A Scalable Method for Solving High-Dimensional Continuous POMDPs Using Local Approximation

Tom Erez and William D. Smart
Washington University in St. Louis

Abstract

Partially-Observable Markov Decision Processes (POMDPs) are typically solved by finding an approximate global solution to a corresponding belief-MDP. In this paper, we offer a new method to solve POMDPs with continuous state, action and observation spaces. Since such domains have an inherent notion of locality, we can find an approximate solution using local optimization methods. We parameterize the belief distribution as a Gaussian mixture, and use the Extended Kalman Filter (EKF) to approximate the belief update. Since the EKF is a first-order filter, we can marginalize over the observations analytically. For domains with unilateral constraints, we use the equations of truncated normal distributions to analytically approximate the belief update. Our results demonstrate the scalability of this approach, and include a simulated hand-eye coordination domain with 16 continuous state dimensions and 6 continuous action dimensions.

Introduction

Partially-Observable Markov Decision Processes (POMDPs) offer a framework for studying decision making under uncertainty. The optimal behavior in a POMDP domain is expected to strike a balance between exploring the partially-observable world and acting in a goal-directed manner. Most of the POMDP literature is concerned with discrete domains, but in the past few years, as POMDP tools become more powerful, there is growing interest in tackling continuous domains.

The standard approach to solving POMDPs is to find an approximate solution to the fully-observable *belief*-MDP, whose states are probability distributions over the state space of the original POMDP. In the discrete case, the resulting belief space is continuous but finite-dimensional, and belief update can be carried out exactly. However, the belief space of a continuous POMDP is infinite-dimensional, and must be approximated (Thrun, 2000).

The optimal value function of belief-MDPs is piecewise-linear and convex in the discrete case (Sondik, 1971), and this also holds for some cases of continuous state (Porta et al., 2006), as long as the observations and actions are discrete. This result was used to tackle domains with continuous hybrid-linear dynamics by Brunskill et al. (2008). Other combinations of the discrete and the continuous domains were also considered (Hoey & Poupart, 2005; Spaan & Vlassis, 2005). The richest domain tackled by continuous POMDPs is probably outdoor navigation (Brooks, 2009).

However, in all the examples mentioned above, the belief domain is solved through *global* optimization, as the MDP formalism offers no inherent notion of locality. Since the volume of state space grows exponentially with the number of state dimensions, it is unrealistic to seek a globally-optimal solution in domains above a certain size. Some studies (e.g., Feng & Zilberstein, 2004) try to avoid some of the computational burden by finding parts of belief space that can safely be ignored, but the fundamental problem of exponential scaling remains.

However, continuous domains naturally admit a notion of distance, which opens the door for using *local* optimization methods to approximately solve such domains. Since such methods allow us to focus the computational effort only on the most relevant parts of belief space, they offer greater scalability than the global, belief-MDP-based approach.

We present a method for approximating a locally-optimal solution to a POMDP in which state, action and observation space are continuous. We approximate the belief space with a parametric distribution, specifically a Gaussian mixture, and use the Extended Kalman Filter (EKF) for belief update. By virtue of the EKF being a first-order filter, we can analytically marginalize the belief update over the observations, resulting in a deterministic update scheme (section).

The EKF equations maintain Gaussian beliefs for dynamics without discontinuities. However, POMDPs are often used to tackle domains with unilateral constraints, such as contacts (e.g., Hsiao et al., 2007). In such cases the true belief can be far from Gaussian, since the distribution is truncated by a constraint manifold. We approximate the probability mass that aggregates on the manifold with a Gaussian of lower rank (section). We analytically account for the flow of probability mass between the two Gaussians by using the equations of truncated normal distributions (section). These approximations allow us to cast the infinite-dimensional, stochastic belief domain in terms of a finite-dimensional optimal control problem, which can be solved using continuous methods of local optimization.

The optimal policy for the POMDP is approximated by linear feedback around a locally-optimal trajectory in belief

ICAPS 2010 POMDP Practitioners Workshop, May 12, 2010, Toronto, Canada.

space (section), which is found using Differential Dynamic Programming (section). Since planning takes place in belief space, the resulting policy allows the agent to respond to changes in the estimation uncertainty during policy execution. Finally, the belief approximations mentioned above enable efficient planning, but they can be replaced by more accurate state estimation (e.g., particle filter) during policy execution (i.e., forward simulation or real-world interaction).

