Solving Continuous-State POMDPs via Density Projection

Enlu Zhou, *Student Member, IEEE,* Michael C. Fu, *Fellow, IEEE,*
and Steven I. Marcus, *Fellow, IEEE,*

Abstract

Research on numerical solution methods for partially observable Markov decision processes (POMDPs) has primarily focused on discrete-state models, and these algorithms do not generally extend to continuous-state POMDPs, due to the infinite dimensionality of the belief space. In this paper, we develop a computationally viable and theoretically sound method for solving continuous-state POMDPs by effectively reducing the dimensionality of the belief space via density projections. The density projection technique is also incorporated into particle filtering to provide a filtering scheme for online decision making. We provide an error bound between the value function induced by the policy obtained by our method and the true value function of the POMDP, and also an error bound between projection particle filtering and exact filtering. Finally, we illustrate the effectiveness of our method through an inventory control problem.

I. Introduction

Partially observable Markov decision processes (POMDPs) model sequential decision making under uncertainty with partially observed state information. At each stage or period, an action is taken based on a partial observation of the current state along with the history of observations and actions, and the state transitions probabilistically. The objective is to minimize (or maximize) a cost (or reward) function, where costs (or rewards) are accrued in each stage. Clearly, POMDPs suffer from the same curse of

E. Zhou and S.I. Marcus are with the Department of Electrical and Computer Engineering, and Institute for Systems Research, University of Maryland, College Park, MD, 20742 USA e-mail: enluzhou@umd.edu, marcus@umd.edu.

M.C. Fu is with Robert H. Smith School of Business, and Institute for Systems Research, University of Maryland, College Park, MD, 20742 USA e-mail: mfu@umd.edu.

This work has been supported in part by the National Science Foundation under Grants DMI-0540312 and DMI-0323220, and by the Air Force Office of Scientific Research under Grant FA9550-07-1-0366.

dimensionality as fully observable MDPs, so efficient numerical solution of problems with large state spaces is a challenging research area.

A POMDP can be converted to a continuous-state Markov decision process (MDP) by introducing the notion of the belief state [4], which is the conditional distribution of the current state given the history of observations and actions. For a discrete-state model, the belief space is finite dimensional (i.e., a simplex), whereas for a continuous-state model, the belief space is an infinite dimensional space of continuous probability distributions. This difference suggests that simple generalizations of many of the discrete-state algorithms to continuous-state models are not appropriate or applicable. For example, discretization of the continuous-state space may result in a discrete-state POMDP of dimension either too huge to solve computationally or not sufficiently precise. Taking another example, many algorithms for solving discrete-state POMDPs (see [13] for a survey) are based on discretization of the finite-dimensional probability simplex; however, it is usually not feasible to discretize an infinite-dimensional probability distribution space.

Despite the abundance of algorithms for discrete-state POMDPs, the aforementioned difficulty has motivated some researchers to look for efficient algorithms for continuous-state POMDPs [20] [25] [22] [7]. Assuming discrete observation and action spaces, Portal et al. [20] showed that the optimal finite-horizon value function is defined by a finite set of "α-functions", and model all functions of interest by Gaussian mixtures. However, the number of Gaussian mixtures in representing belief states and α-functions grows exponentially in value iteration as the number of iterations increases. Thrun [25] addressed continuous-state POMDPs using particle filtering to simulate the propagation of belief states and represent the belief states by a finite number of samples. The number of samples determines the dimension of the belief space, and the dimension could be very high in order to approximate the belief states closely.

Roy [22] and Brooks et al. [7] used sufficient statistics to reduce the dimension of the belief space, which is often referred to as belief compression in the Artificial Intelligence literature. Roy [22] proposed an augmented MDP (AMDP), using maximum likelihood state and entropy to characterize belief states, which are usually not sufficient statistics except for the linear Gaussian model. As shown by Roy himself, the algorithm fails in a simple robot navigation problem, since the two statistics are not sufficient for distinguishing between a unimodal distribution and a bimodal distribution. Brooks et al. [7] proposed a parametric POMDP, representing the belief state as a Gaussian distribution with the parameters of mean and standard deviation, so as to convert the POMDP to a problem of computing the value function over a two-dimensional continuous space. The restriction to the Gaussian representation has the same problem as the AMDP. There are some other belief compression algorithms designed for discrete-state POMDP, such as value-directed compression [21] and the exponential family principle components analysis (E-PCA) belief compression [23]. They are not suitable for generalization to continuous-state models, since they are based on a fixed set of support points.

Motivated by the work of [25], [22], and [7], we develop a computationally tractable algorithm that effectively reduces the dimension of the belief state and has the flexibility to represent arbitrary belief states, such as multimodal or heavy tail distributions. The idea is to project the original high/infinite-dimensional belief space to a low-dimensional family of parameterized distributions by minimizing the Kullback-Leibler (KL) divergence between the belief state and its projection on that family of distributions. For an exponential family, the minimization of KL divergence can be carried out in analytical form, making the method very easy to implement. The belief MDP can then be solved on the parameter space by using simulation-based algorithms, or can be further approximated by a discrete-state MDP via a suitable discretization of the parameter space and thus solved by using standard solution techniques such as value iteration and policy iteration. Our method can be viewed as a generalization of the AMDP in [22] and the parametric POMDP in [7], where the exponential family is chosen to be the family of Gaussian distributions. In addition, we will provide theoretical results on the error bound of the value function and the performance of the near-optimal policy generated by our method.

We also develop a projection particle filter for online filtering and decision making, by incorporating the density projection technique into particle filtering. The projection particle filter we propose here is a modification of the projection particle filter in [2]. Unlike in [2] where the *predicted* conditional density is projected, we project the *updated* conditional density, so as to ensure the projected belief state remains in the given family of densities. Although seemingly a small modification in the algorithm, we prove under much less restrictive assumptions a similar bound on the error between our projection particle filter and the exact filter.

The rest of the paper is organized as follows. Section II describes the formulation of a continuous-state POMDP and its transformation to a belief MDP. Section III describes the density projection technique, and uses it to develop the projected belief MDP. Section IV develops the projection particle filter. Section V computes error bounds for the value function approximation and the projection particle filter. Section VI applies the method to a simulation example of an inventory control problem. Section VII concludes the paper.

II. Continuous-state POMDP

A discrete-time continuous-state POMDP can be formulated as a set of system equations and observation equations [4]:

$$x_{k+1} = f(x_k, a_k, u_k), k = 0, 1, \ldots, \tag{1}$$

$$y_k = h(x_k, a_{k-1}, v_k), k = 1, 2, \ldots, \qquad y_0 = h_0(x_0, v_0), \tag{2}$$

where for all k, the state x_k is in a continuous state space $S \in R^{n_x}$, the action a_k is in a finite action space $A \in R^{n_a}$, the observation y_k is in a continuous observation space $O \in R^{n_y}$, the random disturbances $\{u_k\} \in R^{n_x}$ and $\{v_k\} \in R^{n_y}$ are sequences of i.i.d. continuous random vectors with known distributions, and n_x, n_a and n_y are the dimensions of x_k, a_k and y_k, respectively. Assume that $\{u_k\}$ and $\{v_k\}$ are independent of each other, and independent of x_0, which follows a distribution p_0. Also assume that $f(x, a, u)$ is continuous in x for every $a \in A$ and $u \in R^{n_x}$, $h(x, a, v)$ is continuous in x for every $a \in A$ and $v \in R^{n_x}$, and $h_0(x, v)$ is continuous in x for every $v \in R^{n_x}$.

All the information available to the decision maker at time k can be summarized by means of an *information vector* I_k, which is defined as

$$\begin{aligned} I_k &= (y_0, y_1, \ldots, y_k, a_0, a_1, \ldots, a_{k-1}), k = 1, 2, \ldots, \\ I_0 &= y_0. \end{aligned}$$

The objective is to find a policy π consisting of a sequence of functions $\pi = \{\mu_0, \mu_1, \ldots\}$, where each function μ_k maps the information vector I_k onto the action space A, that minimizes the *value function*

$$J_\pi = E_{x_0, \{u_k\}, \{v_k\}} \left\{ \sum_{k=0}^{\infty} \gamma^k g(x_k, \mu_k(I_k)) \right\},$$

where $g : S \times A \to R$ is the *one-step cost function*, $\gamma \in (0, 1)$ is the *discount factor*, and $E_{x_0, \{u_k\}, \{v_k\}}$ denotes the expectation with respect to the joint distribution of x_0, $\{u_k\}$, and $\{v_k\}$. We assume g is bounded for all $(x, a) \in S \times A$. The *optimal value function* is defined by

$$J_* = \min_{\pi \in \Pi} J_\pi,$$

where Π is the set of all admissible policies. An *optimal policy*, denoted by π^*, is an admissible policy that achieves J_*. A *stationary policy* is an admissible policy of the form $\pi = \{\mu, \mu, \ldots\}$, referred to as the stationary policy μ for brevity, and its corresponding value function is denoted by J_μ.

The information vector I_k grows as the history expands. A well-known approach to encode historical information is the use of the *belief state*, which is the conditional probability density of the current state

x_k given the past history, i.e.,

$$b_k(\cdot) = p_k(\cdot|I_k),$$

Given our assumptions on (1) and (2), b_k exists. b_k can be computed recursively via Bayes' rule:

$$\begin{aligned} b_{k+1}(x_{k+1}) &= p(x_{k+1}|I_k, a_k, y_{k+1}) \\ &\propto p(y_{k+1}|x_{k+1}, a_k) \int_{x \in S} p(x_{k+1}|I_k, a_k, x) p(x|I_k, a_k) dx \\ &\propto p(y_{k+1}|x_{k+1}, a_k) \int_{x \in S} p(x_{k+1}|a_k, x) b_k(x) dx, k = 0, 1, \ldots, \\ b_0(x_0) &= p(x_0|y_0). \end{aligned} \tag{3}$$

The second line follows from the Markovian property. The third line follows from the Markovian property of $\{x_k\}$ and the fact that x_k does not depend on a_k. Hence, the evolution of b_k depends on a_k and y_{k+1}, summarized as

$$b_{k+1} = \psi(b_k, a_k, y_{k+1}), \tag{4}$$

where y_{k+1} is characterized by a conditional distribution $P_Y(y_{k+1}|b_k, u_k)$ (that does not depend on $\{y_0, \ldots, y_k\}$) induced by (1) and (2). Moreover, P_Y does not depend on time.

