Making the Impossible Possible: Strategies for Fast POMDP Monitoring

Richard Washington
Caelum Research
NASA Ames Research Center
MS 269-2
Moffett Field, CA 94035
richw@ptolemy.arc.nasa.gov

Abstract.

Systems modeled as partially observable Markov decision processes (POMDPs) can be tracked quickly with three restrictions: all actions are grouped together, the out-degree of each system state is bounded by a constant, and the number of non-zero elements in the belief state is bounded by a (different) constant. With these restrictions, the tracking algorithm operates in constant time and linear space. The first restriction assumes that the action itself is unobservable. The second restriction defines a subclass of POMDPs that covers however a wide range of problems. The third restriction is an approximation technique that can lead to a potentially vexing problem: an observation may be received that has probability 0 according to the restricted belief state. This problem of *impossibility* will cause the belief state to collapse.

In this paper we discuss the tradeoffs between the constant bound on the belief state and the quality of the solution. We concentrate on strategies for overcoming the impossibility problem and demonstrate initial experimental results that indicate promising directions.

1 Introduction

Systems operating in dynamic, uncertain environments are difficult to monitor and control. The changing world may make action effects difficult to predict, and the uncertainty complicates this by obscuring the current state of the system. Stochastic representations of systems operating in such environments allow these systems to be monitored and controlled robustly despite the uncertainty and uncontrollable effects.

The ability to monitor a system state under uncertainty is useful for control. But it is also useful for diagnosis, where traditionally a fault-diagnosis system uses the fault information to deduce the possible states of the system [3]. By intelligently tracking the state of the system, more accurate fault diagnosis is possible, and thus more appropriate corrective action.

Markov models [1] have been a popular representation for capturing the dynamics of the environment [14, 4, 7, 9]. A Markov decision process (MDP) describes a system as a discrete set of states. The effect of an action is represented as a probability distribution over the states, reflecting both the range of possible outcomes and their likelihoods. A system's behavior is a sequence of states and actions. The usual way to plan a behavior in an MDP is to construct a *policy*: a mapping from states to actions, indicating for each state which action will be executed when the system is in that state (note that a policy allows multiple behaviors, since there are multiple possible outcomes of actions). Rewards associated with states and actions specify the local utility of a policy; the total reward of a state for a given policy is simply the sum of the rewards of all the possible behaviors allowed by the policy (weighted by the probabilities). An optimal policy can be computed for an MDP, specifying for each state the optimal action to take to maximize the total utility of the plan.

The classic MDP, however, does not account for uncertainty in the system's state. Often this state is known only indirectly through observations. If there is inaccuracy or uncertainty in the observation, this indirect information reflects only imprecisely the actual process state. To account for the state uncertainty, MDPs have been extended to partially observable MDPs (POMDPs) [8]. In this model, the underlying system is an MDP, but the state is only indirectly known; an *observation* is produced on each state transition. The model specifies the probability of seeing an observation in a state (this can be produced in practice by experimental study). Instead of an exact state, the knowledge of the process can be represented as a probability distribution over states, called the *belief state*. A policy in a POMDP is a mapping from belief states to actions. An optimal POMDP policy is thus a mapping from belief states to actions, indicating the optimal action to take in each belief state. This can be useful when the system state is only incompletely known.

The control problem for POMDPs is computationally intractable for large problems [7], even with approximation algorithms [14, 9, 11, 12, 5]. The monitoring problem is much simpler: here the goal is to follow the state of the underlying system. Nonetheless, for a problem with N states, the time and space complexity is $\mathcal{O}(N^2)$, which for large problems may still be too high for fast calculations. By restricting the class of POMDPs to a subclass called *sequential POMDPs*, by agglomerating action effects, and by restricting the belief state to a constant number of non-zero entries, we can achieve constant time complexity and space complexity that scales linearly with the size of the state space [13].

This constant-time performance comes at a price, however. The restrictions allow only a subclass of POMDPs, albeit an interesting one. More troubling is the problem of *impossibility*, where an observation may be received that has probability 0 according to the reduced belief state.