Definitions

We consider a discrete-time POMDP defined by a tuple $\langle S, A, Z, T, \Omega, R, N \rangle$, where: S, A and Z are the state space, action space and observation space, respectively; $T(s', s, a) = \Pr(s'|s, a)$ is a transition function describing the probability of the next state given the current state and action; $\Omega(z, s, a) = \Pr(z|s, a)$ is the observation function, describing the probability of an observation given the current state and action; and R is a time-dependent reward function $R^i(s, a)$, with a terminal reward $R^N(s)$. In this paper we consider an undiscounted optimality criterion, where the agent's goal is to maximize the expected cumulative reward within a fixed time horizon N. This formulation is a deviation from the common focus on discounted horizons, and we adopt it because it is useful for the local optimal control algorithm we employ (section).

The Stochastic Belief Domain

The *belief state* $b \in B$ is a probability distribution over S, where $b^i(s)$ is the likelihood of the true state being s at time i given the history of a particular trial (which consists of $i-1$ observation-action pairs). In order to construct the belief domain of a given POMDP, we need to find a representation for b, and define the reward function and dynamics (belief update) over this space.

The reward associated with a belief is simply the expected value over this state distribution:

$$R^i(b, a) = \mathop{\mathrm{E}}_{s \sim b} \left[R^i(s, a) \right]. \tag{1}$$

Given the current belief b, an action a and observation z, the updated belief b' can be calculated by applying Bayes's rule. In the discrete case, the belief is fully represented by a normalized vector of size $|S|$, representing the likelihood of every state in S, and the distribution of the expected next state is:

$$b'(s') \propto \sum_s b(s) T(s', s, a) \Omega(z, s, a)$$

which is readily computable. However, in the continuous case B is infinite-dimensional, and the belief update is an integral:

$$b'(s') \propto \int_s b(s) T(s', s, a) \Omega(z, s, a) ds.$$

In order to make this function computationally tractable, we must employ some approximation $\hat{b}$ to the true belief b, and commit to some state estimation filter to update the approximated belief.

Since our optimality criterion employs a finite-horizon, our optimization focuses on the *time-dependent policy* $\pi(\hat{b}, i)$, mapping beliefs and time to actions. The optimal policy maximizes the cumulative reward:

$$\pi^* = \mathop{\mathrm{argmax}}_{\pi} \mathrm{E}\Big[\sum_{i=1}^{N} R^i(b^i, \pi(b^i, i)) \Big]. \tag{2}$$

The Deterministic Belief Domain

In this paper we propose an alternative construction of the belief domain. During planning, we employ two approximation steps: first, we approximate B as a Gaussian mixture. Second, we update the belief *deterministically* by analytically marginalizing over the observation z. It is important to note that these approximations facilitate planning using local methods, but during policy execution they can be replaced by any other estimation process (see section).

Smooth Dynamics

In this section, we focus on nonlinear stochastic dynamics of the form:

$$ds = f(s, a)dt + q(s, a)d\xi, \tag{3}$$

where ξ is a Wiener process. For a given state s and action a, integrating these dynamics over a small time-step τ results in a normal distribution over the next state s':

$$T(s', s, a) = \mathcal{N}(s'|F(s, a), Q(s, a)), \tag{4}$$

where the mean is propagated with the Euler integration

$$F = s + \tau f(s, a), \tag{5}$$

and the covariance $Q = \tau q^{\mathsf{T}} q$ is a time-scaling of the continuous process $qd\xi$. Similarly, we focus on observation distributions of the form:

$$\Omega(z, s, a) = \mathcal{N}(z|w(s), W(s, a)), \tag{6}$$

where w deterministically maps states to observations, and W describes how the current state and action affect the observation noise.

