Solving Continuous-State POMDPs via Density Projection

Enlu Zhou, *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 finite-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 projection. 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.**

***Index Terms*—Partially observable Markov decision processes, particle filtering, decision making, density projection, belief state, value function.**

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 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 [6], which is the conditional distribution of the current state given the history of observations and actions. For a finite-state POMDP, the belief space is finite dimensional (i.e., a probability simplex), whereas for a continuous-state POMDP, the belief space is an infinite-dimensional space of continuous probability distributions. This difference suggests that simple generalizations of many of the finite-state algorithms to continuous-state models are not appropriate or applicable. For example, discretization of the continuous-state space may result in a finite-state POMDP of dimension either too large to solve computationally or not sufficiently precise. Taking another example, many algorithms for solving finite-state POMDPs (see [17] 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. Throughout the paper, when we use the word "dimension" or "dimensional", we refer to the dimension of the belief space/state.

Despite the abundance of algorithms for finite-state POMDPs, the aforementioned difficulty has motivated some researchers to look for efficient algorithms for continuous-state POMDPs [24] [25] [31] [28] [8] [9] [10]. Assuming discrete observation and action spaces, Porta et al. [24] 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. In a later work [25], they extended their result and method to continuous observation and action spaces using sampling strategies. 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 [31] 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. Brunskill et al. [10] used weighted sums of Gaussians to approximate the belief states and value functions in a class of switching state models.

Roy [28] and Brooks et al. [8] 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 [28] proposed an augmented MDP (AMDP), characterizing belief states using maximum likelihood state and entropy, which are usually not sufficient statistics except for a 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. [8] proposed a parametric POMDP, representing the belief state as a Gaussian distribution so as to convert the POMDP to a problem of computing the value function over a two-dimensional continuous space, and using the extended Kalman filter to estimate the transition of the approximated belief state.

E. Zhou is with the Department of Industrial & Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, IL, 61801 USA email: enluzhou@illinois.edu.

S.I. Marcus is with the Department of Electrical and Computer Engineering, and Institute for Systems Research, University of Maryland, College Park, MD, 20742 USA e-mail: 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.

The restriction to the Gaussian representation has the same problem as the AMDP. The algorithm recently proposed in Brooks and Williams [9] is similar to ours, in that they also approximate the belief state by a parameterized density and solve the approximate belief MDP on the parameter space using Monte Carlo simulation-based methods. However, they did not specify how to compute the parameters except for Gaussian densities, whereas we explicitly provide an analytical way to calculate the parameters for exponential families of densities. Moreover, we develop rigorous theoretical error bounds for our algorithm. There are some other belief compression algorithms designed for finite-state POMDPs, such as value-directed compression [26] and the exponential family principle components analysis (E-PCA) belief compression [29], but they cannot be directly generalized to continuous-state models, since they are based on a fixed set of support points.

Motivated by the work of [31] [28] and [8], 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 that family of distributions. For an exponential family, the minimization of KL divergence can be carried out in analytical form, making the method easy to implement. The projected belief MDP can then be solved on the parameter space by using simulation-based algorithms, or can be further approximated by a finite-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 [28] and the parametric POMDP in [8], which considers only the family of Gaussian distributions. In addition, we provide theoretical results on the error bounds of the value function and the performance of the policy generated by our method with respect to the optimal ones.

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 discusses scalability and computational issues of the method, and applies the method to a simulation example of an inventory control problem. Section VII concludes the paper. Proofs of all results are contained in the Appendix.

II. Continuous-state POMDP

Consider a discrete-time continuous-state POMDP:

$$\begin{aligned} x_k &= f(x_{k-1}, a_{k-1}, u_{k-1}), \quad k = 1, 2 \ldots, && (1)\\ y_k &= h(x_k, a_{k-1}, v_k), \quad k = 1, 2, \ldots, && (2)\\ y_0 &= h_0(x_0, v_0), \end{aligned}$$

where for all k, the state x_k is in a continuous state space $S \subseteq \mathbb{R}^{n_x}$, the action a_k is in a finite action space $A \subset \mathbb{R}^{n_a}$, the observation y_k is in a continuous observation space $O \subseteq \mathbb{R}^{n_y}$, the random disturbances $u_k \in \mathbb{R}^{n_u}$ and $v_k \in \mathbb{R}^{n_v}$ are sequences of i.i.d. continuous random vectors with known distributions. Assume that $\{u_k\}$ and $\{v_k\}$ are independent of each other, and are 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 $u \in \mathbb{R}^{n_u}$, $h(x, a, v)$ is continuous in x for every $a \in A$ and $v \in \mathbb{R}^{n_v}$, and $h_0(x, v)$ is continuous in x for every $v \in \mathbb{R}^{n_v}$. Eqn. (1) is often referred to as the state equation, and (2) as the observation equation.

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

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

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, to minimize the *value function*

$$J_\pi = \lim_{H\to\infty} E_{x_0, \{u_k\}_{k=0}^{H-1}, \{v_k\}_{k=0}^{H}} \left\{ \sum_{k=0}^{H} \gamma^k g(x_k, \mu_k(I_k)) \right\},$$

where $g : S \times A \to \mathbb{R}$ is the *one-step cost function*, $\gamma \in (0, 1)$ is the *discount factor*, and $E_{x_0, \{u_k\}_{k=0}^{H-1}, \{v_k\}_{k=0}^{H}}$ denotes the expectation with respect to the joint distribution of $x_0, u_0, \ldots, u_{H-1}, v_0, \ldots, v_H$. For simplicity, we assume that the above limit exists. 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. The standard 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.,

$$b_k(x) \triangleq p(x_k = x | I_k).$$

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

$$\begin{aligned} & b_k(x) \;=\; p(x_k = x | I_{k-1}, a_{k-1}, y_k) \\ = \;& \frac{p(y_k | x_k = x, I_{k-1}, a_{k-1}) p(x_k = x | I_{k-1}, a_{k-1})}{p(y_k | I_{k-1}, a_{k-1})} \\ \propto \;& p(y_k | x_k = x, a_{k-1}) \int_S p(x_k = x | I_{k-1}, a_{k-1}, x_{k-1}) \dots \\ & \qquad p(x_{k-1} | I_{k-1}, a_{k-1}) dx_{k-1} \\ \propto \;& p(y_k | x_k = x, a_{k-1}) \int_S p(x_k = x | a_{k-1}, x_{k-1}) \dots \\ & \qquad b_{k-1}(x_{k-1}) dx_{k-1}. \end{aligned} \tag{3}$$

The third line follows from the Markovian property of y_k induced by (2), and the fact that the denominator $p(y_k|I_{k-1}, a_{k-1})$ does not explicitly depend on x_k and k; the fourth line follows from the Markovian property of x_k induced by (1), and the fact that a_{k-1} is a function of I_{k-1}. The right-hand side of (3) can be expressed in terms of b_{k-1}, a_{k-1} and y_k. Hence,

$$b_k = \psi(b_{k-1}, a_{k-1}, y_k), \tag{4}$$

where y_k is characterized by the time-homogeneous conditional distribution $P_Y(y_k|b_{k-1})$ that is induced by (1) and (2), and does not depend on $\{y_0, \dots, y_{k-1}\}$.