A POMDP can be converted to an MDP by conditioning on the information vectors, and the converted MDP is called the *belief MDP*. The states of the belief MDP are the belief states, which follow the system dynamics (4), where y_k can be seen as the system noise with the distribution P_Y. The state space of the belief MDP is the *belief space*, denoted by B, which is the set of all belief states, i.e., a set of probability densities. A policy π is a sequence of functions $\pi = \{\mu_0, \mu_1, \ldots\}$, where each function μ_k maps the belief state b_k into the action space A. Notice that

$$E_{x_0, \{u_i\}_{i=0}^k, \{v_i\}_{i=0}^k} \{g(x_k, a_k)\} = E\left\{E_{x_k}\{g(x_k, a_k)|I_k\}\right\},$$

thus the one-step cost function can be written in terms of the belief state as the *belief one-step cost function*

$$\begin{aligned} \tilde{g}(b_k, a_k) &\triangleq E_{x_k}\{g(x_k, a_k)|I_k\} \\ &= \int_{x \in S} g(x, a_k) b_k(x) dx \\ &\triangleq \langle g(\cdot, a), b \rangle. \end{aligned}$$

Assuming there exists a stationary optimal policy, we can apply value iteration to solve the belief MDP, that is, we apply the *dynamic programming (DP) mapping* to any bounded function $J : S \rightarrow R$. We denote it by

$$TJ(b) = \min_{a \in A} [\langle g(\cdot, a), b \rangle + \gamma E_Y\{J(\psi(b, a, Y))\}], \tag{5}$$

where E_Y denotes the expectation with respect to the distribution P_Y. The optimal value function is obtained by

$$J_*(b) = \lim_{k \to \infty} T^k J(b), \qquad \forall b \in B.$$

For finite-state POMDPs, the belief state b is a vector with each entry being the probability of being at one of the states. Hence, the belief space B is a finite-dimensional probability simplex, and the value function is a piecewise linear convex function after a finite number of iterations, provided that the one-step cost function is piecewise linear and convex [24]. This feature has been exploited in various exact and approximate value iteration algorithms such as those found in [13], [18], and [24] .

For continuous-state POMDPs, the belief state b is a continuous density, and thus, the belief space B is this infinite-dimensional space that contains all sorts of continuous densities. For continuous-state POMDPs, the value function preserves convexity [26], but value iteration algorithms are not directly applicable because the belief space is infinite dimensional. The infinite-dimensionality of the belief space also creates difficulties in applying the approximate algorithms that were developed for finite-state POMDPs. For example, one straightforward and commonly used approach is to approximate a continuous-state POMDP by a discrete-state one via discretization of the state space. In practice, this could lead to computational difficulties, either resulting in a belief space that is of huge dimension or in a solution that is not accurate enough. In addition, note that even for a relatively nice prior distribution b_k (e.g., a Gaussian distribution), the exact evaluation of the posterior distribution b_{k+1} is computationally intractable; moreover, the update b_{k+1} may not have any structure, and therefore can be very difficult to handle. Therefore, for practical reasons, we often wish to have a low-dimensional belief space and to have a posterior distribution b_{k+1} that stays in the same distribution family as the prior b_k.

To address the aforementioned difficulties, we apply the density projection technique to project the infinite-dimensional belief space onto a finite/low-dimensional parameterized family of densities, so as to derive a so-called projected belief MDP, which is an MDP with a finite/low-dimensional state space and therefore can be solved by many existing methods. In the next section, we describe density projection in details and develop the formulation of a projected belief MDP.

III. Projected Belief MDP

We define a *projection mapping* from the belief space B to a family of parameterized densities Ω, denoted as $Proj_\Omega : B \to \Omega$, by

$$Proj_\Omega(b) \triangleq \arg\min_{f \in \Omega} D_{KL}(b \| f), \qquad b \in B, \tag{6}$$

where $D_{KL}(b\|f)$ denotes the *Kullback-Leibler (KL) divergence* (or *relative entropy*) between b and f, which is

$$D_{KL}(b\|f) \triangleq \int \log \frac{b(x)}{f(x)} b(x)dx. \tag{7}$$

Hence, the projection of b on Ω has the minimum KL divergence from b among all the densities in Ω.

When Ω is an exponential family of densities, the minimization (6) has an analytical solution and can be carried out easily. The exponential families include many common families of densities, such as Gaussian, binomial, Poisson, Gamma, etc. An *exponential family of densities* is defined as follows:

Definition 1: Let $\{c_1(\cdot), \ldots, c_m(\cdot)\}$ be affinely independent scalar functions defined on R^n. Assuming that $\Theta_0 = \{\theta \in R^m : \varphi(\theta) = \log \int \exp(\theta^T c(x))dx < \infty\}$ is a convex set with a nonempty interior, where $c(x) = [c_1(x), \ldots, c_m(x)]^T$, then Ω defined by

$$\begin{aligned} \Omega &= \{f(\cdot, \theta), \theta \in \Theta\}, \\ f(x, \theta) &= \exp[\theta^T c(x) - \varphi(\theta)], \end{aligned}$$

where $\Theta \subseteq \Theta_0$ is open, is called *an exponential family of probability densities.* θ is called the natural parameter and $c(x)$ is the sufficient statistic of the probability density.

Substituting $f(x) = f(x, \theta)$ into (7) and expressing it further as

$$\begin{aligned} D_{KL}(b\|f(\cdot, \theta)) &= \int \log \frac{b(x)}{f(x, \theta)} b(x)dx \\ &= \int \log b(x)b(x)dx - \int \log f(x, \theta)b(x)dx, \end{aligned}$$

we can see that the first term does not depend on $f(\cdot, \theta)$, hence $\min D_{KL}(b\|f(\cdot, \theta))$ is equivalent to

$$\max \int \log f(x, \theta)b(x)dx,$$

which by Definition 1 is the same as

$$\max \int (\theta^T c(x) - \varphi(\theta))b(x)dx. \tag{8}$$

Recall the fact that the log-likelihood $l(\theta) = \theta^T c(x) - \varphi(\theta)$ is strictly concave in θ [17], and therefore, $\int (\theta^T c(x) - \varphi(\theta))b(x)dx$ is also strictly concave in θ. Hence, (8) has a unique maximum and the maximum is achieved when the first-order condition is satisfied, i.e.,

$$\int (c_j(x) - \frac{\int c_j(x) \exp(\theta^T c(x))dx}{\int \exp(\theta^T c(x))dx})b(x)dx = 0.$$

Therefore, b and its projection $f(\cdot, \theta)$ is related by

$$E_b[c_j(X)] = E_\theta[c_j(X)], j = 1, \ldots, m, \tag{9}$$

where E_b and E_θ denote the expectations with respect to b and $f(\cdot,\theta)$, respectively.

Density projection is a useful idea to approximate an arbitrary (most likely, infinite-dimensional) density as accurately as possible by a density in a chosen family that is characterized by only a few parameters. Using this idea, we can transform the belief MDP to another MDP confined on a low-dimensional belief space, and then solve this MDP problem. We call such an MDP the *projected belief MDP*. Its state is the *projected belief state* $b_k^p \in \Omega$ that satisfies the system dynamics

$$\begin{aligned} b_0^p &= Proj_\Omega(b_0), \\ b_{k+1}^p &= \psi(b_k^p, a_k, y_{k+1})^p, k = 0, 1, \ldots, \end{aligned}$$

where $\psi(b_k^p, a_k, y_{k+1})^p = Proj_\Omega(\psi(b_k^p, a_k, y_{k+1}))$, and the dynamic programming mapping on the projected belief MDP is

$$T^p J(b^p) = \min_{a\in A}[\langle g(\cdot,a), b^p\rangle + \gamma E_Y\{J(\psi(b^p,a,Y)^p)\}]. \tag{10}$$

For the projected belief MDP, a policy is denoted as $\pi^p = \{\mu_0^p, \mu_1^p, \ldots\}$, where each function μ_k^p maps the projected belief state b_k^p into the action space A. Similarly, a stationary policy is denoted as μ^p; an optimal stationary policy is denoted as μ_*^p; and the optimal value function is denoted as $J_*^p(b^p)$.

The projected belief MDP is in fact a low-dimensional continuous-state MDP, and can be solved in numerous ways. One common approach is to use value iteration or policy iteration by converting the projected belief MDP to a discrete-state MDP problem via a suitable discretization of the projected belief space (i.e., the parameter space) and then estimating the one-step cost function and transition probabilities on the discretized mesh. We describe this approach in detail below.

Discretization of the projected belief space Ω is equivalent to discretization of the parameter space Θ, which yields a set of grid points, denoted by $G = \{\theta_i, i = 1, \ldots, N\}$. Let $\tilde{g}(\theta_i, a)$ denote the one-step cost function associated with taking action a at the projected belief state $b^p = f(\cdot, \theta_i)$. Let $\tilde{P}(\theta_i, a)(\theta_j)$ denote the transition probability from the current projected belief state $b_k^p = f(\cdot,\theta_i)$ to the next projected belief state $b_{k+1}^p = f(\cdot,\theta_j)$ by taking action a. $\tilde{g}(\theta_i, a)$ and $\tilde{P}(\theta_i, a)(\theta_j)$ can be estimated via Monte-Carlo simulation as follows:

Algorithm 1: Estimation of the one-step cost function $\tilde{g}(\theta_i, a)$.

- Input: θ_i, a, N; Output: $\tilde{g}(\theta_i, a)$.
- Step 1: Sampling. Sample $x_1, \ldots, x_N$ i.i.d. from $p(\cdot, \theta_i)$.
- Step 2: Estimation. $\tilde{g}(\theta_i, a) = \frac{1}{N}\sum_{i=1}^N g(x_i, a)$.

The following algorithm is adapted from projection particle filtering (Algorithm 4) described in the next section, so we omit the explanation of the steps here.

Algorithm 2: Estimation of the transition probabilities $\tilde{P}(\theta_i, a)(\theta_j), j = 1, \ldots, N$.

- Input: θ_i, a, N; Output: $\tilde{P}(\theta_i, a)(\theta_j), j = 1, \ldots, N$.
- Step 1. Sampling: Sample $x_1, \ldots, x_N$ from $f(\cdot, \theta_i)$.
- Step 2. Prediction: Compute $\tilde{x}_1, \ldots, \tilde{x}_N$ by propagating $x_1, \ldots, x_N$ according to the system dynamics (1) using the action a and randomly generated noise $\{u_i\}_{i=1}^N$.
- Step 3. Sampling observation: Compute $y_1, \ldots, y_N$ from $\tilde{x}_1, \ldots, \tilde{x}_N$ according to the observation equation (2) using randomly generated noise $\{v_i\}_{i=1}^N$.
- Step 4. Bayes' updating: For each $y_k, k = 1, \ldots, N$, the updated belief state is
$$\tilde{b}_k = \sum_{i=1}^{N} w_i^k \delta(x - \tilde{x}_i),$$
where
$$w_i^k = \frac{p(y_k|\tilde{x}_i, a)}{\sum_{i=1}^N p(y_k|\tilde{x}_i, a)}, i = 1, \ldots, N.$$
- Step 5. Projection: For $k = 1, \ldots, N$, project each $\tilde{b}_k$ to the exponential family, i.e., finding $\tilde{\theta}_k$ that satisfies (9).
- Step 6. Estimation: For $k = 1, \ldots, N$, find the nearest-neighbor of $\tilde{\theta}_k$ in G. For each $\theta_j \in G$, count the frequency $\tilde{P}(\theta_i, a)(\theta_j) =$ (number of $\theta_j)/N$.

Remark 1: The approach for solving the projected belief MDP described here is probably the most intuitive, but not necessarily the most computationally efficient. Other more efficient techniques for solving continuous-state MDPs can be used to solve the projected belief MDP, such as the linear programming approach [12], neuro-dynamic programming methods [5], and simulation-based methods [9].

