Automatica 151 (2023) 110894

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

An optimal control approach to particle filtering[☆]

Qinsheng Zhang[a], Amirhossein Taghvaei[b], Yongxin Chen[c,*]

[a] Machine Learning Center, Georgia Institute of Technology, Atlanta, GA, USA
[b] Department of Aeronautics and Astronautics, University of Washington, Seattle, WA, USA
[c] School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ARTICLE INFO

Article history:
Received 2 October 2021
Received in revised form 19 October 2022
Accepted 29 December 2022
Available online 17 February 2023

Keywords:
Particle filtering
Optimal control
Path integral
Nonlinear filtering
Stochastic control

ABSTRACT

We present a novel particle filtering framework for the continuous-time dynamical systems with continuous-time measurements. Our approach is based on the duality between estimation and optimal control, which allows for reformulating the estimation problem over a fixed time window into an optimal control problem. The resulting optimal control problem has a cost function that depends on the measurements, and the closed-loop dynamics under optimal control coincides with the posterior distribution over the trajectories for the corresponding estimation problem. By recursively solving these optimal control problems approximately as new measurements become available, we obtain an optimal control based particle filtering algorithm. Our algorithm uses path integrals to compute the weights of the particles and is thus termed the path integrals particle filter (PIPF). A distinguishing feature of the proposed method is that it uses the measurements over a finite-length time window instead of a single measurement for the estimation at each time step, resembling the batch methods of filtering, and improving fault tolerance. The efficacy of our algorithm is illustrated with several numerical examples.

1. Introduction

In systems and control, filtering refers to estimating the true state of a dynamical system using raw sensor measurements. It is a critical component in feedback control and plays an indispensable role in almost all applications related to control. Many theories and algorithms for filtering have been developed. For instance, the celebrated Kalman filter is for linear dynamics driven by Gaussian noise. It is optimal in the sense of mean-squared error. It also computes the exact posterior distribution of the state given the available measurements. For nonlinear systems, the filtering problem is much more challenging; the posterior distribution of the state rarely has a simple parametrization. To attain the posterior distribution, one needs to solve a stochastic partial differential equation known as the Kushner–Stratonovich equation. The methods relying on discretizing the state space and the Kushner–Stratonovich equation are computationally infeasible for high dimensional problems. There are several algorithms that approximate the posterior distributions with Gaussian distributions, including the extended Kalman filter (EKF) and the unscented Kalman filter (UKF). However, the performance of this type of methods deteriorates as the posterior distribution drifts away from the Gaussian family.

One idea that avoids brute force discretization of the state space, while still retaining the richness of the posterior distributions, is to represent the distributions with an ensemble of particles. This type of methods are known as particle filtering. Over the last decades, many different versions of particle filtering algorithms have been proposed (Del Moral, 1997; Guarniero, Johansen, & Lee, 2017; Pitt & Shephard, 1999; Ruiz & Kappen, 2017; Särkkä & Sottinen, 2008; Taghvaei, Mehta, & Meyn, 2020). In the standard setup of particle filtering, the posterior distribution at the current step is approximated by K weighted particles. These particles are propagated forward following a proposal density and then combined with the next measurement to estimate the posterior distribution at the next time step. The implementation of particle filtering is extremely easy if the proposal density is simple, which makes particle filtering a popular method for nonlinear filtering. Theoretically, it can be shown that as the number of particles K goes to infinity, the empirical distribution of the particles converges to the true posterior distribution at each time step in some suitable sense (Del Moral, 1997). In practice, however, due to the potentially large difference between proposal distributions and posterior distributions, the weights of

[☆] This work was supported by the NSF under grant 1942523 and 2008513. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Mattia Zorzi under the direction of Editor Alessandro Chiuso.

[*] Corresponding author.

E-mail addresses: qzhang419@gatech.edu (Q. Zhang), amirtag@uw.edu (A. Taghvaei), yongchen@gatech.edu (Y. Chen).

https://doi.org/10.1016/j.automatica.2023.110894
0005-1098/© 2023 Elsevier Ltd. All rights reserved.

the particles degenerate quickly (Doucet, Briers, & Sénécal, 2006), that is, the weights of most of the particles become negligible and the mass of the particles concentrates only on a few particles, rendering a small effective particle size. A resampling step is adopted to mitigate the effects of degenerate weights. However, both in theory and in practice, the choice of a proper proposal density is critical and most particle filtering algorithms still perform poorly in high-dimensional problems (Bengtsson, Bickel, & Li, 2008; Beskos, Crisan, Jasra, & Whiteley, 2014), largely due to the particle degeneracy.

In this work, we consider the nonlinear filtering problem for the continuous-time diffusion dynamics with continuous-time measurements. We present a new particle filtering method based on a duality between estimation and optimal control (Kim & Mehta, 2020; Mitter & Newton, 2003). Building on this duality, we are able to obtain a superior proposal density by (approximately) solving an optimal control problem and thus establishing a particle filtering algorithm with better performance. Moreover, this duality makes it natural to resample the particles from the past; this is different from most particles filtering algorithms that only sample in the present. This extra flexibility of updating samples in the past provides us the opportunity to correct numerical errors or errors induced by outlier in the previous filtering steps and makes the algorithm more robust to mistakes and outlier measurements. Empirically, we also observe that extending the sampling to the past, with proper proposals, can significantly mitigate the particle degeneracy issue.

The proposed algorithm is mostly related to those proposed in Ruiz and Kappen (2017), Doucet et al. (2006), and Balaji (2009). In Doucet et al. (2006) a block sampling strategy is proposed to resample particles in the past as in our algorithm. However, they focus on an abstract framework for general discrete-time systems. How to leverage the structure of the underlying dynamics to construct a proper proposal distribution is not studied. In Ruiz and Kappen (2017), an optimal control approach to smoothing is proposed. However, they consider the smoothing problem over a fixed-time window. Moreover, though the dynamics they use is continuous-time diffusion, their measurement model is discrete-time. The same setting with continuous-time diffusion and discrete-time measurement is used in Balaji (2009). Even though the path integral idea is used, the algorithm in Balaji (2009) is grid-based, not particle based. There are also some other particle filtering algorithms such as feedback particle filtering (Taghvaei et al., 2020; Yang, Mehta, & Meyn, 2013), particle flow filter (Daum, Huang, & Noushin, 2010), and learning based methods (Le, Igl, Rainforth, Jin, & Wood, 2018; Naesseth, Linderman, Ranganath, & Blei, 2018) that aim to improve the performance by using a better proposal.

The rest of the paper is structured as follows. In Section 2, we provide a brief introduction to a stochastic optimal control problem and present a duality relationship between filtering and optimal control. We then use this optimal control formulation of filtering to derive our particle filtering algorithm in Section 3. The algorithm is illustrated in Section 4 through several numerical examples. This is followed by concluding remarks in Section 5.

2. Background

In this section we provide the background on the optimal control and estimation that is necessary to present our proposed method.

2.1. Stochastic optimal control

Consider the stochastic dynamics described by the stochastic differential equation (SDE)[1] (Särkkä & Solin, 2019)

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)(u_t dt + dW_t) \tag{1}$$

where $X_t \in \mathbb{R}^n, u_t \in \mathbb{R}^m$ denotes the state and control input respectively, and $W_t \in \mathbb{R}^m$ represents a standard Wiener process. The drift $b(\cdot, \cdot)$ and the input channel matrix $\sigma(\cdot, \cdot)$ are assumed to be Lipschitz continuous and bounded.

