Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

A Density Projection Approach to Dimension Reduction for Continuous-State POMDPs

Enlu Zhou, Michael C. Fu, and Steven I. Marcus

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 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. 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.

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 finite-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 [11] 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 [15] [19] [17] [6]. Thrun [19] addressed continuous-state POMDPs using particle filtering to simulate the propagation of belief states and represent the belief states by a 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 [17] proposed 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 the author, 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. [6] 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.

Motivated by the work of [19], [17], and [6], we develop a computationally tractable algorithm that effectively reduces the dimension of the belief state and also 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 projected 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 [17] and the parametric POMDP in [6], which considers only the family of Gaussian distributions. In addition, we provide theoretical results on the error bound of the value function and the performance of the policy generated by our method.

We also develop a projection particle filter for online filtering and decision making, by incorporating the density

E. Zhou is with the Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742 USA enluzhou@umd.edu

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

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

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.

II. Continuous-state POMDP

A discrete-time continuous-state POMDP can be formulated as:

$$\begin{aligned} x_{k+1} &= f(x_k, a_k, u_k), k = 0, 1, \ldots, \quad (1) \\ y_k &= h(x_k, a_{k-1}, v_k), k = 1, 2, \ldots, \quad (2) \\ y_0 &= h_0(x_0, v_0), \end{aligned}$$

where for all k, the state x_k is in a continuous state space $S \in R^{n_x}$, 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. 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 that $h_0(x, v)$ is continuous in x for every $v \in R^{n_x}$.

A *belief state* is defined as the conditional probability density of the current state x_k given the past history, i.e.,

$$b_k(x_k) = p_k(x_k | y_0, y_1, \ldots, y_k, a_0, a_1, \ldots, a_{k-1}).$$

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

$$b_{k+1}(x_{k+1}) \propto p(y_{k+1}|x_{k+1}, a_k) \int p(x_{k+1}|a_k, x_k) b_k(x_k) dx_k. \quad (3)$$

The righthand side of the above expression can be expressed in terms of b_k, a_k and y_{k+1}. Hence,

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

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

A POMDP can be converted to an MDP by conditioning on the history of observations and actions, 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. We assume the *one-step cost function* $g : S \times A \to R$ is bounded for all $(x, a) \in S \times A$. The *belief one-step cost function* $\tilde{g}(b_k, a_k)$ is related to $g(x, a_k)$ by

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

The objective is to find a policy π to minimize the *cost function*

$$J_\pi(b_0) = E_{\{Y_k\}} \left\{ \sum_{k=0}^{\infty} \gamma^k \tilde{g}(x_k, \mu_k(b_k)) \right\},$$

where $\gamma \in (0, 1)$ is the *discount factor*. The *dynamic programming (DP) mapping* to any bounded function $J : S \to R$ is defined by

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

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

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

For finite-state problems, the belief state b lies in a finite-dimensional probability simplex. For continuous state-space problems, the belief space is an infinite-dimensional space of continuous probability densities. The infinite dimensionality prohibits exact value iteration, and also imposes difficulty on applying the approximate algorithms that were developed for finite state-space POMDPs. 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.

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

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

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

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

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 $\{c_1(\cdot), \ldots, c_m(\cdot)\}$ 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*. θ is the parameter and $c(x)$ is the sufficient statistic of the probability density.

For an exponential family, the projection mapping in (6) can be carried out in analytical form, as shown below. Since

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

minimizing $D_{KL}(b\|f(\cdot, \theta))$ is equivalent to maximizing

$$\int \log f(x, \theta) b(x)dx = \int (\theta^T c(x) - \varphi(\theta)) b(x)dx. \quad (8)$$

Recalling the fact that the log-likelihood $l(\theta) = \theta^T c(x) - \varphi(\theta)$ is strictly concave in θ [13], 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,

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

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

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

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. For example, it can be solved using 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.

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 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.