IV. Projection Particle Filtering

Solving the projected belief MDP gives us a near-optimal policy, which tells us what action to take at each projected belief state. In an online implementation, at each time k, the decision maker receives a new observation y_k, estimates the belief state b_k, and then chooses his action a_k according to b_k and the near-optimal policy. Hence, to make our approach work properly for real-life applications, it is also important to address the problem of how to estimate the belief state. Estimation of b_k, or simply called *filtering*, does not have an analytical solution in most cases except linear Gaussian systems, but it can be solved using many approximation methods, such as the extended Kalman filter and particle filtering. Here we focus on particle filtering, for 1) it outperforms the extended Kalman filter in many nonlinear/non-Gaussian systems [1], and 2) we will develop a projection particle filter in particular to be used in conjunction with the projected belief MDP.

A. Particle Filtering

Particle filtering is a Monte Carlo simulation-based method that approximates the belief state by a finite number of particles/samples and mimics the propagation of the belief state [1] [11]. As we have already shown in (3), the belief state evolves recursively as

$$b_k(x_k) \propto p(y_k|x_k, a_{k-1}) \int p(x_k|a_{k-1}, x_{k-1}) b_{k-1}(x_{k-1}) dx_{k-1}. \tag{11}$$

The integration in (11) can be approximated using Monte Carlo simulation, which is the essence of particle filtering. Specifically, suppose $\{x_{k-1}^i\}_{i=1}^N$ are drawn i.i.d. from b_{k-1}, and $x_{k|k-1}^i$ is drawn from $p(x_k|a_{k-1}, x_{k-1}^i)$ for each i; then $b_k(x_k)$ can be approximated by the probability mass function

$$\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_{k|k-1}^i), \tag{12}$$

where

$$w_k^i \propto p(y_k|x_{k|k-1}^i, a_{k-1}), \tag{13}$$

δ denotes the Kronecker delta function, $\{x_{k|k-1}^i\}_{i=1}^N$ are the random support points, and $\{w_k^i\}_{i=1}^N$ are the associated probabilities/weights which sum up to 1.

To avoid sample degeneracy, new samples $\{x_k^i\}_{i=1}^N$ are sampled i.i.d. from the approximate belief state $\hat{b}_k$. At the next time $k+1$, the above steps are repeated to yield $\{x_{k+1|k}^i\}_{i=1}^N$ and corresponding weights $\{w_{k+1}^i\}_{i=1}^N$, which are used to approximate b_{k+1}. It is the basic form of particle filtering, which is also called the bootstrap filter [14]. (Please see [1] for a more rigorous and thorough derivation for a more general form of particle filtering.) The algorithm is as follows:

Algorithm 3: Particle Filtering (Bootstrap Filter).

- Input: a (stationary) policy μ on the belief MDP; a sequence of observations $y_1, y_2, \ldots$ arriving sequentially at time $k = 1, 2, \ldots$. Output: a sequence of approximate belief states $\hat{b}_1, \hat{b}_2, \ldots$.
- Step 1. Initialization: Sample $x_0^1, \ldots, x_0^N$ i.i.d. from the approximate initial belief state $\hat{b}_0$. Set $k = 1$.
- Step 2. Prediction: Compute $x_{k|k-1}^1, \ldots, x_{k|k-1}^N$ by propagating $x_{k-1}^1, \ldots, x_{k-1}^N$ according to the system dynamics (1) using the action $a_{k-1} = \mu(\hat{b}_{k-1})$ and randomly generated noise $\{u_{k-1}^i\}_{i=1}^N$, i.e., sample $x_{k|k-1}^i$ from $p(\cdot|x_{k-1}^i, a_{k-1})$, $i = 1, \ldots, N$. The empirical predicted belief state is
$$\hat{b}_{k|k-1}(x) = \frac{1}{N} \sum_{i=1}^N \delta(x - x_{k|k-1}^i).$$
- Step 3. Bayes' updating: Receive a new observation y_k. The empirical updated belief state is
$$\hat{b}_k(x) = \sum_{i=1}^N w_k^i \delta(x - x_{k|k-1}^i),$$

where

$$w_k^i = \frac{p(y_k|x_{k|k-1}^i, a_{k-1})}{\sum_{i=1}^N p(y_k|x_{k|k-1}^i, a_{k-1})}, i = 1, \ldots, N.$$

- Step 4. Resampling: Sample $x_k^1, \ldots, x_k^N$ i.i.d. from $\hat{b}_k$.
- Step 5. $k \leftarrow k + 1$ and go to step 2.

It has been proved that the approximate belief state $\hat{b}_k$ converges to the true belief state b_k in certain sense as the sample number N increases to infinity [10] [16]. However, uniform convergence in time has only been proved for the special case, where the system dynamics has a mixing kernel which ensures that any error is forgotten (exponentially) in time. Usually, as time k increases, an increasing number of samples is required to ensure a given precision of the approximation $\hat{b}_k$ for all k.

B. Projection Particle Filtering

To get a reasonable approximation of the belief state, particle filtering needs a large amount of samples/particles. Since the number of samples/particles is the dimensionality of the approximate belief state $\hat{b}$, particle filtering is not very helpful in reducing the dimensionality of the belief space. Moreover, the near-optimal policy we obtained by solving the projected belief MDP is a function on the projected belief space Ω, and hence, the policy is *immediately* applicable if the approximate belief state is in Ω.

We incorporate the idea of density projection into particle filtering, so as to approximate the belief state by a density in Ω. The projection particle filter we propose here is a modification of the one in [2]. Their projection particle filter projects the empirical *predicted* belief state, not the empirical *updated* belief state, onto a parametric family of densities, so after Bayes' updating, the approximate belief state might not be in that family. We will the project empirical *updated* belief state onto a parametric family by minimizing the KL divergence between the empirical density and the projected one. In addition, we will need much less restrictive assumptions than [2] to obtain similar error bounds. Since resampling is from a continuous distribution instead of an empirical (discrete) one, the proposed projection particle filter also overcomes the difficulty of sample impoverishment [1] that occurs in bootstrap filter.

Applying the density projection technique we described in last section, projecting the empirical belief state $\hat{b}_k$ onto an exponential family Ω is to find a $f(\cdot, \theta)$ with the parameter θ satisfying (9). Hence, letting $b = \hat{b}_k$ in (9) and plugging in (12), θ should satisfy

$$\sum_{i=1}^N w_i c_j(x_{k|k-1}^i) = E_\theta[c_j], j = 1, \ldots, m. \tag{14}$$

(14) constitutes the projection step in the projection particle filtering.

Algorithm 4: Projection particle filtering for an exponential family of densities (PPF).

- Input: a (stationary) policy μ^p on the projected belief MDP; a family of exponential densities $\Omega = \{f(\cdot,\theta), \theta \in \Theta\}$; a sequence of observations $y_1, y_2, \ldots$ arriving sequentially at time $k = 1, 2, \ldots$. Output: a sequence of approximate belief states $f(\cdot,\hat{\theta}_1), f(\cdot,\hat{\theta}_2), \ldots$.
- Step 1. Initialization: Sample $x_0^1, \ldots, x_0^N$ i.i.d. from the approximate initial belief state $f(\cdot,\hat{\theta}_0)$. Set $k = 1$.
- Step 2. Prediction: Compute $x_{k|k-1}^1, \ldots, x_{k|k-1}^N$ by propagating $x_{k-1}^1, \ldots, x_{k-1}^N$ according to the system dynamics (1) using the action $a_{k-1} = \mu^p(f(\cdot,\hat{\theta}_{k-1}))$ and randomly generated noise $\{u_{k-1}^i\}_{i=1}^N$, i.e., sample $x_{k|k-1}^i$ from $p(\cdot|x_{k-1}^i, a_{k-1})$, $i = 1, \ldots, N$.
- Step 3. Bayes' updating: Receive a new observation y_k. Calculate weights as
$$w_k^i = \frac{p(y_k|x_k^i, a_{k-1})}{\sum_{i=1}^N p(y_k|x_k^i, a_{k-1})}, i = 1, \ldots, N.$$
- Step 4. Projection: The approximate belief state is $f(\cdot,\hat{\theta}_k)$, where $\hat{\theta}_k$ satisfies the equations
$$\sum_{i=1}^N w_k^i c_j(x_{k|k-1}^i) = E_{\hat{\theta}_k}[c_j], j = 1, \ldots, m.$$
- Step 5. Resampling: Sample $x_k^1, \ldots, x_k^N$ from $f(\cdot,\hat{\theta}_k)$.
- Step 6. $k \leftarrow k+1$ and go to Step 2.

In an online implementation, at each time k, PPF approximates b_k by $f(\cdot,\hat{\theta}_k)$, and then decides an action a_k according to $a_k = \mu^p(f(\cdot,\hat{\theta}_k))$, where μ^p is the near-optimal policy solved for the projected belief MDP.

V. Analysis of Error Bounds

A. Value Function Approximation

Our method solves the projected belief MDP instead of the original belief MDP, and that raises two questions: How well does the optimal value function of the projected belief MDP approximate the optimal value function of the original belief MDP? How well does the optimal policy obtained by solving the projected belief MDP perform on the original belief MDP? To answer these questions, we first need to rephrase them mathematically.

Here we assume perfect computation of the belief states and the projected belief states. We also assume the stationarity of optimal policies as stated below.

Assumption 1: There is a stationary optimal policy for the belief MDP, denoted by μ_*, and a stationary optimal policy for the projected belief MDP, denoted by μ_*^p.

Assumption 1 holds under some mild conditions [4], [15]. Using the stationarity, and the dynamic programming mapping on the belief MDP and the projected belief MDP given by (5) and (10), the optimal value function $J_*(b)$ for the belief MDP can be obtained by

$$J_*(b) \triangleq J_{\mu_*}(b) = \lim_{k\to\infty} T^k J_0(b),$$

and the optimal value function for the projected belief MDP obtained by

$$J_*^p(b^p) \triangleq J_{\mu_*^p}^p(b^p) = \lim_{k\to\infty} (T^p)^k J_0(b^p).$$

Therefore, the questions posed at the beginning of this section can be formulated mathematically as:

1. How well the optimal value function of the projected belief MDP approximates the true optimal value function can be measured by

$$|J_*(b) - J_*^p(b^p)|.$$

2. How well the optimal policy μ_*^p for the projected belief MDP performs on the original belief space can be measured by

$$|J_*(b) - J_{\bar{\mu}_*^p}(b)|,$$

where $\bar{\mu}_*^p(b) \triangleq \mu_*^p \circ Proj_\Omega(b) = \mu_*^p(b^p)$.

The next assumption assumes bounds on the difference between the belief state b and its projection b^p, and also the difference between their one-step evolutions $\psi(b,a,y)$ and $\psi(b^p,a,y)^p$. It is an assumption on the projection error.

Assumption 2: There exist $\epsilon_1 > 0$ *and* $\delta_1 > 0$ *such that for all* $a \in A, y \in O$ *and* $b \in B$,

$$|\langle g(\cdot,a), b - b^p\rangle| \le \epsilon_1,$$

$$|\langle g(\cdot,a), \psi(b,a,y) - \psi(b^p,a,y)^p\rangle| \le \delta_1.$$

The following assumption can be seen as a continuity property of the value function.

Assumption 3: For all $b, b' \in B$, *if* $|\langle g(\cdot,a), b - b'\rangle| \le \delta$, *then there exists* $\epsilon > 0$ *such that* $|J_k(b) - J_k(b')| \le \epsilon, \forall k$, *and there exists* $\tilde{\epsilon} > 0$ *such that* $|J_\mu(b) - J_\mu(b')| \le \tilde{\epsilon}, \forall \mu \in \Pi$.