In the finite horizon stochastic optimal control problem, Lewis, Vrabie, and Syrmos (2012) one seeks an optimal feedback control strategy that minimizes the cost function

$$J(u) = \mathbb{E}\left\{\int_0^T l(t, X_t, u_t)dt + \Psi_T(X_T)\right\} \tag{2}$$

over a time interval $[0, T]$. Here, l and Ψ_T represent the running cost and terminal cost respectively. This problem can be solved via dynamic programming (Bertsekas, 1995; Evans, 1998), which boils down to solving the Hamilton–Jacobi–Bellman (HJB) equation (Evans, 1998)

$$\frac{\partial V_t}{\partial t} + \min_{u\in\mathbb{R}^m}\{\mathcal{L}_t^u V_t + l(t, x, u)\} = 0, \quad V_T(\cdot) = \Psi_T(\cdot), \tag{3}$$

where $\mathcal{L}_t^u$ denotes the generator of the controlled process (1) defined as (Fleming & Rishel, 1975)

$$\mathcal{L}_t^u f(x) = (b(t, x) + \sigma(t, x)u) \cdot \nabla f(x) + \frac{1}{2}\operatorname{Tr}(\sigma\sigma'\nabla^2 f(x))$$

for any sufficiently smooth $f(\cdot)$. Here $(\cdot)'$ denotes transpose and ∇ is with respect to x. The space–time function $V_t(x)$ is known as the cost-to-go function (Fleming & Rishel, 1975), capturing the minimum cost J but over the time window $[t, T]$ conditioned on $X_t = x$. The optimal control strategy is of state feedback form $u_t^\star = u^\star(t, X_t)$ with

$$u^\star(t, x) = \operatorname*{argmin}_{u\in\mathbb{R}^m}\{\mathcal{L}_t^u V_t(x) + l(t, x, u)\}. \tag{4}$$

As we will see, the filtering algorithm developed in this work is closely related to the special case of stochastic control problems where the running cost is of the form

$$l(t, x, u) = g(t, x) + \frac{1}{2}\|u\|^2, \tag{5}$$

where $g(t, x)$ is a running cost depending only on the state x, not the control. With this running cost, the minimization in (3) can be solved in closed-form, yielding the optimal policy

$$u_t^\star = -\sigma(t, X_t)'\nabla V_t(X_t), \tag{6}$$

and the HJB equation (3) simplifies to

$$\frac{\partial V_t}{\partial t} + b \cdot \nabla V_t + g - \frac{1}{2}\nabla V_t'\sigma\sigma'\nabla V_t + \frac{1}{2}\operatorname{Tr}(\sigma\sigma'\nabla^2 V_t) = 0. \tag{7}$$

The running cost (5) plays a crucial role in our framework. The quadratic cost in control u in (5) quantifies the difference of the controlled and the uncontrolled ($u_t \equiv 0$) process. More specifically, let $\mathcal{P}_u$ denote the measure over the path space $\Omega = C([0, T]; \mathbb{R}^n)$ induced by the dynamics (1), and $\mathcal{P}_0$ the measure

[1] A stochastic control problem may involve a more general form of the dynamics where the noise and control enter through different channels. However, this special form facilitates the path integral formulation and it is sufficient to represent the optimal control formulation of the smoothing problem discussed in Section 2.2.

associated with the uncontrolled process. Here $C([0,T];\mathbb{R}^n)$ denotes the space of continuous functions from the interval $[0,T]$ to $\mathbb{R}^n$. Then, by the Girsanov theorem (Särkkä & Sottinen, 2008),

$$\frac{d\mathcal{P}_u}{d\mathcal{P}_0} = \exp\left\{\int_0^T \frac{1}{2}\|u_t\|^2 dt + u_t' dW_t\right\}. \tag{8}$$

It follows that the Kullback–Leibler divergence between $\mathcal{P}_u$ and $\mathcal{P}_0$ is Särkkä and Sottinen (2008)

$$\mathrm{KL}(\mathcal{P}_u \parallel \mathcal{P}_0) := \int_\Omega d\mathcal{P}_u \log\frac{d\mathcal{P}_u}{d\mathcal{P}_0} = \mathbb{E}\left\{\int_0^T \frac{1}{2}\|u_t\|^2 dt\right\}, \tag{9}$$

where the expectation is with respect to the controlled process. Here we have used the fact that $\mathbb{E}\{u_t' dW_t\} = 0$. Thus, the optimal control problem with running cost (5) can be equivalently written as

$$\min_{\mathcal{P}_u} \mathbb{E}_{\mathcal{P}_u}\left\{\int_0^T g(t,X_t)dt + \Psi_T(X_T)\right\} + \mathrm{KL}(\mathcal{P}_u \parallel \mathcal{P}_0). \tag{10}$$

Note that the optimization variable becomes $\mathcal{P}_u$ instead of the control policy; the two are equivalent as the control policy fully determines the measure $\mathcal{P}_u$ and vice versa (Thijssen & Kappen, 2015).

When the cost is of the form (5), it turns out that the above nonlinear optimal control problem can be solved in a linear manner (Doucet et al., 2006; Guarniero et al., 2017; Heng, Bishop, Deligiannidis, & Doucet, 2017; Reich, 2018; Ruiz & Kappen, 2017; Särkkä & Sottinen, 2008; Thalmeier, Kappen, Totaro, & Gómez, 2020; Thijssen & Kappen, 2015; Williams, Aldrich, & Theodorou, 2017; Zhang, Wang, Hartmann, Weber, & Schütte, 2014). One way to see it is through the logarithmic transformation (Fleming & Rishel, 1975) of the HJB equation (7). More specifically, let

$$\varphi(t,x) := \exp(-V_t(x)), \tag{11}$$

then a straightforward calculation leads to

$$\frac{\partial\varphi}{\partial t} + b\cdot\nabla\varphi - g\varphi + \frac{1}{2}\mathrm{Tr}(\sigma\sigma'\nabla^2\varphi) = 0,\ \varphi(T,\cdot) = \exp\{-\Psi_T\}. \tag{12}$$

The associated optimal control strategy is

$$u_t^\star = \sigma(t,X_t)'\nabla\log\varphi(t,X_t). \tag{13}$$

Note that unlike the HJB (7) which is nonlinear, (12) is a linear partial differential equation (PDE); it is the Backward Kolmogorov equation (Särkkä & Solin, 2019) associated with the (uncontrolled $u_t \equiv 0$) process (1) and killing rate g.

2.2. Smoothing as stochastic control

Consider a diffusion process with noisy measurements governed by the SDEs

$$dX_t = b(t,X_t)dt + \sigma(t,X_t)dW_t, \quad X_0 \sim \nu_0 \tag{14a}$$

$$dY_t = h(t,X_t)dt + \sigma_B dB_t, \quad Y_0 = 0 \tag{14b}$$

where the measurement $Y_t \in \mathbb{R}^p$ is corrupted by the standard Wiener process $B_t \in \mathbb{R}^p$ weighted by $\sigma_B > 0$ and the initial state X_0 follows the prior distribution ν_0. The smoothing problem is a particular type of Bayesian inference problem that aims at estimating the distribution of X_t for $0 \le t \le T$ given the full history of measurement $\{Y_t,\ 0 \le t \le T\}$.