Given a Gaussian prior on the initial state, we approximate the infinite-dimensional b by a single Gaussian:

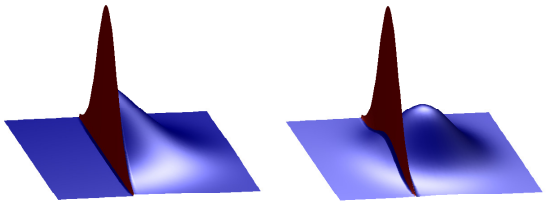
$$\hat{b}(s) = \mathcal{N}(s|\hat{s}, \Sigma),$$

and denote its parameterization by:

$$\nu = \{\hat{s}, \Sigma\} \tag{7}$$

where the covariance Σ belongs to the space of symmetric, positive-semidefinite matrices $\mathcal{M} \subset \mathbb{R}^{n \times n}$. Therefore, the belief space $\hat{B}$ is parameterized in this case by the product space $\nu \in S \times \mathcal{M}$. In the limit of $\tau \to 0$, this approximation is accurate.

In order to approximate the belief update, we use the Extended Kalman Filter (EKF). Given the current belief $\hat{b} = \{\hat{s}, \Sigma\}$, action a and observation z, we calculate the partial derivatives around $\hat{s}$: $w_s = \partial w / \partial s, F_s = \partial F / \partial s$. We find the uncorrected estimation uncertainty $H = F_s \Sigma F_s^{\mathsf{T}} +$



(a) (b)

Figure 1: **(a)** A visualization of a truncated distribution, with probability mass aggregating on the constraint manifold. **(b)** A two-Gaussian mixture approximation of this distribution. Since a Gaussian has infinite support, some probability mass appears below the constraint.

$Q(\hat{s}, a)$ and calculate the new mean $\hat{s}'$ by the innovation process:

$$\hat{s}' = F(\hat{s}, a) - K(z - w(\hat{s})). \tag{8}$$

where $K = Hw_s(w_s^\mathsf{T} H w_s + W(\hat{s}, a))^{-1}$ is the *Kalman gain*. Finally, the new covariance Σ' is given by:

$$\Psi(\hat{s}, \Sigma, a) = H - Hw_s(w_s^\mathsf{T} H w_s + W(\hat{s}, a))^{-1} w_s^\mathsf{T} H^\mathsf{T}. \tag{9}$$

The deterministic belief update is obtained by marginalizing equations (8) and (9) over the observation z. Equation (8) is linear in z, and so we can take the expectation by simply replacing z with its mean $w(\hat{s})$. The second term of equation (8) vanishes, and so the mean follows (5). By virtue of the EKF being a first-order filter, the calculation in (9) is independent of z. In summary, the deterministic belief update is formed by the combination of (5) and (9):

$$\hat{b}' = \{F(\hat{s}, a), \Psi(\hat{s}, \Sigma, a)\}. \tag{10}$$

Dynamics with Unilateral Constraints

The assumptions that T and Ω are Gaussian is too restrictive for some domains. In particular, it excludes discontinuous dynamics that occur due to unilateral constraints. Since this category includes interesting domains of disambiguation by contact, object manipulation and locomotion, we extend our method to handle the non-Gaussian belief that come about in such cases.

In this section we consider domains with non-penetration constraints Γ:

$$\begin{aligned} ds &= f(s, a)dt + Q(s, a)d\xi, \\ \Gamma(s) &\geq 0. \end{aligned} \tag{11}$$

In the general case, the reaction forces that enforce these constraints can be calculated using complementarity methods (Stewart, 2000) or penalty methods (Drumwright, 2008). When $\Gamma(s) = 0$, we say that the constraint is *active*. In this paper, we consider domains where at most one constraint is active at any one time, and so we may focus on cases where $\Gamma(s)$ is scalar.