A POMDP can be converted to an MDP by conditioning on the information vectors ([6], Chapter 5), 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 viewed 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 $\pi = \{\mu_0, \mu_1, \dots\}$, where each function μ_k maps the belief state b_k onto the action space A. Noticing that

$$E_{x_0, \{u_i\}_{i=0}^{k-1}, \{v_i\}_{i=0}^{k}} \{g(x_k, a_k)\} = E\left\{E_{x_k}\{g(x_k, a_k)|I_k\}\right\},$$

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, the optimal value function is given by

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

where T is the *dynamic programming (DP) mapping* that operates on any bounded function $J : S \to \mathbb{R}$ according to

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

where E_Y denotes the expectation with respect to the distribution P_Y.

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 [30]. This feature has been exploited in various exact and approximate value iteration algorithms such as those found in [17], [22], and [30].

For continuous-state POMDPs, the belief state b is a continuous density, and thus, the belief space B is an infinite-dimensional space that contains all sorts of continuous densities. For continuous-state POMDPs, the value function preserves convexity [32], but value iteration algorithms are not computationally feasible 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 finite-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 detail and develop the formulation of a projected belief MDP.

III. Projected Belief MDP

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

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

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

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

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 [3]:

Definition 1: Let $\{c_1(\cdot), \dots, c_m(\cdot)\}$ be affinely independent scalar functions defined on $\mathbb{R}^n$, i.e., for distinct points

$x_1, \ldots, x_{m+1}$, $\sum_{i=1}^{m+1} \lambda_i c(x_i) = 0$ and $\sum_{i=1}^{m+1} \lambda_i = 0$ implies $\lambda_1, \ldots, \lambda_{m+1} = 0$, where $c(x) = [c_1(x), \ldots, c_m(x)]^T$. Assuming that $\Theta_0 = \{\theta \in \mathbb{R}^m : \varphi(\theta) = \log \int \exp(\theta^T c(x))dx < \infty\}$ is a convex set with a nonempty interior, 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*, with θ its parameter and $c(x)$ its sufficient statistic.

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

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

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

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

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 [21], 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 optimality condition is satisfied, i.e.,

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

With a little rearranging of the terms and the expression of $f(x, \theta)$, the above equation can be rewritten as

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

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

$$\begin{aligned} b_0^p &= Proj_\Omega(b_0), \\ b_k^p &= \psi(b_{k-1}^p, a_{k-1}, y_k)^p, \quad k = 0, 1, \ldots, \end{aligned}$$

where $\psi(b_{k-1}^p, a_{k-1}, y_k)^p = Proj_\Omega(\psi(b_{k-1}^p, a_{k-1}, y_k))$, 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 onto 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. The effect of the discretization procedure on dynamic programming has been studied in [5]. 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. Estimation of $\tilde{P}(\theta_i, a)(\theta_j)$ is done using a variation of the projection particle filtering algorithm, to be described in the next section. $\tilde{g}(\theta_i, a)$ can be estimated by its sample mean:

$$\tilde{g}(\theta_i, a) = \frac{1}{N} \sum_{j=1}^{N} g(x_j, a), \tag{11}$$

where $x_1, \ldots, x_N$ are sampled i.i.d. from $f(\cdot, \theta_i)$.

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 [15], neuro-dynamic programming methods [7], and simulation-based methods [12].

IV. Projection Particle Filtering

Solving the projected belief MDP gives us a 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 that policy. Hence, to implement our approach requires addressing the problem of estimating 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, because 1) it outperforms the extended Kalman filter in many nonlinear/non-Gaussian systems [1], and 2) we will develop a projection particle filter 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] [14]. As we have already shown in (3), the belief state evolves recursively as

$$b_k(x_k) \quad \propto \quad p(y_k|x_k, a_{k-1}) \int_S p(x_k|a_{k-1}, x_{k-1}) \dots b_{k-1}(x_{k-1}) dx_{k-1}. \quad (12)$$

The integration in (12) can be approximated using Monte Carlo simulation, which is the essence of particle filtering. Specifically, suppose $\{x^i_{k-1}\}^N_{i=1}$ are drawn i.i.d. from b_{k-1}, and $x^i_{k|k-1}$ is drawn from $p(x_k|a_{k-1}, x^i_{k-1})$ for each i; then $b_k(x_k)$ can be approximated by the probability mass function

$$\hat{b}_k(x_k) = \sum_{i=1}^{N} w^i_k \delta(x_k - x^i_{k|k-1}), \quad (13)$$

where

$$w^i_k \propto p(y_k|x^i_{k|k-1}, a_{k-1}), \quad (14)$$

δ denotes the Kronecker delta function, $\{x^i_{k|k-1}\}^N_{i=1}$ are the random support points, and $\{w^i_k\}^N_{i=1}$ are the associated probabilities/weights which sum up to 1.

