Bayesian Reinforcement Learning in Continuous POMDPs with Application to Robot Navigation

Stéphane Ross[1], Brahim Chaib-draa[2] and Joelle Pineau[1]

[1]School of Computer Science
McGill University
Montreal, Canada
{sross12,jpineau}@cs.mcgill.ca

[2]Department of Computer Science
Laval University
Quebec City, Canada
chaib@ift.ulaval.ca

***Abstract*— We consider the problem of optimal control in continuous and partially observable environments when the parameters of the model are not known exactly. Partially Observable Markov Decision Processes (POMDPs) provide a rich mathematical model to handle such environments but require a known model to be solved by most approaches. This is a limitation in practice as the exact model parameters are often difficult to specify exactly. We adopt a Bayesian approach where a posterior distribution over the model parameters is maintained and updated through experience with the environment. We propose a particle filter algorithm to maintain the posterior distribution and an online planning algorithm, based on trajectory sampling, to plan the best action to perform under the current posterior. The resulting approach selects control actions which optimally trade-off between 1) exploring the environment to learn the model, 2) identifying the system's state, and 3) exploiting its knowledge in order to maximize long-term rewards. Our preliminary results on a simulated robot navigation problem show that our approach is able to learn good models of the sensors and actuators, and performs as well as if it had the true model.**

I. INTRODUCTION

Many robot planning and control problems are characterized by uncertainty inherent in the robot's sensors and actuators. Partially Observable Markov Decision Processes (POMDPs) allow to take these uncertainties into account during the action selection process, such that optimal actions (or controls) are selected. Many researchers have proposed efficient algorithms for planning in large scale domains, provided an exact model of the robot's sensors and actuators [1], [2], [3], [4], [5]. These approaches are unfortunately of limited use when models of the robot's sensors and dynamics are poor or unavailable. A few approaches have been proposed to cope with domains lacking such a model [6], [7], [8], [9] but these approaches usually require very large amounts of data, and do not address the problem of how to gather this data efficiently, or how to compose with partially specified models during the planning phase.

Bayesian reinforcement learning approaches [10], [11], [12] have successfully address the joint problem of optimal action selection under parameter uncertainty. In Bayesian reinforcement learning, the robot starts with a prior distribution over model parameters, the posterior distribution is updated as the robot interacts with its environment, and action selection is optimized with respect to the posterior over model parameters. This framework is particularly interesting as it provides a theoretically optimal solution to the well-known exploration-exploitation problem in reinforcement learning, i.e. finding a policy which, given a prior distribution over model parameters, will maximize expected return over a finite (or infinite) planning horizon. Recent work has extended these techniques to domains where the state can only be partially inferred through an observation function [13], as well as applying these ideas to simple robot tasks [14]. However these methods have thus far been limited to domains with finite state and action spaces.

The main contribution of this paper is to extend the bayes-optimal reinforcement learning framework to the case of multi-dimensional continuous POMDPs. Our approach assumes that the dimensionnality of the state, action and observation spaces are known, and that the robot can be modeled as a general Gaussian system (not necessarily linear). Normal-Wishart priors are used to model the prior knowledge of the unknown means and covariance matrices. The posterior distribution is approximated by a finite mixture using a particle filter algorithm. Planning is conducted online, using the current posterior distribution, by sampling sequences of actions and observations, and selecting actions which maximize rewards over a fixed planning horizon. While we assume the reward function is known, and dynamics are Gaussian, the approach can be easily extended to cases where these assumptions are removed, as long as a family of distributions specifying these functions can be defined. We validate our approach on a simple robot navigation problem. Results indicate that our approach is able to learn a good model of the system, and achieves near-optimal performance after a short learning time.

II. BACKGROUND

A. Partially Observable Markov Decision Processes

A POMDP is defined by a set of states S, a set of actions A and a set of observations Z. The dynamics of the system is specified by a discrete-time transition function $T : S \times A \times S \rightarrow [0, \infty]$, where $T(s, a, s') = f(s'|s, a)$ defines the conditional probability density over the next state s' of the system, given the current state is s and action

a was executed. The perception of the system's state is specified by the observation function $O : S \times A \times Z \rightarrow [0, \infty]$, where $O(s', a, z) = f(z|s', a)$ defines the conditional probability density over the observation z obtained when entering next state s' after doing action a. For convenience of notation, we can define a joint transition-observation function: $P(s, a, s', z) = T(s, a, s')O(s', a, z)$. Finally, the reward function $R : S \times A \rightarrow \mathbb{R}$ specifies the reward obtained by the agent at each time step. The system is assumed to be time-invariant.

