{n-1}\, \mathrm{d}x$. Then, using the E-L equation (B.1),

$$\hat{f}_n - \hat{f}_{n-1} = \Delta t_n\, \beta \int \nabla h \cdot \xi_n \rho_n\, \mathrm{d}x,$$

and upon summing,

$$\hat{f}_{\bar{N}} = \hat{f}_0 + \beta \sum_{n=1}^{\bar{N}} \Delta t_n \int \nabla h \cdot \xi_n \rho_n\, \mathrm{d}x. \tag{B.3}$$

(iii) Integrating by parts and using (B.2),

$$\int \nabla h \cdot \xi_n \rho_n \, \mathrm{d}x = -\int h(f - \hat{f}_{n-1}) \rho_{n-1} \, \mathrm{d}x + \mathscr{E}_n,$$

where the error term $\mathscr{E}_n = O(\Delta t_n)$. Equation (B.3) thus becomes

$$\hat{f}_{\bar{N}} = \hat{f}_0 - \beta \sum_{n=1}^{\bar{N}} \Delta t_n \int h(f - \hat{f}_{n-1}) \rho_{n-1} \, \mathrm{d}x + \sum_{n=1}^{\bar{N}} \Delta t_n \mathscr{E}_n.$$

(iv) On taking the limit as $\Delta t_n \downarrow 0$, the limiting density $p^*(x,t)$ satisfies

$$\begin{aligned} \hat{f}_t &= \hat{f}_0 - \beta \int_0^t \int h(x)(f(x) - \hat{f}_s) p^*(x,s) \, \mathrm{d}x \, \mathrm{d}s \\ &= \hat{f}_0 - \beta \int_0^t \int (h(x) - \hat{h}_s) f(x) p^*(x,s) \, \mathrm{d}x \, \mathrm{d}s \end{aligned} \tag{B.4}$$

for all test functions f, showing that $p^*(x,t)$ is a weak solution of the replicator pde (3.8). For additional details on these calculations, see [204].

B.2 Optimal Control

Preliminaries: Consider a functional $E : \mathscr{P} \to \mathbb{R}$ mapping densities to real numbers. For a fixed $\rho \in \mathscr{P}$, the (Gâteaux) derivative of E is a real-valued function on $\mathbb{R}^d$, and an element of the function space $L^2(\mathbb{R}^d; \rho)$ [16]. This function is denoted as $\frac{\partial E}{\partial \rho}(\rho, t)(x)$ for $x \in \mathbb{R}^d$, and defined as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} E(\rho_t)\Big|_{t=0} = -\int_{\mathbb{R}^d} \frac{\partial E}{\partial \rho}(\rho)(x) \nabla \cdot (\rho(x) u(x)) \, \mathrm{d}x,$$

where ρ_t is a path in $\mathscr{P}$ such that $\frac{\partial \rho_t}{\partial t} = -\nabla \cdot (\rho_t u)$ with $\rho_0 = \rho$, and u is any arbitrary vector-field on $\mathbb{R}^d$. Similarly, $\frac{\partial^2 E}{\partial \rho^2}(\rho) \in L^2(\mathbb{R}^d \times \mathbb{R}^d)$ is the second (Gâteaux) derivative of the functional E if

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial E}{\partial \rho}(\rho_t)(x)\Big|_{t=0} = -\int_{\mathbb{R}^d} \frac{\partial^2 E}{\partial \rho^2}(\rho)(x,y) \nabla \cdot (\rho(y) u(y)) \, \mathrm{d}y.$$

The optimal control problems (3.10) and (3.14) are examples of the mean-field type control problem introduced in [16]. The notation and the methodology for the following proofs is based in part on [16].

Proof of Theorem 3.2: The value function $V(\rho,t)$, defined in (3.12), is the solution of the DP equation (i.e., the HJB equation):

$$\begin{aligned} &\frac{\partial V}{\partial t}(\rho,t)+\inf_{u\in L^2}\ \mathsf{H}(\rho,\frac{\partial V}{\partial \rho}(\rho,t),u)=0, \quad t\in[0,T),\\ &V(\rho,T)=\beta\int h(x)\rho(x)\,\mathrm{d}x. \end{aligned} \tag{B.5}$$

In the following, we use the notation

$$\Theta=\Theta(\rho,t)(x):=\frac{\partial V}{\partial \rho}(\rho,t)(x).$$

For a fixed $\rho\in\mathscr{P}$ and $t\in[0,T)$, Θ is a function on $\mathbb{R}^d$.

A necessary condition is obtained by considering the first variation of H. Suppose u is a minimizing control function. Then u satisfies the first order optimality condition:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathsf{H}(\rho,\Theta,u+\varepsilon v)\Big|_{\varepsilon=0}=0,$$

where v is an arbitrary vector field on $\mathbb{R}^d$. Explicitly,

$$\int\nabla(-\frac{1}{\rho}\nabla\cdot(\rho u)+\Theta)\cdot v\,\rho\,\mathrm{d}x=0,$$

or in its strong form

$$-\frac{1}{\rho}\nabla\cdot(\rho u)+\Theta=(\text{constant}).$$

Multiplying both sides by ρ and integrating yields the value of the constant as $\int\Theta(\rho,t)(x)\rho(x)\,\mathrm{d}x=:\hat{\Theta}(\rho,t)$. Therefore, the minimizing control solves the pde

$$\frac{1}{\rho}\nabla\cdot(\rho u)=\Theta-\hat{\Theta}.$$

On substituting the optimal control law into (B.5), the HJB equation for the value function is given by

$$\begin{aligned} &\frac{\partial V}{\partial t}(\rho,t)+\frac{\beta^2}{2}\int|h-\hat{h}|^2\rho\,\mathrm{d}x-\frac{1}{2}\int|\Theta(\rho,t)-\hat{\Theta}(\rho,t)|^2\rho\,\mathrm{d}x=0, \quad t\in[0,T),\\ &V(\rho,T)=\beta\int h(x)\rho(x)\,\mathrm{d}x. \end{aligned}$$

The equation involves both V and Θ. One obtains the so-called master equation (see [16]) involving only Θ by differentiating with respect to ρ

$$\begin{aligned}
&\frac{\partial\Theta}{\partial t}(\rho,t)(x)+\frac{\beta^2}{2}|h(x)-\hat{h}|^2-\frac{1}{2}|\Theta(\rho,t)(x)-\hat{\Theta}(\rho,t)|^2\\
&-\int(\Theta(\rho,t)(y)-\hat{\Theta}(\rho,t))\frac{\partial\Theta}{\partial\rho}(\rho,t)(y,x)\rho(y)\,\mathrm{d}y=0,\ t\in[0,T),\\
&\Theta(\rho,T)=\beta h.
\end{aligned}$$

It is easily verified that $\Theta(\rho,t)=\beta h$ solves the master equation. The corresponding value function $V(\rho,t)=\beta\int h\rho\,\mathrm{d}x$.

Sufficiency: The proof that the proposed control law is a minimizer is as follows. Consider any arbitrary control law v_t with the resulting density ρ_t. Taking the time derivative of $-\beta\int h\rho_t\,\mathrm{d}x$:

$$\begin{aligned}
-\beta\frac{\mathrm{d}}{\mathrm{d}t}\int h\rho_t\,\mathrm{d}x&=\beta\int h\,\nabla\cdot(\rho_t v_t)\,\mathrm{d}x\\
&=\int\beta(h-\hat{h}_t)\,(\frac{1}{\rho_t}\nabla\cdot(\rho_t v_t))\,\rho_t\,\mathrm{d}x\\
&\le\int\left(\frac{\beta^2}{2}|h-\hat{h}|^2+\frac{1}{2}\left|\frac{1}{\rho_t}\nabla\cdot(\rho_t v_t)\right|^2\right)\rho_t\,\mathrm{d}x\\
&=\mathsf{L}(\rho_t,v_t).
\end{aligned}$$

On integrating both sides with respect to time,

$$\beta\int_{\mathbb{R}^d}h\rho_0\,\mathrm{d}x\ \le\ \int_0^T\mathsf{L}(\rho_t,v_t)\,\mathrm{d}t\ +\ \beta\int_{\mathbb{R}^d}h\rho_T\,\mathrm{d}x,$$

where the equality holds with $v_t=u_t$ (defined as solution of (3.13)). Therefore,

$$J(u)=\beta\int h\rho_0\,\mathrm{d}x\le J(v).$$

This also confirms that $V(\rho,t)=\beta\int h\rho\,\mathrm{d}x$ is the value function, and completes the proof of Theorem 3.2. □

The analysis for the infinite horizon optimal control problem (3.14) is similar and described next.

Proof of Theorem 3.3: The infinite-horizon value function $V^\infty(\rho):=\inf_u\int_0^\infty\mathsf{L}(\rho_t,u_t)\,\mathrm{d}t$ is a solution of the

DP equation:

$$\inf_{u \in L^2} \mathsf{H}(\rho, \Theta^\infty(\rho), u) = 0, \tag{B.6}$$

where $\Theta^\infty(\rho) := \frac{\partial V^\infty}{\partial \rho}(\rho)$. By carrying out the first order analysis in an identical manner, it is readily verified that:

(i) A minimizing control u is a solution of the pde (3.13);

(ii) $V^\infty(\rho) = \beta \int h\rho \, \mathrm{d}x - \beta h(\bar{x})$ is a solution of the DP equation (B.6).