To avoid sample degeneracy, new samples $\{x^i_k\}^N_{i=1}$ 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 $\{x^i_{k+1|k}\}^N_{i=1}$ and corresponding weights $\{w^i_{k+1}\}^N_{i=1}$, which are used to approximate b_{k+1}. This is the basic form of particle filtering, which is also called the bootstrap filter [18]. (Please see [1] for a rigorous and thorough derivation for a more general form of particle filtering.) The algorithm is as follows:

Algorithm 1: (Particle Filtering (Bootstrap Filter))

- Input: a (stationary) policy μ on the belief MDP; a sequence of observations $y_1, y_2, \dots$ arriving sequentially at time $k = 1, 2, \dots$.
- Output: a sequence of approximate belief states $\hat{b}_1, \hat{b}_2, \dots$.
- Step 1. Initialization: Sample $x^1_0, \dots, x^N_0$ i.i.d. from the approximate initial belief state $\hat{b}_0$. Set $k = 1$.
- Step 2. Prediction: Compute $x^1_{k|k-1}, \dots, x^N_{k|k-1}$ by propagating $x^1_{k-1}, \dots, x^N_{k-1}$ according to the system dynamics (1) using the action $a_{k-1} = \mu(\hat{b}_{k-1})$ and randomly generated noise $\{u^i_{k-1}\}^N_{i=1}$, i.e., sample $x^i_{k|k-1}$ from $p(\cdot|x^i_{k-1}, a_{k-1})$, $i = 1, \dots, N$. The empirical predicted belief state is
$$\hat{b}_{k|k-1}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x^i_{k|k-1}).$$
- Step 3. Bayes' updating: Receive a new observation y_k. The empirical updated belief state is
$$\hat{b}_k(x) = \sum_{i=1}^{N} w^i_k \delta(x - x^i_{k|k-1}),$$
where
$$w^i_k = \frac{p(y_k|x^i_{k|k-1}, a_{k-1})}{\sum_{i=1}^N p(y_k|x^i_{k|k-1}, a_{k-1})}, \quad i = 1, \dots, N.$$
- Step 4. Resampling: Sample $x^1_k, \dots, x^N_k$ 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 as the sample number N increases to infinity [13] [20]. 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 obtain a reasonable approximation of the belief state, particle filtering needs a large number of samples/particles. Since the number of samples/particles is the dimension of the approximate belief state $\hat{b}$, particle filtering is not very helpful in reducing the dimensionality of the belief space. Moreover, particle filtering does not give us an approximate belief state in the projected belief space Ω, hence the policy we obtained by solving the projected belief MDP is not immediately applicable.

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 project the 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 the bootstrap filter.

Applying the density projection technique we described in the last section, projecting the empirical belief state $\hat{b}_k$ onto an exponential family Ω involves finding a $f(\cdot, \theta)$ with the parameter θ satisfying (9). Hence, plugging (13) into (9), yields

$$\sum_{i=1}^{N} w_i c_j(x^i_{k|k-1}) = E_\theta [c_j], \quad j = 1, \dots, m,$$

which constitutes the projection step in the projection particle filtering.

Algorithm 2: (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 $y_1, y_2, \dots$ arriving sequentially at time $k = 1, 2, \dots$.
- Output: a sequence of approximate belief states $f(\cdot, \hat{\theta}_1), f(\cdot, \hat{\theta}_2), \dots$.
- Step 1. Initialization: Sample $x^1_0, \dots, x^N_0$ i.i.d. from the approximate initial belief state $f(\cdot, \hat{\theta}_0)$. Set $k = 1$.

- Step 2. Prediction: Compute $x^1_{k|k-1},\ldots,x^N_{k|k-1}$ by propagating $x^1_{k-1},\ldots,x^N_{k-1}$ according to the system dynamics (1) using the action $a_{k-1}=\mu^p(f(\cdot,\hat{\theta}_{k-1}))$ and randomly generated noise $\{u^i_{k-1}\}_{i=1}^N$, i.e., sample $x^i_{k|k-1}$ from $p(\cdot|x^i_{k-1},a_{k-1})$, $i=1,\ldots,N$.
- Step 3. Bayes' updating: Receive a new observation y_k. Compute weights according to

$$w^i_k=\frac{p(y_k|x^i_{k|k-1},a_{k-1})}{\sum_{i=1}^N p(y_k|x^i_{k|k-1},a_{k-1})},\quad i=1,\ldots,N.$$

- Step 4. Projection: The approximate belief state is $f(\cdot,\hat{\theta}_k)$, where $\hat{\theta}_k$ satisfies the equations

$$\sum_{i=1}^N w^i_k c_j(x^i_{k|k-1})=E_{\hat{\theta}_k}[c_j],\quad j=1,\ldots,m.$$

- Step 5. Resampling: Sample $x^1_k,\ldots,x^N_k$ 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, the 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 policy solved for the projected belief MDP.

As mentioned in the last section, PPF can be varied slightly for estimating the transition probabilities of the discretized projected belief MDP, as follows:

Algorithm 3: (Estimation of the transition probabilities)

- 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

$$\tilde{b}_k(x)=\sum_{i=1}^N w^k_i\delta(x-\tilde{x}_i),$$

where

$$w^k_i=\frac{p(y_k|\tilde{x}_i,a)}{\sum_{i=1}^N p(y_k|\tilde{x}_i,a)},\quad i=1,\ldots,N.$$

- Step 5. Projection: For $k=1,\ldots,N$, project each $\tilde{b}_k$ onto the exponential family, i.e., finding $\tilde{\theta}_k$ that satisfies (9).
- Step 6. Estimation: For $k=1,\ldots,N$, find the nearest-neighbor grid point of $\tilde{\theta}_k$ in G. For each $\theta_j\in G$, count the frequency $\tilde{P}(\theta_i,a)(\theta_j)=$ (number of θ_j)$/N$.

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, and the following:

Assumption 1: There exist 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 [6], [19]. 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

$$J_*(b)\triangleq J_{\mu_*}(b)=\lim_{k\to\infty}T^kJ_0(b),$$

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

$$J^p_*(b^p)\triangleq J^p_{\mu^p_*}(b^p)=\lim_{k\to\infty}(T^p)^kJ_0(b^p).$$

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

$$\left|J_*(b)-J^p_*(b^p)\right|.$$

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

$$\left|J_*(b)-J_{\bar{\mu}^p_*}(b)\right|,$$

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

The next assumption bounds 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 any $\delta>0$ that satisfies $|\langle g(\cdot,a),b-b'\rangle|\le\delta$, $\forall b,b'\in B$, there exists $\epsilon>0$ such that $|J_k(b)-J_k(b')|\le\epsilon$, $\forall b,b'\in B$, $\forall k$, and there exists $\tilde{\epsilon}>0$ such that $|J_\mu(b)-J_\mu(b')|\le\tilde{\epsilon}$, $\forall b,b'\in B$, $\forall\mu\in\Pi$.