While the state is only partially observable, it can be tracked using Baye's rule. Starting from a prior probability density function (p.d.f.) b_0 over the initial state of the environment, the conditional p.d.f. over the current state of the environment given the entire history of actions and observations, called the belief state, is computed as follows:

$$b_t(s') = \frac{1}{f(z_t|b_{t-1}, a_{t-1})} \int_S P(s, a_{t-1}, s', z_t) b_{t-1}(s) ds, \tag{1}$$

where $f(z|b, a) = \int_S \int_S P(s, a, s', z) b(s) ds ds'$ is the conditional probability density of observing z after doing action a in belief b. $f(z_t|b_{t-1}, a_{t-1})$ acts as a normalization constant such that $\int_S b_t(s) ds = 1$.

The goal of the robot, when modeled as a POMDP, is to choose an action selection strategy which maximizes its expected sum of discounted rewards over the infinite horizon. A policy that achieves this (for any given belief b) can be computed by solving Bellman's equation (Eqn 2):

$$V^*(b) = \max_{a \in A} \left[\int_S R(s, a) b(s) ds + \gamma \int_Z f(z|b, a) V^*(b^{a,z}) dz \right] \tag{2}$$

where $b^{a,z}$ is the next belief state obtained after performing action a in belief b and then observing z.

The best action for a given belief b follows directly from Eqn 2 (simply replacing the $\max$ by an argmax).

In the continuous case, we consider $S \subseteq \mathbb{R}^m$, $A \subseteq \mathbb{R}^n$ and $Z \subseteq \mathbb{R}^p$, where m, n and p represent respectively the dimensionality of the state, action and observation spaces. We assume, as in [15], that m, n and p are finite and that the action space A is bounded. For the transition function, we assume a Gaussian model (not necessarily linear), such that $s_t = g_T(s_{t-1}, a_{t-1}, V_{t-1})$ where $V_t \sim N_k(\mu_v, \Sigma_v)$, a k-variate Normal distribution with mean vector μ_v and covariance matrix Σ_v, and g_T is a function yielding s_t given the previous state s_{t-1}, the previous action a_{t-1} and vector random variable V_{t-1}. Here we assume that given a state s and action a, the function $g_{T|s,a}(v) = g_T(s, a, v)$ is a 1-1 mapping from $\mathbb{R}^k$ to S, such that its inverse $g^{-1}_{T|s,a}(s')$ exists. Similarly, we also assume that the observation function is specified by a Gaussian model of the form $z_t = g_O(s_t, a_{t-1}, W_t)$, where $W_t \sim N_l(\mu_w, \Sigma_w)$ and g_O is a function yielding the observation z_t given the current state s_t, previous action a_{t-1} and vector random variable W_t such that $g_{O|s',a}(w) = g_O(s', a, w)$ is invertible. The framework we propose can be easily extended to other families of distributions, assuming the posterior distribution can be computed. Note that any linear model (e.g. additive Gaussian noise) satisfies the assumptions we make on g_T and g_O.

B. Bayesian Reinforcement Learning

The standard Bayesian reinforcement learning framework uses Dirichlet distributions to represent the prior and posterior distributions over the unknown transition probabilities defining the model [10], [11], [12]. Dirichlet distributions are convenient because they represent the probability that some random variable follows a particular discrete distribution, given the number of times each event has been observed thus far. Dirichlet parameters can be estimated exactly by simply counting the number of times each state transition occurred. Planning is achieved by specifying an extended MDP model, called Bayes-Adaptive MDP (BAMDP), where the Dirichlet distribution parameters are included in the state space. The transition function models how these parameters are updated given a particular state transition.

When modeling a robot domain as a finite POMDP, Dirichlet distributions can also be used to represent the prior distribution over the unknown transition and observation probabilities. However, an added complication arises due to the fact that the state is not observable, therefore it is not possible to know the exact values of the Dirichlet parameters. To overcome this problem, some approaches assume access to an oracle which can be queried to reveal the exact state of the environment, and thus know exactly which Dirichlet parameters should be udpated [14]. This assumption is difficult to meet in many domains. Fortunately, an oracle is not necessary since Baye's rule can be applied to compute the distribution over both the current state and the current values of the Dirichlet parameters, yielding a posterior over POMDP models that is represented by a mixture of Dirichlet distributions [13]. Such methods have been shown to learn good models of simple POMDP problems, without an oracle.

However, these approaches cannot be directly applied in continuous spaces since Dirichlet distributions are limited to discrete state spaces. In the continuous case, we need to turn to other families of distributions to define the transition and observation functions. In this paper, we focus on the case where these functions depend on multivariate normal random variables, with unknown mean vector and covariance matrices. In such cases, the prior and posterior distribution can be represented using a Normal-Wishart distribution.