The sufficiency also follows similarly. With any arbitrary control v_t,

$$\beta \int_{\mathbb{R}^d} h\rho_0 \, \mathrm{d}x \le \int_0^\infty \mathsf{L}(\rho_t, v_t) \, \mathrm{d}t + \beta \limsup_{t \to \infty} \int_{\mathbb{R}^d} h\rho_t \, \mathrm{d}x,$$

with equality if $v_t = u_t$ solves the pde (3.13). Using the boundary condition, $\limsup_{t\to\infty} \int h\rho_t \, \mathrm{d}x = h(\bar{x})$,

$$J(u) = \beta \int h\rho_0 \, \mathrm{d}x - \beta h(\bar{x}) \le J(v).$$

□

Proof of Lemma 3.1: Suppose ϕ is the unique solution of the Poisson equation (3.5) (Theorem 4.1 in Chapter 4 and Theorem 2.2 in [100]). Then $u = -\beta \nabla \phi$ is a particular solution of the pde (3.13). The general solution is then given by $u = v - \beta \nabla \phi$ where v is a null solution, i.e., $\nabla \cdot (\rho v) = 0$. The L^2 optimality of the gradient solution follows from the simple calculation:

$$\begin{aligned} \int |u|^2 \rho \, \mathrm{d}x &= \int \beta^2 |\nabla \phi|^2 \rho \, \mathrm{d}x + \int |v|^2 \rho \, \mathrm{d}x - 2\beta \int v \cdot \nabla \phi \rho \, \mathrm{d}x \\ &= \beta^2 \|\nabla \phi\|_2^2 + \|v\|_2^2, \end{aligned}$$

because $\int \nabla \phi \cdot v \rho \, \mathrm{d}x = -\int \phi \nabla \cdot (\rho v) \, \mathrm{d}x = 0$. □

B.3 Hamiltonian Formulation

The Hamiltonian H is defined in (3.11). Suppose u_t is the optimal control and ρ_t is the corresponding optimal trajectory. Denote the trajectory for the momentum (co-state) as q_t. Using the Pontryagin's minimum

principle, (ρ_t, q_t) satisfy the following Hamilton's equations:

$$\begin{aligned}
\frac{\partial \rho_t}{\partial t} &= \frac{\partial \mathsf{H}}{\partial \mathsf{q}}(\rho_t, \mathsf{q}_t, u_t), \quad \rho_0 = p_0^*, \\
\frac{\partial \mathsf{q}_t}{\partial t} &= -\frac{\partial \mathsf{H}}{\partial \rho}(\rho_t, \mathsf{q}_t, u_t), \quad \mathsf{q}_T = \frac{\partial}{\partial \rho}\left(\beta \int h(x)\rho(x)\,\mathrm{d}x\right), \\
0 &= \mathsf{H}(\rho_t, \mathsf{q}_t, u_t) = \min_{v \in L^2} \mathsf{H}(\rho_t, \mathsf{q}_t, v).
\end{aligned}$$

The calculus of variation argument in the proof of Theorem 3.2 shows that the minimizing control u_t solves the first order optimality equation

$$\frac{1}{\rho_t}\nabla \cdot (\rho_t u_t) = \mathsf{q}_t - \hat{\mathsf{q}}_t, \tag{B.7}$$

where $\hat{\mathsf{q}}_t := \int \mathsf{q}_t(x)\rho_t(x)\,\mathrm{d}x$.

The explicit form of the Hamilton's equations are obtained by explicitly evaluating the derivatives along the optimal trajectory:

$$\begin{aligned}
\frac{\partial \mathsf{H}}{\partial \mathsf{q}}(\rho_t, \mathsf{q}_t, u_t) &= -\nabla \cdot (\rho_t u_t), \\
\frac{\partial \mathsf{H}}{\partial \rho}(\rho_t, \mathsf{q}_t, u_t) &= \frac{\beta^2}{2}(h - \hat{h}_t)^2 - \frac{1}{2}(\mathsf{q}_t - \hat{\mathsf{q}}_t)^2.
\end{aligned}$$

It is easy to verify that $\mathsf{q}_t \equiv \beta h(x)$ satisfies both the boundary condition and the evolution equation for the momentum. This results in a simpler form of the Hamilton's equations:

$$\begin{aligned}
\frac{\partial \rho_t}{\partial t} &= -\nabla \cdot (\rho_t u_t), \\
0 &= \mathsf{H}(\rho_t, \beta h, u_t) = \min_{v \in L^2} \mathsf{H}(\rho_t, \beta h, v).
\end{aligned}$$

In a particle filter implementation, the minimizing control $u_t = -\nabla\phi$ is obtained by solving the first order optimality equation (B.7) with $\mathsf{q}_t = \beta h$.

B.4 Exactness and Convergence

Before proving the Theorem 3.4, we state and prove the following technical Lemma:

Lemma B.1 *Suppose the prior density $p_0^*(x)$ satisfies Assumption 3.1 and the objective function $h(x)$ satisfies Assumption 3.2. Then for each fixed time $t \geq 0$:*

(i) *The posterior density $p^*(x,t)$, defined according to (3.1), admits a spectral bound;*

(ii) *The objective function $h \in L^2(\mathbb{R}^d; p^*(\cdot,t))$.*

Proof B.1 *Define $V_t(x) := -\log p^*(x,t) = V_0(x) + t\beta h(x) + \gamma_t$ where $\gamma_t := \log(\int e^{-V_0(y)-th(y)}\,\mathrm{d}y)$. It is directly verified that $V_t \in C^2$ with $D^2V_t \in L^\infty$ and $\liminf_{|x|\to\infty} \nabla V_t(x) \cdot \frac{x}{|x|} = \infty$. Therefore, the density $p^*(x,t)$ admits a spectral bound [Thm 4.6.3 in [6]]. The function h is square-integrable because*

$$\int |h(x)|^2 p^*(x,t)\,\mathrm{d}x \leq e^{-\beta t\bar{h}-\gamma_t} \int |h(x)|^2 e^{-V_0(x)}\,\mathrm{d}x < \infty.$$

□

Proof of Theorem 3.4: Given any C^1 smooth and compactly supported (test) function f, using the elementary chain rule,

$$\mathrm{d}f(X_t^i) = -\beta \nabla\phi(X_t^i) \cdot \nabla f(X_t^i).$$

On integrating and taking expectations,

$$\begin{aligned} \mathsf{E}[f(X_t^i)] &= \mathsf{E}[f(X_0^i)] - \beta \int_0^t \mathsf{E}[\nabla\phi(X_s^i) \cdot \nabla f(X_s^i)]\,\mathrm{d}s \\ &= \mathsf{E}[f(X_0^i)] - \beta \int_0^t \mathsf{E}[(h(X_s^i) - \hat{h}_s) f(X_s^i)]\,\mathrm{d}s, \end{aligned} \tag{B.8}$$

which is the weak form of the replicator pde (3.8). Note that the weak form of the Poisson equation (4.11) is used to obtain the second equality. Since the test function f is arbitrary, we conclude that the evolution of p and p^* are identical by comparing the equations (B.4) and (B.8). That the control function is well-defined for each time follows from Theorem 4.1 based on apriori estimate of h and the spectral bound in Lemma B.1 for $p = p^*$.

The convergence proof is presented next. The proof here is somewhat more general than needed to prove the Theorem. For a function h, we define the minimizing set:

$$A_0 := \{x \in \mathbb{R}^d \mid h(x) = \bar{h}\},$$

where it is recalled that $\bar{h} = \inf_{x\in\mathbb{R}^d} h(x)$. In the following it is shown that for *any* open neighborhood U of A_0,

$$\liminf_{t\to\infty} \int_U p(x,t)\,\mathrm{d}x = 1. \tag{B.9}$$

It then follows that X_t^i converges in distribution where the limiting distribution is supported on A_0 [Thm. 3.2.5 in [52]]. If the minimizer is unique (i.e., $A_0 = \{\bar{x}\}$), X_t^i converges to $\bar{x}$ in probability.

The key to prove the convergence is the following property of the function h:

(P1): For each $\delta > 0$, $\exists\ \varepsilon > 0$ such that:

$$|h(x) - \bar{h}| \le \varepsilon \quad \Rightarrow \quad \mathrm{dist}(x, A_0) \le \delta \quad \forall x \in \mathbb{R}^d,$$

where $\mathrm{dist}(x,A_0)$ denotes the Euclidean distance of point x from set A_0. If the minimizer $\bar{x}$ is unique, it equals $|x-\bar{x}|$.

Any lower semi-continuous function satisfying Assumption 3.3 also satisfies the property (P1): Suppose $\{x_n\}$ is a sequence such that $h(x_n) \to \bar{h}$. Then $\{x_n\}$ is compact because $h(x) > \bar{h} + r$ outside some compact set. Therefore, the limit set is non-empty and because h is lower semi-continuous, for any limit point z, $\bar{h} \le h(z) \le \liminf_{x_n\to z} h(x_n) = \bar{h}$. That is, $z \in A_0$.

The proof for (B.9) is based on construction of a Lyapunov function: Denote $A_\varepsilon := \{x \in \mathbb{R}^d \mid h(x) \le \bar{h} + \varepsilon\}$ where $\varepsilon > 0$. By property (P1), given any open neighborhood U containing A_0, $\exists\ \varepsilon > 0$ such that $A_\varepsilon \subset U$. A candidate Lyapunov function $V_{A_\varepsilon}(\mu) := -\beta^{-1}\log(\mu(A_\varepsilon))$ is defined for measure μ with everywhere positive density. By construction $V_{A_\varepsilon}(\mu) \ge 0$ with equality iff $\mu(A_\varepsilon) = 1$.

Let μ_t be the probability measure associated with $p(x,t)$, i.e, $\mu_t(B) := \int_B p(x,t)\,\mathrm{d}x$ for all Borel measurable set $B \subset \mathbb{R}^d$. Since $p(x,t)$ is a solution of the replicator pde,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} V_{A_\varepsilon}(\mu_t) &= \frac{\mathrm{d}}{\mathrm{d}t}\Big[-\frac{1}{\beta}\log(\mu_t(A_\varepsilon))\Big] \\
&= \frac{1}{\mu_t(A_\varepsilon)} \int_{A_\varepsilon} (h(x) - \hat{h}_t)\,\mathrm{d}\mu_t(x) \\
&= (1 - \mu_t(A_\varepsilon)) \left(\frac{\int_{A_\varepsilon} h\,\mathrm{d}\mu_t}{\mu_t(A_\varepsilon)} - \frac{\int_{A_\varepsilon^c} h\,\mathrm{d}\mu_t}{\mu_t(A_\varepsilon^c)} \right) \\
&\le 0
\end{aligned}$$

with equality iff $\mu_t(A_\varepsilon) = 1$.