Now we present our main result.

Theorem 1: Under Assumptions 1, 2 and 3, for all $b \in B$,

$$|J_*(b) - J_*^p(b^p)| \le \frac{\epsilon_1 + \gamma\epsilon_2}{1-\gamma}, \quad (15)$$

$$|J_*(b) - J_{\bar{\mu}_*^p}(b)| \le \frac{2\epsilon_1 + \gamma(\epsilon_2 + \epsilon_3)}{1-\gamma}, \quad (16)$$

where ϵ_1 is the constant in Assumption 2, and ϵ_2, ϵ_3 are the constants ϵ and $\tilde{\epsilon}$, respectively, in Assumption 3 corresponding to $\delta = \delta_1$.

Remark 2: In (15) and (16), ϵ_1 is a projection error, and ϵ_2 and ϵ_3 are both due to the projection error δ_1. Therefore, as the projection error decreases, $J_*^p(b^p)$ converges to the optimal value function $J_*(b)$, and $\bar{\mu}_*^p$ converges to the optimal policy μ_*. Roughly speaking, the projection error decreases as the number of sufficient statistics in the chosen exponential family increases (for a rigorous result, please see [3]).

B. Projection Particle Filtering

In the above analysis, we assumed perfect computation of the belief states and the projected belief states. In this section, we consider the filtering error, and compute an error bound on the approximate belief state generated by the projection particle filter (PPF).

1) Notations: Let $C_b(R^n)$ be the set of all continuous bounded functions on R^n. Let $B(R^n)$ be the set of all bounded measurable functions on R^n. Let $\|\cdot\|$ denote the supremum norm on $B(R^n)$, i.e., $\|\phi\| \triangleq \sup_{x\in R^n} |\phi(x)|, \phi \in B(R^n)$. Let $\mathcal{M}^+(R^n)$ and $\mathcal{P}(R^n)$ be the sets of nonnegative measures and probability measures on R^n, respectively. If $\eta \in \mathcal{M}^+(R^n)$ and $\phi : R^n \to R$ is an integrable function with respect to η, then

$$\langle \eta, \phi \rangle \triangleq \int \phi d\eta.$$

Moreover, if $\eta \in \mathcal{P}(R^n)$,

$$\begin{aligned} E_\eta[\phi] &= \langle \eta, \phi \rangle, \\ Var_\eta(\phi) &= \langle \eta, \phi^2 \rangle - \langle \eta, \phi \rangle^2. \end{aligned}$$

We will use the two representations on the two sides of the above equalities interchangeably in the sequel.

The belief state and the projected belief state are probability densities; however, we will prove our results in terms of their corresponding probability measures, which we refer as “conditional distributions” (belief states are conditional densities). The two representations are essentially the same once we assume the probability measures admit probability densities. Therefore, from now on we use the same notations for probability densities before to denote the probability measures. Namely, we use b denote a probability measure on R^{n_x} and assume it admits a probability density, which is the belief state, with respect to Lebesgue measure. Similarly, we use $f(\cdot, \theta)$ to denote a probability measure on R^{n_x} and assume it admits a probability density with respect to Lebesgue measure in the chosen exponential family with parameter θ.

A probability transition kernel $K : \mathcal{P}(R^{n_x}) \times R^{n_x} \rightarrow R$ is defined by

$$K\eta(E) \triangleq \int_{R^{n_x}} \eta(dx)K(E,x),$$

where E is a set in the Borel σ-algebra on R^{n_x}. For $\phi : R^{n_x} \rightarrow R$, an integrable function with respect to $K(\cdot, x)$,

$$K\phi(x) \triangleq \int_{R^{n_x}} \phi(x')K(dx', x).$$

Let $K_k(dx_k, x_{k-1})$ denote the probability transition kernel of the system (1) at time k, which satisfies

$$b_{k|k-1}(dx_k) = K_k b_{k-1}(dx_{k|k-1}) = \int_{R^{n_x}} b_{k-1}(dx_{k-1})K_k(dx_{k|k-1}, x_{k-1}).$$

We let Ψ_k denote the likelihood function associated with the observation equation (2) at time k, and assume that $\Psi_k \in C_b(R^{n_x})$. Hence,

$$b_k = \frac{\Psi_k b_{k|k-1}}{\langle b_{k|k-1}, \Psi_k \rangle}.$$

2) Main Idea: The exact filter (EF) at time k can be described as

$$b_{k-1} \quad \underset{prediction}{\longrightarrow} \quad b_{k|k-1} = K_k b_{k-1} \quad \underset{updating}{\longrightarrow} \quad b_k = \frac{\Psi_k b_{k|k-1}}{\langle b_{k|k-1}, \Psi_k \rangle}.$$

The PPF at time k can be described as

$$\hat{f}(\cdot, \hat{\theta}_{k-1}) \quad \underset{prediction}{\longrightarrow} \quad \hat{b}_{k|k-1} = K_k f(\cdot, \hat{\theta}_{k-1}) \quad \underset{updating}{\longrightarrow} \quad \hat{b}_k = \frac{\Psi_k \hat{b}_{k|k-1}}{\langle \hat{b}_{k|k-1}, \Psi_k \rangle} \quad \underset{projection}{\longrightarrow} \quad f(\cdot, \hat{\theta}_k) \quad \underset{resampling}{\longrightarrow} \quad \hat{f}(\cdot, \hat{\theta}_k).$$

To facilitate our analysis, we introduce a conceptual filter (CF), which at each time k is reinitialized by $f(\cdot, \hat{\theta}_{k-1})$, performs exact prediction and updating to yield $b'_{k|k-1}$ and b'_k, respectively, and does projection to get $f(\cdot, \theta'_k)$. It can be described as

$$f(\cdot, \hat{\theta}_{k-1}) \quad \underset{prediction}{\longrightarrow} \quad b'_{k|k-1} = K_k f(\cdot, \hat{\theta}_{k-1}) \quad \underset{updating}{\longrightarrow} \quad b'_k = \frac{\Psi_k b'_{k|k-1}}{\langle b'_{k|k-1}, \Psi_k \rangle} \quad \underset{projection}{\longrightarrow} \quad f(\cdot, \theta'_k).$$

The CF serves as an bridge to connect the EF and PPF, as we describe below.

We are interested in the difference between the true conditional distribution b_k and the PPF generated projected conditional distribution $f(\cdot, \hat{\theta}_k)$ for each time k. The difference between the two can be decomposed as follows:

$$b_k - f(\cdot, \hat{\theta}_k) = (b_k - b'_k) + (b'_k - f(\cdot, \theta'_k)) + (f(\cdot, \theta'_k) - f(\cdot, \hat{\theta}_k)). \tag{17}$$

The first term $(b_k - b'_k)$ is the error due to the inexact initial condition of the CF, i.e., $(b_{k-1} - f(\cdot, \hat{\theta}_{k-1})$, which is also the total error at time $k-1$. The second term $(b'_k - f(\cdot, \theta'_k))$ evaluates the minimum deviation from the exponential family generated by one step of exact filtering, since $f(\cdot, \theta'_k)$ is the orthogonal projection of b'_k. The third term $(f(\cdot, \theta'_k) - f(\cdot, \hat{\theta}_k))$ is purely due to Monte Carlo simulation, since $f(\cdot, \theta'_k)$ and $f(\cdot, \hat{\theta}_k)$ are obtained using the same steps from $f(\cdot, \hat{\theta}_{k-1})$ and its empirical version $\hat{f}(\cdot, \hat{\theta}_{k-1})$, respectively. We will find error bounds on each of the three terms respectively, and finally find the total error at time k by induction.

3) Error Bound: We shall look at the the case in which the observation process has an arbitrary but fixed value $y_{0:k} = \{y_0, \ldots, y_k\}$. Hence, all the expectations E in this section are with respect to the sampling in the algorithm only. We consider the test function $\phi \in B(R^{n_x})$. It can be seen that $K\phi \in B(R^{n_x})$ and $\|K\phi\| \leq \|\phi\|$, since

$$\begin{aligned} |K\phi(x)| &= |\int_{R^{n_x}} \phi(x')K(dx', x)| \\ &\leq \int_{R^{n_x}} |\phi(x')K(dx', x)| \\ &\leq \|\phi\| \int_{R^{n_x}} K(dx', x) \\ &= \|\phi\|. \end{aligned}$$

Since $\Psi \in C_b(R^{n_x})$, we know that $\Psi \in B(R^{n_x})$ and $\Psi\phi \in B(R^{n_x})$.

We also need the following assumptions.

Assumption 4: All the projected distributions are in a compact subset of the given exponential family. In other words, there exists a compact set Θ' such that $\hat{\theta}_k \in \Theta'$, and $\theta'_k \in \Theta'$, $\forall k$.

Assumption 5: For all $k \in \mathbb{N}$,

$$\begin{aligned} \langle b_{k|k-1}, \Psi_k \rangle &> 0, \\ \langle b'_{k|k-1}, \Psi_k \rangle &> 0, \quad w.p.1, \\ \langle \hat{b}_{k|k-1}, \Psi_k \rangle &> 0, \quad w.p.1. \end{aligned}$$

Remark 3: Assumption 5 is to guarantee that the normalizing constant in the Bayes' updating is nonzero, so that the conditional distribution is well defined. Under Assumption 4, the second inequality in Assumption 5 can be strengthened using the compactness of Θ'. Since $f(\cdot, a_k, u_k)$ in (1) is continuous in x, K_k is weakly continuous (pp. 175-177, [15]). Hence, $\langle b'_{k|k-1}, \Psi_k \rangle = \langle K_k f(\cdot, \hat{\theta}_{k-1}), \Psi_k \rangle = \langle f(\cdot, \hat{\theta}_{k-1}), K_k \Psi_k \rangle$ is continuous in $\hat{\theta}_{k-1}$, where $\hat{\theta}_{k-1} \in \Theta'$. Since Θ' is compact, there exists a constant

$\delta > 0$ such that for each k

$$\langle b'_{k|k-1}, \Psi_k \rangle \geq \frac{1}{\delta}, \quad w.p.1. \tag{18}$$

The assumption below is to guarantee that the conditional distribution stays close to the given exponential family after one step of exact filtering if the initial distribution is in the exponential family. Recall that starting with initial distribution $f(\cdot, \hat{\theta}_{k-1})$, one step of exact filtering yields b'_k, which is then projected to yield $f(\cdot, \theta'_k)$, where $\hat{\theta}_{k-1} \in \Theta', \theta'_k \in \Theta'$.

Assumption 6: For all $\phi \in B(R^{n_x})$ and all $k \in \mathbb{N}$, there exists a constant $\epsilon > 0$ such that

$$E[|\langle b'_k, \phi \rangle - \langle f(\cdot, \theta'_k), \phi \rangle|] \leq \epsilon \|\phi\|.$$

Remark 4: Assumption 6 is our main assumption, which essentially assumes an error bound on the projection error. Our assumptions are much less restrictive than the assumptions in [2], while our conclusion is similar to but slightly different from that in [2], which will be seen later. Although Assumption 6 appears similar to Assumption 3 in [2], it is essentially different. Assumption 3 in [2] says that the optimal conditional density stays close to the given exponential family for *all* time, whereas Assumption 6 only assumes that if the exact filter starts in the given exponential family, after *one* step the conditional distribution stays close to the family. Moreover, we do not need any assumption like the restrictive Assumption 4 in [2].