It was discovered in Kim and Mehta (2020), Mitter and Newton (2003) that the smoothing problem can be reformulated as a stochastic optimal control problem whose cost function depends on the measurements. To see this, denote the measure over the path space Ω induced by the process (14a) by $\mathcal{P}$. This serves as the prior measure for this Bayesian inference problem. Denote the posterior distribution over Ω by $\mathcal{Q}^Y$. By the Kallianpur–Striebel formula (Klebaner, 2005),

$$\frac{d\mathcal{Q}^Y}{d\mathcal{P}} \propto \exp\left\{-\frac{1}{\sigma_B^2}\left[\int_0^T Y_t dh + \frac{1}{2}\|h\|^2 dt - Y_T h(T,X_T)\right]\right\}. \tag{15}$$

The right hand side of (15) is the likelihood of the measurement. The variational form of the smoothing problem seeks a distribution $\tilde{\mathcal{P}}$ on the path space that minimizes

$$\begin{aligned}\mathbb{E}_{\tilde{\mathcal{P}}}\left\{\log\frac{d\tilde{\mathcal{P}}}{d\mathcal{Q}^Y}\right\} &= \mathbb{E}_{\tilde{\mathcal{P}}}\left\{\log\frac{d\tilde{\mathcal{P}}}{d\mathcal{P}} - \log\frac{d\mathcal{Q}^Y}{d\mathcal{P}}\right\} \\ &= \mathrm{KL}(\tilde{\mathcal{P}} \parallel \mathcal{P}) - \mathbb{E}_{\tilde{\mathcal{P}}}\left\{\log\frac{d\mathcal{Q}^Y}{d\mathcal{P}}\right\}.\end{aligned} \tag{16}$$

Let $\tilde{\mathcal{P}}$ be parametrized by the diffusion process $\tilde{X}_t$ with dynamics

$$d\tilde{X}_t = b(t,\tilde{X}_t)dt + \sigma(t,\tilde{X}_t)(u_t dt + dW_t),\ \tilde{X}_0 \sim \pi_0. \tag{17}$$

By Girsanov theorem (8),

$$\mathrm{KL}(\tilde{\mathcal{P}} \parallel \mathcal{P}) = \mathbb{E}\left\{\int_0^T \frac{1}{2}\|u_t\|^2 dt\right\} + \mathrm{KL}(\pi_0 \parallel \nu_0). \tag{18}$$

Note that (18) is slightly different from (9) since in the control problem $\mathcal{P}_u$ and $\mathcal{P}_0$ share the same initial distribution while (17) and (14a) do not. Plugging (15) and (18) into (16) yields an optimal control formulation (Kim & Mehta, 2020)

$$\begin{aligned}\min_{u,\pi_0}\ \mathbb{E}&\left\{\int_0^T [\frac{1}{2}\|u_t\|^2 + \frac{1}{2\sigma_B^2}\|h(t,\tilde{X}_t)\|^2]dt\right. \\ &\left.+\frac{1}{\sigma_B^2}Y_t dh(t,\tilde{X}_t) - \frac{1}{\sigma_B^2}Y_T h(T,\tilde{X}_T)\right\} + \mathrm{KL}(\pi_0 \parallel \nu_0)\end{aligned} \tag{19}$$

for the smoothing problem. Apart from an extra term $\mathrm{KL}(\pi_0 \parallel \nu_0)$ related to the initial distributions, (19) coincides with the optimal control problem (1)–(2)–(5) if we take

$$g(t,x)dt = \frac{1}{2\sigma_B^2}\|h(t,x)\|^2 dt + \frac{1}{\sigma_B^2}Y_t dh(t,x), \tag{20a}$$

$$\Psi_T(x) = -\frac{1}{\sigma_B^2}Y_T h(T,x). \tag{20b}$$

Remark 1. The optimal control formulation (19) relies heavily on the relation between Kullback–Leibler divergence and the control energy in (18), which is itself due to the Girsanov theorem. Its counterpart in the discrete-time setting is much more complicated.

3. Path integral particle smoothing and filtering

Building on the control formulation of the smoothing (19), we propose a new particle filtering algorithm under the name “path integral particle filter (PIPF)”. At each iteration, our algorithm solves a smoothing problem over a sliding window based on the duality between smoothing and optimal control presented in Section 2.2.

3.1. Path integral particle smoothing

We begin with the smoothing problem to estimate the posterior distribution $\mathcal{Q}^Y$ for the system (14) over the time-window $[0,T]$. As discussed in Section 2.2, this smoothing problem amounts to an optimal control problem

$$\min_{u,\pi_0}\mathbb{E}\left\{\int_0^T [\frac{1}{2}\|u_t\|^2 + g(t,X_t)]dt + \Psi_T(X_T)\right\} + \mathrm{KL}(\pi_0 \parallel \nu_0) \tag{21}$$

where g and Ψ_T are given in (20). The only difference to a standard optimal control problem is that the initial distribution π_0, apart from the control u, is also an optimization variable.

Since the minimization over u is independent of π_0, the optimal control strategy remains to be $u^\star(t,x) = \sigma(t,x)'\nabla \log\varphi(t,x)$ with φ as in (12). Plugging this optimal control into (21) we arrive at the optimization over π_0,

$$\min_{\pi_0} \mathbb{E}_{\pi_0} \{-\log\varphi(0,X_0) - \log\nu_0(X_0) + \log\pi_0(X_0)\}. \tag{22}$$

The optimal solution is

$$\pi_0^\star(\cdot) \propto \nu_0(\cdot)\varphi(0,\cdot). \tag{23}$$

Note that $\pi_0^\star$ is exactly the posterior distribution of X_0 given the full observation $\{Y_t,\ 0 \le t \le T\}$. Thus, to sample from the posterior distribution $\mathcal{Q}^Y$, one can sample K trajectories $\{X_t^k\}_{k=1}^K$ of the diffusion process (17) for $t \in [0,T]$ under the optimal control strategy $u^\star$ with the initial distribution $\pi_0^\star$, i.e.

$$dX_t^k = b(t,X_t^k)dt + \sigma(t,X_t^k)(u^\star(t,X_t^k)dt + dW_t),\ \ X_0^k \sim \pi_0^\star$$

for $k = 1,\ldots,K$. The empirical distribution formed by these K trajectories on the path space Ω is an approximation of the posterior distribution $\mathcal{Q}^Y$. Moreover, for any $0 \le t \le T$, the empirical distribution

$$\frac{1}{K}\sum_{k=1}^K \delta_{X_t^k} \tag{24}$$

forms an approximation of the conditional distribution of X_t given the full observation $\{Y_t,\ 0 \le t \le T\}$.

The above sampling strategy requires the exact posterior distribution $\pi_0^\star$ of X_0 and the exact optimal control strategy $u^\star$. This can be made possible by solving the PDE (12) but is still computationally demanding for high dimensional problems.

Our strategy to sample from $\mathcal{Q}^Y$ is to sample trajectories with a suboptimal initial distribution π_0 and a suboptimal control strategy u, and then weight the trajectories through importance sampling. More precisely, let $\tilde{\mathcal{P}}$ be the measure over the path space Ω associated with initial distribution π_0 and a suboptimal control strategy u, and $\{X_t^k\}_{k=1}^K$ be K trajectories independently sampled from $\tilde{\mathcal{P}}$. By Girsanov theorem, in view of (15),

$$\frac{d\mathcal{Q}^Y}{d\tilde{\mathcal{P}}} = \frac{d\mathcal{Q}^Y}{d\mathcal{P}}\frac{d\mathcal{P}}{d\tilde{\mathcal{P}}} \propto \frac{d\nu_0}{d\pi_0}\exp[-S_u(0,T)] \tag{25}$$

where S_u denotes the path integral defined as

$$S_u(t,s) = \int_t^s [\frac{1}{2}\|u_\tau\|^2 + \frac{1}{2\sigma_B^2}\|h(\tau,X_\tau)\|^2]d\tau + \frac{1}{\sigma_B^2}Y_\tau dh$$
$$+u_\tau' dW_\tau - \frac{1}{\sigma_B^2}Y_s h(s,X_s) + \frac{1}{\sigma_B^2}Y_t h(t,X_t). \tag{26}$$