For the objective function h, a direct calculation also shows:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int h(x)p(x,t)\,\mathrm{d}x &= -\beta\int h(x)(h(x)-\hat{h}_t)p(x,t)\,\mathrm{d}x\\ &= -\beta\int (h(x)-\hat{h})^2 p(x,t)\,\mathrm{d}x \le 0,\end{aligned}$$

with equality iff $h = \hat{h}$ almost everywhere (with respect to the measure μ_t). □

B.5 Quadratic Gaussian Case

Proof of Lemma 3.2: We are interested in obtaining an explicit solution of the Poisson equation,

$$-\nabla\cdot(\rho(x)\nabla\phi(x)) = (h(x)-\hat{h})\rho(x). \tag{B.10}$$

Consider the solution ansatz:

$$\nabla\phi(x) = \mathsf{K}(x-m)+b, \tag{B.11}$$

where the matrix $\mathsf{K} = \mathsf{K}^T \in \mathbb{R}^{d\times d}$ and the vector $b\in\mathbb{R}^d$ are determined as follows:

(i) Multiply both sides of (B.10) by vector x and integrate (element-by-element) by parts to obtain

$$b = \int x(h(x)-\hat{h})\rho(x)\,\mathrm{d}x. \tag{B.12}$$

(ii) Multiply both sides of (B.10) by matrix $(x-m)(x-m)^T$ and integrate by parts to obtain

$$\Sigma\mathsf{K}+\mathsf{K}\Sigma = \int (x-m)(x-m)^T(h(x)-\hat{h})\rho(x)\,\mathrm{d}x. \tag{B.13}$$

We have thusfar not used the fact that the density ρ is Gaussian and the function h is quadratic. In the following, it is shown that the solution thus defined in fact *solves* the pde (B.10) under these conditions.

A radially unbounded quadratic function is of the general form:

$$h(x) = \frac{1}{2}(x-\bar{x})^T H(x-\bar{x})+c$$

where the matrix $H = H^T \succ 0$ and c is some constant. For a Gaussian density ρ with mean m and variance $\Sigma \succ 0$, the integrals are explicitly evaluated to obtain

$$b = \int x(h(x) - \hat{h})\rho(x)\,\mathrm{d}x = \Sigma H(m - \bar{x}), \tag{B.14a}$$

$$\Sigma \mathsf{K} + \mathsf{K}\Sigma = \int (x-m)(x-m)^T (h(x) - \hat{h})\rho(x)\,\mathrm{d}x = \Sigma H \Sigma. \tag{B.14b}$$

A unique positive-definite symmetric solution K exists for the Lyapunov equation (B.14b) because $\Sigma \succ 0$ and $\Sigma H \Sigma \succ 0$ [50].

On substituting the solution (B.11) into the Poisson equation (B.10) and dividing through by ρ, the two sides are:

$$-\frac{1}{\rho}\nabla \cdot (\rho \nabla \phi) = (x-m)^T \Sigma^{-1}(\mathsf{K}(x-m) + b) - \mathrm{tr}(\mathsf{K}),$$
$$h - \hat{h} = \frac{1}{2}(x-\bar{x})^T H (x-\bar{x}) - \frac{1}{2}(m-\bar{x})^T H (m-\bar{x}) - \frac{1}{2}\mathrm{tr}(H\Sigma).$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace. Using formulae (B.14a)-(B.14b) for b and K, the two sides are seen to be equal. □

Proof of Proposition 3.1: Using the affine control law (3.16), the particle filter is a linear system with a Gaussian prior:

$$\frac{\mathrm{d}X_t^i}{\mathrm{d}t} = -\beta \mathsf{K}_t (X_t^i - m_t) - \beta b_t, \quad X_0^i \sim \mathcal{N}(m_0, \Sigma_0). \tag{B.15}$$

Therefore, the density of X_t^i is Gaussian for all $t > 0$. The evolution of the mean is obtained by taking an expectation of both sides of the ode (B.15):

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathsf{E}[X_t^i] = -\beta b_t = -\beta \mathsf{E}[X_t^i (h(X_t^i) - \hat{h}_t)],$$

where (3.17) is used to obtain the second equality. The equation for the variance Σ_t of X_t^i is similarly obtained:

$$\begin{aligned}\frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} &= -\beta(\mathsf{K}_t \Sigma_t + \Sigma_t \mathsf{K}_t) \\ &= -\beta \mathsf{E}\left[(X_t^i - m_t)(X_t^i - m_t)^T (h(X_t^i) - \hat{h}_t)\right],\end{aligned}$$

where (3.18) has been used. □

Proof of Corollary 3.1: The closed-form odes (3.20a) and (3.20b) are obtained by using explicit formulae (B.14a) and (B.14b) for b and K, respectively. The explicit formulae of m_t and Σ_t (3.21) are obtained by inspecting the Bayes' model (3.1) that yields a Gaussian distribution at any time t when h takes the assumed quadratic form. Direct calculation shows that the solution (3.21) solves the odes (3.20a) and (3.20b). □

B.6 Parametric Case

Proof of Theorem 3.2: The natural gradient ode (3.26) is obtained by applying the chain rule. In its parameterized form, the density $p(x,t) = \varrho\,(x;\theta_t)$ evolves according to the replicator pde:

$$\frac{\partial\ \varrho}{\partial t}(x;\theta_t) = -\beta\,(h(x) - \hat{h}_t)\ \varrho\,(x;\theta_t).$$

Now, using the chain rule,

$$\frac{\partial\ \varrho}{\partial t}(x,\theta_t) = \varrho\,(x,\theta_t)\left[\frac{\partial}{\partial \vartheta}(\log \varrho\,(x;\theta_t))\right]^T \frac{\mathrm{d}\theta_t}{\mathrm{d}t},$$

where $\frac{\partial}{\partial \vartheta}(\log \varrho)$ and $\frac{\mathrm{d}\theta_t}{\mathrm{d}t}$ are both $M \times 1$ column vectors. Therefore, the replicator pde is given by

$$\left[\frac{\partial}{\partial \vartheta}(\log \varrho\,(x;\theta_t))\right]^T \frac{\mathrm{d}\theta_t}{\mathrm{d}t}\ \varrho\,(x;\theta_t) = -\beta(h(x) - \hat{h}_t)\ \varrho\,(x;\theta_t).$$

Multiplying both sides by the column vector $\frac{\partial}{\partial \vartheta}(\log \varrho)$, integrating over the domain, and using the definitions (3.24) of the Fisher information matrix G and (3.25) for ∇e, one obtains

$$G_{(\theta_t)}\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = -\beta \nabla e(\theta_t).$$

The ode (3.26) is obtained because G is assumed invertible. □

Appendix C

Proof for Chapter 4

C.1 Proof of Theorem 4.1

Define the inner product on $H_0^1(G;\pi)$,

$$\langle \phi, \psi \rangle_H := \pi\big(\langle \text{grad}(\phi), \text{grad}(\psi) \rangle\big).$$

On account of the Poincaré inequality (4.3), the norm defined by the inner product $\langle \cdot, \cdot \rangle_H$ is equivalent to the standard norm in $H^1(G;\pi)$.

Consider the Poisson equation (4.1). Using (4.3) and Assumption 4.1,

$$|\pi\big((h-\hat{h})\psi\big)|^2 \le \pi\big(|h-\hat{h}|^2\big)\pi\big(|\psi|^2\big) \le C\,\pi\big(|\text{grad}(\psi)|^2\big),$$

where C is a constant. That is, the integral on the right-hand-side of (2.22) is a bounded linear functional for $\psi \in H_0^1(G;\pi)$.

It then follows from the Riesz representation theorem (Section 2.4 in [23]) that there exists a unique $\phi \in H_0^1(G;\pi)$ such that

$$\langle \phi, \psi \rangle_H = \pi\big((h-\hat{h})\psi\big)$$

for all $\psi \in H_0^1(G;\pi)$. In addition, this formula also holds if any constant is added to ψ, thus it holds for all $\psi \in H^1(G;\pi)$. That is, the Poisson equation (2.22) has a unique solution.

Let ϕ be a weak solution of the Poisson equation (2.22). On taking $\psi = \phi$ and using the Poincaré

inequality, we have,

$$\begin{aligned}\pi\big(|\mathrm{grad}(\phi)|_G^2\big) &= \pi\big((h-\hat{h})\,\phi\big)\\ &\le \Big(\pi\big(|h-\hat{h}|^2\big)\Big)^{\frac{1}{2}}\Big(\pi\big(|\phi|^2\big)\Big)^{\frac{1}{2}}\\ &\le \Big(\pi\big(|h-\hat{h}|^2\big)\Big)^{\frac{1}{2}}\Big(\frac{1}{\bar{\lambda}}\pi\big(|\mathrm{grad}(\phi)|_G^2\big)\Big)^{\frac{1}{2}},\end{aligned}$$

and the estimate (4.4) follows.

By the definition of the gain function $\mathrm{grad}(\phi)(x) = x\mathsf{K}(x)$ (see (2.23)) and the fact that $|\mathsf{K}|_{\mathscr{G}} = |\mathrm{grad}(\phi)|_G$, the bound (4.5) is simply an equivalent form of the estimate (4.4).