The resulting belief b can no longer be described by a simple normal distribution: Γ describes an $(n-1)$-dimensional constraint manifold, and the belief distribution is truncated at this manifold, with some probability mass aggregating on it (figure 1(a)). We approximate this truncated distribution with a weighted mixture of two Gaussians (figure 1(b)): one describing the belief distribution in the unconstrained volume, and the other describing the aggregated belief on the constraint (hence degenerate in the direction locally perpendicular to the manifold). Using ν to parameteize a single Gaussian as in (7), We denote the parameterized belief

$$\hat{b}(s) = \alpha\mathcal{N}(s|\hat{s}_1, \Sigma_1) + (1 - \alpha)\mathcal{N}(s|\hat{s}_2, \Sigma_2)$$

by the shorthand

$$\hat{b} = \{\nu_1, \nu_2, \alpha\},$$

where the weight $\alpha \in [0, 1]$. This is not an exact representation of the true belief; a Gaussian has infinite support, and therefore the unconstrained Gaussian has non-zero probability mass beyond the constraint. However, this mass is small enough that, in practice, it has had no noticeable effect on our results.

Belief update is done in two stages (as outlined in algorithm 1). In the first stage, we update the belief of each Gaussian independently using (10). Assuming that there is noise in the direction locally-perpendicular to the constraint, the second Gaussian is now full-rank. In the second stage, we re-approximate this two-Gaussian mixture, ensuring that the resulting mixture maintains the form described above — the probability mass above the constraint manifold is approximated with one Gaussian, and the belief that lies below the constraint is approximated with a second, degenerate Gaussian that lies on the manifold. The details of the computations required for the second stage are detailed in the next two subsections.

Truncation In order to re-adjust the belief to the constraint, we linearize the constraint function $\Gamma \approx Js + e \geq 0$ around the mean of each Gaussian. We compute the distributions on either side of the constraint analytically by considering truncated normal distributions (Toussaint, 2009). We can linearly rotate and re-scale the state space so as to ensure that the constraint manifold is locally perpendicular to the k^{th} dimension of s, and that the uncertainty in this dimension is independent of the others. Therefore, we can focus our analysis on the one-dimensional case, assuming without loss of generality that the constraint does not affect any dimension but k.

Let $x \sim \mathcal{N}(\mu, \sigma^2)$. When bound to an interval $x \in [l, u]$, its distribution becomes:

$$Pr(x) \propto \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(x-\mu)^2}{2\sigma^2}\Big)\Theta(x - l)\Theta(u - x),$$

where Θ is the Heaviside function. The first two moments of the resulting distribution are:

$$\mathrm{E}(X \mid l < X < u) = \mu + \sigma\frac{\phi(\bar{l}) - \phi(\bar{u})}{\Phi(\bar{u}) - \Phi(\bar{l})} \tag{12a}$$

$$\mathrm{Var}(X \mid l < X < u) = \sigma^2\left[1 + \frac{\bar{l}\phi(\bar{l}) - \bar{u}\phi(\bar{u})}{\Phi(\bar{u}) - \Phi(\bar{l})} - \left(\frac{\phi(\bar{l}) - \phi(\bar{u})}{\Phi(\bar{u}) - \Phi(\bar{l})}\right)^2\right] \tag{12b}$$

Algorithm 1 Deterministic Belief Update with Unilateral Constraints

Input: $\hat{b} = \{\nu_1, \nu_2, \alpha\}$, action a
for $i = 1, 2$ **do**
  Marginalized EKF: Calculate ν_i' by (10).
  Truncation: Calculate $\{\nu_i^u, \nu_i^l, \alpha_i^u\}$ by (12).
end for
Reduction: Calculate ν_1'', ν_2'' by (13).
Adjustment: Project ν_2'' onto constraint by (14).
Weight update: Calculate α' by (15).
Output: $\hat{b}' = \{\nu_1'', \nu_2'', \alpha'\}$.

where $\bar{l} = \frac{l-\mu}{\sigma}, \bar{u} = \frac{u-\mu}{\sigma}$, and $\phi(\bar{x})$, $\Phi(\bar{x})$ are the PDF and CDF of the normal distribution with zero mean and unit variance. The probability masses that aggregate on the constraints are $\Phi(\bar{l})$ and $1 - \Phi(\bar{u})$. We are interested in distributions over one-sided intervals, so either $l = -\infty$ or $u = \infty$, which further simplifies (12).

Mixture Reduction We use the truncation procedure described above to split each Gaussian in two, across the constraint. In order to maintain our form (one Gaussian unconstrained, one Gaussian on the constraint manifold), we reduce this four-Gaussian mixture back to two, and project the second Gaussian onto the constraint.