Denote the value of S_u along the trajectory X_t^k by S_u^k and define the weights

$$w^k = \frac{d\nu_0}{d\pi_0}(X_0^k)\exp[-S_u^k(0,T)]. \tag{27}$$

It follows from (25) that $\mathcal{Q}^Y$ can be approximated by the empirical distribution formed by the trajectories $\{X_t^k\}_{k=1}^K$ and with weights $\{w^k\}_{k=1}^K$, that is,

$$d\mathcal{Q}^Y \approx \sum_{k=1}^K \hat{w}^k \delta_{X_{(\cdot)}^k}, \tag{28}$$

where

$$\hat{w}^k = \frac{w^k}{\sum_{k=1}^K w^k}$$

are the normalized weights. Similarly, the posterior distribution of X_t for any $0 \le t \le T$ is approximated by

$$\sum_{k=1}^K \hat{w}^k \delta_{X_t^k}. \tag{29}$$

The effectiveness of the above approximation (28) depends on the variance of the weights $\{\hat{w}^k\}_{k=1}^K$. The lower the variance is, the better the approximation becomes. This variance reduces to zero when π_0 and u are optimal, that is, $\pi_0 = \pi_0^\star, u = u^\star$. In general, computing the exact optimal solution is too expensive and one has to use a suboptimal solution that is easier to compute. There are many methods that can generate suboptimal controller for (21), including differential dynamic programming (DDP) (Jacobson & Mayne, 1970) and iterative linear quadratic regulator (iLQR) (Li & Todorov, 2004). One can also start from the original smoothing problem for (14) and adopt suboptimal smoothing methods such as extended Rauch–Rung–Striebel (ERTS) (Rauch, Tung, & Striebel, 1965). These suboptimal smoothing methods induce suboptimal π_0 and u for (21).

To summarize, our path integral particle smoothing method consists of a proposal initial distribution π_0 and a proposal feedback u. They should be designed such that the distribution on the path space induced by π_0 and u is an approximation of the posterior distribution $\mathcal{Q}^Y$. A better proposal gives a better estimation with lower variance. In our experiments, we found iLQR to be an effective proposal for u. A natural proposal for π_0 is ν_0. Once the proposal is chosen, we can sample trajectories from the controlled diffusion process (17) under the proposal control strategy u with the proposal initial distribution π_0. The posterior distribution $\mathcal{Q}^Y$ is then approximated by (28) whose weights are computed using the path integral (26).

3.2. Path integral particle filtering

We next move to the filtering problem. We are interested in the problem of approximating the posterior distribution of X_t conditioned on the past observations $\{Y_\tau, 0 \le \tau \le t\}$. More precisely, let $\mathcal{Y}_t = \sigma(Y_\tau : 0 \le \tau \le t)$ denote the σ-field generated by the observation up to the time t. Then the objective of the filtering is to approximate $\mathcal{P}(X_t \in \cdot \mid \mathcal{Y}_t)$.

The path integral particle smoothing algorithm proposed in Section 3.1 is suitable for smoothing problem over a fixed time window $[0,T]$. To use this method for the filtering problem where new measurements keep coming in, one naive strategy is to carry out the smoothing task over the time window $[0,t]$. However, this requires recursively implementing the smoothing algorithm over a larger and larger time window. As t increases, the computational complexity of the smoothing problem grows and will eventually become computationally infeasible.

We propose to use a sliding window implementation of the smoothing algorithm to solve the filtering problem. More specifically, consider the smoothing problem over the time window $[t-H,t]$ of size $H > 0$. It is equivalent to the optimal control problem

$$\min_{u,\pi_{t-H}} \mathbb{E}\left\{\int_{t-H}^t [\frac{1}{2}\|u_\tau\|^2 + g(\tau,X_\tau)]d\tau + \Psi_t(X_t)\right\} \tag{30}$$
$$+\ \mathrm{KL}(\pi_{t-H}\ \|\ \nu_{t-H}),$$

where g is as in (20a) and

$$\Psi_t(x) = -\frac{1}{\sigma_B^2}Y_t h(t,x). \tag{31}$$

The prior distribution ν_{t-H} for this smoothing problem is the posterior distribution $\mathcal{P}(X_{t-H} \in \cdot \mid \mathcal{Y}_{t-H})$. Since ν_{t-H} already accounts for all the observations $\{Y_\tau, 0 \le \tau \le t-H\}$, the solution to the smoothing problem (30), in fact, induces the exact posterior distribution over the trajectories $\{X_\tau, t-H \le \tau \le t\}$, conditioned on the full history of observations $\{Y_\tau, 0 \le \tau \le t\}$. Thus, by running the smoothing algorithm presented in Section 3.1 over a fixed-size time window $[t-H,\ t]$, we can obtain the posterior distribution of $\mathcal{P}(X_t \mid \mathcal{Y}_t)$.

To implement the path integral particle smoothing algorithm over the time window $[t-H, t]$, one needs to evaluate $d\nu_{t-H}/d\pi_{t-H}$ as in (27). However, in the proposed path integral particle filtering method, the distribution ν_{t-H} does not have a closed-form expression and is represented by a collection of weighted particles as

$$\nu_{t-H} \approx \sum_{k=1}^{K} \tilde{w}_{t-H}^k \delta_{X_{t-H}^k}. \tag{32}$$

Thus, a natural way to sample trajectories over the time interval $[t-H,\ t]$ is to initialize them with $\{X_{t-H}^k\}_{k=1}^K$ and then follow the closed-loop dynamics (17) under a suboptimal control policy u. With this strategy, the proposal initial distribution π_{t-H} satisfies

$$\frac{d\nu_{t-H}}{d\pi_{t-H}}(X_{t-H}^k) \propto \tilde{w}_{t-H}^k. \tag{33}$$

Let $\{X_{(\cdot)}^k\}_{k=1}^K$ be the K generated trajectories and $S_u^k(t-H, t)$ be the value of (26) of the trajectory $X_{(\cdot)}^k$ over the time interval $[t-H,\ t]$. Then, by (25), the posterior distribution over the trajectory space conditioned on the past observations $\{Y_\tau, 0 \le \tau \le t\}$ is approximated by

$$\sum_{k=1}^{K} \hat{w}^k X_{(\cdot)}^k \tag{34}$$

with $\{\hat{w}^k\}_{k=1}^K$ being the normalized version of the weights

$$\tilde{w}_{t-H}^k \exp[-S_u^k(t-H, t)]. \tag{35}$$

To see the rationale of (35), assume that ν_{t-H} in (30) is obtained using the path integral particle smoothing algorithm over the time interval with proposal initial distribution ν_0. Following the arguments in Section 3.1, by (27), we obtain

$$\tilde{w}_{t-H}^k \propto \exp[-S_u^k(0, t-H)]$$

where $S_u^k(0, t-H)$ is evaluated over some sampled trajectory over the time interval $[0,\ t-H]$. Combining it with (35) we conclude that

$$\hat{w}^k \propto \exp[-S_u^k(0, t-H)] \exp[-S_u^k(t-H, t)] = \exp[-S_u^k(0, t)]$$

where $S_u^k(0, t)$ is evaluated over the concatenated trajectory of $X_\tau^k, 0 \le \tau \le t-H$ and $X_\tau^k, t-H \le \tau \le t$. This is the same as (27) in the smoothing problem when $\pi_0 = \nu_0$. Instead of resampling the whole trajectory starting from the very beginning, in the sliding window filtering, all the past weights are recorded in the particle representation of ν_{t-H} and are combined with the measurement over $[t-H,\ t]$ to estimate the posterior distribution.

3.3. Algorithm and implementation

In this section we provide implementation details of the path integral particle filtering algorithm. Let $0 = t_0 < t_1 < t_2 < \cdots$ be a sequence of time discretization points. It can be a constant stepsize discretization, i.e., $t_{j+1} - t_j = \Delta t$, or any other more flexible discretization scheme. With a slight abuse of notation, let H denote the number of the discretization points included in the sliding time window employed in the path integral particle filtering algorithm.