The bound (4.6) is obtained as follows: Using the definition (2.25) of the control function u,

$$\begin{aligned}\pi\big(\sum_n|\mathsf{u}_n|\big) &= \frac{1}{2}\,\pi\big(\sum_n|\mathsf{k}_n||h+\hat{h}|\big)\\ &\le \Big(\pi\big(|(h+\hat{h})|^2\big)\pi\big(\big(\sum_n|\mathsf{k}_n|\big)^2\big)\Big)^{\frac{1}{2}}\\ &\le C\Big(\pi\big(|(h+\hat{h})|^2\big)\pi\big(\sum_n|\mathsf{k}_n|^2\big)\Big)^{\frac{1}{2}}\\ &= C\Big(\pi\big(|(h+\hat{h})|^2\big)\pi\big(|\mathsf{K}|_{\mathscr{G}}^2\big)\Big)^{\frac{1}{2}}\\ &\overset{(4.5)}{\le} C\Big(\pi\big(|(h+\hat{h})|^2\big)\pi\big(|(h-\hat{h})|^2\big)\Big)^{\frac{1}{2}}\\ &\le C\,\pi\big(|h|^2\big),\end{aligned}$$

where the Cauchy-Schwarz inequality is used in the second and third inequality, and the definition $|\mathsf{K}|_{\mathscr{G}}^2 = \sum_n|\mathsf{k}_n|^2$ is also used where $(\mathsf{k}_1,\ldots,\mathsf{k}_d)$ are the coordinates of K. The constant C is generic in the proof. □

C.2 Galerkin Approximation Error

Spectral representation: Under Assumption that the Lie group is compact and the density is everywhere positive, the spectrum of $-\Delta_\rho$ is known to be discrete with an ordered sequence of eigenvalues $0 = \lambda_0 < \lambda_1 \le \lambda_2 \le \ldots$ and associated eigenfunctions $\{e_l\}_{l=1}^{\infty}$ that form a complete orthonormal basis of $L^2(G,\pi)$

[154, 29]. As a result, for $k, l \in \mathbb{Z}^+$:

$$\langle e_k, e_l \rangle = \delta_{kl}, \qquad \pi\big(\langle \operatorname{grad}(e_k), \operatorname{grad}(e_l) \rangle\big) = \lambda_k \, \delta_{kl}, \tag{C.1}$$

where the second equation follows from integration by parts and using the first equation. The trivial eigenvalue $\lambda_0 = 0$ with associated eigenfunction $e_0(x) = 1$. On the subspace of zero-mean functions, the spectral representation yields: For $\phi \in L_0^2(\mathbb{R}^d, \rho)$,

$$-\Delta_\rho \phi(x) = \sum_{l=1}^{\infty} \lambda_l \langle e_l, \phi \rangle e_l(x). \tag{C.2}$$

Proof of Proposition 4.1: The property (C.1) of the basis implies that $A = \operatorname{diag}(\lambda_1, ..., \lambda_L)$. Hence, the matrix equation (4.8) has a unique solution.

To prove the error bound, by the triangle inequality,

$$\|\mathsf{K} - \mathsf{K}^{(L,N)}\|_{\mathscr{G}} \le \underbrace{\|\mathsf{K} - \mathsf{K}^{(L)}\|_{\mathscr{G}}}_{\text{bias}} + \|\mathsf{K}^{(L)} - \mathsf{K}^{(L,N)}\|_{\mathscr{G}}.$$

The estimates for the bias and for the error due to the empirical approximation are as below. In the remainder of the proof, denote $\tilde{h} := h - \hat{h}$.

Bias: Using the spectral representation (C.2), because $\tilde{h} \in L_0^2$,

$$\phi = -\Delta_\rho^{-1}(\tilde{h}) = \sum_{l=1}^{\infty} \frac{1}{\lambda_l} \langle e_l, \tilde{h} \rangle e_l.$$

With basis functions as eigenfunctions, $\phi^{(L)} = \sum_{l=1}^{L} \kappa_l e_l$. Substituting this into the weak form (4.7) and letting ψ be the eigenfunctions $\{e_l\}_{l=1}^{L}$, we obtain

$$\phi^{(L)} = \sum_{l=1}^{L} \frac{1}{\lambda_l} \langle e_l, \tilde{h} \rangle e_l.$$

Therefore, using the definition of inner product (2.8),

$$\|\mathsf{K}-\mathsf{K}^{(L)}\|_{\mathscr{G}}^2 = \|\mathrm{grad}(\phi)-\mathrm{grad}(\phi^{(L)})\|_G^2 = \sum_{l=L+1}^{\infty} \frac{1}{\lambda_l^2}|\langle e_l,\tilde{h}\rangle|^2\|\mathrm{grad}(e_l)\|_G^2$$
$$= \sum_{l=L+1}^{\infty} \frac{1}{\lambda_l^2}|\langle e_l,\tilde{h}\rangle|^2\lambda_l \;\le\; \frac{1}{\lambda_L}\|\tilde{h}-\Pi_S\tilde{h}\|_2^2,$$

where $\Pi_S\tilde{h}(x) := \sum_{l=1}^{L}\langle e_l,\tilde{h}\rangle e_l(x)$ denotes the projection of $\tilde{h}$ onto S.

Empirical error: Suppose $\{X^i\}_{i=1}^N$ are drawn i.i.d. from the distribution π. The empirical solution is obtained as:

$$\phi^{(L,N)}(x) = \sum_{l=1}^{L}\frac{1}{\lambda_l}\left(\frac{1}{N}\sum_{i=1}^{N} e_l(X^i)\tilde{h}(X^i)\right)e_l(x),$$

and the error,

$$\phi^{(L)}(x)-\phi^{(L,N)}(x) = \sum_{l=1}^{L}\frac{1}{\lambda_l}z_l^{(N)}e_l(x),$$

where $z_l^{(N)} := \langle e_l,\tilde{h}\rangle - \frac{1}{N}\sum_{i=1}^{N} e_l(X^i)\tilde{h}(X^i)$. Therefore,

$$\|\mathsf{K}^{(L)}-\mathsf{K}^{(L,N)}\|_{\mathscr{G}}^2 = \sum_{l=1}^{L}\frac{1}{\lambda_l}|z_l^{(N)}|^2 =: \varepsilon_N^2, \tag{C.3}$$

where $\pi(\langle\mathrm{grad}(e_k),\mathrm{grad}(e_l)\rangle = \lambda_k\delta_{kl}$ is used to simplify the cross-terms. Finally, by applying the Law of Large Numbers (LLN) for the random variable $z_l^{(N)}$, $\varepsilon_N \overset{\text{a.s.}}{\to} 0$ as $N\to\infty$. The LLN applies because

$$\mathsf{E}[|e_l(X^i)\tilde{h}(X^i)|] \le \|e_l\|_2\,\|\tilde{h}\|_2 = \|\tilde{h}\|_2 < \infty,$$

where $\|\tilde{h}\|_2<\infty$ due to Assumption 4.1. This completes the proof of (4.9).

Variance: Under additional restrictions on h, one can obtain sharper estimates. For example, taking the expectation of both sides of (C.3) with respect to the joint distribution of the i.i.d. samples $\{X^i\}_{i=1}^N$,

$$\mathsf{E}[\|\mathsf{K}^{(L)}-\mathsf{K}^{(L,N)}\|_{\mathscr{G}}^2] = \sum_{l=1}^{L}\frac{\mathsf{E}[|z_l^{(N)}|^2]}{\lambda_l}.$$

Now, $\mathsf{E}[|z_l^{(N)}|^2] = \frac{\mathrm{Var}(e_l(X^i)\tilde{h}(X^i))}{N}$. Therefore, supposing $\tilde{h} \in L^\infty$,

$$\mathsf{E}[\|\mathsf{K}^{(L)} - \mathsf{K}^{(L,N)}\|_{\mathscr{G}}^2] \leq \frac{\|\tilde{h}\|_\infty^2}{N} \sum_{l=1}^{L} \frac{1}{\lambda_l},$$

because $\mathrm{Var}(e_l(X^i)) \leq \pi(e_l^2) = 1$.

In summary, for bounded functions h,

$$\mathsf{E}[\|\mathsf{K} - \mathsf{K}^{(L,N)}\|_{\mathscr{G}}] \leq \underbrace{\frac{\|\tilde{h} - \Pi_S \tilde{h}\|_2}{\sqrt{\lambda_L}}}_{\text{bias}} + \underbrace{\frac{\|\tilde{h}\|_\infty}{\sqrt{N}} \sqrt{\sum_{l=1}^{L} \frac{1}{\lambda_l}}}_{\text{variance}}. \qquad \square$$

C.3 Basis Functions on $SO(3)$

The eigenfunctions of the Laplacian on $SO(3)$ are determined by the matrix elements of the *irreducible unitary representations* of $SO(3)$ (see Sec. 9.3 in [37]). The eigenfunctions associated with the smallest non-zero eigenvalue are tabulated in Table C.1, expressed using both the rotation matrix and the quaternion. In order to compute the matrix A in the Galerkin gain function approximation, the formulae for $E_1 \cdot \psi_l$, $E_2 \cdot \psi_l$ and $E_3 \cdot \psi_l$ are also provided, where $\{E_1, E_2, E_3\}$ denote the basis of $so(3)$ given by (2.13).

A detailed derivation of the basis functions in Table C.1 is provided in [37], [69], and is briefly summarized as follows: The complete set of irreducible unitary representations of $SO(3)$ is given by the matrices $\{D^l\}_{l \in \mathbb{N}}$, where each D^l is a $(2l+1) \times (2l+1)$ matrix whose elements, $D^l_{k,m}$, are called the *Wigner D-functions* of degree l and orders k and m ($-l \leq k, m \leq l$). By the Peter-Weyl theorem, the entire set of Wigner D-functions constitute the orthogonal basis of $L^2(SO(3))$. In addition, for each l, the $(2l+1)^2$ elements of D^l form the complete set of orthogonal basis for the eigenspace of the Laplacian operator on $SO(3)$ associated with the eigenvalue $\lambda_l = l(l+1)$, i.e., $-\Delta D^l_{k,m} = l(l+1) D^l_{k,m}$.