Now we present our main result.

Theorem 1: Under Assumptions 1, 2 and 3,

$$|J_*(b)-J^p_*(b^p)|\le\frac{\epsilon_1+\gamma\epsilon_2}{1-\gamma},\quad\forall b\in B,\tag{15}$$

$$\left|J_*(b)-J_{\bar{\mu}^p_*}(b)\right|\le\frac{2\epsilon_1+\gamma(\epsilon_2+\epsilon_3)}{1-\gamma},\quad\forall b\in B,\tag{16}$$

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 $\delta = \delta_1$, where δ_1 is the constant in Assumption 2.

Remark 2: In (15) and (16), ϵ_1 is a projection error, and ϵ_2 and ϵ_3 decrease as the projection error δ_1 decreases. Therefore, as the projection error decreases, the optimal value function of the projected belief MDP $J_*^p(b^p)$ converges to the true optimal value function $J_*(b)$, and the corresponding policy $\bar{\mu}_*^p$ converges to the true optimal policy μ_*. Roughly speaking, the projection error decreases as the number of sufficient statistics in the chosen exponential family increases (for details, please see section V-C: Validation of the Assumptions).

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

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

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

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

We will use the 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 to 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, the notations used for probability densities before are used to denote their corresponding probability measures from now on. Namely, we use b to denote a probability measure on $\mathbb{R}^{n_x}$ and assume it admits a probability density with respect to Lebesgue measure, which is the belief state. Similarly, we use $f(\cdot, \theta)$ to denote a probability measure on $\mathbb{R}^{n_x}$ and assume it admits a probability density with respect to Lebesgue measure in the chosen exponential family with parameter θ.

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

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

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

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

TABLE I
NOTATIONS OF DIFFERENT CONDITIONAL DISTRIBUTIONS

b_k	exact conditional distribution
$\hat{b}_k$	PPF conditional distribution before projection
$f(\cdot, \hat{\theta}_k)$	PPF projected conditional distribution
b'_k	CF conditional distribution before projection
$f(\cdot, \theta'_k)$	CF projected conditional distribution

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

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

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

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

The PPF at time k can be described as

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

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 obtain $f(\cdot, \theta'_k)$. It can be described as

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

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 total error 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 compared to EF, 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 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, 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 in this section are with respect to the sampling in the algorithm only. We consider a test function $\phi \in B(\mathbb{R}^{n_x})$. It is easy to see that $K\phi \in B(\mathbb{R}^{n_x})$ and $\|K\phi\| \leq \|\phi\|$, since

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

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

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 $\Theta' \subseteq \Theta$ such that $\hat{\theta}_k \in \Theta'$, and $\theta'_k \in \Theta', \forall k$.

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

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

Assumption 5 guarantees 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 $f(x, a_k, u_k)$ in (1) is continuous in x, K_k is weakly continuous (pp. 175-177, [19]). 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 guarantees 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: There exists a constant $\epsilon > 0$ such that

$$E\left[|\langle b'_k, \phi \rangle - \langle f(\cdot, \theta'_k), \phi \rangle|\right] \leq \epsilon \|\phi\|, \quad \forall \phi \in B(\mathbb{R}^{n_x}), \forall k \in \mathbb{N}.$$

Remark 3: 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 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 in (17).

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

$$E\left[|\langle b_k - b'_k, \phi \rangle|\right] \leq \tau_k e_{k-1} \|\phi\|, \quad \forall \phi \in B(\mathbb{R}^{n_x}). \tag{19}$$

Lemma 2 considers the bound on the third term in (17) before projection.

Lemma 2: Under Assumptions 4 and 5, for each $k \in \mathbb{N}$,

$$E\left[\left|\langle \hat{b}_k - b'_k, \phi \rangle\right|\right] \leq \tau_k \frac{\|\phi\|}{\sqrt{N}}, \quad \forall \phi \in B(\mathbb{R}^{n_x}),$$

where τ_k is the same constant as in Lemma 1.

Lemma 3 considers the bound on the third term in (17) based on the result of Lemma 2.

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

$$E\left[\left|\langle f(\cdot, \hat{\theta}_k) - f(\cdot, \theta'_k), \phi \rangle\right|\right] \leq d\tau_k \frac{\|\phi\|}{\sqrt{N}}, \quad \forall \phi \in B(\mathbb{R}^{n_x}), \tag{20}$$

where τ_k is the same constant as in Lemmas 1 and 2.

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

Theorem 2: Let e_0 be a nonnegative constant such that $E[|\langle b_0 - f(\cdot, \hat{\theta}_0), \phi \rangle|] \leq e_0 \|\phi\|, \forall \phi \in B(\mathbb{R}^{n_x})$. Under Assumptions 4, 5 and 6, and assuming that $c_j \in B(\mathbb{R}^{n_x}), j = 1, \ldots, m$, then for each $k \in \mathbb{N}$

$$E\left[\left|\langle b_k - f(\cdot, \hat{\theta}_k), \phi \rangle\right|\right] \leq e_k \|\phi\|, \quad \forall \phi \in B(\mathbb{R}^{n_x}),$$

where

$$e_k = \tau_1^k e_0 + \left(\sum_{i=2}^{k} \tau_i^k + 1 \right) \epsilon + \frac{d}{\sqrt{N}} \sum_{i=1}^{k} \tau_i^k, \tag{21}$$

$\tau_i^k = \prod_{j=i}^{k} \tau_j$ for $k \geq i$, $\tau_i^k = 0$ for $k < i$, τ_j is the constant in Lemmas 1, 2, and 3, d is the constant in Lemma 3, and ϵ is the constant in Assumption 6.

Remark 4: As we mentioned in Remark 2, the projection error e_0 and ϵ decrease 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 with 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.

C. Validation of the Assumptions

Assumptions 2 and 6 are the main assumptions of our analysis. They are assumptions on the projection error, assuming that density projection introduces a "small" error. We will show that in certain cases these assumptions hold, and the projection error converges to 0 as the number of sufficient statistics, m, goes to infinity. We will first state a convergence result from [4]. However, as this convergence result is in the sense of KL divergence, we will further show the convergence in the sense employed in our assumptions by using an intermediate result in [4].