The proposed particle filtering algorithm is divided into two stages. For $t_j, j \le H$, the total number of time steps is less than the window size. Therefore, we use the path integral particle smoothing over the time interval $[0, t_j]$ to estimate the posterior distribution $\mathcal{P}(X_{t_j}|\mathcal{Y}_{t_j})$. When $j > H$, we adopt the path integral particle filtering over the time interval $[t_{j-H}, t_j]$.

In the sliding window stage over $[t_{j-H}, t_j]$, the choice of the particle representation for the prior distribution $\nu_{t_{j-H}} = \mathcal{P}(X_{t_{j-H}}|\mathcal{Y}_{t_{j-H}})$ is crucial. We use the trajectories generated in the previous step over the time interval $[t_{j-H-1}, t_{j-1}]$ with appropriate weights to obtain an estimate of $\nu_{t_{j-H}}$. The locations of the particles, that are generated in this way, account for the measurements up to the time t_{j-1} and thus match the posterior distribution $\mathcal{P}(X_{t_{j-H}}|\mathcal{Y}_{t_j})$, which is the ideal proposal initial distribution (Doucet et al., 2006), better. More explicitly, the prior distribution $\nu_{t_{j-H}}$ is updated recursively as follows. Let the particle representation of the prior distribution $\nu_{t_{j-H-1}}$ at the previous step be

$$\nu_{t_{j-H-1}} \approx \sum_{k=1}^{K} \tilde{w}_{t_{j-H-1}}^k X_{t_{j-H-1}}^k, \tag{36}$$

and S_u^k be the values of (26) evaluated over the sampled trajectories over the time window $[t_{j-H-1}, t_{j-1}]$. Then,

$$\nu_{t_{j-H}} \approx \sum_{k=1}^{K} \tilde{w}_{t_{j-H-1}}^k \exp[-S_u^k(t_{j-H-1}, t_{j-H})] X_{t_{j-H}}^k. \tag{37}$$

In the above, to simplify the notation, the normalization for the weights is not displayed explicitly. We remark that even through the trajectories are generated based on the measurements over the time interval $[t_{j-H-1}, t_{j-1}]$, the weights of $\nu_{t_{j-H}}$ in (37) use only the information, encoded in $\exp[-S_u^k(t_{j-H-1}, t_{j-H})]$, over the interval $[t_{j-H-1}, t_{j-H}]$.

The effective size of the samples decreases much slower for PIPF compared to the standard SIR filter as time grows. Yet, a resampling step is needed after a long time horizon when the effective size is too small. For the resampling, we start with the samples in (37). We resample them based on the weights

$$\tilde{w}_{t_{j-H-1}}^k \exp[-S_u^k(t_{j-H-1}, t_{j-1})], \tag{38}$$

obtaining new samples $\hat{X}_{t_{j-H}}^k$. These samples follow approximately the distribution $\mathcal{P}(X_{t_{j-H}}|\mathcal{Y}_{t_{j-1}})$. With these new samples, the prior distribution $\nu_{t_{j-H}}$ in (37) is approximated by

$$\nu_{t_{j-H}} \approx \sum_{k=1}^{K} \exp[S_u^k(t_{j-H}, t_{j-1})] \hat{X}_{t_{j-H}}^k. \tag{39}$$

This is obtained from combining (37), the resampling weights in (38), and the fact that

$$S_u(t, s) = S_u(t, \tau) + S_u(\tau, s), \quad t < \tau < s \tag{40}$$

which itself follows from (26).

Once the particle representation $\sum_{k=1}^K \tilde{w}_{t_{j-H}}^k X_{t_{j-H}}^k$ of $\nu_{t_{j-H}}$ is available, one can start from it and apply the path integral particle smoothing over the time window $[t_{j-H}, t_j]$. This leads to the particle filtering algorithm

$$\mathcal{P}(X_{t_j}|\mathcal{Y}_{t_j}) \approx \sum_{k=1}^{K} \tilde{w}_{t_{j-H}}^k \exp[-S_u^k(t_{j-H}, t_j)] X_{t_j}^k. \tag{41}$$

The overall structure of the proposed algorithm is illustrated in Fig. 1. The complete path integral particle filtering algorithm is presented in Algorithm 1 and a subroutine of it over a given time window is provided in Algorithm 2. The prior distribution at each step is represented by the weighted particles $\{X_p^k, \tilde{w}^k\}$. Here p stands for the prior. The filtering results $\mathcal{P}(X_{t_j}|\mathcal{Y}_{t_j})$ at the current step are represented by weighted particles $\{X^k, \hat{w}^k\}$.

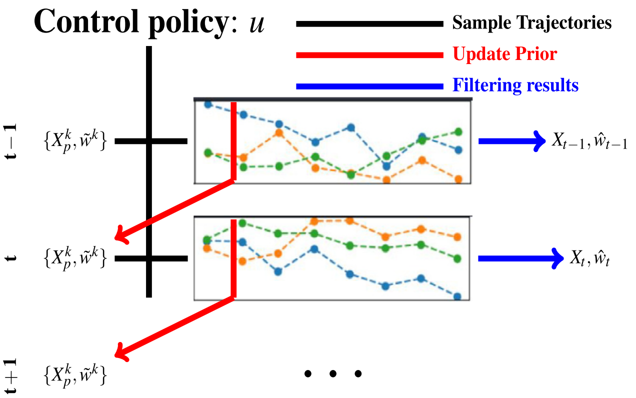

Fig. 1. Pipeline diagram of PIPF.

Algorithm 1 Path Integral Particle Filtering (PIPF)

```
**Input:** $L$: Total Number of time steps
$\{X_p^k\}$: Samples from prior distribution $\nu_0$
$\{\tilde{w}^k\}$: Weight of samples, uniform
$H$: Length of sliding windows
**for** $j \leftarrow 1, \cdots, L$ **do**
    $j$-th time interval $\leftarrow [\max(0, t_{j-H}), t_j]$
    $\{X^k\}, \{\hat{w}^k\}, \{X_p^k\}, \{\tilde{w}^k\} \leftarrow$ Algorithm 2 for $j$-th time interval
**end for**
```

Algorithm 2 One Step of PIPF

```
**Input:** $b, \sigma, \sigma_B$: System model
$g, \Psi$: Cost model as in (20)
$\gamma_{\text{thres}}$: threshold for resampling
$[t_i, t_j]$: sliding window
$K$: Number of samples
$\{X_p^k\}, \{\tilde{w}_{t_i}^k\}$: Samples and weights of the previous step
**Output:** $\{X^k\}, \{\hat{w}^k\}$: Filtering results
$\{X_p^k\}, \{\tilde{w}_{t_{i+1}}^k\}$: Samples and weight for the current step
**for** $k \leftarrow 1, \cdots, K$ **do**
    Sampling $k$-th trajectory $X_{(\cdot)}^k$ initialized by $X_p^k$
    Evaluate the value of $S_u^k$ over the trajectory $X_{(\cdot)}^k$
    Filtering samples: $X^k = X_{t_j}^k$
    Filtering weight: $\hat{w}^k \propto \tilde{w}_{t_i}^k \exp[-S_u^k(t_i, t_j)]$
    # Prior information for usage in next sliding window
    Prior sample: $X_p^k = X_{t_{i+1}}^k$
    Prior weight: $\tilde{w}_{t_{i+1}}^k \propto \tilde{w}_{t_i}^k \exp[-S_u^k(t_i, t_{i+1})]$
**end for**
# Check resampling
Effective ratio: $\gamma = \frac{1}{K \sum_{k=1}^K (\hat{w}^k)^2}$
**if** $\gamma < \gamma_{\text{thres}}$ **then**
    $\{X_p^k\} \sim \text{multinomial}(\{X_p^k\}, \{\tilde{w}_{t_i}^k \exp[-S_u^k(t_i, t_j)]\})\}$
    $\{\tilde{w}_{t_{i+1}}^k\} \propto \exp[S_u^k(t_{i+1}, t_j)]$
**end**
```

The performance of the PIPF algorithm depends on the length H of the sliding window and the choice of proposal suboptimal control u. When $H = 1$ and $u \equiv 0$, our algorithm reduces to the standard sequential importance resampling (SIR) algorithm (Doucet et al., 2006) as explained further in the following remark.