Lemma 1 considers the bound on the first term $(b_k - b'_k)$ in (17).

Lemma 1: For all $\phi \in B(R^{n_x})$ and all $k \in \mathbb{N}$, suppose $E[|\langle b_{k-1} - f(\cdot, \hat{\theta}_{k-1}), \phi \rangle|] \leq e_{k-1}\|\phi\|$, where e_{k-1} is a positive constant. Then under Assumptions 4 and 5, for all $\phi \in B(R^{n_x})$ and all $k \in \mathbb{N}$, there exists a constant $a_k > 0$ such that

$$E[|\langle b_k - b'_k, \phi \rangle|] \leq a_k e_{k-1} \|\phi\|. \tag{19}$$

Lemma 2 considers the bound on the third term in (17) before projection, i.e., $(\langle \hat{b}_k, \phi \rangle - \langle b'_k, \phi \rangle)$.

Lemma 2: Under Assumptions 3 and 5, for all $\phi \in B(R^{n_x})$ and all $k \in \mathbb{N}$, there exists a constant $\tau_k > 0$ such that

$$E[|\langle \hat{b}_k - b'_k, \phi \rangle|] \leq \tau_k \frac{\|\phi\|}{\sqrt{N}}.$$

Lemma 3 considers the bound on the third term in (17) based on the result of Lemma 2. The key idea of proof is to connect the errors before and after projection through (9), which we derived for the density projection that minimizes the KL divergence.

Lemma 3: Let $c_j, j = 1, \ldots, m$ be the sufficient statistics of the exponential family as defined in Definition 1, and assume $c_j \in B(R^{n_x}), j = 1, \ldots, m$. Then under Assumptions 4 and 5, for all $\phi \in B(R^{n_x})$ and all $k \in \mathbb{N}$, there exists a constant $d_k > 0$ such that

$$E[|f(\cdot, \hat{\theta}_k) - \langle f(\cdot, \theta'_k), \phi\rangle|] \leq d_k \frac{\|\phi\|}{\sqrt{N}}. \tag{20}$$

Now we present our main result on the error bound of the projection particle filter.

Theorem 2: For all $\phi \in B(R^{n_x})$, suppose $E[|\langle b_0 - f(\cdot, \hat{\theta}_0), \phi\rangle] \leq e_0\|\phi\|, e_0 \geq 0$. Under Assumptions 4, 5 and 6, and assuming that $c_j \in B(R^{n_x}), j = 1, \ldots, m$, there exist $a_i > 0, d_i > 0, i = 1, \ldots, k$ such that for all $\phi \in B(R^{n_x})$ and all $k \in \mathbb{N}$,

$$E[|\langle b_k - f(\cdot, \hat{\theta}_k), \phi\rangle|] \leq e_k\|\phi\|, k = 1, 2, \ldots,$$

where

$$e_k = a_1^k e_0 + (\sum_{i=2}^{k} a_i^k + 1)\epsilon + (\sum_{i=2}^{k} a_i^k d_{i-1} + d_k)\frac{1}{\sqrt{N}}, \tag{21}$$

$a_i^k = \prod_{j=i}^{k} a_j$ for $k \geq i$, and $a_i^k = 0$ for $k < i$, ϵ is the constant in Assumption 6.

Remark 5: As we mentioned in Remark 2, the projection error e_0 and ϵ decreases as the number of sufficient statistics in the chosen exponential family, m, increases. The error e_k decreases at the rate of $\frac{1}{\sqrt{N}}$, as we increase the number of samples in the projection particle filter. However, notice that the coefficient in front of $\frac{1}{\sqrt{N}}$ grows as time, so we have to use an increasing number of samples as time goes on, in order to ensure a uniform error bound with respect to time.

VI. Numerical Experiments

We consider an inventory control problem, where the inventory level is reviewed at discrete times, but the observations are noisy because of, e.g., inventory spoilage, misplacement, distributed storage. At each period, inventory is either replenished by an order of a fixed amount or not replenished. The customer demands arrive randomly with known distribution. The demand is filled if there is enough inventory remaining. Otherwise, in the case of a shortage, excess demand is not satisfied and a penalty is issued on the lost sales amount. We assume that the demand and the observation noise are both continuous random variables; hence the state, i.e., the inventory level, and the observation, are continuous random variables.

Let x_k denote the inventory level at period k, u_k the i.i.d. random demand at period k, a_k the replenish decision at period k (i.e., $a_k = 0$ or 1), Q the fixed order amount, y_k the observation of inventory level

x_k, v_k the i.i.d. observation noise, h the per period per unit inventory holding cost, s the per period per unit inventory shortage penalty cost. The system equations are as follows

$$\begin{aligned} x_{k+1} &= \max(x_k + a_k Q - u_k, 0), k = 0, 1, \ldots, \\ y_k &= x_k + v_k, k = 0, 1, \ldots. \end{aligned}$$

The cost incurred in period k is

$$g_k(x_k, a_k, u_k) = h \max\left(x_k + a_k Q - u_k, 0\right) + s \max\left(u_k - x_k - a_k Q, 0\right).$$

We consider two objective functions: average cost per period and discounted total cost, given by

$$\lim_{H \to \infty} \frac{\sum_{i=0}^{H} E[g_i]}{H},$$

$$\lim_{H \to \infty} \sum_{i=0}^{H} \gamma^i E[g_i],$$

where $\gamma \in (0, 1)$ is the discount factor.

We compare our algorithm to three other algorithms: (1) Certainty equivalence (CE) using the mean estimate; (2) Certainty equivalence using the maximum likelihood estimate (MLE); (3) Greedy policy. Our algorithm solves the projected belief MDP via value iteration to yield a policy, and then uses the policy to determine the action a_k online according to the filtered belief state obtained using the projection particle filter. CE solves the full observation problem first and treats the state estimate as the true state in the solution to the full observation problem. We use the bootstrap filter to obtain the mean estimate and the MLE of the states for CE. The greedy policy chooses an action a_k that attains the minimum in the expression

$$\min_{a_k \in A} E_{x_k, u_k}[g_k(x_k, a_k Q, u_k) | I_k].$$

Numerical experiments are carried out in the following settings:

- *Problem parameters*: initial inventory level $x_0 = 5$, holding cost $h = 1$, shortage penalty cost $s = 10$, fixed order amount $b = 10$, random demand $u_k \sim exp(5)$, discount factor $\gamma = 0.9$, inventory observation noise $v_k \sim N(0, \sigma^2)$ with σ ranging from 0.1 to 3.3 in steps of 0.2.
- *Algorithm parameters*: The number of particles in both the usual particle filter and the projection particle filter is $N = 200$; the exponential family in the projection particle filter is chosen as the Gaussian family; the set of grids on the projected belief space is $G = \{$ mean = [0 : 0.5 : 15], standard deviation = [0 : 0.2 : 5]$\}$; one run of horizon length $H = 10^5$ for each average cost criterion case, 1000 independent runs of horizon length $H = 40$ for each discounted total cost criterion case.
- *Simulation issues*: We use common random variables among different policies and different σ's.

In order to implement CE, we use Monte Carlo simulation to find the optimal threshold policy for the fully observed problem (i.e., $y_k = x_k$), which means

$$a_k = \begin{cases} 0, & \text{if } x_k > L^*; \\ 1, & \text{if } x_k < L^*. \end{cases} \tag{22}$$

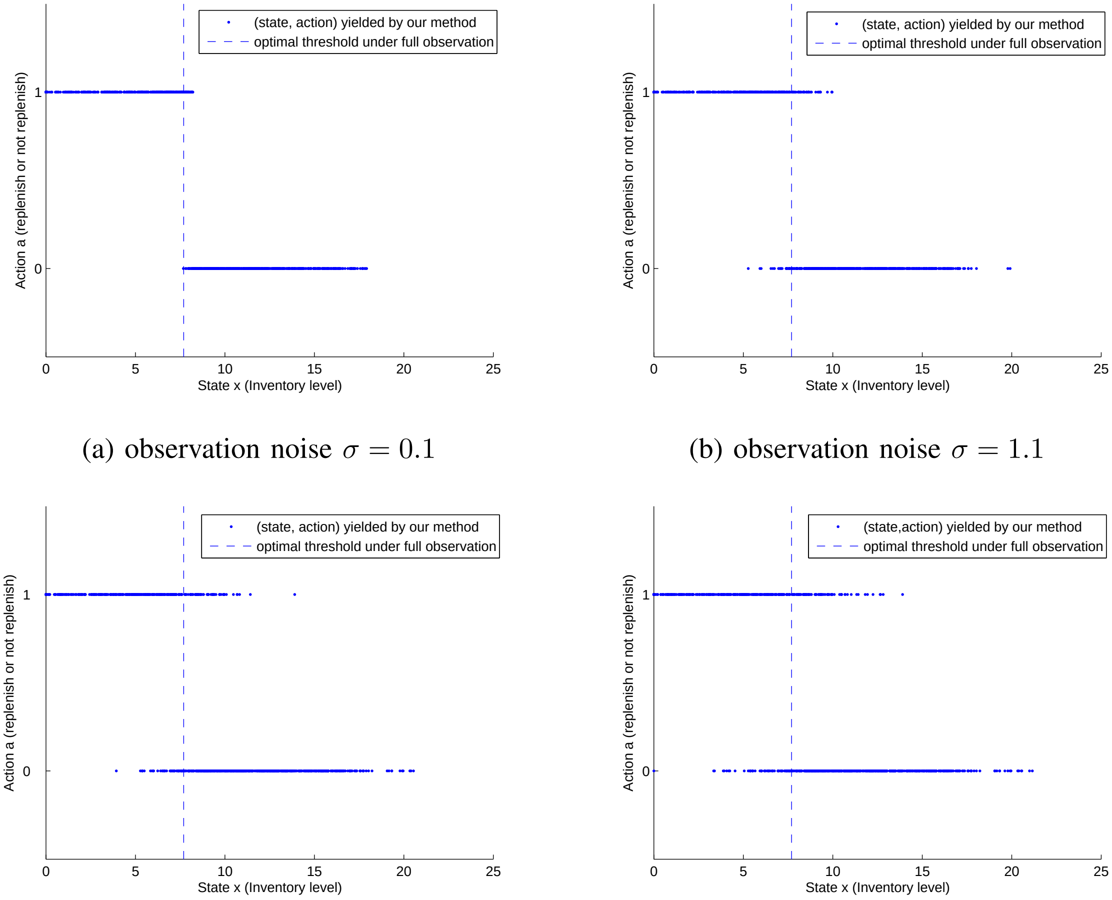
The simulation result indicates the cost is a convex function of the threshold and the minimum is achieved at $L^* = 7.7$ for both average cost and discounted total cost.