C. Normal-Wishart Distribution

In multivariate statistics, the Wishart distribution defines the distribution of the unbiased estimator of the covariance matrix of a multivariate normal distribution. It is parametrised by a degree of freedom n and a covariance matrix Σ. However, it is often more convenient to express its density function in terms of a precision matrix $\tau = \Sigma^{-1}$:

$$f(V|\tau, n) \propto |\tau|^{n/2} |V|^{(n-k-1)/2} \exp(\frac{-1}{2} Tr(\tau V)). \tag{3}$$

The concept of precision matrix $\tau = \Sigma^{-1}$ (the inverse of the covariance matrix) can also be used to define the p.d.f of a multivariate normal distribution with mean vector μ:

$$f(x|\mu,\tau) \propto |\tau|^{1/2} \exp(\frac{-1}{2}(x-\mu)^T\tau(x-\mu)). \quad (4)$$

A normal-Wishart distribution is the product of a multivariate normal and Wishart distributions. It is used in bayesian statistics to represent the joint prior/posterior distribution over the unknown mean vector μ and unknown precision matrix τ of a normally distributed vector random variable $X \sim N_k(\mu, \tau^{-1})$. It is parametrised by four parameters $(\hat{\mu}, \nu, \alpha, \hat{S})$ such that the density over the mean vector $\mu = m$ and precision matrix $\tau = t$ is defined as:

$$f(m,t|\hat{\mu},\nu,\alpha,\hat{S}) = f(m|\hat{\mu},\nu t)f(t|\hat{S},\alpha) \propto |t|^{(\alpha-k)/2} \exp(\tfrac{-1}{2}Tr([\hat{S} + \nu(m-\hat{\mu})(m-\hat{\mu})^T]t)). \quad (5)$$

where $f(m|\hat{\mu},\nu t)$ represents the normal density (Eqn 4) on $\mu = m|\tau = t$ and $f(t|\hat{S},\alpha)$ the Wishart density (Eqn 3) on $\tau = t$.

An important result [16], states that if X follows a multivariate normal distribution with unknown mean vector μ and unknown precision matrix τ, and that the prior joint distribution on (μ,τ) is a normal-Wishart distribution with parameters $(\hat{\mu},\nu,\alpha,\hat{S})$, then the posterior joint distribution on (μ,τ) after observing $X = x$ is also a normal-Wishart distribution with parameters $(\hat{\mu}',\nu',\alpha',\hat{S}')$ defined as follows:

$$\begin{aligned} \hat{\mu}' &= \tfrac{\nu\hat{\mu}+x}{\nu+1}, \\ \nu' &= \nu+1, \\ \alpha' &= \alpha+1, \\ \hat{S}' &= \hat{S} + \tfrac{\nu}{\nu+1}(\hat{\mu}-x)(\hat{\mu}-x)^T. \end{aligned} \quad (6)$$

As we can see, the posterior distribution can be updated quite easily online as new observations are made. The way to interpret these parameters is that $\hat{\mu}$ is the sample mean, ν is the number of samples taken to compute the sample mean, while $\hat{S}$ and α relates to the sample covariance in that $\hat{S}$ is the scatter matrix of the samples and α is such that $\frac{\hat{S}}{\alpha}$ is the sample covariance. Hence specifying a prior for a multivariate normal distribution with unknown mean and unknown precision matrix can be seen as creating an artificial set of samples and taking the sample mean and sample covariance from this set to define the normal-Wishart prior parameters.

III. BAYES-ADAPTIVE CONTINUOUS POMDP

Our objective is to provide an optimal approach for decision-making under model and state uncertainty in continuous domains. More concretely, the goal is to be able to choose optimal action according to the posterior distributions over unknown means and precision matrices defining the POMDP model. To achieve this, we adopt the Bayesian RL perspective, and begin by creating an extended POMDP model, called the *Bayes-Adaptive Continuous POMDP* (BACPOMDP), in which the normal-Wishart parameters are included in the state space. Thus the belief update operation in the BACPOMDP model (Eqn 1) will compute a posterior over both the state of the system and the normal-Wishart parameters. This allows us to track how the parameters evolve as actions and observations are made in the environment, thereby allowing us to consider the value of learning information about the environment, as an integral part of the planning.

A. Model Definition

Let $\mathcal{NW}_k$ be the space of normal-Wishart parameters for the vector random variable V defining the transition probabilities, i.e. 4-tuples $< \hat{\mu},\nu,\alpha,\hat{S} > \in \mathbb{R}^k \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{k^2}$, where k is the dimensionality of V, and $\mathcal{NW}_l$ be the space of normal-Wishart parameters for the vector random variable W defining the observation probabilities, where l is the dimensionality of W. Let $\mathcal{U}$ be the update function, such that for some $< \hat{\mu},\nu,\alpha,\hat{S} > \in \mathcal{NW}_k$, and some observation $v \in \mathbb{R}^k$ then $< \hat{\mu}',\nu',\alpha',\hat{S}' > = \mathcal{U}(\hat{\mu},\nu,\alpha,\hat{S},v)$, where $< \hat{\mu}',\nu',\alpha',\hat{S}' >$ are defined as in equation 6. Similarly, for any $\psi \in \mathcal{NW}_l$ and $w \in \mathbb{R}^l$, then $\mathcal{U}(\psi,w)$ updates the normal-Wishart parameters in ψ with new observation w, according to equation 6.