Remark 2. Without any control, i.e., $u = 0$, the algorithm resembles a sliding window version of the SIR particle filter. Indeed, when $u = 0$, the location of the particles X_t^k is only governed by the open-loop dynamics (14a) similar to the SIR particle filter. By (26), the weights of the particles $w_t^k \propto \exp[-S_0^k(0, t)]$ where

$$S_0^k(0, t) = \int_0^t \frac{1}{2\sigma_B^2} \|h(\tau, X_\tau^k)\|^2 d\tau + \frac{1}{\sigma_B^2} Y_\tau dh(\tau, X_\tau^k) - \frac{1}{\sigma_B^2} Y_t h(t, X_t^k).$$

This is precisely the log-likelihood of the observation signal over the time interval $[0, t]$. With a time discretization of the integral $0 = t_0 < t_1 < \cdots < t_J = t$, the weights can be expressed as multiplication of the likelihoods $\prod_{j=1}^J p(Y_{t_j} | X_{t_j}^k)$, which is similar to how SIR particle filter updates the weights.

4. Numerical examples

In this section we present several numerical examples to demonstrate the efficacy of the proposed path integral particle filtering algorithm. In the first example, we test the proposed algorithm in the linear Gaussian setting. In the second example, we consider a nonlinear filtering example where the optimal filter solution can be obtained in closed form. These examples are used to show that PIPF is able to approximate this optimal filter solution well.

4.1. Linear filtering examples

We first consider the following one-dimensional state space model

$$dX_t = -\kappa X_t dt + dW_t, \ X_0 \sim N(m_0, P_0)$$
$$dY_t = X_t dt + \sigma_B dB_t,$$

where $\kappa > 0$. The model corresponds to an *Ornstein–Uhlenbeck* process, whose measurements are corrupted with Gaussian noise. The posterior distribution is Gaussian, that is, $\mathcal{P}(X_t | \mathcal{Y}_t) = N(m_t, P_t)$ with

$$dm_t = -\kappa m_t dt + \frac{P_t}{\sigma_B}(dY_t - m_t dt),$$

$$\frac{dP_t}{dt} = -2\kappa P_t - \frac{P_t^2}{\sigma_B} + 1.$$

Three filtering algorithms are employed for this problem: (i) the SIR particle filter; (ii) the path integral particle filter with zero controller (PIPF-zero) $u \equiv 0$; (iii) the path integral particle filter with linear quadratic regulator controller (PIPF-LQR). LQR is designed based on the cost function given in (20). The simulations are executed for $L = 600$ times-steps with step-size $\Delta t = 0.01$. Both PIPF-zero and PIPF-LQR employ a sliding window size $H = 20$. All three algorithms use $K = 500$ particles.

It is well known the resampling procedure helps to mitigate the particle degeneracy issue (Doucet et al., 2006). We benchmark the performance of the algorithms in terms of the mean squared errors (m.s.e.) between the exact mean (covariance) and the estimated mean (covariance), with and without the resampling step. The results are depicted in Figs. 2 and 3 respectively. It is observed that the PIPF with the LQR controller outperforms the PIPF without control and the SIR by a large margin when the resampling step is not used. This advantage becomes less significant but still exists when the resampling is used. In the experiments, each algorithm is repeated for 50 trials with different random seeds. The solid curves represent the average of 50 trials and the shaded regions represent the corresponding 1× standard

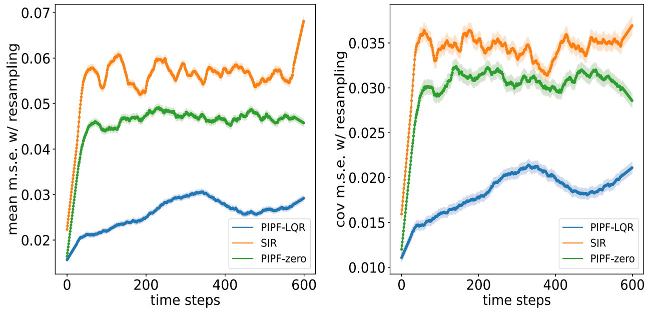

Fig. 2. Performance comparison w/ resampling. PIPF-LQR outperforms both PIPF-zero and SIR.

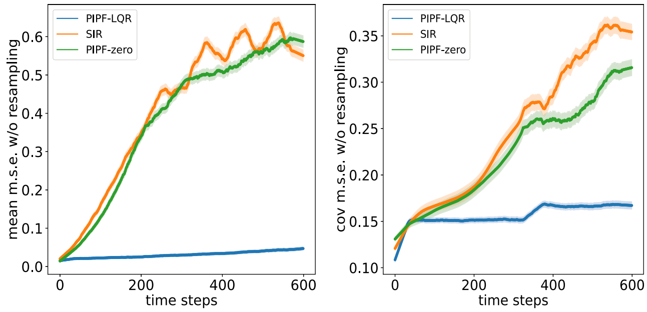

Fig. 3. Performance comparison w/o resampling. PIPF-LQR outperforms both PIPF-zero and SIR by a large margin.

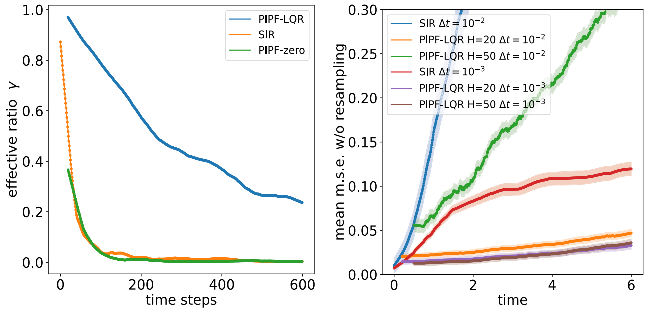

Fig. 4. Left: Comparison of the effective ratio. The level of particle degeneracy is much lower in PIPF-LQR, compared with SIR and PIPF-zero. Right: Performance comparison with different stepsize Δt. Small Δt can improve accuracy and make the PIPF-LQR less sensitive to the window size H.

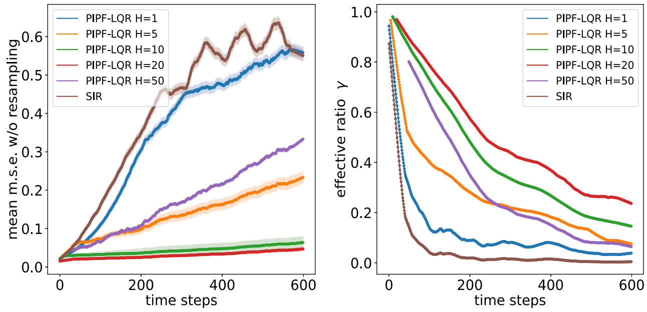

Fig. 5. Performance comparison for different H.