Table I and Table II list the simulated average costs and discounted total cost using different policies under different observation noises, respectively. Each entry shows the average cost/discounted total cost, and in the parentheses the percentage error from the average cost/discounted total cost under full observation using the optimal threshold policy. Our algorithm generally outperforms all other algorithms under all observation noise levels. CE also performs very well, and slightly outperforms CE-MLE. The greedy policy is much worse than all other algorithms. For all the algorithms, the average cost/discounted total cost increases as the observation noise increases. That is consistent with the intuition that we cannot perform better with less information. Fig.1 shows the actions taken by our algorithm as a function of the true inventory levels in the average cost case (the discounted total cost case is similar and is omitted here). The dotted vertical line is the optimal threshold under full observation, so the optimal threshold policy would yield action $a = 1$ when the inventory level falls below the threshold and yields $a = 0$ otherwise when there is no observation noise. Our algorithm yields a policy that picks actions very close to those of the optimal threshold policy when the observation noise is small (cf. Fig.1(a)), indicating that our algorithm is indeed finding the optimal policy. As the observation noise increases, more actions picked by our policy violate the optimal threshold, and that again shows the value of information in determining the actions.

Although the performance of CE is comparable to that of our method, we should notice that CE with mean estimate is generally a suboptimal policy except in some special cases (cf. section 6.1 in [4]), and it does *not* have a theoretical error bound. Moreover, to use CE requires solving the full observation problem, which is also very difficult in many cases. In contrast, our algorithm has a proven error bound on the performance, and works with the belief MDP directly without having to solve the MDP problem under full observation.

VII. Conclusion

In this paper, we developed a method that effectively reduces the dimension of the belief space via the orthogonal projection of the belief states onto a parameterized family of probability densities. For an



(a) observation noise $\sigma = 0.1$

(b) observation noise $\sigma = 1.1$

(c) observation noise $\sigma = 2.1$

(d) observation noise $\sigma = 3.1$

Fig. 1. Our algorithm: actions taken for different inventory levels under different observation noise variances.

exponential family, the orthogonal projection has an analytical form and can be carried out efficiently. The exponential family is fully represented by a finite (small) number of parameters, hence the belief space is mapped to a low-dimensional parameter space and the resultant belief MDP is called the projected belief MDP. The projected belief MDP can then be solved in numerous ways, such as using standard value iteration or policy iteration, to generate a policy. This policy is used in conjunction with the projection particle filter for online decision making.

We analyzed the performance of the policy generated by solving the projected belief MDP in terms of the difference between the value function associated with this policy and the optimal value function of the POMDP. We also provided a bound on the error between our projection particle filter and exact filtering.

We applied our method to an inventory control problem, and it generally outperformed other methods.

TABLE I

OPTIMAL AVERAGE COST ESTIMATE FOR THE INVENTORY CONTROL PROBLEM USING DIFFERENT POLICIES. EACH ENTRY REPRESENTS THE AVERAGE COST OF A RUN OF HORIZON 10^5 (DEVIATION ABOVE OPTIMUM IN PARENTHESES).

observation noise σ	Our method	CE policy	CE-MLE policy	Greedy policy
0.1	12.849 (0.12%)	12.842 (0.06%)	12.837 (0.02%)	25.454 (98.34%)
0.3	12.845 (0.08%)	12.857 (0.18%)	12.861 (0.21%)	25.467 (98.43%)
0.5	12.864 (0.23%)	12.867 (0.26%)	12.884 (0.39%)	25.457 (98.36%)
0.7	12.881 (0.37%)	12.882 (0.37%)	12.890 (0.44%)	25.452 (98.31%)
0.9	12.904 (0.55%)	12.908 (0.57%)	12.940 (0.82%)	25.450 (98.30%)
1.1	12.938 (0.81%)	12.945 (0.87%)	12.969 (1.05%)	25.428 (98.13%)
1.3	12.973 (1.08%)	12.977 (1.12%)	12.993 (1.24%)	25.356 (97.57%)
1.5	13.016 (1.41%)	13.034 (1.56%)	13.029 (1.52%)	25.293 (97.08%)
1.7	13.066 (1.81%)	13.100 (2.07%)	13.117 (2.20%)	25.324 (97.32%)
1.9	13.110 (2.15%)	13.159 (2.53%)	13.172 (2.64%)	25.343 (97.47%)
2.1	13.123 (2.25%)	13.183 (2.72%)	13.227 (3.06%)	25.332 (97.38%)
2.3	13.210 (2.93%)	13.263 (3.34%)	13.292 (3.57%)	25.355 (97.56%)
2.5	13.250 (3.24%)	13.314 (3.74%)	13.333 (3.89%)	25.402 (97.92%)
2.7	13.323 (3.81%)	13.382 (4.27%)	13.454 (4.83%)	25.428 (98.13%)
2.9	13.374 (4.21%)	13.458 (4.86%)	13.497 (5.17%)	25.478 (98.52%)
3.1	13.444 (4.75%)	13.527 (5.40%)	13.580 (5.81%)	25.553 (99.10%)
3.3	13.512 (5.28%)	13.603 (6.00%)	13.655 (6.40%)	25.655 (99.90%)

When the observation noise is small, our algorithm yields a policy that picks the actions very closely to the optimal threshold policy for the fully observed problem. Although we only proved theoretical results for discounted cost problems, the simulation results indicate that our method also works well on average cost problems. We should point out that our method is also applicable to finite horizon problems, and is suitable for large-state POMDPs in addition to continuous-state POMDPs.

VIII. APPENDIX

Proof of Theorem 1: Denote $J_k(b) \triangleq T^k J_0(b), J_k^p(b^p) \triangleq (T^p)^k J_0(b^p), k = 0, 1, \ldots$, and define

$$\begin{aligned} b_k(b,a) &= \langle g(\cdot,a), b\rangle + \gamma E_Y\{J_{k-1}(\psi(b,a,Y))\}, \\ \mu_k(b) &= \arg\min_{a\in A} Q_k(b,a), \end{aligned}$$

$$\begin{aligned} b_k^p(b,a) &= \langle g(\cdot,a), b^p\rangle + \gamma E_Y\{J_{k-1}(\psi(b^p,a,Y)^p)\}, \\ \mu_k^p(b) &= \arg\min_{a\in A} Q_k^p(b,a). \end{aligned}$$

TABLE II

Optimal discounted cost estimate for the inventory control problem using different policies. Each entry represents the discounted cost on 1000 independent runs of horizon 40 (Deviation above optimum in parentheses).

observation noise σ	Our method	CE policy	CE-MLE policy	Greedy policy
0.1	129.126 (13.57%)	129.120 (13.56%)	129.090 (13.54%)	241.667 (112.55%)
0.3	129.016 (13.47%)	129.169 (13.61%)	129.105 (13.55%)	242.079 (112.91%)
0.5	129.097 (13.54%)	129.122 (13.57%)	129.164 (13.60%)	242.656 (113.42%)
0.7	129.474 (13.88%)	129.299 (13.72%)	129.623 (14.01%)	243.327 (114.01%)
0.9	129.868 (14.22%)	129.593 (13.98%)	129.762 (14.13%)	244.002 (114.61%)
1.1	129.940 (14.29%)	130.192 (14.51%)	130.229 (14.54%)	244.804 (115.31%)
1.3	130.336 (14.63%)	130.493 (14.77%)	130.543 (14.82%)	245.673 (116.08%)
1.5	130.575 (14.84%)	130.738 (14.99%)	131.085 (15.29%)	246.708 (116.99%)
1.7	130.724 (14.98%)	130.952 (15.18%)	131.446 (15.61%)	247.701 (117.86%)
1.9	131.266 (15.45%)	131.294 (15.48%)	131.595 (15.74%)	248.545 (118.60%)
2.1	131.778 (15.90%)	131.758 (15.88%)	132.235 (16.30%)	249.452 (119.40%)
2.3	132.176 (16.25%)	132.222 (16.29%)	132.763 (16.77%)	250.070 (119.94%)
2.5	132.741 (16.75%)	132.536 (16.57%)	133.467 (17.39%)	250.492 (120.31%)
2.7	133.070 (17.04%)	133.184 (17.14%)	133.978 (17.84%)	250.763 (120.55%)
2.9	133.484 (17.40%)	133.606 (17.51%)	134.558 (18.35%)	250.811 (120.59%)
3.1	133.961 (17.82%)	134.088 (17.93%)	135.827 (19.46%)	250.887 (120.66%)
3.3	134.502 (18.30%)	134.807 (18.57%)	136.117 (19.72%)	250.767 (120.56%)

Hence,

$$\begin{aligned} J_k(b) &= \min_{a\in A} Q_k(b,a) = Q_k(b,\mu_k(b)), \\ J_k^p(b^p) &= \min_{a\in A} Q_k^p(b,a) = Q_k^p(b,\mu_k^p(b)). \end{aligned}$$

Denote $err_k \triangleq \max_{b\in B} |J_k(b) - J_k^p(b^p)|, k = 1, 2, \ldots$.

We consider the first iteration. Initialize with $J_0(b) = J_0^p(b^p) = 0$. By Assumption 2, $\forall a \in A$,

$$|Q_1(b,a) - Q_1^p(b,a)| = |\langle g(\cdot,a), b - b^p\rangle| \le \epsilon_1, \quad \forall b \in B. \tag{23}$$

Hence, with $a = \mu_1^p(b)$, the above inequality yields $Q_1(b,\mu_1^p(b)) \le J_1^p(b^p) + \epsilon_1$. Using $J_1(b) \le Q_1(b,\mu_1^p(b))$, we get

$$J_1(b) \le J_1^p(b^p) + \epsilon_1, \quad \forall b \in B. \tag{24}$$

With $a = \mu_1(b)$, (23) yields $Q_1^p(b,\mu_1(b)) - \epsilon_1 \le J_1(b)$. Using $J_1^p(b^p) \le Q_1^p(b,\mu_1(b))$, we get

$$J_1^p(b^p) - \epsilon_1 \le J_1(b), \quad \forall b \in B. \tag{25}$$

From (24) and (25), we conclude

$$|J_1(b) - J_1^p(b^p)| \leq \epsilon_1, \quad \forall b \in B.$$

Taking the maximum over b on both sides of the above inequality yields

$$err_1 \leq \epsilon_1. \tag{26}$$

Now we consider the $(k+1)^{th}$ iteration. For a fixed $Y = y$, by Assumption 2, $|\langle g(\cdot, a), \psi(b, a, y) - \psi(b^p, a, y)^p\rangle| \leq \delta_1$. Let δ_1 be the δ in Assumption 3 and denote the corresponding ϵ by ϵ_2. Then

$$|J_k(\psi(b, a, y)) - J_k(\psi(b^p, a, y)^p)| \leq \epsilon_2, \quad \forall b \in B. \tag{27}$$

Therefore, $\forall a \in A$,

$$\begin{aligned}
& |Q_{k+1}(b, a) - Q_{k+1}^p(b, a)| \\
\leq\ & |\langle g(\cdot, a), b - b^p\rangle| + \gamma E_Y\{|J_k(\psi(b, a, Y)) - J_k^p(\psi(b^p, a, Y)^p)|\} \\
\leq\ & \epsilon_1 + \gamma E_Y\{|J_k(\psi(b, a, Y)) - J_k(\psi(b^p, a, Y)^p)| + |J_k(\psi(b^p, a, Y)^p) - J_k^p(\psi(b^p, a, Y)^p)|\} \\
\leq\ & \epsilon_1 + \gamma(\epsilon_2 + err_k), \quad \forall b \in B.
\end{aligned}$$

The third inequality follows from (27) and the definition of err_k. Using an argument similar to that used to prove (26) from (23), we conclude that

$$err_{k+1} \leq \epsilon_1 + \gamma(\epsilon_2 + err_k). \tag{28}$$

Using induction on (28) with initial condition (26) and taking $k \to \infty$, we obtain

$$\begin{aligned}
|J_*(b) - J_*^p(b^p)| &\leq \sum_{k=0}^{\infty} \gamma^k \epsilon_1 + \sum_{k=1}^{\infty} \gamma^k \epsilon_2 \\
&= \frac{\epsilon_1 + \gamma\epsilon_2}{1 - \gamma}.
\end{aligned} \tag{29}$$

Therefore, (15) is proved.