Consider a probability density function b defined on a bounded interval, and approximate it by b^p, a density function in an exponential family, whose sufficient statistics consist of polynomials, splines or trigonometric series. The following theorem is proved in [4].

Theorem 3: If $\log b$ has r square-integrable derivatives, i.e., $\int |D^r \log b|^2 < \infty$, then $D_{KL}(b||b^p)$ converges to 0 at rate m^{-2r} as $m \to \infty$.

Theorem 3 says the projected density b^p converges to b in the sense of KL divergence, as m goes to infinity. An intermediate result (see (6.6) in [4]) is:

$\|\log b/b^p\| \leq \nu_m$, *where* ν_m *is a constant that depends on* m, *and* $\nu_m \to 0$ *as* $m \to \infty$.

Since b is bounded and $log(\cdot)$ is a continuously differentiable function, there exists a constant ξ such that $\|b - b^p\| \leq \xi \|\log b - \log b^p\|$. Hence, with the intermediate result above,

$$\begin{aligned} |\langle \phi, b - b^p \rangle| &\leq \|\phi\| \int \|b - b^p\| dx \\ &\leq \|\phi\| \int \xi \|log \frac{b}{b^p}\| dx \leq \|\phi\| \xi l \nu_m, \end{aligned}$$

where l is the length of the bounded interval that b is defined on. Since ν_m can be made arbitrarily small by taking large enough m, it is easy to see that Assumptions 2 and 6 hold in the cases that we consider.

VI. Numerical Experiments

A. Scalability and Computational Issues

Estimation of the one-step cost function (11) and transition probabilities (Algorithm 3) are executed for every belief-action pair that is in the discretized mesh G and the action space A. Hence, the algorithms scale according to $O(|G||A|N)$ and $O(|G||A|N^2)$, respectively, where $|G|$ is the number of grid points, $|A|$ is the number of actions, and N is the number of samples specified in the algorithms. In implementation, we found that most of the computation time is spent on executing Algorithm 3 over all belief-action pairs. However, estimation of cost functions and transition probabilities can be pre-computed and stored, and hence only needs to be done once.

The advantage of the algorithms is that the scalability is independent of the size of the actual state space, since G is a grid mesh on the parameter space of the projected belief space. That is exactly what is desired by employing density projection. However, to get a better approximation, more parameters in the exponential family should be used, and that will lead to a higher-dimensional parameter space to discretize. Increasing the number of parameters in the exponential family also makes sampling more difficult. Sampling from a general exponential family is usually not easy, and may require some advanced techniques, such as the adaptive rejection sampling (ARS) [16], and hence more computation time. This difficulty can be avoided by resampling from the discrete particles instead of the projected density, which is equivalent to using the plain particle filter and then doing projection outside the filter. However, this may lead to sample degeneracy. The trade-off between a better approximation and less computation time is complicated and requires more research. We plan to study how to appropriately choose the exponential family and improve the simulation efficiency in the future.

B. Simulation Results

Since most of the benchmark POMDP problems in the literature assume a discrete state space, it is difficult to compare against the state of the art. Here we consider an inventory control problem by adding a partial observation equation to a fully observable inventory control problem. The fully observable problem has an optimal threshold policy [27], which allows us to verify our method in the limiting case by setting the observation noise very close to 0. In our inventory control problem, the inventory level is reviewed at discrete times, and 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

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

The cost incurred in period k is

$$\begin{aligned} g_k(x_k, a_k, u_k) &= h \max(x_k + a_k Q - u_k, 0) \ldots \\ &\quad + s \max(u_k - x_k - a_k Q, 0). \end{aligned}$$

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

$$\limsup_{H \to \infty} \frac{E\left[\sum_{k=0}^{H} g_k\right]}{H}, \qquad \lim_{H \to \infty} E\left[\sum_{k=0}^{H} \gamma^k g_k\right],$$

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

In the simulation, we first choose an exponential family and specify a grid mesh on its parameter space, then implement (11) and Algorithm 3 on the grid mesh, and use value iteration to solve for a policy. These are done offline. In an online run, Algorithm 2 (PPF) is used for filtering and making decisions with the policy obtained offline. We also consider a small variation of this method: instead of using PPF, we use Algorithm 1 (PF) and do density projection outside the filter each time. We compare our two methods (called "Ours 1" and "Ours 2", respectively) described above to four other algorithms: (1) Certainty equivalence using the mean estimate (CE); (2) Certainty equivalence using the maximum likelihood estimate (CE-MLE); (3) EKF-based Parametric POMDP (EKF-PPOMDP) in [8]; (4) Greedy policy. CE 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. EKF-PPOMDP approximates the belief state by a Gaussian distribution, and uses the extended Kalman filter to estimate the transition of the belief state. Similar to our method, it also solves a discretized MDP defined on the parameter space of the Gaussian density. 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 grid points on the projected belief space is $G = \{$mean = $[0 : 0.5 : 15]$, standard deviation = $[0 : 0.2 : 5]\}$ for both our methods and EKF-PPOMDP; 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; nearest neighbor as the value function approximator in both our methods and EKF-PPOMDP.
- *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., $y_k = x_k$): if the inventory level is below the threshold L, the store/warehouse should order to replenish its inventory; otherwise, if the inventory level is above L, the store/warehouse should not order. That is,

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

The simulation result indicates both the average and discounted cost functions are convex in the threshold and the minimum is achieved at $L = 7.7$ for both.

Table II and Table III list the simulated average costs and discounted total cost using different policies under different observation noises, respectively. Our methods generally outperforms all the other algorithms under all observation noise levels. CE also performs very well, and slightly outperforms CE-MLE. EKF-PPOMDP performs better in the average cost case than the discounted cost case. The greedy policy is much worse than all other algorithms. While our methods and the EKF-PPOMDP involve offline computation, the more critical online computation time of all the simulated methods is approximately the same.

For all the algorithms, the average cost/discounted cost increases as the observation noise increases. That is consistent with the intuition that we cannot perform better with less information. Fig.1 shows 1000 actions taken by our method versus 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 L. 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.