Reducing a mixture of two Gaussians $\{\nu_1, \nu_2, \alpha\}$ results in a single Gaussian whose mean $\hat{s}$ and covariance Σ are:

$$\hat{s} = \alpha\hat{s}_1 + (1-\alpha)\hat{s}_2, \tag{13a}$$

$$\Sigma = \alpha\Sigma_1 + (1-\alpha)\Sigma_2 + \alpha(1-\alpha)(\hat{s}_1 - \hat{s}_2)(\hat{s}_1 - \hat{s}_2)^\mathsf{T} \tag{13b}$$

Using these equations, we combine the two Gaussians above the constraint into a single Gaussian ν_1'', and the two Gaussians below the constraint into ν_2''. Assuming that the constraint is locally perpendicular to the k^{th} dimension as above, we project ν_2'' onto the constraint by setting:

$$(\hat{s}_2'')_k = \Gamma(\hat{s}_2''), \text{ and } (\Sigma_2'')_{k,k} = 0. \tag{14}$$

Finally, the weight of the unconstrained Gaussian in the adjusted mixture is:

$$\alpha' = \alpha\alpha_1^u + (1-\alpha)\alpha_2^u. \tag{15}$$

Policy parametrization

Since a policy for a continuous POMDP is infinite-dimensional, it also needs to be parameterized. In this paper we focus on policies that are locally-linear:

$$\pi(\hat{b}, i) = \bar{a}^i + L^i(\hat{b} - \bar{b}^i) \tag{16}$$

for some parameterized belief states $\bar{b}^{1:N}$, actions $\bar{a}^{1:N-1}$ and feedback gain matrices $L^{1:N-1}$. The optimal values for these parameters can be found using a variety of local optimization techniques, and in this paper we use Differential Dynamic Programming, as described in the next section.

During policy execution, we can use a more accurate filter (e.g., particle filter) for state approximation, using a different representation $\tilde{b}$. In order to combine an arbitrary filter with the above parameterization, we follow Brooks (2009, ch. 6) and define a distance function $D(\tilde{b}, \hat{b})$ between the runtime beliefs and planned beliefs. This allows us to use the points of the planned trajectory $\bar{b}^{1:N}$ as nodes for nearest-neighbor control. The time-dependence of the policy can be integrated into this framework by including the time as another dimension of $\hat{b}$ and $\tilde{b}$ when calculating the distance D.

Differential Dynamic Programming

The combination of (1) and the belief update schemes of the previous section define a problem of optimal control in a high-dimensional continuous space, with non-linear dynamics and non-quadratic reward. To find a locally-optimal solution, we turn to a local optimization scheme called Differential Dynamic Programming (DDP), an algorithm that has been successfully applied to real-world high-dimensional, non-linear control domains (e.g., Abbeel & Ng, 2005). Here, we only provide an overview of DDP; the interested reader may find an in-depth description of the algorithm in Jacobson & Mayne (1970).

DDP finds a locally-optimal trajectory emanating from a fixed starting point. The algorithm makes iterative improvements to a nominal trajectory of length N, until a local minimum is found. DDP forms a quadratic approximation, and so it has Newton-method-like convergence properties. After convergence, DDP outputs the locally-optimal trajectory, the open-loop action sequence which realizes this trajectory, and a sequence of linear feedback gain matrices. These parameterize the policy (16) to create a near-optimal policy for the original POMDP.

Results

We apply our method to three example domains, of increasing dimensionality. The first is a simple one-dimensional navigation problem, considered by Brunskill et al. (2008) and Porta et al. (2006). The second is a two-dimensional navigation domain, similar to those considered by Brooks (2009). Finally, we solve a 16-dimensional problem, presented by Erez & Smart (2009).

One-Dimensional Navigation

In this problem, a robot must locate a power socket in a one-dimensional corridor, blocked at either end. The robot cannot sense the plug nor its own position. In Brunskill et al. (2008), the robot can move left or right in discrete steps, while we consider continuous actuation.