The BACPOMDP is defined as follows: the new state space $S' = S \times \mathcal{NW}_k \times \mathcal{NW}_l$, where S is the original state space of the POMDP with unknown model parameters, and the new action and observation space are the same as in the original POMDP. To avoid confusion, we refer to these extended states as hyperstates.

The transition-observation function P' of the BACPOMDP specify the conditional probability density $f(s',\phi',\psi',z|s,\phi,\psi,a)$ of moving from one hyperstate (s,ϕ,ψ) to another hyperstate (s',ϕ',ψ') by doing some action a, and then observing z after such a transition. Using our assumptions on g_T and g_O, we can recover the values of V and W as follows: $v = g^{-1}_{T|s,a}(s')$ and $w = g^{-1}_{O|s',a}(z)$. Hence P' must ensure that the parameters ϕ,ψ are updated properly for this transition: i.e. $\phi' = \mathcal{U}(\phi, g^{-1}_{T|s,a}(s'))$ and $\psi' = \mathcal{U}(\psi, g^{-1}_{O|s',a}(z))$, otherwise the probability density of transition is 0. Given that these parameters are well updated, we have that the joint density $f(s',z|s,a,\phi,\psi) = f(s'|s,a,\phi)f(z|s',a,\psi) = f_V(g^{-1}_{T|s,a}(s')|\phi)J_{T|s,a}(s')f_W(g^{-1}_{O|s',a}(z)|\psi)J_{O|s',a}(z)$. Here $f_V(g^{-1}_{T|s,a}(s')|\phi)$ is the conditional p.d.f. of V given normal-Wishart posterior ϕ and $J_{T|s,a}(s')$ is the *jacobian*[1] *of the 1-1 transformation* $g^{-1}_{T|s,a}$ evaluated at s'; similarly for $f_W(g^{-1}_{O|s',a}(z)|\psi)$ and $J_{O|s',a}(z)$.

Hence, we define the joint transition-observation function

[1]The absolute value of the determinant of the Jacobian matrix.

(analogous to the one in Sec. II-A) as:

$$P'(s,\phi,\psi,a,s',\phi',\psi',z) = \\ f_V(g^{-1}_{T|s,a}(s')|\phi)J_{T|s,a}(s')f_W(g^{-1}_{O|s',a}(z)|\psi)J_{O|s',a}(z) \\ \text{Given } \phi' = \mathcal{U}(\phi, g^{-1}_{T|s,a}(s')),\ \psi' = \mathcal{U}(\psi, g^{-1}_{O|s',a}(z)). \tag{7}$$

Note that $f_V(v|\phi) = \int_{\mathbb{R}^k}\int_{\mathbb{R}^{k^2}} f(v|m,t)f(m,t|\phi)dtdm$ where $f(v|m,t)$ represents the multivariate normal density function (Eqn 4) and $f(m,t|\phi)$ the normal-Wishart density function (Eqn 5). $f_W(w|\psi)$ is obtained similarly by integrating over all mean vector and precision matrix.

The reward function of the BACPOMDP is taken directly from the original POMDP: $R'(s,\phi,\psi,a) = R(s,a)$ (though it could be learned through Bayesian updating as well).

The tuple $(S', A, Z, P', R', \gamma)$ defines formally the BACPOMDP. If $b_0 \in \Delta S$ is the initial belief of the original POMDP, and that $\phi_0 \in \mathcal{NW}_k$ and $\psi_0 \in \mathcal{NW}_l$ are the normal-Wishart prior parameters, then the initial belief of the BACPOMDP is defined as $b'_0(s,\phi,\psi) = b_0(s)I_{\phi_0}(\phi)I_{\psi_0}(\psi)$, where I is the indicator function. Here, the belief state in the BACPOMDP is a distribution over both state of the environment and values of the normal-Wishart parameters. The model of the POMDP is effectively learned by monitoring the belief state of the BACPOMDP.

Note that the BACPOMDP has a known model and is an instance of a continuous POMDP. Therefore the belief update (Eqn 1) and Bellman equation (Eqn 2), can be applied directly to update the belief and compute the value function of the BACPOMDP. Of course computing these complex integrals in closed-form will usually be intractable. Thus the next section describes a sampling-based approximation for monitoring the belief, followed in the subsequent section by an online planning approach suitable for the BACPOMDP model.