Using the ZXZ Euler angles (α, β, γ) (spin, nutation, and precession angle), the Wigner D-functions of degree l have the explicit expression,

$$D^l_{k,m}(\alpha, \beta, \gamma) = P^l_{k,m}\big(\cos(\beta)\big)\, e^{-i(k\alpha + m\gamma)}, \tag{C.4}$$

Table C.1: Basis functions on $SO(3)$

	expression in R	expression in q	$E_1\cdot$	$E_2\cdot$	$E_3\cdot$
ψ_1	R_{33}	$2(q_0^2+q_3^2)-1$	$2(-q_0q_1-q_2q_3)$	$2(-q_0q_2+q_1q_3)$	0
ψ_2	R_{13}	$2(q_0q_2+q_1q_3)$	$2(q_0q_3-q_1q_2)$	$2(q_0^2+q_1^2)-1$	0
ψ_3	$-R_{23}$	$2(q_0q_1-q_2q_3)$	$2(q_0^2+q_2^2)-1$	$2(-q_0q_3-q_1q_2)$	0
ψ_4	R_{31}	$2(-q_0q_2+q_1q_3)$	0	$-2(q_0^2+q_3^2)+1$	$2(q_0q_1+q_2q_3)$
ψ_5	R_{32}	$2(q_0q_1+q_2q_3)$	$2(q_0^2+q_3^2)-1$	0	$2(q_0q_2-q_1q_3)$
ψ_6	$(1/2)(R_{21}-R_{12})$	$2q_0q_3$	$-q_0q_2-q_1q_3$	$q_0q_1-q_2q_3$	$q_0^2-q_3^2$
ψ_7	$(1/2)(R_{11}+R_{22})$	$q_0^2-q_3^2$	$-q_0q_1+q_2q_3$	$-q_0q_2-q_1q_3$	$-2q_0q_3$
ψ_8	$(1/2)(R_{21}+R_{12})$	$2q_1q_2$	$q_0q_2+q_1q_3$	$q_0q_1-q_2q_3$	$q_2^2-q_1^2$
ψ_9	$(1/2)(R_{11}-R_{22})$	$q_1^2-q_2^2$	$q_0q_1-q_2q_3$	$-q_0q_2-q_1q_3$	$2q_1q_2$

where the polynomials $P^l_{k,m}$ are given by (here $t=\cos(\beta)$),

$$P^l_{k,m}(t) = C\,(1-t)^{-\frac{m-k}{2}}\,(1+t)^{-\frac{m+k}{2}}\,\frac{\mathrm{d}^{l-m}}{\mathrm{d}t^{l-m}}\big((1-t)^{l-k}\,(1+t)^{l+k}\big),$$

with constant

$$C = \frac{(-1)^{l-k}\,i^{m-k}}{2^l\,(l-k)!}\sqrt{\frac{(l-k)!\,(l+m)!}{(l+k)!\,(l-m)!}}.$$

Note that the pair of Wigner D-functions $D^l_{k,m}$ and $D^l_{-k,-m}$ gives rise to the pair of basis functions,

$$P^l_{k,m}\big(\cos(\beta)\big)\sin(k\alpha+m\gamma), \qquad P^l_{k,m}\big(\cos(\beta)\big)\cos(k\alpha+m\gamma).$$

The nine basis functions in Table C.1 are then obtained by letting $l=1$, evaluating the polynomials $P^1_{k,m}$ ($k,m\in\{-1,0,1\}$), and expressing the basis functions using the rotation matrix according to the conversion formula,

$$R(\alpha,\beta,\gamma) = \begin{bmatrix} c_\alpha c_\gamma - s_\alpha c_\beta s_\gamma & -c_\alpha s_\gamma - s_\alpha c_\beta c_\gamma & s_\alpha s_\beta \\ s_\alpha c_\gamma + c_\alpha c_\beta s_\gamma & -s_\alpha s_\gamma + c_\alpha c_\beta c_\gamma & -c_\alpha s_\beta \\ s_\beta s_\gamma & s_\beta c_\gamma & c_\beta \end{bmatrix},$$

where $s_\alpha := \sin(\alpha)$, $c_\alpha := \cos(\alpha)$, and the same notation is used for β and γ.

The Galerkin scheme can also be attempted using a subset of the basis functions in Table C.1 and their combinations. For example, the basis $\{\psi_1, \psi_2, ..., \psi_7\}$ produce the following four basis,

$$\varphi_1 = 2\,q_0\,q_1, \quad \varphi_2 = 2\,q_0\,q_2, \quad \varphi_3 = 2\,q_0\,q_3, \quad \varphi_4 = 2q_0^2-1.$$

Indeed, $\varphi_1 = \frac{1}{2}(\psi_3 + \psi_5)$, $\varphi_2 = \frac{1}{2}(\psi_2 - \psi_4)$, $\varphi_3 = \psi_6$, and $\varphi_4 = \frac{1}{2}\psi_1 + \psi_7 - \frac{1}{2}$. Using the definition of quaternion (2.15), these four basis functions are also written as,

$$\psi_1 = \omega_1 \sin(\theta), \quad \psi_2 = \omega_2 \sin(\theta), \quad \psi_3 = \omega_3 \sin(\theta), \quad \psi_4 = \cos(\theta),$$

which form a natural extension of the basis functions on $SO(2)$ given by (4.10). □

Appendix D

Proof for Chapter 7

D.1 Density Projection for Gaussians

Recall that, given the samples $\{X_n^i\}_{i=1}^N$ at the n-th time step, the parameter update using the density projection entails solving the following optimization problem with respect to the parameter θ,

$$\theta_{n+1} := \arg\max_{\theta\in\Theta} \sum_{i=1}^{N} w_n^i \log q(X_n^i;\theta), \tag{D.1}$$

where

$$w_n^i \propto \frac{p_{n+1}^*(X_n^i)}{q(X_n^i;\theta_n)}, \qquad \sum_{i=1}^{N} w_n^i = 1.$$

When the density $q(x;\theta)$ is chosen as Gaussian, i.e.,

$$q(x;\theta) = \frac{1}{(2\pi)^{d/2}|\Sigma|} \exp\big(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\big),$$

closed-form solution of (D.1) for the updated mean μ_{n+1} and covariance Σ_{n+1} can be obtained as follows:

The objective function in (D.1) is expanded as,

$$\begin{aligned} l(\theta) &:= \sum_{i=1}^{N} w_n^i \log q(X_n^i;\theta) \\ &= -\frac{1}{2}\sum_{i=1}^{N} w_n^i \log(|\Sigma|) - \frac{1}{2}\sum_{i=1}^{N} w_n^i (X_n^i-\mu)^T\Sigma^{-1}(X_n^i-\mu) + (\text{const.}) \end{aligned} \tag{D.2}$$

Since the objective function is quadratic, the optimal solution of μ_n and Σ_n is obtained by solving the first-order condition:

1) The derivative of $l(\theta)$ with respect to μ is given by,

$$\frac{\partial l(\theta)}{\partial \mu} = \sum_{i=1}^{N} w_n^i \Sigma^{-1}(X_n^i - \mu_n).$$

Setting $\frac{\partial l(\theta)}{\partial \mu} = 0$, we obtain

$$\mu_{n+1} = \sum_{i=1}^{N} w_n^i X_n^i.$$

2) We calculate the derivative of $l(\theta)$ with respect to Σ^{-1}. First,

$$\frac{\partial \log(|\Sigma|)}{\partial \Sigma^{-1}} = -\frac{\partial \log(|\Sigma^{-1}|)}{\partial \Sigma^{-1}} = -\Sigma, \tag{D.3}$$

where we used the fact that $|A| = 1/|A^{-1}|$ and $\frac{\partial \log(|A|)}{\partial A} = A^{-T}$ for an invertible matrix A. Secondly,

$$\begin{aligned}
&\frac{\partial \left[(X_n^i - \mu_{n+1})^T \Sigma^{-1} (X_n^i - \mu_{n+1})\right]}{\partial \Sigma^{-1}} \\
=&\frac{\partial \left[\mathrm{Tr}\big((X_n^i - \mu_{n+1})^T \Sigma^{-1} (X_n^i - \mu_{n+1})\big)\right]}{\partial \Sigma^{-1}} \\
=&\frac{\partial \left[\mathrm{Tr}\big(\Sigma^{-1} (X_n^i - \mu_{n+1})(X_n^i - \mu_{n+1})^T\big)\right]}{\partial \Sigma^{-1}} \\
=&(X_n^i - \mu_{n+1})(X_n^i - \mu_{n+1})^T,
\end{aligned} \tag{D.4}$$

where we used the fact that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ and $\frac{\partial(\mathrm{Tr}(AB))}{\partial A} = B^T$ for two matrices A and B. Using the formulas (D.2), (D.3) and (D.4), we obtain,

$$\frac{\partial l(\theta)}{\partial \Sigma^{-1}} = \frac{1}{2}\sum_{i=1}^{N} w_n^i \Sigma - \frac{1}{2}\sum_{i=1}^{N} w_n^i (X_n^i - \mu_{n+1})(X_n^i - \mu_{n+1})^T.$$

Setting $\frac{\partial l(\theta)}{\partial \Sigma^{-1}} = 0$, we obtain,

$$\Sigma_{n+1} = \sum_{i=1}^{N} w_n^i (X_n^i - \mu_{n+1})(X_n^i - \mu_{n+1})^T.$$

The derivation in this section is similar to obtaining the maximum likelihood estimation for Gaussian distributions. The formulae for the relevant matrix derivatives can be found in [142]. □

References

[1] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. Global Optim.*, 31(4):635–672, 2005.

[2] F. Aluffi-Pentini, V. Parisi, and F. Zirilli. Global optimization and stochastic differential equations. *J. Optim. Theory App.*, 47(1):1–16, 1985.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[4] S. N. Atluri and S. Shen. The basis of meshless domain discretization: the meshless local Petrov–Galerkin (MLPG) method. *Adv. Comput. Math.*, 23(1-2):73–93, 2005.