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] [9]. As we have already shown, the belief state evolves recursively as (3). The integration in (3) 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

$$\hat{b}_k(x_k) = \sum_{i=1}^N w_k^i \delta(x_k - x_{k|k-1}^i), \quad (11)$$

where

$$w_k^i \propto p(y_k|x_{k|k-1}^i, a_{k-1}), \tag{12}$$

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}. This is the basic form of particle filtering, which is also called the bootstrap filter [10]. (Please see [1] for a rigorous and thorough derivation for a more general form of 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 near-optimal policy we obtained by solving the projected belief MDP is *not immediately* applicable. Therefore, we incorporate the idea of density projection into particle filtering, so as to approximate the belief state by a density in Ω.

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, letting $b = \hat{b}_k$ in (9) and plugging in (11), θ should satisfy

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

which constitutes the projection step in the projection particle filtering.

Algorithm 1: 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, 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 near-optimal policy solved for the projected belief MDP.

V. Analysis of Error Bounds

Assuming perfect computation of the original and projected belief states, our method solves the projected belief MDP instead of the original belief MDP. That raises two questions: 1. How well the optimal value function of the projected belief MDP approximates the true optimal value function, which is 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, which is measured by

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

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

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 certain mild conditions [4], [12].

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

$$|\langle g(\cdot,a), b - b^p\rangle| \le \epsilon_1,$$
$$|\langle g(\cdot,a), \psi(b,a,y) - \psi(b^p,a,y)^p\rangle| \le \delta_1.$$

Assumption 2 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 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 $|J_\mu(b) - J_\mu(b')| \le \tilde{\epsilon}, \forall \mu \in \Pi$.

Assumption 3 can be seen as a continuity property of the value function. Now we present our main result.

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

$$|J_*(b) - J_*^p(b)| \le \frac{\epsilon_1 + \gamma\epsilon_2}{1-\gamma}, \tag{13}$$

$$|J_*(b) - J_{\mu_*^p}(b)| \le \frac{2\epsilon_1 + \gamma(\epsilon_2 + \epsilon_3)}{1-\gamma}, \tag{14}$$

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 $\delta = \delta_1$. (Proof: [21].)

VI. Numerical Experiments

We consider an inventory control problem, where the observations are noisy, 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 arriving random 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.

Let x_k denote the inventory level, u_k the i.i.d. random demand, a_k the replenishment decision ($a_k = 1$ or 0), 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. We assume that the demand u_k and the observation noise v_k are both continuous random variables; hence the state x_k and the observation y_k are continuous. 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

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

We consider two objective functions: average cost per period and discounted total cost.

We compare our algorithm to two other algorithms: Certainty equivalence (CE) policy and Greedy policy. 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 $Q = 10$, 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 0.4.
- *Algorithm parameters*: We use the usual particle filter to obtain the mean estimate of the states for CE. 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.

Table I and Table II list the simulated average costs and discounted total cost under increasing observation noises, respectively. Each entry shows the simulated average cost/discounted total cost, and in the parentheses the percentage error from the average cost/discounted total cost using the optimal threshold policy under full observation. Our algorithm generally outperforms the other two algorithms under all observation noise levels. CE also performs very well, and the greedy policy is much worse. 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 and Fig. 2 show the actions taken by our algorithm as a function of the true inventory levels in the average cost case under small and large observation noise, respectively (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. When the observation noise is small, our algorithm yields a policy that picks actions very close to the optimal threshold policy (see Fig.1). As the observation noise increases, more actions picked by our policy violate the optimal threshold (see Fig. 2), 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 policy 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, not like here a simple threshold policy. 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.

TABLE I: Optimal average cost estimates using different policies. Each entry represents the average cost of a run of horizon 10^5 (Deviation above optimum under full observation in parentheses).

σ	Our method	CE policy	Greedy policy
0.1	12.849 (0.12%)	12.842 (0.06%)	25.454 (98.34%)
0.5	12.864 (0.23%)	12.867 (0.26%)	25.457 (98.36%)
0.9	12.904 (0.55%)	12.908 (0.57%)	25.450 (98.30%)
1.3	12.973 (1.08%)	12.977 (1.12%)	25.356 (97.57%)
1.7	13.066 (1.81%)	13.100 (2.07%)	25.324 (97.32%)
2.1	13.123 (2.25%)	13.183 (2.72%)	25.332 (97.38%)
2.5	13.250 (3.24%)	13.314 (3.74%)	25.402 (97.92%)
2.9	13.374 (4.21%)	13.458 (4.86%)	25.478 (98.52%)
3.3	13.512 (5.28%)	13.603 (6.00%)	25.655 (99.90%)

TABLE II: Optimal discounted cost estimates using different policies. Each entry represents the discounted cost on 1000 independent runs of horizon 40 (Deviation above optimum under full observation in parentheses).