B. Belief updating in the BACPOMDP via particle filtering

Particle filters using Monte Carlo sampling methods have been widely used for sequential state estimation in POMDPs [17], [18]. Given a current belief b, action a and observation z, a standard approach to estimate the next belief b', after a and z are made, is to sample K states from the distribution b, and for each of the sampled state s, sample a successor state s' according to the distribution $T(s,a,\cdot)$ and add probability density $O(s',a,z)$ to $b'(s')$. b' is renormalized at the end so that it represents a probability distribution, over at most K particles.

Applying the same principles to the BACPOMDP, the next belief b' obtained after doing action a in belief b and observing z can be estimated via by particle filter described in Algorithm 1.

Sampling a precision matrix t and mean vector m from a normal-Wishart distribution with parameters $(\hat{\mu}, \nu, \alpha, \hat{S})$ is achieved by first sampling t from a Wishart distribution with α degrees of freedom and precision matrix $\hat{S}$ and then sampling m from a multivariate normal distribution with mean $\hat{\mu}$ and precision matrix νt. Details on how to sample these distributions can be found in [19]. Note that these methods require the covariance matrices for both the multivariate normal and the Wishart distribution, hence these procedures must be executed with covariance $\hat{S}^{-1}$ to sample the Wishart distribution and covariance $\frac{t^{-1}}{\nu}$ to sample the multivariate normal distribution.

Algorithm 1 PARTICLEFILTER(b, a, z, K)

1: Define b' as a 0 vector.
2: $\eta \leftarrow 0$
3: **for** $i = 1$ to K **do**
4: Sample hyperstate (s,ϕ,ψ) from distribution b.
5: Sample (m,t) from normal-Wishart parametrised by ϕ.
6: Sample v from multivariate normal distribution $N_k(m,t)$.
7: Compute successor state $s' = g_T(s,a,v)$.
8: Compute $w = g^{-1}_{O|s',a}(z)$.
9: Compute $\phi' = \mathcal{U}(\phi, v)$ and $\psi' = \mathcal{U}(\psi, w)$.
10: Sample (m',t') from Normal-Wishart parametrised by ψ.
11: Add density $f(w|m',t')J_{O|s',a}(z)$ (Eqn 4) to (s',ϕ',ψ') in b'.
12: $\eta \leftarrow \eta + f(w|m',t')J_{O|s',a}(z)$
13: **end for**
14: **return** $\eta^{-1}b'$

The complexity of generating a single new particle is in $O(\log K + k^3 + l^3 + C_T + C_O)$, where k is the dimensionality of V, l the dimensionality of W, C_T the complexity of evaluating $g_T(s,a,v)$ and C_O the complexity of evaluating $g^{-1}_{O|s',a}(z)$. Sampling a hyperstate from b is in $\log K$, as b can be maintained as a cumulative distribution, and the k^3 and l^3 complexity comes from the inversion of precision matrices and the sampling procedure for the normal-Wishart distribution. Hence performing a belief update with K particles is achieved in $O(K(\log K + k^3 + l^3 + C_T + C_O))$.

C. Online planning in the BACPOMDP

To ensure tractability, we focus on online methods for action selection, which means that we try to find the optimal action (over a fixed planning horizon) for the current belief state. Several online planning algorithms have been developed for finite POMDPs [5], [20], [21]. Most of these require complete enumeration of the action and observation spaces, which cannot be done in our continuous setting. For this reason, we adopt a sampling-based approach [22], [20]. Algorithm 2 provides a brief outline of the online planning method.

At each time step, $V(b, D, M, N, K)$ is executed with current belief b and then action $bestA$ is performed in the environment. The algorithm proceeds by recursively expanding a tree of reachable beliefs by sampling uniformly a subset of M actions and a subset of N observations at each belief node, until it reaches a tree of depth D. The particle filter is used to approximate the belief states. Sampling the observation from $f(z|b,a)$ is achieved similarly to how the particle filter works, i.e. from Algorithm 1: proceed with lines 4-7 to obtain a successsor state s', then do line 10 to sample a mean m' and precision matrix t' from normal-Wisart posterior ψ, draw w from $N_l(m',t')$ and then the sampled observation is $g_O(s',a,w)$. An approximate value function $\hat{V}$ is used at the fringe node to approximate the