[5] A. Atrsaei, H. Salarieh, and A. Alasty. Human arm motion tracking by orientation-based fusion of inertial sensors and Kinect using unscented Kalman filter. *J. Biomech. Eng.*, 138(9), 2016.

[6] D. Bakry, I. Gentil, and M. Ledoux. *Analysis and Geometry of Markov Diffusion Operators*, volume 348. Springer, 2013.

[7] I. Y. Bar-Itzhack and M. Idan. Recursive attitude determination from vector observations: Euler angle estimation. *J. Guid. Control Dynam.*, 10(2):152–157, 1987.

[8] I. Y. Bar-Itzhack and Y. Oshman. Attitude determination from vector observations: Quaternion estimation. *IEEE Trans. Aerosp. Electron. Syst.*, (1):128–136, 1985.

[9] M. Barczyk, S. Bonnabel, J. Deschaud, and F. Goulette. Invariant EKF design for scan matching-aided localization. *IEEE Trans. Control Syst. Technol.*, 23(6):2440–2448, 2015.

[10] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robot.*, 30(3):679–693, 2014.

[11] A. Barrau. *Nonlinear state error based extneded Kalman filters with applications to navigation*. PhD thesis, Mines ParisTech, 2015.

[12] A. Barrau and S. Bonnabel. Invariant particle filtering with application to localization. In *Proc. 53rd IEEE Conf. Decision Control*, pages 5599–5605, 2014.

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

[14] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, 2003.

[15] V. E. Beneš. Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, 5(1-2):65–92, 1981.

[16] A. Bensoussan, J. Frehse, and S. C. P. Yam. The master equation in mean field theory. *Journal de Mathématiques Pures et Appliquées*, 103(6):1441–1474, 2015.

[17] K. Berntorp. Feedback particle filter: Application and evaluation. In *Proc. 18th Int. Conf. Inform. Fusion*, pages 1633–1640, 2015.

[18] K. Berntorp and P. Grover. Data-driven gain computation in the feedback particle filter. In *Proc. Amer. Control Conf.*, pages 2711–2716, 2016.

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

[20] 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 held jointly with the 28th Chinese Control Conf. (CDC/CCC)*, pages 1297–1304, 2009.

[21] 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.*, 51(1):209–228, 2015.

[22] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu. Discrete extended Kalman filter on Lie groups. In *Proc. 21st Eur. Signal Process. Conf.*, pages 1–5, 2013.

[23] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15. Springer, 2007.

[24] R. Brockett. Optimal control of the Liouville equation. *AMS IP Studies in Advanced Mathematics*, 39:23, 2007.

[25] A. Cavallo, A. Cirillo, P. Cirillo, G. De Maria, P. Falco, C. Natale, and S. Pirozzi. Experimental comparison of sensor fusion algorithms for attitude estimation. *IFAC Proc. Volume.*, 47(3):7585–7591, 2014.

[26] J. Ćesić, I. Marković, I. Cvišić, and I. Petrović. Radar and stereo vision fusion for multitarget tracking on the special Euclidean group. *Robot. and Auton. Syst.*, 83:338–348, 2016.

[27] J. Césić, I. Marković, and I. Petrović. Moving object tracking employing rigid body motion on matrix Lie groups. In *Proc. 19th Int. Conf. Inform. Fusion (FUSION)*, pages 2109–2115, 2016.

[28] N. Charalambous and Z. Lu. Heat kernel estimates and the essential spectrum on weighted manifolds. *J. Geom. Anal.*, 25(1):536–563, 2015.

[29] N. Charalambous, Z. Lu, and J. Rowlett. Eigenvalue estimates on Bakry–émery manifolds. In *Elliptic and Parabolic Equations*, pages 45–61. Springer, 2015.

[30] Y. Chen, T. Georgiou, and M. Pavon. On the relation between optimal transport and Schródinger bridges: A stochastic control viewpoint. *J. Optimiz. Theory App.*, 169(2):671–691, 2016.

[31] Y. Chen, T. Georgiou, and M. Pavon. Optimal steering of a linear stochastic system to a final probability distribution, part I. *IEEE Trans. Autom. Control*, 61(5):1158–1169, 2016.

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

[33] T-S. Chiang, C-R. Hwang, and S-J. Sheu. Diffusion for global optimization. *SIAM J. Control Optim.*, 25(3):737–753, 1987.

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

[35] G. S. Chirikjian and M. Kobilarov. Gaussian approximation of non-linear measurement models on Lie groups. In *Proc. 53rd IEEE Conf. Decision Control*, pages 6401–6406, 2014.

[36] G. S. Chirikjian and A. B. Kyatkin. *Engineering Applications of Noncommutative Harmonic Analysis: with Emphasis on Rotation and Motion Groups*. CRC press, 2000.

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

[38] A. Chiuso and S. Soatto. Monte Carlo filtering on Lie groups. In *Proc. 39th IEEE Conf. Decision Control*, pages 304–309, 2000.

[39] 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.*, 31(4):498–519, 2012.

[40] R. R. Coifman and S. Lafon. Diffusion maps. *Appl. and Comput. Harmon. Anal.*, 21(1):5–30, 2006.

[41] J. P. Condomines, C. Seren, and G. Hattenberger. Nonlinear state estimation using an invariant unscented Kalman filter. In *AIAA Guid. Nav. Control Conf.*, pages 1–15, 2013.

[42] J. L. Crassidis and F. L. Markley. Unscented filtering for spacecraft attitude estimation. *J. Guid. Control Dynam.*, 26(4):536–542, 2003.

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

[44] F. Daum, J. Huang, and A. Noushin. Exact particle flow for nonlinear filters. In *SPIE Defense, Security, and Sensing*, pages 769704–769704, 2010.

[45] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst., Man, Cybern., Part B (Cybern.)*, 26(1):29–41, 1996.

[46] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[47] A. M. Doucet, A.and Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.

[48] Y. Duan. Meshless Galerkin method using radial basis functions based on domain decomposition. *Appl. Math. Comput.*, 179(2):750–762, 2006.

[49] M. Duflot and H. Nguyen-Dang. A truly meshless Galerkin method based on a moving least squares quadrature. *Int. J. Numer. Method. Biomed. Eng.*, 18(6):441–449, 2002.

[50] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*, volume 36. Springer Science & Business Media, 2013.

[51] T. E. Duncan. Some filtering results in Riemann manifolds. *Inform. Control*, 35(3):182–195, 1977.

[52] R. Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2010.

[53] The Economist. The quantified self: Counting every moment, March 2012.

[54] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

[55] N. Enayati, E. De Momi, and G. Ferrigno. A quaternion-based unscented Kalman filter for robust optical/inertial motion tracking in computer-assisted surgery. *IEEE Trans. Instrum. Meas.*, 64(8):2291–2301, 2015.

[56] G Evensen. *Data Assimilation: The Ensemble Kalman Filter*. Springer, 2009.

[57] J. Faraut. *Analysis on Lie Groups: An Introduction*. Cambridge University Press, 2008.

[58] J. R. Forbes, A. H.J. de Ruiter, and D. E. Zlotnik. Continuous-time norm-constrained Kalman filtering. *Automatica*, 50(10):2546–2554, 2014.

[59] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H-P. Seidel. Interacting and annealing particle filters: Mathematics and a recipe for applications. *J. Math. Imaging Vis.*, 28(1):1–18, 2007.

[60] C. W. Gardiner. *Handbook of Stochastic Methods*. Springer, 1985.

[61] S. Germano. Under Armour overtakes Adidas in U.S. sportswear market. *The Wall Street Journal*, January 2015.

[62] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, 1989.

[63] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proc. F (Radar Signal Process.)*, volume 140, pages 107–113, 1993.

[64] A. Grigor'yan. *Heat Kernel and Analysis on Manifolds*, volume 47. American Mathematical Society, 2009.

[65] A. Grigoryan. Heat kernels on weighted manifolds and applications. *Cont. Math*, 398:93–191, 2006.

[66] H. F. Grip, T. I. Fossen, T. A. Johansen, and Ali. Saberi. Attitude estimation using biased gyro and vector measurements with time-varying reference vectors. *IEEE Trans. Autom. Control*, 57(5):1332–1338, 2012.

[67] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Trans. Signal process.*, 50(2):425–437, 2002.

[68] B. C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, volume 222. Springer, 2015.

[69] T. Hangelbroek and D. Schmid. Surface spline approximation on SO(3). *Appl. and Comput. Harmon. Anal.*, 31(2):169–184, 2011.

[70] S. Hauberg, F. Lauze, and K. S. Pedersen. Unscented Kalman filtering on Riemannian manifolds. *J. Math. Imaging Vis.*, 46(1):103–120, 2013.

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

[72] 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.*, 33(1):182–201, 2013.

[73] S. Ho, S. Yang, H. C. Wong, and G. Ni. A meshless collocation method based on radial basis functions and wavelets. *IEEE Trans. Magn.*, 40(2):1021–1024, 2004.

[74] E. P. Hsu. *Stochastic Analysis on Manifolds*, volume 38. American Mathematical Society, 2002.

[75] J. Hu. Model-based stochastic search methods. In *Handbook of Simulation Optimization*, pages 319–340. Springer, 2015.

[76] J. Hu, M. C. Fu, and S. I. Marcus. A model reference adaptive search method for global optimization. *Oper. Res.*, 55(3):549–568, 2007.

[77] J. Hu and P. Hu. Annealing adaptive search, cross-entropy, and stochastic approximation in global optimization. *Nav. Res. Log.*, 58(5):457–477, 2011.

[78] J. Hu and P. Hu. A stochastic approximation framework for a class of randomized optimization algorithms. *IEEE Trans. Autom. Control*, 57(1):165–178, 2012.