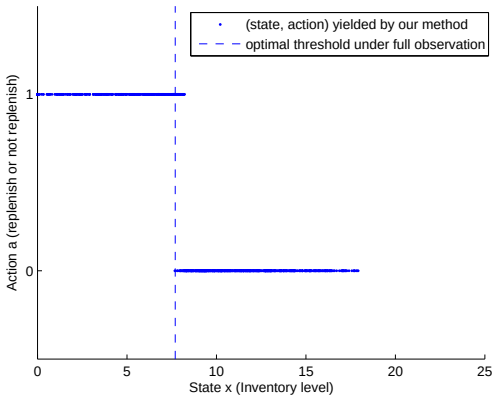
The performances of our two methods are very close, with one slightly better than the other. Solely for the purpose of filtering, doing projection outside the filter is easier to implement if we want to use a general exponential family, and also gives a better estimate of the belief state, since the projection error will not accumulate. However, for solving POMDPs, we conjecture that PPF would work better in conjunction with the policy solved from the projected belief MDP, since the projected belief MDP assumes that the transition of the belief state is also projected. However, we do not know which one is better.

Our method outperforms the EKF-PPOMDP, mainly because the projection particle filter used in our method is better than the extend Kalman filter used in the EKF-PPOMDP for estimating the belief transition probabilities. This agrees with the results in [9], which also observed that Monte Carlo simulation of the belief transitions is better than the EKF estimate.

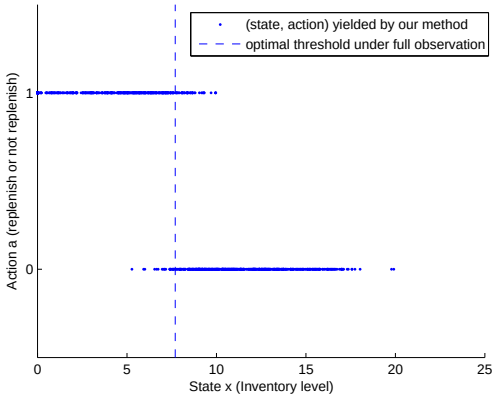
Although the performance of CE is comparable to that of our methods for this particular example, CE is generally a suboptimal policy except in some special cases (cf. section 6.1 in [6]), and it does *not* have a theoretical error bound. Moreover, to use CE requires solving the full observation problem, which is also very difficult in many cases. In contrast, our method has a proven error bound on the performance, and works with the POMDP directly without having to solve the MDP problem under full observation.

VII. Conclusion

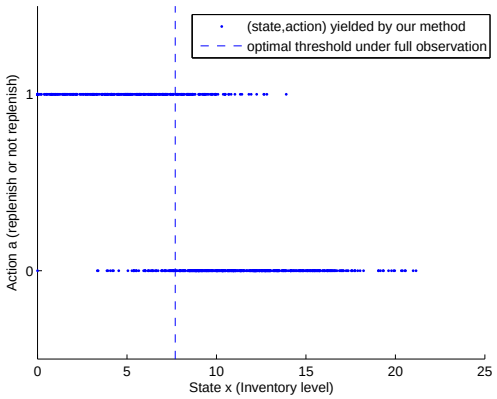
In this paper, we developed a method that effectively reduces the dimension of the belief space via the orthogonal projection of the belief states onto a parameterized family of



(a) observation noise $\sigma = 0.1$



(b) observation noise $\sigma = 1.1$



(c) observation noise $\sigma = 3.1$

Fig. 1. Our method (Ours 1): actions taken for different inventory levels under different observation noise variances.

probability densities. For an exponential family, the density 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 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.

TABLE II
OPTIMAL AVERAGE COST ESTIMATES FOR THE INVENTORY CONTROL PROBLEM USING DIFFERENT METHODS. EACH ENTRY REPRESENTS THE AVERAGE COST OF A RUN OF HORIZON 10^5.

σ	Ours 1	Ours 2	CE	CE-MLE	EKF-P-POMDP	Greedy Policy
0.1	12.849	12.849	12.842	12.837	12.941	25.454
0.3	12.845	12.837	12.857	12.861	12.950	25.467
0.5	12.864	12.862	12.867	12.884	12.964	25.457
0.7	12.881	12.884	12.882	12.890	12.990	25.452
0.9	12.904	12.918	12.908	12.940	13.006	25.450
1.1	12.938	12.943	12.945	12.969	13.059	25.428
1.3	12.973	12.986	12.977	12.993	13.105	25.356
1.5	13.016	13.017	13.034	13.029	13.141	25.293
1.7	13.066	13.067	13.100	13.117	13.182	25.324
1.9	13.110	13.105	13.159	13.172	13.214	25.343
2.1	13.123	13.140	13.183	13.227	13.255	25.332
2.3	13.210	13.201	13.263	13.292	13.307	25.355
2.5	13.250	13.246	13.314	13.333	13.380	25.402
2.7	13.323	13.324	13.382	13.454	13.441	25.428
2.9	13.374	13.384	13.458	13.497	13.491	25.478
3.1	13.444	13.459	13.527	13.580	13.553	25.553
3.3	13.512	13.525	13.603	13.655	13.637	25.655

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. 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.

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

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

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

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, $\forall a \in A$,

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

TABLE III
OPTIMAL DISCOUNTED COST ESTIMATE FOR THE INVENTORY CONTROL PROBLEM USING DIFFERENT METHODS. EACH ENTRY REPRESENTS THE DISCOUNTED COST (MEAN, STANDARD ERROR IN PARENTHESES) BASED ON 1000 INDEPENDENT RUNS OF HORIZON 40.