σ	Our method	CE policy	Greedy policy
0.1	129.126 (13.57%)	129.120 (13.56%)	241.667 (112.55%)
0.5	129.097 (13.54%)	129.122 (13.57%)	242.656 (113.42%)
0.9	129.868 (14.22%)	129.593 (13.98%)	244.002 (114.61%)
1.3	130.336 (14.63%)	130.493 (14.77%)	245.673 (116.08%)
1.7	130.724 (14.98%)	130.952 (15.18%)	247.701 (117.86%)
2.1	131.778 (15.90%)	131.758 (15.88%)	249.452 (119.40%)
2.5	132.741 (16.75%)	132.536 (16.57%)	250.492 (120.31%)
2.9	133.484 (17.40%)	133.606 (17.51%)	250.811 (120.59%)
3.3	134.502 (18.30%)	134.807 (18.57%)	250.767 (120.56%)

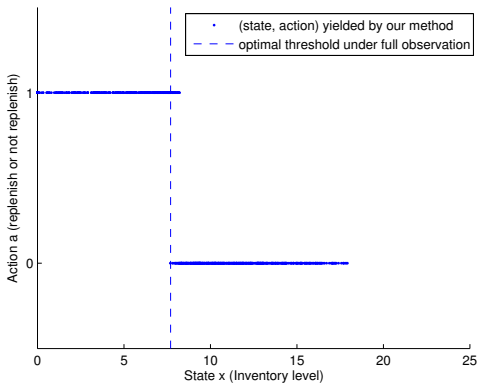

Fig. 1: When the observation noise is small ($\sigma = 0.1$), our method picks actions very closely to the optimal threshold policy for the full observation case.

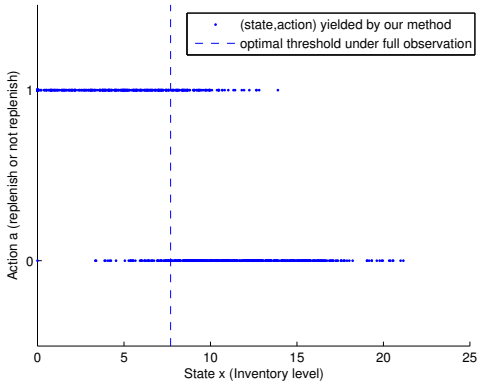

Fig. 2: When the observation noise is large ($\sigma = 3.1$), more actions picked by our method violate the optimal threshold for the full observation case.

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 exponential family, the orthogonal projection has an analytical form and can be carried out efficiently. An 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 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. 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.

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(2), 2002, pp. 174-188.

[2] B. Azimi-Sadjadi, and P. S. Krishnaprasad, "Approximate Nonlinear Filtering and its Application in Navigation," *Automatica*, vol. 41(6), 2005, pp. 945-956.

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

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

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

[6] A. Brooks, A. Makarenkoa, S. Williamsa, and H. Durrant-Whytea, "Parametric POMDPs for Planning in Continuous State Spaces", *Robotics and Autonomous Systems* , vol. 54(11), 2006, pp. 887-897.

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

[8] D. Crisan, and A. Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners", *IEEE Transaction on Signal Processing*, vol. 50(3), 2002, pp. 736-746.

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

[10] 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.

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

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

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

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

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

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

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

[18] R. D. Smallwood, and E. J. Sondik, "The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon", *Operations Research*, vol. 21(5), 1973, pp. 1071-1088.

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

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

[21] E. Zhou, M.C. Fu, and S.I. Marcus, "Solving Continuous-State POMDPs via Density Projection", *TR 2008-6*, Institute for Systems Research, University of Maryland, College Park, MD, 2008.