Algorithm 2 $\mathrm{V}(b, d, M, N, K)$

if $d = 0$ **then**
  return $\hat{V}(b)$
end if
$maxQ \leftarrow -\infty$
for $i = 1$ to M **do**
  Sample a uniformly in A
  $q \leftarrow \sum_{(s,\phi,\psi)} b(s, \phi, \psi) R(s, a)$
  for $j = 1$ to N **do**
    Sample z from $f(z|b, a)$
    $b' \leftarrow \textsc{ParticleFilter}(b, a, z, K)$
    $q \leftarrow q + \frac{\gamma}{N} \mathrm{V}(b', d-1, M, N, K)$
  end for
  if $q > maxQ$ **then**
    $maxQ \leftarrow q$
    $maxA \leftarrow a$
  end if
end for
if $d = D$ **then**
  $bestA \leftarrow maxA$
end if
return $maxQ$

value V^* of the fringe beliefs. The fringe nodes' values are propagated to the parents' nodes using an approximate version of Bellman equation, where the maximization is taken over sampled actions, and expectation over future rewards is taken over sampled observations. This yields a value estimate for each sampled action at the current belief. The action with highest value estimate is stored in the variable $bestA$. After executing $bestA$ in the environment, the agent updates its current belief b with the new observation z obtained using the particle filter. The planning algorithm is then run again on this new belief to compute the next action to take.

The overall complexity of this algorithm is in $O((MN)^D(C_p + C_v))$, where C_p denotes the complexity of doing the particle filter update, as given in the previous section, and C_v denotes the complexity of evaluating the approximate value function $\hat{V}$. A nice property of our approach is that the complexity depends almost entirely on user specified parameters, such that the parameters (M,N,D,K) can be adjusted to meet problem specific real-time constraints.

Recent analysis has shown that it is possible to achieve ϵ-optimal performance with an online POMDP planner [21] by using lower and upper bounds on V^* at the fringe node. In our case, the use of sampling and particle filtering to track the belief over state and model introduces additional error that prevents us from guaranteeing lower and upper bounds. However, it may still be possible to guarantee ϵ-optimality with high probability, provided that one chooses sufficiently large M,N,D and K, as was shown for the particular cases of discrete MDPs and POMDPs by [22], [20]. That remains an open question for our particular framework.

IV. EXPERIMENTS

To validate our approach, we experimented on a simple simulated robot navigation problem where the simulated robot must learn the drift induced by its imperfect actuators and the noise of its sensors. The robot moves in an open 2D area and tries to reach a specific goal location. We considered the state to be the robot's (x, y) position; actions are defined by (d, θ), where $d \in [0, 1]$ relates to the displacement and $\theta \in [0, 2\pi]$ is the angle toward which the robot moves; the observations correspond to the robot's position with additive Gaussian noise, i.e. $g_O(s, a, w) = s + w$. The dynamics of the robot are assumed to be of the form:

$$g_T(s, < d, \theta >, v) = s + d \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} v$$

The exact parameters of the normal distributions for v are the mean, $\mu_v = (0.8; 0.3)$, and covariance $\Sigma_v = [0.04, -0.01; -0.01, 0.01]$. Similarly, the observation noise, w, is parameterized by $\mu_w = (0; 0)$ and $\Sigma_w = [0.01, 0; 0, 0.01]$. Both v and w must be estimated from data. The robot starts with the incorrect assumption that for v, $\hat{\mu}_v = (1, 0)$ and $\hat{\Sigma}_v = [0.04, 0; 0, 0.16]$, and for w, $\hat{\mu}_w = (0; 0)$ and $\hat{\Sigma}_w = [0.16, 0; 0, 0.16]$. The normal-Wishart prior parameters used for v are the tuple $\phi_0 = (\hat{\mu}_v, 10, 9, 9\hat{\Sigma}_v)$ and for w, $\psi_0 = (\hat{\mu}_w, 10, 9, 9\hat{\Sigma}_w)$. This is equivalent to giving an initial sample of 10 observations to the robot, in which the random variable v has sample mean $\hat{\mu}_v$ and sample covariance $\hat{\Sigma}_v$ and the variable w has sample mean $\hat{\mu}_w$ and sample covariance $\hat{\Sigma}_w$.

Initially the robot starts at the known position $(0; 0)$, and the goal is a circular area of radius 0.25 unit where the center position is randomly picked at a distance of 5 units. As soon as the robot reaches a position inside the goal, a new goal center position is chosen randomly (within a distance of 5 units from the previous goal). The robot always knows the position of the current goal, and receives a reward of 1 when it reaches it. A discount factor $\gamma = 0.85$ is used.

For the planning, we used a horizon of $D = 1$, and sampled $M = 10$ actions, $N = 5$ observations and $K = 100$ particles to maintain the belief. The approximate value function $\hat{V}$ at the planning fringe was computed as $\hat{V}(b) = \sum_{(s,\phi,\psi)} b(s, \phi, \psi)\gamma^{G(s,\phi)}$, where $G(s, \phi)$ is the number of steps required to reach the goal from s if the robot moves in a straight-line towards the goal with distance $||\phi_{\hat{\mu}}||_2$ per step.

The average return as a function of the number of training episodes (averaged over 1000 episodes) is plotted in figure 1. We also compare it to the average return obtained by planning only with the prior (with no learning), and planning with the exact model, using the same parameters for M,N,D,K.