σ	Ours 1	Ours 2	CE	CE-MLE	EKF-P-POMDP	Greedy Policy
0.1	126.79 (1.64)	127.26 (1.63)	129.12 (1.81)	129.09 (1.81)	137.41 (1.65)	241.67 (2.99)
0.3	126.86 (1.63)	126.95 (1.63)	129.17 (1.78)	129.11 (1.78)	137.64 (1.62)	242.08 (2.98)
0.5	126.61 (1.63)	126.35 (1.62)	129.12 (1.77)	129.16 (1.78)	138.16 (1.60)	242.66 (2.98)
0.7	126.42 (1.62)	126.99 (1.61)	129.30 (1.77)	129.62 (1.79)	141.78 (1.55)	243.33 (2.98)
0.9	126.78 (1.63)	126.86 (1.63)	129.59 (1.76)	129.76 (1.78)	138.23 (1.60)	244.00 (2.97)
1.1	127.49 (1.64)	127.74 (1.63)	130.19 (1.77)	130.23 (1.75)	140.86 (1.57)	244.81 (2.97)
1.3	128.74 (1.65)	128.30 (1.64)	130.49 (1.76)	130.54 (1.72)	146.02 (1.52)	245.67 (2.96)
1.5	129.30 (1.68)	129.45 (1.66)	130.74 (1.75)	131.09 (1.77)	144.88 (1.52)	246.71 (2.95)
1.7	129.71 (1.67)	128.93 (1.67)	130.95 (1.76)	131.45 (1.77)	146.80 (1.52)	247.70 (2.96)
1.9	130.11 (1.69)	129.85 (1.67)	131.29 (1.75)	131.60 (1.73)	147.16 (1.56)	248.55 (2.93)
2.1	130.67 (1.69)	130.17 (1.67)	131.76 (1.74)	132.24 (1.79)	144.67 (1.54)	249.45 (2.95)
2.3	130.96 (1.68)	130.36 (1.67)	132.22 (1.75)	132.76 (1.78)	145.35 (1.55)	250.07 (2.97)
2.5	131.90 (1.68)	130.86 (1.68)	132.54 (1.76)	133.47 (1.78)	145.06 (1.58)	250.49 (2.96)
2.7	131.81 (1.68)	131.66 (1.68)	133.18 (1.75)	133.98 (1.78)	148.39 (1.54)	250.76 (2.96)
2.9	132.36 (1.68)	131.78 (1.68)	133.61 (1.75)	134.56 (1.83)	146.27 (1.57)	250.81 (2.96)
3.1	132.95 (1.70)	133.51 (1.70)	134.09 (1.76)	135.83 (1.79)	147.96 (1.54)	250.89 (2.95)
3.3	133.08 (1.69)	132.76 (1.69)	134.81 (1.76)	136.12 (1.84)	145.32 (1.60)	250.77 (2.94)

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

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

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

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

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

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

Therefore, $\forall a \in A$,

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

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

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

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

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,

$$\begin{aligned} J_*^p(b^p) &= T_{\mu_*^p}^p J_*^p(b^p), \\ J_{\bar{\mu}_*^p}(b) &= T_{\bar{\mu}_*^p} J_{\bar{\mu}_*^p}(b). \end{aligned}$$

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 right-hand sides respectively, we get

$$\begin{aligned} & J_*^p(b^p) - J_{\bar{\mu}_*^p}(b) = \left\langle g\left(\cdot, \mu_*^p(b^p)\right), b^p - b \right\rangle \ldots \\ & + \gamma E_Y \left\{ J_*^p(\psi(b^p, \mu_*^p(b^p), Y)^p) - J_{\bar{\mu}_*^p}\left(\psi(b, \mu_*^p(b^p), Y)\right) \right\}. \end{aligned} \tag{32}$$

For a fixed $Y = y$,

$$\begin{aligned} & \left|J_*^p(\psi(b^p, \mu_*^p(b^p), y)^p) - J_{\bar{\mu}_*^p}(\psi(b, \mu_*^p(b^p), y))\right| \\ \leq\ & \left|J_*^p(\tilde{b}) - J_{\bar{\mu}_*^p}(\tilde{b})\right| \ldots \\ & + \left|J_{\bar{\mu}_*^p}\left(\psi(b^p, \mu_*^p(b^p), y)^p\right) - J_{\bar{\mu}_*^p}\left(\psi(b, \mu_*^p(b^p), y)\right)\right|, \end{aligned}$$

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

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

Denoting $err \triangleq \max_{b \in B} \left|J_*^p(b^p) - J_{\mu_*^p}(b)\right|$, we obtain

$$\left|J_*^p\left(\psi(b^p, \mu_*^p(b^p), y)^p\right) - J_{\bar{\mu}_*^p}\left(\psi(b, \mu_*^p(b^p), y)\right)\right| \leq err + \epsilon_3.$$

Therefore, (32) becomes

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

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

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

Hence,

$$err \leq \frac{\epsilon_1 + \gamma\epsilon_3}{1-\gamma}. \tag{33}$$

With (29) and (33), we obtain

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

Therefore, (16) is proved. ∎

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

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

Under Assumptions 4 and 5, we have showed (18). Using (18) and (34), the Bayes' updating step yields

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

where $\tau_k = 2\delta\|\Psi_k\|$. ∎

Proof of Lemma 2: This lemma uses essentially the same proof technique as Lemmas 3 and 4 in [13]. 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, $\hat{f}(\cdot, \hat{\theta}_{k-1}) = \frac{1}{N}\sum_{i=1}^N \delta(x - x_{k-1}^i)$, where $x_{k-1}^i, i = 1, \ldots, N$ are i.i.d. samples from $f(\cdot, \hat{\theta}_{k-1})$. Using the Cauchy-Schwartz inequality, we have

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

The Bayes' updating step yields

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

Under Assumptions 4 and 5, we have shown (18). Using the Cauchy-Schwartz inequality, (18), and (35), the first term can be simplified as

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

and the second term can be simplified as

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

Therefore, adding these two terms yields

$$E\left[\left|\langle \hat{b}_k - b'_k, \phi\rangle\right|\right] \leq 2\delta\|\Psi_k\|\frac{\|\phi\|}{\sqrt{N}} \ =\ \tau_k\frac{\|\phi\|}{\sqrt{N}},$$

where $\tau_k = 2\delta\|\Psi_k\|$, the same constant as in Lemma 1. ■

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

$$E\left[\left|E_{\hat{\theta}_k}(c_j(X)) - E_{\theta'_k}(c_j(X))\right|\right] = E\left[\left|\langle \hat{b}_k - b'_k, c_j\rangle\right|\right],$$

for $j = 1,\ldots,m$. Taking summation over j, we obtain

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

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

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

Therefore,

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

where $\|\cdot\|_1$ denotes the L_1 norm on $\mathbb{R}^{n_x}$, $c = [c_1,\ldots,c_m]^T$, 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)

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

Taking expectation on both sides yields

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

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

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

Hence,

$$\left\|\frac{d}{d\theta}E_\theta[\phi(X)]\right\|_1 \leq \left(\sum_{i=1}^m\|c_i\|\right)\|\phi\|.$$