In this paper, we discuss the restrictions on POMDP monitoring that allow constant time and linear space algorithms. We begin with a discussion of Sequential POMDPs, followed by a review of On-Line Markov Tracking. We then discuss new work on finding strategies to overcome the impossibility problem, which include strategies that recover and strategies to avoid the problem altogether. We illustrate the relative performance of the strategies with a set of initial experiments, which indicate some promising paths of future exploration.

2 Sequential POMDPs

In this section we briefly review Markov processes, and in particular POMDPs. Then we discuss Sequential POMDPs [13].

We assume that the underlying process, the *core process*, is described by a finite-state, stationary Markov chain. The core process is captured by the following information:

- a finite set $\mathcal{N} \equiv \{1, \ldots, N\}$, representing the possible states of the process.
- a variable $X_t \in \mathcal{N}$ representing the state of the core process at time t.
- a finite set $\mathcal{A}$ of actions available.
- a matrix $P = [p_{ij}], i, j \in \mathcal{N}$ specifying transition probabilities of the core process: $P(a) = [p_{ij}(a)]$ specifies the transition probabilities when action $a \in \mathcal{A}$ is chosen.
- a reward matrix $R = [r_{ij}], i, j \in \mathcal{N}$ specifying the immediate rewards of the core process: $R(a) = [r_{ij}(a)]$ specifies the reward received when the action $a \in \mathcal{A}$ is executed, moving the process from state i to state j.

So at time t, the core process is in state $X_t = i$, and if an action $a \in \mathcal{A}$ is taken, the core process transitions to state $X_{t+1} = j$ with probability $p_{ij}(a)$, receiving immediate reward $r_{ij}(a)$.

However, in a partially observable MDP, the progress of the core process is not known, but can only be inferred through a finite set of observations. The observations are captured with the following information:

- a finite set $\mathcal{M} \equiv \{1, \ldots, M\}$ representing the possible observations.
- a variable $Y_t \in \mathcal{M}$ representing the observation at time t.
- a matrix $Q = [q_{ij}], i \in \mathcal{N}, j \in \mathcal{M}$ specifying the probability of seeing observations in given states: $Q(a) = [q_{ij}(a)]$, where $q_{ij}(a)$ denotes the probability of observing j from state i when action $a \in \mathcal{A}$ has been taken.
- a state distribution $\pi(t) = \{\pi_1(t), \ldots, \pi_N(t)\}$, where $\pi_i(t)$ is the probability of $X_t = i$ given the information about actions and observations.
- an initial state distribution $\pi(0)$.

At time t, the observation of the core process will be Y_t. If action $a \in \mathcal{A}$ is taken, we can define a function to determine Y_{t+1}. In particular, we define

$$\gamma(j|\pi(t), a) = \sum_{i \in \mathcal{N}} q_{ij}(a) \sum_{k \in \mathcal{N}} p_{ki}(a)\pi_k(t) \qquad (1)$$

as the probability that $Y_{t+1} = j$ given that action $a \in \mathcal{A}$ is taken at time t and the state distribution at that time is $\pi(t)$.

To determine the state distribution variable $\pi(t+1)$, we define the transformation T as follows:

$$\pi(t+1) = T(\pi(t)|j, a) = \{T_1(\pi(t)|j, a), \ldots, T_N(\pi(t)|j, a)\}$$

where

$$T_i(\pi(t)|j, a) = \frac{q_{ij}(a) \sum_{k \in \mathcal{N}} p_{ki}(a)\pi_k(t)}{\sum_{l \in \mathcal{N}} q_{lj}(a) \sum_{k \in \mathcal{N}} p_{kl}(a)\pi_k(t)}, \qquad (2)$$

for $i \in \mathcal{N}$, and where $\pi(t)$ is the state distribution at time t, $a \in \mathcal{A}$ is the action taken at that time, resulting in observation $j \in \mathcal{M}$.