As we can see, our approach is able to quickly reach performance very close to the case where it was given the exact model. Average running time for the planning was 0.074 seconds per time step on an Intel Xeon 2.4Ghz processor.

To measure the accuracy of the learned model, we computed a weighted L1-distance of the estimate (sample mean and sample covariance) in each particle compared to the true model parameters as follows:

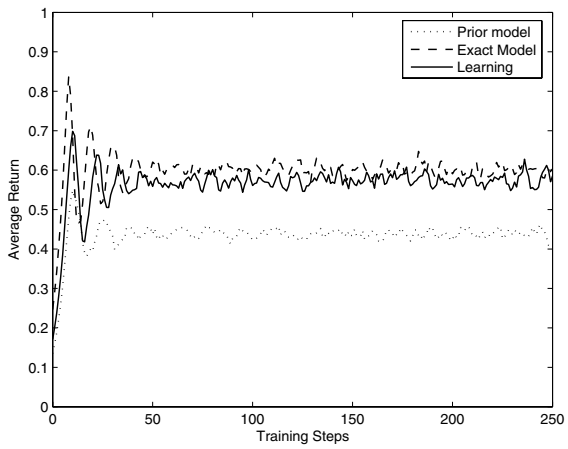

Fig. 1. Average return as a function of the number of training steps.

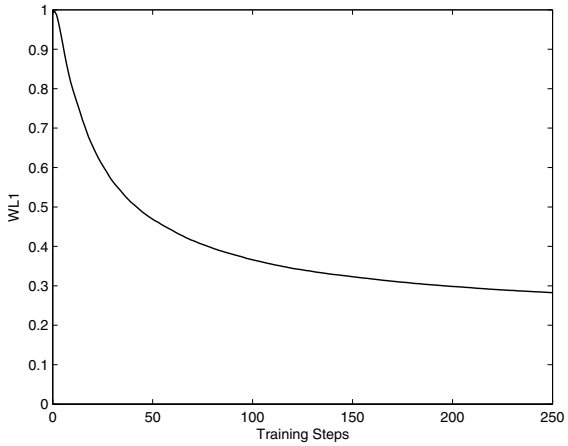

Fig. 2. Average weighted L1-distance as a function of the number of training steps.

$$WL1(b) = \sum_{(s,\phi,\psi)} b(s,\phi,\psi) \left[||\phi_{\hat{\mu}} - \mu_v||_1 + ||\phi_{\hat{S}}/\phi_\alpha - \Sigma_v||_1 + ||\psi_{\hat{\mu}} - \mu_w||_1 + ||\psi_{\hat{S}}/\psi_\alpha - \Sigma_w||_1 \right] \quad (8)$$

In figure 2, we show how the average weighted L1-distance to the true model evolves over 250 steps.

We observe that the weighted L1-distance decreases quickly and thus the robot is able to quickly improve the model of its actuators and sensors through selected learning.

V. RELATED WORK

The problem of optimal control under uncertain model parameters was originally introduced by Feldbaum [23], as the theory of dual control, also sometimes refered to as adaptive control or adaptive dual control. Extensions of this theory has been developed for time-varying systems [24]. Several authors have studied this problem for different kinds of dynamical systems : linear time invariant systems under partial observability [25], linear time varying Gaussian models under partial observability [26], nonlinear systems with full observability [27], and more recently a similar approach to ours, using particle filters, has been proposed for nonlinear systems under partial observability [28]. Our proposed approach differs from [28] in that we use normal-Wishart distributions to maintain the posterior, and use a different particle filtering algorithm. Furthermore, their planning algorithm proceeds by evaluating a particular policy on the underlying MDP defined by each particle, and then averaging the value of the policy over all particles. Contrary to our approach, this does not reflect the value of information gained by actions that help identify the state or the parameters of the model, as it does not consider how the posterior distribution evolves in the future, for different actions and observations.

VI. DISCUSSION

The problem of optimal control in stochastic and partially observable environments with unknown or uncertain model parameters is a very important problem that commonly arizes in practice, as the mathematical model of the system is rarely known exactly. We have proposed a new Bayesian approach to solve this problem, based on particle filtering to maintain the posterior over states and models, and online planning, using trajectory sampling, to find the best action. The method we propose is able to trade-off between: (1) exploration to learn model parameters, (2) identification of the system's state, and (3) exploitation to gather rewards; such that we maximize return over the infinite horizon. The approach requires significant computation, but the complexity of this can be flexibly managed by increasing or decreasing precision of the approximation, in order to meet problem specific real-time constraints. Our preliminary experimental results have shown that our approach is able to achieve good control performance on a simple robot-navigation problem, while learning online a good model of the system.