[79] J. Hu, Y. Wang, E. Zhou, M. C. Fu, and S. I. Marcus. A survey of some model-based methods for global optimization. In H. Daniel and M. Adolfo, editors, *Optimization, Control, and Applications of Stochastic Systems: In Honor of Onésimo Hernández-Lerma*, Systems & Control: Foundations & Applications, pages 157–179. Birkhäuser, 2012.

[80] 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.*, 22(1):201–213, 2014.

[81] M. Huang, P. E. Caines, and R. P. Malhamé. Large-population cost-coupled LQG problems with nonuniform agents: Individual-mass behavior and decentralized ε-Nash equilibria. *IEEE Trans. Autom. Control*, 52(9):1560–1571, 2007.

[82] D. Q. Huynh. Metrics for 3D rotations: Comparison and analysis. *J. Math. Imaging Vis.*, 35(2):155–164, 2009.

[83] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE J. Sel. Areas in Commun.*, 23(4):809–819, 2005.

[84] M. Isard and A. Blake. CONDENSATION-conditional density propagation for visual tracking. *Int. J. Comput. Vision*, 29(1):5–28, 1998.

[85] M. Izadi. *Stable estimation of rigid body motion using geometric mechanics*. PhD thesis, New Mexico State University, 2015.

[86] M. Izadi, E. Samiei, A. K. Sanyal, and V. Kumar. Comparison of an attitude estimator based on the Lagrange-d'Alembert principle with some state-of-the-art filters. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2848–2853, 2015.

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

[88] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Dover Publications, 1970.

[89] C. Ji, Y. Zhang, M. Tong, and S. Yang. Particle filter with swarm move for optimization. In *Int. Conf. Parallel Problem Solving from Nature*, pages 909–918, 2008.

[90] S. Kakade. A natural policy gradient. In *NIPS*, volume 14, pages 1531–1538, 2001.

[91] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Network.*, pages 1942–1948, 1995.

[92] J. Kim, S. Yang, and M. Gerla. StrokeTrack: wireless inertial motion tracking of human arms for stroke telerehabilitation. In *Proc. First ACM Workshop on Mobile Systems, Applications, and Services for Healthcare*, page 4, 2011.

[93] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

[94] P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 2010.

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

[96] 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.*, 36(4):625–643, 2014.

[97] O. Lablée. *Spectral Theory in Riemannian Geometry*. European Mathematical Society, 2015.

[98] C. Lageman, J. Trumpf, and R. Mahony. Gradient-like observers for invariant dynamics on a Lie group. *IEEE Trans. Autom. Control*, 55(2):367–377, 2010.

[99] P. Larranaga and J. A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer, 2002.

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

[101] Q. T. Le Gia. Galerkin approximation for elliptic PDEs on spheres. *J. Approx. Theory*, 130(2):125–149, 2004.

[102] Q. T. Le Gia. Numerical solutions of a boundary value problem on the sphere using radial basis functions. *arXiv preprint:1402.3353*, 2014.

[103] J. K. Lee, E. J. Park, and S. N. Robinovitch. Estimation of attitude and external acceleration using inertial sensor measurement during various dynamic conditions. *IEEE Trans. Instrum. Meas.*, 61(8):2262–2273, 2012.

[104] J. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2012.

[105] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman filtering for spacecraft attitude estimation. *J. Guid. Control Dynam.*, 5(5):417–429, 1982.

[106] R. C. Leishman, J. C. Macdonald, R. W. Beard, and T. W. McLain. Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Syst. Mag.*, 34(1):28–41, 2014.

[107] S. Leung, J. Lowengrub, and H. Zhao. A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion. *J. Comput. Phys*, 230(7):2540–2561, 2011.

[108] D. Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998.

[109] S. Li and W. K. Liu. *Meshfree Particle Methods*. Springer, 2007.

[110] J. Liang and H. Zhao. Solving partial differential equations on point clouds. *SIAM J. Sci. Comput.*, 35(3):A1461–A1486, 2013.

[111] K. Lightman. Next-gen sensors make golf clubs, tennis rackets, and baseball bats smarter than ever. *IEEE Spectr.*, March 2016.

[112] G. Ligorio and A. M. Sabatini. A novel Kalman filter for human motion tracking with an inertial-based dynamic inclinometer. *IEEE Trans. Biomed. Eng.*, 62(8):2033–2043, 2015.

[113] B. Liu. Posterior exploration based sequential Monte Carlo for global optimization. *arXiv preprint:1509.08870*, 2016.

[114] B. Liu, S. Cheng, and Y. Shi. Particle filter optimization: A brief introduction. In *Proc. 7th Int. Conf. Swarm Intell.*, pages 95–104, 2016.

[115] W. K. Liu, S. Li, and T. Belytschko. Moving least-square reproducing kernel methods (I) Methodology and convergence. *Comput. method appl. M.*, 143(1-2):113–154, 1997.

[116] Y. M. Lui. Advances in matrix manifolds for computer vision. *Image Vision Comput.*, 30(6):380–388, 2012.

[117] N. Madinehi. *Rigid body attitude estimation: An overview and comparative study*. PhD thesis, Western University, 2013.

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

[119] J. H. Manton. A primer on stochastic differential geometry for signal processing. *IEEE J. Sel. Topics Signal Process.*, 7(4):681–699, 2013.

[120] 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*, pages 7143–7150, 2015.

[121] G. Marjanovic, M. J. Piggott, and V. Solo. Numerical methods for stochastic differential equations in the Stiefel manifold made simple. In *Proc. 55th IEEE Conf. Decision Control*, pages 2853–2860, 2016.

[122] 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.*, pages 3969–3973, 2016.

[123] F L. Markley. Fast quaternion attitude estimation from two vector measurements. *J. Guid. Control Dynam.*, 25(2):411–414, 2002.

[124] F. L. Markley. Attitude error representations for Kalman filtering. *J. Guid. Control Dynam.*, 26(2):311–317, 2003.

[125] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman. Averaging quaternions. *J. Guid. Control Dynam.*, 30(4):1193–1197, 2007.

[126] I. Marković, J. Ćesić, and I. Petrović. Von Mises mixture PHD filter. *IEEE Signal Process. Lett.*, 22(12):2229–2233, 2015.

[127] I. Marković, J. Ćesić, and I. Petrović. On wrapping the Kalman filter and estimating with the SO(2) group. In *Proc. 19th Int. Conf. Inform. Fusion (FUSION)*, pages 2245–2250, 2016.

[128] S. Martin, R. Pinnau, C. Totzeck, and O. Tse. A consensus-based model for global optimization and its mean-field limit. *arXiv preprint:1604.05648*, 2016.

[129] 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.*, 2016.

[130] J. Míguez, D. Crisan, and P. M. Djurić. On the convergence of two sequential Monte Carlo methods for maximum a posteriori sequence estimation and stochastic global optimization. *Stat. Comput.*, 23(1):91–107, 2013.

[131] S. K. Mitter and N. J. Newton. A variational approach to nonlinear estimation. *SIAM J. Control Optim.*, 42(5):1813–1833, 2003.

[132] M. Moakher. Means and averaging in the group of rotations. *SIAM J. Matrix Anal. A.*, 24(1):1–16, 2002.

[133] O. Molvalioglu, Z. B. Zabinsky, and W. Kohn. Meta-control of an interacting-particle algorithm for global optimization. *Nonlinear Ana. Hybrid Syst.*, 4(4):659–671, 2010.

[134] C. K. Monson and K. D. Seppi. The Kalman swarm. In *Proc. Genetic Evol. Comput. Conf.*, pages 140–150, 2004.

[135] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proc. AAAI National Conf. Artif. Intell.*, pages 593–598, 2002.

[136] P. D. Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

[137] J. Musić, R. Kamnik, and M. Munih. Model based inertial sensing of human body motion kinematics in sit-to-stand movement. *Simul. Model. Pract. Theory*, 16(8):933–944, 2008.

[138] F. Narcowich, S. Rowe, and J. Ward. A novel Galerkin method for solving PDEs on the sphere using highly localized kernel bases. *Math. Comput.*, 86(303):197–231, 2017.

[139] S. K. Ng and P. E. Caines. Nonlinear filtering in Riemannian manifolds. *IMA J. Math. Control Inform.*, 2(1):25–36, 1985.

[140] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, 2007.

[141] Y. Oshman and A. Carmi. Attitude estimation from vector observations using a genetic-algorithm-embedded quaternion particle filter. *J. Guid. Control Dynam.*, 29(4):879–891, 2006.

[142] K. B. Petersen and M. S. Pedersen. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

[143] M. J. Piggott and V. Solo. Stochastic numerical analysis for brownian motion on SO(3). In *Proc. 53rd IEEE Conf. Decision Control*, pages 3420–3425, 2014.

[144] M. J. Piggott and V. Solo. Geometric Euler-Maruyama schemes for stochastic differential equations in SO(n) and SE(n). *SIAM J. Numer. Anal.*, 54(4):2490–2516, 2016.

[145] M. E. Pittelkau. Rotation vector in attitude estimation. *J. Guid. Control Dynam.*, 26(6):855–860, 2003.

[146] M. Pontier and J. Szpirglas. Filtering on manifolds. In *Stochastic Modelling and Filtering*, pages 147–160. Springer, 1987.

[147] A. Radhakrishnan, A. M. Devraj, and S. P. Meyn. Learning techniques for feedback particle filter design. In *Proc. 55th IEEE Conf. Decision Control*, pages 5452–5459, 2016.

[148] S. Reich. A dynamical systems framework for intermittent data assimilation. *BIT Numerical Mathematics*, 51(1):235–249, 2011.

[149] H. E. Romeijn and R. L. Smith. Simulated annealing for constrained global optimization. *J. Global Optim.*, 5(2):101–126, 1994.

[150] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999.

[151] A. M. Sabatini. Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing. *Sensors*, 11(2):1489–1525, 2011.

[152] A. Saccon, J. Trumpf, R. Mahony, and A. P. Aguiar. Second-order-optimal filters on Lie groups. In *Proc. 52nd IEEE Conf. Decision Control*, pages 4434–4441, 2013.