The solution of Brunskill et al. has the robot drive to one of the walls (to localize itself), and then back up to the location of the power supply. Our solution is qualitatively the same (figure 2): as the robot drives towards the wall, the variance of the approximated position collapses, indicating that the robot now has a good idea of where it is. It then backtracks to the location of the target using odometry.

We cannot compare the performance of our method with Brunskill et al.'s directly, because of our use of continuous actions. However, we note that their method is reported to take approximately 40 minutes to find an optimal solution, while ours takes $\sim$5 seconds.

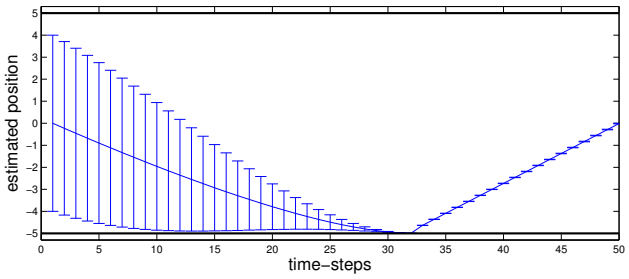
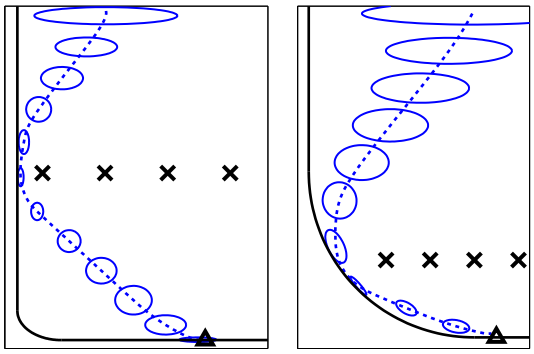


Figure 2: The 1D corridor. The robot begins with high estimation uncertainty (error bars show one standard deviation). As the robot approaches the wall (at time-step 32), the uncertainty vanishes. Certain of its position, the robot can now steer to the target at zero using odometry alone.



(a) (b)

Figure 3: 2D navigation. **(a)** The robot (black line) localizes itself by approaching the wall (dashed blue) while avoiding the obstacles (X) and reaching the target (triangle). Blue ellipses depict the covariance. **(b)** An optimal solution that interacts with the curved part of the constraint manifold.

Planar Navigation

In this problem, a robot must move in a closed room from a start point to a target while avoiding obstacles.

As before, the robot cannot sense its position, but may localize itself by making contact with the walls. The resulting optimal behavior (figure 3(a))is found in less than a minute: the robot avoids the obstacles by approaching the side wall, and then "cut" the corner on its way to the target at the target at the bottom wall. This behavior can be seen as a precursor of coastal navigation (Roy & Thrun, 1999), a technique long-used by roboticists.

In order to study the effect of linearizing the constraint, we tested a case where the agent interacts with the curved segment of the constraint. As figure 3(b) shows, the optimal path in this case follows the round corner without difficulty.

A direct comparison with the results of Brooks (2009) is not possible, but we note that the calculation times reported there range between 7 and 25 minutes, while our method finds a solution in less than one minute.

Hand-eye coordination

This domain simulates the problem of an agent coordinating two "hands" and an "eye". The task requires the agent to bring the hands from their starting positions to a target point at a specific time, while avoiding four obstacles in the planar scene. State transitions are subject to a fixed Gaussian process noise. The obstacles are in a fixed position during a single trial, but can move between trials, so the agent must observe and estimate their positions. Our results are best understood by watching the movie submitted as supplemental material.

The planar scene is illustrated in figure 4(a). The state is defined in terms of the following variables: s_e is the eye's two-dimensional position, s_{h_1} and s_{h_2} are the positions of the hands, s_t is the target's position, and $\{s_{l_i}, i = 1 \dots 4\}$ are the positions of four obstacles. Therefore, the state space has 16 continuous dimensions. Every state s is a concatenation of the 8 planar positions above. The action space A is 6-dimensional, specifying planar velocities for the hands and eye.