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.,

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

Taking expectation on both sides yields

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

where $d = \alpha\beta\sum_{j=1}^m\|c_j\|$. ■

Proof of Theorem 2: Applying Lemma 1, Assumption 6, and Lemma 3, we have that for each $k \in \mathbb{N}$

$$\begin{aligned} & E\left[\left|\langle b_k - f(\cdot,\hat{\theta}_k),\phi\rangle\right|\right] \\ \leq\ & E\left[|\langle b_k - b'_k,\phi\rangle|\right] + E\left[|\langle b'_k - f(\cdot,\theta'_k),\phi\rangle|\right]\ldots \\ & +\ E\left[\left|\langle f(\cdot,\theta'_k) - f(\cdot,\hat{\theta}_k),\phi\rangle\right|\right] \\ \leq\ & \left(\tau_k e_{k-1} + \epsilon + \frac{d\tau_k}{\sqrt{N}}\right)\|\phi\| \ =\ e_k\|\phi\|, \quad \forall\phi \in B(\mathbb{R}^{n_x}), \end{aligned}$$

where τ_k is the constant in Lemmas 1 and 3, d is the constant in Lemma 3, and ϵ is the constant in Assumption 6. It is easy to deduce by induction that

$$e_k = \tau_1^k e_0 + \left(\sum_{i=2}^k\tau_i^k + 1\right)\epsilon + \frac{d}{\sqrt{N}}\sum_{i=1}^k\tau_i^k,$$

where $\tau_i^k = \prod_{j=i}^k\tau_j$ for $k \geq i$, $\tau_i^k = 0$ for $k < i$. ■

ACKNOWLEDGEMENT

We thank the associate editor and anonymous reviewers for their careful reading of the paper and very constructive comments that led to a substantially improved paper. We thank one of the anonymous reviewers for bringing the work of Brooks and Williams [9] to our attention.

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] O. E. Barndorff-Nielsen, *Information and Exponential Families in Statistical Theory*, Wiley, New York, 1978.

[4] 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.

[5] D. P. Bertsekas, "Convergence of Discretization Procedures in Dynamic Programming", *IEEE Transactions on Automatic Control*, vol. 20, no. 3, pp. 415-419, 1975.

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

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

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

[9] A. Brooks, and S. Williams, "A Monte Carlo Update for Parametric POMDPs", *International Symposium of Robotics Research*, 2007.

[10] E. Brunskill, L. Kaelbling, T. Lozano-Perez, and N. Roy, "Continuous-State POMDPs with Hybrid Dynamics", in *Proceedings of the Tenth International Symposium on Artificial Intelligence and Mathematics*, 2008.

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

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

[13] 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.

[14] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo Methods In Practice*, Springer, New York, 2001.

[15] D. de Farias, and B. Van Roy, "The Linear Programming Approach to Approximate Dynamic Programming", *Operations Research*, vol. 51, no. 6, 2003.

[16] W. R. Gilks, "Derivative-Free Adaptive Rejection Sampling for Gibbs Sampling", in *Bayesian Statistics 4*, J. Bernardo, J. Berger, A. Dawid, and A. Smith, editors, pp. 641-649, Oxford University Press, Oxford, 1992.

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

[18] 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)*, vol. 140, no. 2, pp. 107–113, 1993.

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

[20] 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.

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

[22] 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.

[23] M. K. Murray, and J. W. Rice, *Differential Geometry and Statistics*, Chapman & Hall, New York, 1993.

[24] J. M. Porta, M. T. J. Spaan, and N. Vlassis, "Robot Planning in Partially Observable Continuous Domains", in *Proc. Robotics: Science and Systems*, 2005.

[25] J. M. Porta, N. Vlassis, M. T. J. Spaan, and P. Poupart, "Point-Based Value Iteration for Continuous POMDPs", *Journal of Machine Learning Research*, vol. 7, pp. 2329-2367, 2006.

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

[27] M. Resh, and P. Naor, "An Inventory Problem with Discrete Time Review and Replenishment by Batches of Fixed Size", *Management Science*, vol. 10, no. 1, pp. 109-118, 1963.

[28] N. Roy, "Finding Approximate POMDP Solutions through Belief Compression", *Ph.D. thesis*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2003.

[29] 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.

[30] 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.

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

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

Enlu Zhou (S'07-M'10) received the B.S. degree with highest honors in electrical engineering from Zhejiang University, China, in 2004, and received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2009. She is currently an Assistant Professor in the Department of Industrial and Enterprise Systems Engineering, at the University of Illinois at Urbana-Champaign. Her research interests include stochastic control, nonlinear filtering, and simulation optimization.

Michael C. Fu (S'89-M'89-SM'06-F'08) received a bachelor's degree in mathematics and bachelor's and master's degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1985, and a master's and Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA in 1986 and 1989, respectively. Since 1989, he has been with the University of Maryland, College Park, currently as Ralph J. Tyser Professor of Management Science in the Robert H. Smith School of Business, with a joint appointment in the Institute for Systems Research and an affiliate faculty appointment in the Department of Electrical and Computer Engineering. His research interests include simulation optimization and applied probability, with applications in manufacturing, supply chain management, and financial engineering. He served as the Simulation Area Editor of *Operations Research* from 2000–2005, and as the Stochastic Models and Simulation Department Editor of *Management Science* from 2006–2008. He is co-author (with J. Q. Hu) of the book, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications* (Kluwer, 1997), which received the INFORMS Simulation Society's Outstanding Publication Award in 1998, and co-author (with H. S. Chang, J. Hu, and S. I. Marcus) of the book, *Simulation-based Algorithms for Markov Decision Processes* (Springer, 2007).

Steven I. Marcus (S'70-M'75-SM'83-F'86) received the B.A. degree in electrical engineering and mathematics from Rice University in 1971 and the S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology in 1972 and 1975, respectively. From 1975 to 1991, he was with the Department of Electrical and Computer Engineering at the University of Texas at Austin, where he was the L.B. (Preach) Meaders Professor in Engineering. He was Associate Chairman of the Department during the period 1984-89. In 1991, he joined the University of Maryland, College Park, as Professor in the Electrical and Computer Engineering Department and the Institute for Systems Research. He was Director of the Institute for Systems Research from 1991 to 1996 and Chair of the Electrical and Computer Engineering Department from 2000 to 2005. Currently, his research is focused on stochastic control, estimation, and optimization.