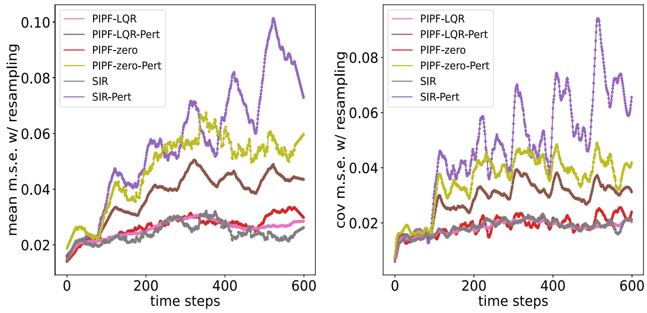

Fig. 6. Robustness of PIPF against observation perturbation. PIPF-LQR is less sensitive to perturbations in the observation model.

deviation. The effective size of the particles can be measured by the ratio $\gamma = 1/(K\sum_{k=1}^{K}(\hat{w}^k)^2)$. This is depicted in Fig. 4 (left) in the absence of resampling. From these results we can clearly see the advantage of the path integral particle filter. Compared with SIR, the introduction of the sliding window in PIPF-zero, which only requires negligible memory footprint, can help reduce the bias of the estimation. The significant improvement of PIPF-LQR owes to the high effective ratio and high-quality particles controlled by the optimal control policy.

To investigate the influence of the size H of the sliding window, we test the PIPF algorithm (with LQR proposal) with different H values. The performance in terms of m.s.e. (of means) with resampling and effective ratio without resampling is depicted in Fig. 5. We notice that increasing H improves the filter performance while very large H may deteriorate the results. The latter could be due to the accumulation of the time discretization error as our algorithm is designed for continuous-time models. To illustrate this point, we carry out an experiment with different stepsizes. As shown in Fig. 4 (right), with a smaller stepsize $\Delta t = 0.001$, the performance of both SIR and PIPF-LQR improves. Moreover, with this stepsize, the performance of PIPF-LQR with $H = 20, 50$ is almost the same. This is in contrast to the case with larger stepsize $\Delta t = 0.01$ where a longer sliding window $H = 50$ is accompanied with a larger error. All simulation parameters, other than H, are kept the same in this comparison.

We also conduct experiments to test the robustness of PIPF with the LQR controller when the observation model is not accurate. We consider the case where the observation model suffers from consistent biases and extra random noise. To make the comparison fair, we first increase K for SIR and PIPF-zero such that they match the performance of PIPF-LQR when the perfect observation model is available. We then perturb the observation model by an extra randomly generated shift and additional Gaussian noise with intensity $0.2\sigma_B$ occasionally. The results are depicted in Fig. 6.[2] Though PIPF with LQR controller uses less particles, it is still less sensitive to the perturbation of observation models, compared with PIPF-zero and SIR.

To study the effects of problem dimension on the filtering performance, we consider the multi-dimensional state space model

$$\begin{aligned} dX_t &= AX_t dt + dW_t,\ \ X_0 \sim N(m_0, P_0) \\ dY_t &= CX_t dt + \sigma_B dB_t, \end{aligned}$$

where $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{p\times n}$ represent the dynamics matrix and output matrix respectively. In our experiments, A, C are randomly generated for each dimension size n and are then fixed. In Fig. 7 we display the performance of PIPF-LQR and SIR for different n in terms of m.s.e (of means) and effective ratio without resampling. Clearly, PIPF-LQR scales better than SIR as n increases.

[2] We disturb observation at $t = 100, 200, 300, 400, 500$. In the plots we use savgol_filter smooth curves for better visualization.

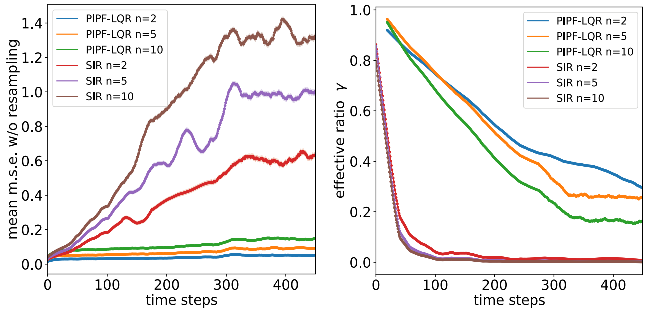

Fig. 7. Performance comparison for different n. PIPF-LQR demonstrates better scalability to high dimensional problems.

4.2. Nonlinear filtering example

We next evaluate our PIPF algorithm on the Benes filter problem (Taghvaei et al., 2020)

$$\begin{aligned} dX_t &= \mu\sigma_W \tanh(\frac{\mu}{\sigma_W}X_t)dt + \sigma_W dW_t, \quad X_0 = x_0, \\ dY_t &= (h_1 X_t + h_1 h_2)dt + dB_t, \end{aligned}$$

where x_0 is a given constant, and the constants $\mu, \sigma_W, h_1, h_2 \in \mathbb{R}$ are the model parameters. Regardless of being a nonlinear filtering problem, its posterior distribution admits the analytical expression

$$\omega_t N(a_t - b_t, \sigma_t^2) + (1-\omega_t)N(a_t + b_t, \sigma_t^2), \tag{42}$$

where

$$\begin{aligned} a_t &= \sigma_W \Psi_t \tanh(h_1\sigma_W t) + \frac{h_2 + x_0}{\cosh(h_1)\sigma_W t} - h_2, \\ b_t &= \frac{\mu}{h_1}\tanh(h_1\sigma_W t), \quad \sigma_t^2 = \frac{\sigma_W}{h_1}\tanh(h_1\sigma_W t), \\ \Psi_t^2 &= \int_0^t \frac{\sinh(h_1\sigma_W s)}{\sinh(h_1\sigma_W t)}dY_s, \\ \omega_t &= \frac{1}{1+\exp\{\frac{2a_t b_t}{\sigma_W}\coth(h_1\sigma_W t)\}}. \end{aligned}$$

In the experiments, we use the model parameters $\mu = 1, h_1 = 1, h_2 = 0, \sigma_W = 1, x_0 = -5.0$. The simulation is carried out for $L = 6000$ time steps with step-size $\Delta t = 0.001$. All the algorithms use $K = 2000$ particles. We compare the performance of our PIPF algorithm and SIR. Since the dynamics are nonlinear, we use a suboptimal control policy, iterative linear quadratic regulator for PIPF. iLQR (Li & Todorov, 2004) approximates the nonlinear dynamics by linearizing it around a nominal trajectory and the cost by a quadratic function, yielding a LQR problem. The iLQR algorithm then solves the resulting LQR problem. The nominal trajectory is calculated by minimizing control cost equation (21) locally under a noise-free version of the dynamics (1). In PIPF-iLQR, the linearization for iLQR is done at the beginning of each sliding window.

The results are displayed in Figs. 8–9. The sliding window size for PIPF-iLQR and PIPF-zero is set to be $H = 10$. Fig. 8 displays the estimated posterior distributions of the state at several time points when resampling is used. These distributions are approximated using the KDE density estimator (Davis, Lii, & Politis, 2011) with bandwidth 0.2. In Fig. 9, we show the m.s.e. of the means with resampling and the effective ratio without resampling. From the experiments we see that even though the optimality of the controller is not promised, PIPF-iLQR still outperforms the other algorithms.

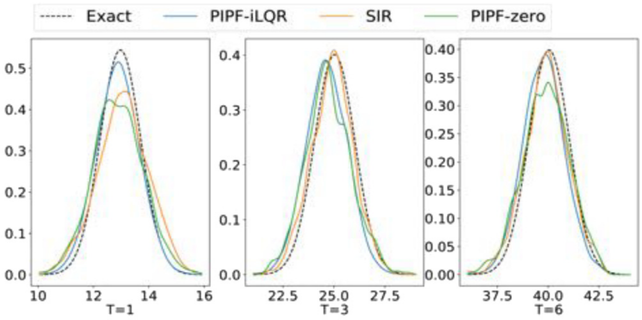

Fig. 8. Estimated posterior distribution vs ground truth.