Z, the observation space, is identical to the state space. The observation noise covariance W is diagonal, allowing independent observation of each scene element. W is state- and action-dependent: the eye has the capacity to produce unambiguous observations in a small region around its current position, conceptually modelling foveated vision. The eye's gaze locally reduces the observation noise:

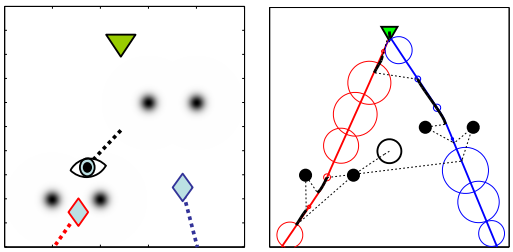
$$W_\star(s, a) = 1 - e^{-\|s_e - s_\star\|^2/2\eta} + 0.01 a_e^\mathsf{T} a_e \qquad (17)$$

where $\star$ stands for one of the scene elements: h_1, h_2, t, or any of the obstacles l_i, the parameter η determines the size of the fovea, and a_e is the current actuation of the eye. The last RHS term in (17) models visual inhibition during saccadic eye movement, effectively eliminating the eye's effect during high-velocity eye movements. Thus, the eye can disambiguate the state only when it is close to an object, and moving slowly.

The reward function in Erez & Smart (2009) penalizes for distance between the hands and the target at the final time step, and for proximity between the hands and the obstacles at all other time steps, and action incurs a quadratic cost.

The covariance of the process noise Q is a constant diagonal matrix, where the noise in the X- and Y-direction are equal for every scene element. The process noise that affects the eye, obstacles and target is negligibly small, and kept away from zero only enough to prevent singularities in equation (9). From the agent's perspective, this means that once observed, the positions of the target and obstacles can be trusted to remain unmoved, allowing the eye's position to provide grounding for locating all other elements of the scene. Since process noise is uncorrelated between state dimensions and symmetric in both planar directions, the belief covariance can be decomposed and succinctly represented by 8 numbers[1], denoting the "planar uncertainty" of each of the scene's elements. In all, $\hat{B}$ has 24 dimensions.

[1]Formally speaking, the covariance is an 8-by-8 block matrix of 2-by-2 matrices, where the 8 diagonal blocks are multiples of the identity matrix, and all other blocks are zero.



(a) (b)

Figure 4: Hand-eye coordination. **(a)** A schematic of the scene. The hands (red and blue diamonds) aim for the target (green triangle) while avoiding the obstacles whose position is uncertain (black blurs), assisted by the eye (middle). **(b)** An illustration of the learned optimal trajectory. The size of the eye's fovea can be seen as a black circle around its starting place (center of the figure). The uncertainty in the hands' positions are depicted as a series of circles. The eye's trajectory alternates between following the hands in smooth pursuit (thick black line) and saccadic motion (thin dashed line) between the hands and the obstacles. This behavior is better illustrated by the movie, attached as supplementary material.

The locally-optimal solution for this problem was found through the use of shaping techniques. Initially, the size of the notional fovea is large ($\eta = 10$), allowing a relatively unambiguous view of the entire scene. As learning progresses, the size of the foveal region is gradually reduced, making the eye movements more important. Every new problem instance is solved using the previous solution as a starting point. This process repeats for decreasing fovea radius ($\eta = [1, 0.3, 0.05]$) until we generate a solution to the final desired problem instance.

Figure 4(b) shows the resulting locally-optimal trajectories for both hands and the eye. Notice how the eye tracks each hand in turn as it passes close the obstacles, and how the eye alternates between saccadic motion and smooth pursuit. This behavior is best illustrated by a video which is included as supplementary material, and we encourage the reader to see it.

This domain was not cast as a POMDP originally, and the authors do not report time estimates for convergence, making direct comparison impossible. The shaping sequence required running DDP to convergence 4 times, yet the optimal solution for this 16-dimensional domain was found in less than 3 minutes of MATLAB running on a single-core desktop computer.

Discussion

This paper offers a new perspective on solving continuous POMDPs. Instead of using global approximation in a belief-MDP, we cast the belief domain in terms of optimal control. This allows us to use computationally efficient methods developed in control theory. While this paper offers only an initial exploration of this approach, our experiments try to highlight its merits in terms of scalability.