While the model we have considered uses Gaussian distributions to model its dynamics, our approach can be generalized to other types of distributions. It is particularly appropriate for domains where it is possible to use the conjugate family of distributions to define the prior and maintain the posterior. The BACPOMDP method can also be generalized to learn the reward function, provided information on this function is contained in the observation space of the system.

Many interesting research problems remain open. In particular, we would like to investigate how to deal with such problems when the form of the functions g_T and g_O, that specify the dynamics of the transition and observation function, are unknown. It would also be interesting to adapt our approach to time-varying systems, i.e. systems where the parameters of the model may change with time. This often happens in practice due to aging sensors and actuators. Finally, we would also like to develop more efficient planning algorithms that can better scale to larger domains. On a more practical level, we intend to implement our approach on more complex problems, such as the control of a robotic

wheelchair, which we are developing for people with disabilities.

REFERENCES

[1] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: an anytime algorithm for POMDPs," in *IJCAI*, Acapulco, Mexico, 2003, pp. 1025–1032.

[2] M. Spaan and N. Vlassis, "Perseus: randomized point-based value iteration for POMDPs," *JAIR*, vol. 24, pp. 195–220, 2005.

[3] T. Smith and R. Simmons, "Point-based POMDP algorithms: improved analysis and implementation," in *UAI*, 2005.

[4] R. Ross and B. Chaib-draa, "AEMS: An Anytime Online Search Algorithm for Approximate Policy Refinement in Large POMDPs," in *IJCAI*, 2007.

[5] S. Paquet, L. Tobin, and B. Chaib-draa, "An online POMDP algorithm for complex multiagent environments," in *AAMAS*, 2005.

[6] A. McCallum, "Instance-based utile distinctions for reinforcement learning with hidden state," in *International Conference on Machine Learning*, 1995, pp. 387–395.

[7] S. Koenig and R. Simmons, "Unsupervised learning of probabilistic models for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996.

[8] J. Baxter and P. Bartlett, "Direct gradient-based reinforcement learning," 1999.

[9] E. Even-Dar, S. M. Kakade, and Y. Mansour, "Reinforcement learning in pomdps without reset," in *IJCAI*, 2005.

[10] R. Dearden, N. Friedman, and N. Andre, "Model based bayesian exploration," in *UAI*, 1999.

[11] M. Duff, "Optimal learning: Computational procedure for bayes-adaptive markov decision processes," Ph.D. dissertation, University of Massachusetts, Amherst, USA, 2002.

[12] P. Poupart, N. Vlassis, J. Hoey, and K. Regan, "An analytic solution to discrete bayesian reinforcement learning," in *Proc. ICML*, 2006.

[13] S. Ross, J. Pineau, and B. Chaib-draa, "Bayes-adaptive pomdps," in *NIPS (To appear)*, 2007.

[14] R. Jaulmes, J. Pineau, and D. Precup, "A formal framework for robot learning and control under model uncertainty," in *ICRA*, 2007.

[15] J. Porta, N. Vlassis, M. Spaan, and P. Poupart, "Point-based value iteration for continuous pomdps," *Journal of Machine Learning Research*, vol. 7, 2006.

[16] M. H. DeGroot, *Optimal Statistical Decisions*. McGraw-Hill, 1970.

[17] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[18] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods In Practice*. Springer, 2001.

[19] M. E. Johnson, *Multivariate Statistical Simulation*. New York, NY, USA: John Wiley & Sons, Inc., 1987.

[20] D. McAllester and S. Singh, "Approximate Planning for Factored POMDPs using Belief State Simplification," in *UAI*, 1999.

[21] S. Ross, J. Pineau, and B. Chaib-draa, "Theoretical analysis of heuristic search methods for online pomdps," in *NIPS (To appear)*, 2007.

[22] M. Kearns, Y. Mansour, and A. Y. Ng, "A sparse sampling algorithm for near-optimal planning in large Markov decision processes," *Machine Learning*, vol. 49, no. 2-3, pp. 193–208, 2002.

[23] A. A. Feldbaum, "Dual control theory, parts i and ii," *Automation and Remote Control*, vol. 21, 1961.

[24] N. M. Filatov and H. Unbehauen, "Survey of adaptive dual control methods," in *IEE Control Theory and Applications*, vol. 147, 2000.

[25] I. Rusnak, "Optimal adaptive control of uncertain stochastic discrete linear systems," in *IEEE International Conference on Systems, Man and Cybernetics*, 1995.

[26] R. Ravikanth, S. Meyn, and L. Brown, "Bayesian adaptive control of time varying systems," in *IEEE Conference on Decision and Control*, 1992.

[27] O. Zane, "Discrete-time bayesian adaptive control problems with complete information," in *IEEE Conference on Decision and Control*, 1992.

[28] A. Greenfield and A. Brockwell, "Adaptive control of nonlinear stochastic systems by particle filtering," in *International Conference on Control and Automation (ICCA)*, 2003.