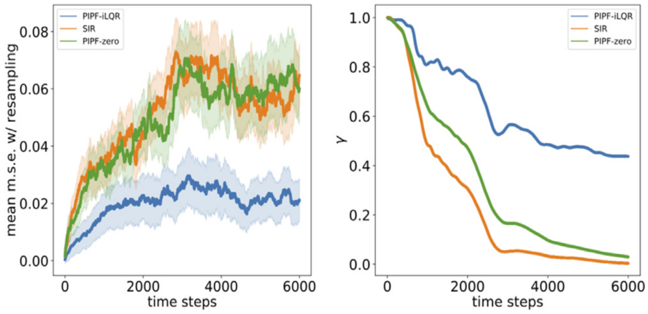

Fig. 9. Performance comparison between PIPF and SIR. PIPF-iLQR outperforms PIPF-zero and SIR in terms of m.s.e. and effective ratio.

5. Conclusion

In this paper, building on the duality between optimal estimation and optimal control theory, we developed a novel particle filtering algorithm. This algorithm has several distinguishing features compared with standard particle filtering algorithms, including a high effective sample size and the ability to update samples in the past to improve robustness. Our algorithm can also be combined with the most of the existing filtering algorithms such as the EKF and UKF to improve their performance. In the future, we plan to extend our approach to solve filtering problems involving diffusion processes with jumps.

References

Balaji, Bhashyam (2009). Continuous-discrete path integral filtering. *Entropy*, *11*(3), 402–430.

Bengtsson, Thomas, Bickel, Peter, & Li, Bo (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman* (pp. 316–334). Institute of Mathematical Statistics.

Bertsekas, Dimitri P. (1995). *Dynamic programming and optimal control, vol. 1, no. 2.* MA: Athena Scientific Belmont.

Beskos, Alexandros, Crisan, Dan O, Jasra, Ajay, & Whiteley, Nick (2014). Error bounds and normalising constants for sequential Monte Carlo samplers in high dimensions. *Advances in Applied Probability*, *46*(1), 279–306.

Daum, Fred, Huang, Jim, & Noushin, Arjang (2010). Exact particle flow for nonlinear filters. In *Signal processing, sensor fusion, and target recognition XIX, vol. 7697*. International Society for Optics and Photonics, Article 769704.

Davis, Richard A., Lii, Keh-Shin, & Politis, Dimitris N. (2011). Remarks on some nonparametric estimates of a density function. In *Selected works of Murray Rosenblatt* (pp. 95–100). Springer.

Del Moral, P. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de Sciences-Series*, *325*(6), 653–658.

Doucet, Arnaud, Briers, Mark, & Sénécal, Stéphane (2006). Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, *15*(3), 693–711.

Evans, Lawrence C. (1998). Partial differential equations. *Graduate Studies in Mathematics*, *19*(2).

Fleming, Wendell, & Rishel, Raymond (1975). *Deterministic and stochastic optimal control*. Springer.

Guarniero, Pieralberto, Johansen, Adam M., & Lee, Anthony (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association*, *112*(520), 1636–1647.

Heng, Jeremy, Bishop, Adrian N., Deligiannidis, George, & Doucet, Arnaud (2017). Controlled sequential Monte Carlo. arXiv preprint arXiv:1708.08396.

Jacobson, David H., & Mayne, David Q. (1970). *Differential dynamic programming*. North-Holland.

Kim, Jin Won, & Mehta, Prashant G. (2020). An optimal control derivation of nonlinear smoothing equations. In *Proceedings of the workshop on dynamics, optimization and computation held in honor of the 60th birthday of Michael Dellnitz* (pp. 295–311). Springer.

Klebaner, Fima C. (2005). *Introduction to stochastic calculus with applications*. World Scientific Publishing Company.

Le, Tuan Anh, Igl, Maximilian, Rainforth, Tom, Jin, Tom, & Wood, Frank (2018). Auto-encoding sequential Monte Carlo. In *International conference on learning representations*.

Lewis, Frank L., Vrabie, Draguna, & Syrmos, Vassilis L. (2012). *Optimal control*. John Wiley & Sons.

Li, Weiwei, & Todorov, Emanuel (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)* (pp. 222–229). Citeseer.

Mitter, Sanjoy K., & Newton, Nigel J. (2003). A variational approach to nonlinear estimation. *SIAM Journal on Control and Optimization*, *42*(5), 1813–1833.

Naesseth, Christian, Linderman, Scott, Ranganath, Rajesh, & Blei, David (2018). Variational sequential Monte Carlo. In *International conference on artificial intelligence and statistics* (pp. 968–977). PMLR.

Pitt, Michael K., & Shephard, Neil (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, *94*(446), 590–599.

Rauch, Herbert E., Tung, F., & Striebel, Charlotte T. (1965). Maximum likelihood estimates of linear dynamic systems. *American Institute of Aeronautics and Astronautics*, *3*(8), 1445–1450.

Reich, Sebastian (2018). Data assimilation: The Schrödinger perspective. arXiv preprint arXiv:1807.08351.

Ruiz, Hans-Christian, & Kappen, Hilbert J. (2017). Particle smoothing for hidden diffusion processes: Adaptive path integral smoother. *IEEE Transactions on Signal Processing*, *65*(12), 3191–3203.

Särkkä, Simo, & Solin, Arno (2019). *Applied stochastic differential equations, vol. 10*. Cambridge University Press.

Särkkä, Simo, & Sottinen, Tommi (2008). Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems. *Bayesian Analysis*, *3*(3), 555–584.

Taghvaei, Amirhossein, Mehta, Prashant G., & Meyn, Sean P. (2020). Diffusion map-based algorithm for gain function approximation in the feedback particle filter. *SIAM/ASA Journal on Uncertainty Quantification*, *8*(3), 1090–1117.

Thalmeier, Dominik, Kappen, Hilbert J., Totaro, Simone, & Gómez, Vicenç (2020). Adaptive smoothing path integral control. arXiv preprint arXiv:2005.06364.

Thijssen, Sep, & Kappen, H. J. (2015). Path integral control and state-dependent feedback. *Physical Review E*, *91*(3), Article 032104.

Williams, Grady, Aldrich, Andrew, & Theodorou, Evangelos A. (2017). Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, *40*(2), 344–357.

Yang, Tao, Mehta, Prashant G., & Meyn, Sean P. (2013). Feedback particle filter. *IEEE Transactions on Automatic Control*, *58*(10), 2465–2480.

Zhang, Wei, Wang, Han, Hartmann, Carsten, Weber, Marcus, & Schütte, Christof (2014). Applications of the cross-entropy method to importance sampling and optimal control of diffusions. *SIAM Journal on Scientific Computing*, *36*(6), A2654–A2672.

Qinsheng Zhang received his B.Sc. from Shanghai Jiao Tong University, Shanghai, China in 2019. Currently he is working on his Ph.D. degree in Robotics at Georgia Institute of Technology, Atlanta, United States. His research focuses on the intersection of robotics, optimization, and machine learning.

Amirhossein Taghvaei is an Assistant Professor in the William E. Boeing department of Aeronautics and Astronautics at the University of Washington (UW) Seattle. He obtained his Ph.D. degree from University of Illinois at Urbana-Champaign where he was a member of Decision and Control Lab (DCL) in the Coordinated Science Laboratory (CSL). He was a Postdoctoral Scholar with Tryphon Georgiou at University of California, Irvine before joining UW. His research interests lie at the intersection of control theory, machine learning, and physics, with the aim of designing scalable and reliable computational algorithms and understanding fundamental limitations.

Yongxin Chen received his B.Sc. from Shanghai Jiao Tong University in 2011 and Ph.D. from University of Minnesota in 2016, both in Mechanical Engineering. He is currently an Assistant Professor in the School of Aerospace Engineering at Georgia Institute of Technology. He has served on the faculty at Iowa State University (2017–2018). He received the George S. Axelby Best Paper Award in 2017 for his joint work with Tryphon Georgiou and Michele Pavon. He received the NSF Faculty Early Career Development Program (CAREER) Award in 2020. His current research focuses on the intersection of control theory, machine learning, robotics and optimization.