POMDPs are challenging because the domain inherently couples estimation and control. Our method realizes this coupling by incorporating the continuous dynamics and the EKF equations into a single dynamical system, and performing optimal control in this augmented domain. This EKF-based coupling has been studied in the context of control theory (e.g., Tse et al., 1973) and robotics (Prentice & Roy, 2009), but not for POMDPs.

While this method scales very well with state dimensionality, we chose to focus on domains where only one constraint is active at a time. Such cases are amenable to analytic manipulation using truncated normal distributions, as described above. If we extended this type of analysis to cases where more than one constraint may be active at once, we would be assigning a Gaussian to every combination of active constraints, and accounting for the flow of probability mass between all of them. This would introduce yet another set of approximations, and be computationally reasonable only for a small number of jointly-active constraints.

One natural extension of this work could employ local optimization from multiple starting points, creating a controller that uses a trajectory library (Stolle & Atkeson, 2006). In particular, a multi-modal prior can be handled by finding the optimal behavior for each of the modes, and using state estimation during policy execution to choose the relevant case.

In many real-life cases, an active constraint results in frictional forces, in addition to the reaction forces that maintain non-penetration. This can be incorporated into our method by using a different dynamical model for the initial belief update of the constrained Gaussian ν_2, in particular one that incorporates friction. In cases where making contact (i.e., collision) is associated with a non-negligible impact dynamics of other degrees of freedom beyond the constrained one (e.g., foot-ground impact, or ball-racket impact), these impulses can be considered as we project the Gaussian that lies below the constraint manifold onto the linearized hyperplane.

References

Abbeel, Pieter and Ng, Andrew Y. Exploration and apprenticeship learning in reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 1–8, 2005.

Brooks, Alex. *Parametric POMDPs*. VDM Verlag, 2009.

Brunskill, Emma, Kaelbling, Leslie, Lozano-Perez, Tomas, and Roy, Nicholas. Continuous-state POMDPs with hybrid dynamics. In *Tenth International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, January 2008.

Drumwright, E. A fast and stable penalty method for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):231–240, 2008.

Erez, Tom and Smart, William D. Coupling perception and action using minimax optimal control. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 58–65, 2009.

Feng, Zhengzhu and Zilberstein, Shlomo. Region-based incremental pruning for POMDPs. In *The 20th conference on Uncertainty in artificial intelligence (UAI)*, pp. 146–153, 2004.

Hoey, Jesse and Poupart, Pascal. Solving POMDPs with continuous or large discrete observation spaces. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1332–1338, 2005.

Hsiao, Kaijen, Kaelbling, Leslie Pack, and Lozano-Perez, Tomas. Grasping POMDPs. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4685–4692, 2007.

Jacobson, D. H. and Mayne, D. Q. *Differential Dynamic Programming*. Elsevier, 1970.

Porta, Josep M., Vlassis, Nikos, Spaan, Matthijs T.J., and Poupart, Pascal. Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research*, 7: 2329–2367, December 2006.

Prentice, S. and Roy, N. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 28(11-12):1448–1465, 2009.

Roy, Nicholas and Thrun, Sebastian. Coastal navigation with mobile robots. In *Advances in Neural Processing Systems (NIPS)*, volume 12, pp. 1043–1049, 1999.

Sondik, E.J. *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford, 1971.

Spaan, Matthijs T. J. and Vlassis, Nikos A. Planning with continuous actions in partially observable environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3458–3463, 2005.

Stewart, David E. Rigid-body dynamics with friction and impact. *SIAM Reviews*, 42(1):3–39, 2000.

Stolle, Martin and Atkeson, Chris. Policies based on trajectory libraries. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

Thrun, S. Monte carlo POMDPs. In Solla, S.A., Leen, T.K., and Müller, K.-R. (eds.), *Advances in Neural Information Processing Systems (NIPS)*, volume 12, pp. 1064–1070. MIT Press, 2000.

Toussaint, Marc. Pros and cons of truncated gaussian ep in the context of approximate inference control. NIPS workshop on Probabilistic Approaches for Robotics and Control, 2009.

Tse, E., Bar-Shalom, Y., and Meier, L., III. Wide-sense adaptive dual control for nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 18(2):98–108, 1973.