Fixing a policy μ on the original belief MDP, define the mappings under policy μ on the belief MDP and the projected belief MDP as

$$T_\mu J(b) = \langle g(\cdot, \mu(b)), b\rangle + \gamma E_Y\{J(\psi(b, \mu(b), Y))\}, \tag{30}$$

$$T_\mu^p J(b) = \langle g(\cdot, \mu(b)), b^p\rangle + \gamma E_Y\{J(\psi(b^p, \mu(b), Y)^p)\}, \tag{31}$$

respectively. Since μ_*^p is a stationary policy for the projected belief MDP, $\bar{\mu}_*^p = \mu_*^p \circ Proj_\Omega$ is stationary for the original belief MDP. Hence,

$$J_*^p(b^p) = T_{\mu_*^p}^p J_*^p(b^p),$$

$$J_{\bar{\mu}_*^p}(b) = T_{\bar{\mu}_*^p} J_{\bar{\mu}_*^p}(b).$$

Subtracting both sides of the above two equations, and substituting in the definitions of T^p and T (i.e., (31) and (30)) for the righthand sides respectively, we get

$$J^p_*(b^p) - J_{\bar{\mu}^p_*}(b) = \langle g(\cdot, \mu^p_*(b^p)), b^p - b\rangle + \gamma E_Y\{J^p_*(\psi(b^p, \mu^p_*(b^p), Y)^p) - J_{\bar{\mu}^p_*}(\psi(b, \mu^p_*(b^p), Y))\} \quad (32)$$

For a fixed $Y = y$,

$$\begin{aligned} & |J^p_*(\psi(b^p, \mu^p_*(b^p), y)^p) - J_{\bar{\mu}^p_*}(\psi(b, \mu^p_*(b^p), y))| \\ \leq \;& |J^p_*(\tilde{b}) - J_{\bar{\mu}^p_*}(\tilde{b})| + |J_{\bar{\mu}^p_*}(\psi(b^p, \mu^p_*(b^p), y)^p) - J_{\bar{\mu}^p_*}(\psi(b, \mu^p_*(b^p), y))|, \end{aligned}$$

where $\tilde{b} = \psi(b^p, \mu^p_*(b^p), y)^p \in B$. Since $|\langle g(\cdot, a), \psi(b^p, \mu^p_*(b^p), y)^p - \psi(b, \mu^p_*(b^p), y)\rangle| \leq \delta_1$ by Assumption 2, letting $\delta = \delta_1$ in Assumption 3 and denoting the corresponding $\tilde{\epsilon}$ by ϵ_3, we get the second term

$$|J_{\bar{\mu}^p_*}(\psi(b^p, \mu^p_*(b^p), y)^p) - J_{\bar{\mu}^p_*}(\psi(b, \mu^p_*(b^p), y))| \leq \epsilon_3.$$

Denoting $err \triangleq \max_{b\in B} |J^p_*(b^p) - J_{\mu^p_*}(b)|$, we obtain

$$|J^p_*(\psi(b^p, \mu^p_*(b^p), y)^p) - J_{\bar{\mu}^p_*}(\psi(b, \mu^p_*(b^p), y))| \leq err + \epsilon_3.$$

Therefore, (32) becomes

$$|J^p_*(b^p) - J_{\bar{\mu}^p_*}(b)| \leq \epsilon_1 + \gamma(err + \epsilon_3).$$

Taking the maximum over b on both sides of the above inequality yields

$$err \leq \epsilon_1 + \gamma(err + \epsilon_3).$$

Hence,

$$err \leq \frac{\epsilon_1 + \gamma\epsilon_3}{1-\gamma}. \quad (33)$$

With (29) and (33), we obtain

$$\begin{aligned} |J_*(b) - J_{\bar{\mu}^p_*}(b)| &\leq |J_*(b) - J^p_*(b^p)| + |J^p_*(b^p) - J_{\bar{\mu}^p_*}(b)| \\ &\leq \frac{2\epsilon_1 + \gamma(\epsilon_2 + \epsilon_3)}{1-\gamma}, \quad \forall b \in B. \end{aligned}$$

Therefore, (16) is proved. $\square$

Proof of Lemma 1: $E[|\langle b_{k-1} - f(\cdot, \hat{\theta}_{k-1}), \phi\rangle|]$ is the error from time $k-1$, which is also the initial error for time k. Hence, the prediction step gives

$$\begin{aligned} E[|\langle b_{k|k-1} - b'_{k|k-1}, \phi\rangle|] &= E[|\langle K_k(b_{k-1} - f(\cdot, \hat{\theta}_{k-1})), \phi\rangle|] \\ &= E[|\langle b_{k-1} - f(\cdot, \hat{\theta}_{k-1}), K_k\phi\rangle|] \\ &\leq e_{k-1}\|K_k\phi\| \\ &\leq e_{k-1}\|\phi\|. \end{aligned} \quad (34)$$

The Bayes' updating step gives

$$\begin{aligned}
E[|\langle b_k - b'_k, \phi\rangle|] &= E[|\frac{\langle b_{k|k-1}, \Psi_k\phi\rangle}{\langle b_{k|k-1}, \Psi_k\rangle} - \frac{\langle b'_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle}|] \\
&= E[|\frac{\langle b_{k|k-1}, \Psi_k\phi\rangle}{\langle b_{k|k-1}, \Psi_k\rangle} - \frac{\langle b_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle} + \frac{\langle b_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle} - \frac{\langle b'_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle}|] \\
&\leq E[|\frac{\langle b_{k|k-1}, \Psi_k\phi\rangle\langle b_{k|k-1} - b'_{k|k-1}, \Psi_k\rangle}{\langle b_{k|k-1}, \Psi_k\rangle\langle b'_{k|k-1}, \Psi_k\rangle}|] + E[|\frac{\langle b_{k|k-1} - b'_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle}|] \\
&\leq \delta\frac{|\langle b_{k|k-1}, \Psi_k\phi\rangle|}{\langle b_{k|k-1}, \Psi_k\rangle}E[|\langle b_{k|k-1} - b'_{k|k-1}, \Psi_k\rangle|] + \delta E[|\langle b_{k|k-1} - b'_{k|k-1}, \Psi_k\phi\rangle|] \\
&\leq \delta\|\phi\|e_{k-1}\|\Psi_k\| + \delta e_{k-1}\|\Psi_k\phi\| \\
&\leq 2\delta\|\Psi_k\|e_{k-1}\|\phi\| \\
&= a_k e_{k-1}\|\phi\|,
\end{aligned}$$

where $a_k = 2\delta\|\Psi_k\|$. The second inequality follows from (18), and the third inequality follows from (34). □

Proof of Lemma 2: This lemma uses essentially the same proof technique as Lemmas 3 and 4 in [10]. However, it is not quite obvious how these lemmas imply our lemma here. Therefore, we state the proof to make this paper more accessible. After the resampling step, $\hat{f}(\cdot, \hat{\theta}_{k-1}) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x^i_{k-1})$, where $x^i_{k-1}, i = 1, \ldots, N$ are i.i.d. samples from $f(\cdot, \hat{\theta}_{k-1})$. Using the Cauchy-Schwartz inequality, we have

$$\begin{aligned}
& (E[\langle \hat{f}(\cdot, \hat{\theta}_{k-1}) - f(\cdot, \hat{\theta}_{k-1}), \phi\rangle^2])^{1/2} \\
= \;& (E[(\frac{1}{N}\sum_{i=1}^{N}(\phi(x^i_{k-1}) - \langle f(\cdot, \hat{\theta}_{k-1}), \phi\rangle))^2])^{1/2} \\
= \;& \frac{1}{\sqrt{N}}(E[\frac{1}{N}\sum_{i=1}^{N}(\phi(x^i_{k-1}) - \langle f(\cdot, \hat{\theta}_{k-1}), \phi\rangle)^2])^{1/2} \\
= \;& \frac{1}{\sqrt{N}}(\langle f(\cdot, \hat{\theta}_{k-1}), \phi^2\rangle - \langle f(\cdot, \hat{\theta}_{k-1}), \phi\rangle^2)^{1/2} \\
\leq \;& \frac{1}{\sqrt{N}}\langle f(\cdot, \hat{\theta}_{k-1}), \phi^2\rangle^{1/2} \\
\leq \;& \frac{1}{\sqrt{N}}\|\phi\|. \qquad (35)
\end{aligned}$$

The Bayes' updating step gives

$$\begin{aligned}
E[|\langle \hat{b}_k - b'_k, \phi\rangle|] &= E[|\frac{\langle \hat{b}_{k|k-1}, \Psi_k\phi\rangle}{\langle \hat{b}_{k|k-1}, \Psi_k\rangle} - \frac{\langle b'_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle}|] \\
&= E[|\frac{\langle \hat{b}_{k|k-1}, \Psi_k\phi\rangle}{\langle \hat{b}_{k|k-1}, \Psi_k\rangle} - \frac{\langle \hat{b}_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle} + \frac{\langle \hat{b}_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle} - \frac{\langle b'_{k|k-1}, \Psi_k\phi\rangle}{\langle b'_{k|k-1}, \Psi_k\rangle}|]
\end{aligned}$$

$$\leq \quad E[|\frac{\langle \hat{b}_{k|k-1}, \Psi_k \phi \rangle \langle \hat{b}_{k|k-1} - b'_{k|k-1}, \Psi_k \rangle}{\langle \hat{b}_{k|k-1}, \Psi_k \rangle \langle b'_{k|k-1}, \Psi_k \rangle}|] + E[|\frac{\langle \hat{b}_{k|k-1} - b'_{k|k-1}, \Psi_k \phi \rangle}{\langle b'_{k|k-1}, \Psi_k \rangle}|].$$

Using the Cauchy-Schwartz inequality, (18) and (35), the first term can be simplified as

$$\begin{aligned}
& E[|\frac{\langle \hat{b}_{k|k-1}, \Psi_k \phi \rangle \langle \hat{b}_{k|k-1} - b'_{k|k-1}, \Psi_k \rangle}{\langle \hat{b}_{k|k-1}, \Psi_k \rangle \langle b'_{k|k-1}, \Psi_k \rangle}|] \\
\leq \quad & \delta(E[\frac{\langle \hat{b}_{k|k-1}, \Psi_k \phi \rangle^2}{\langle \hat{b}_{k|k-1}, \Psi_k \rangle^2}])^{1/2}(E[\langle \hat{b}_{k|k-1} - b'_{k|k-1}, \Psi_k \rangle^2])^{1/2} \\
= \quad & \delta(E[\frac{\langle \hat{b}_{k|k-1}, \Psi_k \phi \rangle^2}{\langle \hat{b}_{k|k-1}, \Psi_k \rangle^2}])^{1/2}(E[\langle f(\cdot, \hat{\theta}'_{k-1}) - f(\cdot, \theta'_{k-1}), K_k \Psi_k \rangle^2])^{1/2} \\
\leq \quad & \delta \|\phi\| \frac{1}{\sqrt{N}} \|\Psi_k\|,
\end{aligned}$$

and the second term can be simplified as

$$\begin{aligned}
& E[|\frac{\langle \hat{b}_{k|k-1} - b'_{k|k-1}, \Psi_k \phi \rangle}{\langle b'_{k|k-1}, \Psi_k \rangle}|] \\
\leq \quad & \delta(E[\langle \hat{b}_{k|k-1} - b'_{k|k-1}, \Psi_k \phi \rangle^2])^{1/2} \\
= \quad & \delta(E[\langle \hat{f}(\cdot, \hat{\theta}_{k-1}) - f(\cdot, \hat{\theta}_{k-1}), K_k \Psi_k \phi \rangle^2])^{1/2} \\
\leq \quad & \delta \frac{1}{\sqrt{N}} \|\Psi_k \phi\| \\
\leq \quad & \delta \frac{1}{\sqrt{N}} \|\Psi_k\| \|\phi\|.
\end{aligned}$$