[153] S. Said and J. H. Manton. On filtering with observation in a manifold: Reduction to a classical filtering problem. *SIAM J. Control Optim.*, 51(1):767–783, 2013.

[154] A. G. Setti. Eigenvalue estimates for the weighted Laplacian on a Riemannian manifold. *Rend. Sem. Mat. Univ. Padova*, 100:27–55, 1998.

[155] I. Sharf, A. Wolf, and M. B. Rubin. Arithmetic and geometric solutions for average rigid-body rotation. *Mech. Mach. Theory*, 45(9):1239–1251, 2010.

[156] J. Shen, T. Tang, and L. Wang. *Spectral Methods: Algorithms, Analysis and Applications*. Springer, 2011.

[157] S. Shen, H. Wang, and R. R. Choudhury. I am a smartwatch and I can track my users arm. In *Proc. 14th Annual Int. Conf. Mobile Syst. Appl. Serv. (MobiSys)*, 2016.

[158] L. Shi and S. Ólafsson. Nested partitions method for global optimization. *Oper. Res.*, 48(3):390–407, 2000.

[159] M. D. Shuster. Uniform attitude probability distributions. *J. Astronaut. Sci.*, 51(4):451–475, 2003.

[160] H. Snoussi. Particle filtering on riemannian manifolds. application to covariance matrices tracking. In *Matrix Information Geometry*, pages 427–449. Springer, 2013.

[161] V. Solo. On nonlinear state estimation in a Riemannian manifold. In *Proc. 48th IEEE Conf. Decision Control*, pages 8500–8505, 2009.

[162] A. Srivastava. A Bayesian approach to geometric subspace estimation. *IEEE Trans. Signal Process.*, 48(5):1390–1400, 2000.

[163] A. Srivastava and E. Klassen. Monte Carlo extrinsic estimators of manifold-valued parameters. *IEEE Trans. Signal Process.*, 50(2):299–308, 2002.

[164] 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.*, 24:67–78, 2014.

[165] P. Stinis. Stochastic global optimization as a filtering problem. *J. Comput. Phys.*, 231(4):2002–2014, 2012.

[166] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11(4):341–359, 1997.

[167] Y. S. Suh. Orientation estimation using a quaternion-based indirect Kalman filter with adaptive estimation of external acceleration. *IEEE Trans. Instrum. and Meas.*, 59(12):3296–3305, 2010.

[168] S. C. Surace, A. Kutschireiter, and J-P. Pfister. How to avoid the curse of dimensionality: scalability of particle filters with and without importance weights. arXiv preprint: 1703.07879.

[169] A. Taghvaei and P. G. Mehta. Gain function approximation in the feedback particle filter. In *IEEE 55th Conf. Decision Control*, pages 5446–5452, 2016.

[170] A. Taghvaei, P. G. Mehta, and S. P. Meyn. Error estimates for the kernel gain function approximation in the feedback particle filter. To appear in Amer. Control Conf., 2017. arXiv preprint: 1612.05606.

[171] Y. Tao and H. Hu. A novel sensing and data fusion system for 3-D arm motion tracking in telerehabilitation. *IEEE Trans Instrum. Meas.*, 57(5):1029–1040, 2008.

[172] Y. Tao, H. Hu, and H. Zhou. Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. *Int. J. Robot. Res.*, 26(6):607–624, 2007.

[173] Y. Tian, X. Meng, D. Tao, D. Liu, and C. Feng. Upper limb motion tracking with the integration of IMU and Kinect. *Neurocomputing*, 159:207–218, 2015.

[174] H. Tidefelt and T. B. Schön. Robust point-mass filters on manifolds. *IFAC Proc. Volume*, 42(10):540–545, 2009.

[175] A. K. Tilton, S. Ghiotto, and P. G. Mehta. A comparative study of nonlinear filtering techniques. In *Proc. 16th IEEE Int. Conf. Inform. Fusion*, pages 1827–1834, July 2013.

[176] A. K. Tilton, E. T. Hsiao-Wecksler, and P. G. Mehta. Filtering with rhythms: Application to estimation of gait cycle. In *Proc. Amer. Control Conf.*, pages 3433–3438, 2012.

[177] D. Ting, L. Huang, and M. Jordan. An analysis of the convergence of graph Laplacians. *arXiv preprint:1101.5435*, 2011.

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

[179] J. Trumpf, R. Mahony, T. Hamel, and C. Lageman. Analysis of non-linear attitude observers for time-varying reference measurements. *IEEE Trans. Autom. Control*, 57(11):2789–2800, 2012.

[180] A. Van Rhijn, R. Van Liere, and J. D. Mulder. An analysis of orientation prediction and filtering methods for VR/AR. In *IEEE Proc. Virtual Reality*, pages 67–74, 2005.

[181] 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.*, 59(3):155–166, 2010.

[182] V. Verma, G. Gordon, R. Simmons, and S. Thrun. Real-time fault diagnosis. *IEEE Robot. Autom. Mag.*, 11(2):56–66, 2004.

[183] C. Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2008.

[184] U. Von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Ann. Stat.*, pages 555–586, 2008.

[185] Y. Wang, S. Chen, and C. Wu. A meshless collocation method based on the differential reproducing kernel interpolation. *Comput. Mech.*, 45(6):585–606, 2010.

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

[187] Y. Wang, M. C. Fu, and S. I. Marcus. Model-based evolutionary optimization. In *Proc. Winter Simulation Conf.*, pages 1199–1210, December 2010.

[188] S. Watanabe and N. Ikeda. *Stochastic Differential Equations and Diffusion Processes*. Elsevier, 1981.

[189] G. F. Welch. History: The use of the kalman filter for human motion tracking in virtual reality. *Presence: Teleoperators and Virtual Environments*, 18(1):72–91, 2009.

[190] H. Wendland. Meshless Galerkin methods using radial basis functions. *Math. of Comput.*, 68(228):1521–1531.

[191] K. Wolfe, M. Mashner, and G. S. Chirikjian. Bayesian fusion on Lie groups. *J. Algebr. Stat.*, 2(1):75–97, 2011.

[192] J. Xiong. *An Introduction to Stochastic Filtering Theory*. Oxford University Press, 2008.

[193] T. Yang. *Feedback particle filter and its applications*. PhD thesis, University of Illinois at Urbana-Champaign, 2014.

[194] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn. Multivariable feedback particle filter. *Automatica*, 71(9):10–23, 2016.

[195] T. Yang, P. G. Mehta, and S. P. Meyn. Feedback particle filter. *IEEE Trans. Autom. Control*, 58(10):2465–2480, 2013.

[196] X-S. Yang. *Nature-inspired Metaheuristic Algorithms*. Luniver press, 2010.

[197] H. Yin, P. G. Mehta, S. P. Meyn, and U. V. Shanbhag. Synchronization of coupled oscillators is a game. *IEEE Trans. Autom. Control*, 57(4):920–935, 2012.

[198] X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. *IEEE Trans. Robot.*, 22(6):1216–1227, 2006.

[199] Z. B. Zabinsky. *Stochastic Adaptive Search for Global Optimization*, volume 72. Springer, 2013.

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

[201] M. Zamani, J. Trumpf, and R. Mahony. Minimum-energy filtering on the unit circle. In *Australian Control Conf. (AUCC)*, pages 236–241, 2011.

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

[203] R. Zanetti, M. Majji, R. H. Bishop, and D. Mortari. Norm-constrained Kalman filtering. *J. Guid. Control Dynam.*, 32(5):1458–1465, 2009.

[204] C. Zhang. A particle system for global optimization. In *Proc. 52nd IEEE Conf. Decision Control*, pages 1714–1719, 2013.

[205] C. Zhang and P. G. Mehta. A comparative study of cooperative localization techniques in sensor networks. In *Proc. Amer. Control Conf.*, 2015.

[206] C. Zhang, A. Taghvaei, and P. G. Mehta. Attitude estimation of a wearable motion sensor. To appear in Amer. Control Conf., 2017.

[207] C. Zhang, A. Taghvaei, and P. G. Mehta. A controlled particle filter for global optimization. arXiv preprint: 1701.02413.

[208] C. Zhang, A. Taghvaei, and P. G. Mehta. Feedback particle filter on matrix Lie groups. arXiv preprint: 1701.02416.

[209] C. Zhang, A. Taghvaei, and P. G. Mehta. Attitude estimation with feedback particle filter. In *IEEE 55th Conf. Decision Control*, pages 5440–5445, 2016.

[210] C. Zhang, A. Taghvaei, and P. G. Mehta. Feedback particle filter on matrix Lie groups. In *Proc. Amer. Control Conf.*, pages 2723–2728, 2016.

[211] X. Zhang, X. Liu, K. Song, and M. Lu. Least-squares collocation meshless method. *Int. J. Numer. Method. Eng.*, 51(9):1089–1100, 2001.

[212] E. Zhou. *Particle Filtering for Stochastic Control and Global Optimization*. PhD thesis, University of Maryland at College Park, 2009.

[213] E. Zhou and X. Chen. Sequential Monte Carlo simulated annealing. *J. Global Optim.*, 55(1):101–124, 2013.

[214] E. Zhou, M. C. Fu, and S. I. Marcus. Solving continuous-state POMDPs via density projection. *IEEE Trans. Autom. Control*, 55(5):1101–1116, 2010.

[215] E. Zhou, M. C. Fu, and S. I. Marcus. Particle filtering framework for a class of randomized optimization algorithms. *IEEE Trans. Autom. Control*, 59(4):1025–1030, 2014.

[216] H. Zhou and H. Hu. Human motion tracking for rehabilitation: A survey. *Biomed. Signal Process. Control*, 3(1):1–18, 2008.

[217] M. Zlochin, N. Birattari, and M. Meuleau, M. Dorigo. Model-based search for combinatorial optimization: A critical survey. *Ann. Oper. Res.*, 131(1-4):373–395, 2004.