Therefore, adding these two terms yields

$$\begin{aligned}
E[|\langle \hat{b}_k - b'_k, \phi \rangle|] \quad & \leq \quad 2\delta \|\Psi_k\| \frac{\|\phi\|}{\sqrt{N}} \\
& = \quad \tau_k \frac{\|\phi\|}{\sqrt{N}},
\end{aligned}$$

where $\tau_k = 2\delta \|\Psi_k\|$. □

Proof of Lemma 3**:** The key idea of the proof for Lemma 4 in [2] is used here. From (9), we know that $E_{\hat{\theta}_k}[c_j(X)] = E_{\hat{b}_k}[c_j(X)]$ and $E_{\theta'_k}[c_j(X)] = E_{b'_k}[c_j(X)]$. Hence, we obtain

$$E[|E_{\hat{\theta}_k}(c_j(X)) - E_{\theta'_k}(c_j(X))|] = E[|\langle \hat{b}_k - b'_k, c_j \rangle|], \quad j = 1, \ldots, m.$$

Taking summation over j, we obtain

$$E[\sum_{j=1}^{m} |E_{\hat{\theta}_k}(c_j(X)) - E_{\theta'_k}(c_j(X))|] = \sum_{j=1}^{m} E[|\langle \hat{b}_k - b'_k, c_j \rangle|].$$

Since $c_j \in B(R^{n_x})$, we apply Lemma 2 with $\phi = c_j$ and thus obtain

$$E[|\langle \hat{b}_k - b'_k, c_j \rangle|] \leq b_k \frac{\|c_j\|}{\sqrt{N}}, j = 1, \ldots, m.$$

Therefore,

$$E[\|E_{\hat{\theta}_k}(c(X)) - E_{\theta'_k}(c(X))\|_1] \leq \frac{\tilde{\tau}_k}{\sqrt{N}},$$

where $\|\cdot\|_1$ denotes the L_1 norm on R^{n_x}, $c = [c_1, \dots, c_m]^T$, and $\tilde{\tau}_k \tau_k \sum_{j=1}^{m} \|c_j\|$. Since Θ' is compact and the Fisher information matrix $[E_\theta[c_i(X)c_j(X)] - E_\theta[c_i(X)]E_\theta[c_j(X)]]_{ij}$ is positive definite, we get (cf. Fact 2 in [2] for a detailed proof)

$$\|\hat{\theta}_k - \theta'_k\|_1 \quad \leq \quad \alpha\|E_{\hat{\theta}_k}(c(X)) - E_{\theta'_k}(c(X))\|_1.$$

Taking the expectation on both sides yields

$$\begin{aligned} E[\|\hat{\theta}_k - \theta'_k\|_1] &\leq \alpha E[\|E_{\hat{\theta}_k}(c(X)) - E_{\theta'_k}(c(X))\|_1] \\ &\leq \alpha\tilde{\tau}_k \frac{1}{\sqrt{N}}. \end{aligned}$$

On the other hand, taking derivative of $E_\theta[\phi(X)]$ with respect to θ_i yields

$$\begin{aligned} |\frac{d}{d\theta_i} E_\theta[\phi(X)]| &= |E_\theta[c_i(X)\phi(X)] - E_\theta[c_i(X)]E_\theta[\phi(X)]| \\ &\leq \sqrt{Var_\theta(c_i)Var_\theta(\phi)} \\ &\leq \sqrt{E_\theta(c_i^2)E_\theta(\phi^2)} \\ &\leq \|c_i\|\|\phi\|. \end{aligned}$$

Hence,

$$\|dE_\theta[\phi(X)]/d\theta\|_1 \leq (\sum_{i=1}^{m} \|c_i\|)\|\phi\|.$$

Since Θ' is compact, there exists a constant $\beta > 0$ such that $E_\theta[\phi(X)]$ is Lipschitz over $\theta \in \Theta'$ with Lipschitz constant $\beta\|\phi\|$ (cf. the proof of Fact 2 in [2]), i.e.,

$$|E_{\hat{\theta}_k}[\phi] - E_{\theta'_k}[\phi]| \leq \beta\|\phi\|\|\hat{\theta}_k - \theta'_k\|_1.$$

Taking expectation on both sides yields

$$\begin{aligned} E[|f(\cdot, \hat{\theta}_k) - f(\cdot, \theta'_k), \phi\rangle|] &\leq \beta\|\phi\|E[\|\hat{\theta}_k - \theta'_k\|_1] \\ &\leq \beta\|\phi\|\alpha\tilde{\tau}_k \frac{1}{\sqrt{N}} \\ &= d_k \frac{|\phi\|}{\sqrt{N}}, \end{aligned}$$

where $d_k = \alpha\beta\tilde{\tau}_k$. □

***Proof of Theorem 2*:** Applying Lemma 1, Assumption 6, and Lemma 3, we have that for all $\phi \in B(R^{n_x})$ and $k \in \mathbb{N}$, there exist $a_i > 0, d_i > 0, i = 1, \dots, k$ such that

$$\begin{aligned} E[|\langle b_k - f(\cdot,\hat{\theta}_k),\phi\rangle|] &\leq E[|\langle b_k - b'_k,\phi\rangle|] + E[|\langle b'_k - f(\cdot,\theta'_k),\phi\rangle|] + E[|\langle f(\cdot,\theta'_k) - f(\cdot,\hat{\theta}_k),\phi\rangle|] \\ &\leq (a_k e_{k-1} + \epsilon + d_k \frac{1}{\sqrt{N}})\|\phi\| \\ &= e_k\|\phi\|. \end{aligned}$$

It is easy to deduce by induction that

$$e_k = a_1^k e_0 + (\sum_{i=2}^{k} a_i^k + 1)\epsilon + (\sum_{i=2}^{k} a_i^k d_{i-1} + d_k)\frac{1}{\sqrt{N}}. \quad \Box$$

References

[1] S. Arulampalam, S. Maskell, N. J. Gordon, and T. Clapp, “A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking”, *IEEE Transactions of Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[2] B. Azimi-Sadjadi, and P. S. Krishnaprasad, “Approximate Nonlinear Filtering and its Application in Navigation,” *Automatica*, vol. 41, no. 6, pp. 945-956, 2005.

[3] A. R. Barron, and C. Sheu, “Approximation of Density Functions by Sequences of Exponential Family”, *The Annals of Statistics*, vol. 19, no. 3, pp. 1347-1369, 1991.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[5] D.P. Bertsekas, and J.N. Tsitsiklis, *Neuro-Dynamic Programming*, Optimization and Neural Computation Series. Athena Scientific, 1st edition, 1996.

[6] D. Brigo, “Filtering by Projection on the Manifold of Exponential Densities”, *Ph.D. Thesis*, Department of Economics and Econometrics, Vrijie Universiteit, Armsterdam, 1996.

[7] A. Brooks, A. Makarenkoa, S. Williamsa, and H. Durrant-Whytea, “Parametric POMDPs for Planning in Continuous State Spaces”, *Robotics and Autonomous Systems* , vol. 54, no. 11, pp. 887-897, 2006.

[8] A. R. Cassandra, “Exact and Approximate Algorithms for Partially Observable Markov Decision Processes”, *Ph.D. thesis*, Brown University, 2006.

[9] J. Hu, H.S. Chang, M.C. Fu and S.I. Marcus. *Simulation-based Algorithms for* M*arkov Decision Processes*. Communications and Control Engineering Series. Springer, New York, 1st edition, 2007.

[10] D. Crisan, and A. Doucet, “A Survey of Convergence Results on Particle Filtering Methods for Practitioners”, *IEEE Transaction on Signal Processing*, vol. 50, no. 3, pp. 736-746, 2002.

[11] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors, *Sequential* M*onte* C*arlo Methods In Practice*, Springer, New York, 2001.

[12] D. de Farias, and B. Van Roy, The linear programming approach to approximate dynamic programming, *Operations Research*, 51(6), 2003.

[13] M. Hauskrecht, “Value-Function Approximations for Partially Observable Markov Decision Processes”, *Journal of Artificiall Intelligence Research*, vol. 13, pp. 33-95, 2000.

[14] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, Novel approach to nonlinear/non-Gaussian bayesian state estimation, In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113, 1993.

[15] O. Hernandez-Lerma, J. B. Lasserre, *Discrete-Time Markov Control Processes Basic Optimality Criteria*, New York: Springer, 1996.

[16] F. Le Gland, and N. Oudjane, “Stability and Uniform Approximation of Nonlinear Filters Using the Hilbert Metric and Application to Particle Filter”, *The Annals of Applied Probability*, vol. 14, no. 1, pp. 144-187, 2004.

[17] E. L. Lemann, and G. Casella, *Theory of Point Estimation*, 2nd edition, New York: Springer, 1998.

[18] M. L. Littman, “The Witness Algorithm: Solving Partially Observable Markov Decision Processes”, *TR CS-94-40*, Department of Computer Science, Brown University, Providence, RI, 1994.

[19] M. K. Murray, and J. W. Rice, *Differential Geometry and Statistics*, Chapman & Hill, 1993.

[20] J. M. Porta, M. T. J. Spaan, and N. Vlassis, “Robot planning in partially observable continuous domains”, *Proc. Robotics: Science and Systems*, 2005.

[21] P. Poupart, C. Boutilier, “Value-Directed Compression of POMDPs”, *Advances in Neural Information Processing Systems*, vol. 15, pp. 1547-1554, 2003.

[22] N. Roy, “Finding Approximate POMDP Solutions through Belief Compression”, *Ph.D. thesis*, Robotics Institute, Carnegie Mellon University, Pittsburg, PA, 2003.

[23] N. Roy and G. Gordon. “Exponential Family PCA for Belief Compression in POMDPs”, *Advances in Neural Information Processing Systems*, vol. 15, pp. 1635-1642, 2003.

[24] R. D. Smallwood, and E. J. Sondik, “The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon”, *Operations Research*, vol. 21, no. 5, pp. 1071-1088, 1973.

[25] S. Thrun, “Monte Carlo POMDPs”, *Advances in Neural Information Processing Systems*, vol. 12, pp. 1064-1070, 2000.

[26] H. J. Yu, “Approximate Solution Methods for Partially Observable Markov and Semi-Markov Decision Processes”, *Ph.D. thesis*, M.I.T., Cambridge, MA, 2006.