A *sequential POMDP* is a restricted POMDP in which the state transitions in the underlying MDP are constrained to the same state or the "following" state:

$$\forall_{i,j \in \mathcal{N}} \forall_{a \in \mathcal{A}} p_{ij}(a) > 0 \rightarrow i \leq j \leq i+1$$

Graphically such a model looks like Figure 1. In fact, this can be generalized a bit: if the number of nonzero transitions from any state is less than a constant bound, the results in this paper will hold. However, in the remainder of the paper, we will hold to the more restrictive definition for ease of explanation and understanding.

Given the restriction on the model, the state distribution update given in Equation 2 can be computed in $\mathcal{O}(N)$, since the denominator can be computed once for all T_i at a cost of $\mathcal{O}(N)$ and the numerator is computed separately for each T_i at constant cost for each (thus $\mathcal{O}(N)$ total). In addition, the memory required for the transition and reward matrices is $\mathcal{O}(N \cdot |\mathcal{A}|)$ and the memory for the observation matrix remains $\mathcal{O}(N \cdot |\mathcal{M}| \cdot |\mathcal{A}|)$. Note that this is for the general case; in the next section we present a restriction on the general POMDP model that makes these updates more efficient.

2.1 On-Line Markov Tracking

Suppose that instead of wanting to find a plan to control a system, we wanted to track the system's state. The belief state is know to be a sufficient statistic for choosing optimal actions [10]. Thus an optimal approach in general for tracking POMDPs is to track the belief state, i.e., to update the state distribution over actions and observations.

Since we are assuming tracking by an external entity, we assume that the individual actions are themselves unobservable. Instead, the external tracking agent will see just the results of the actions, that is, the transitions from state to state. Because of this we use an agglomerated action model. In this model, there is just one set of transition probabilities from a state, representing the possible effects of the actions from that state. This effectively reduces the number of actions, $|\mathcal{A}|$, to 1.

One thing that is lost in this representation of the problem is the dependency between the agent's internal actions and its behaviors. When the system follows a policy internally, reasoning about that policy could allow a more accurate tracking of its behavior.

In POMDPs, the state of knowledge of the current state of the process is represented by the belief state, which is a probability distribution over the set of states. Over time, this distribution may have many non-zero but vanishingly small elements, each of which must be taken into account when

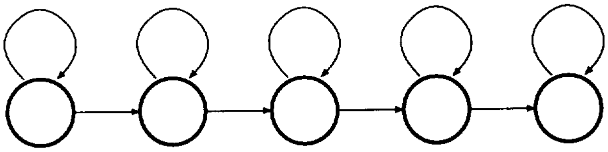


Figure 1. Sequential POMDP. All transitions are to the same or following state.

updating the belief state. Thus much of the computation involved in belief state updates is due to low and vanishing probability elements of the state distribution.

In On-Line Markov Tracking, we limit the distribution to the k most probable states, for some constant k, by zeroing the rest and renormalizing the distribution. The goal of doing this is to restrict attention and computation to the most likely part of the probability distribution. The risks inherent in truncating the distribution are the subject of the remainder of the paper. However, this improves the efficiency of tracking. The computational complexity of tracking in the general POMDP case reduces from $\mathcal{O}(N^2)$ for N states to $\mathcal{O}(N)$, based on the cost of the belief state update in Equation 2.

For sequential POMDPs, given a distribution of k possible states, there are at most $2k$ states that could have a non-zero probability in the following time step (of which k will be retained). This means that the belief state update can be limited to those states, and in fact that makes the belief state update $\mathcal{O}(1)$ (constant). The space required is $\mathcal{O}(N \cdot |\mathcal{M}|)$. This scales linearly with the state space size. Obviously this grows also with the number of observations, but for a given problem class, we assume that the number of observations remains fixed regardless of the number of states (consider robot navigation, where the sensors remain fixed whatever the size of the world explored).

3 Strategies for Handling Impossibility

We have shown that sequential POMDPs with constant-size belief state truncation can be tracked in constant time. However, restricting the distribution to a constant size is not without risks: the part of the belief state that is truncated is still possible with some small but non-zero probability. So in some small percentage of cases, a valid transition will be lost. This in turn may lead to an observation that is impossible with respect to the remaining, truncated belief state. If we were to push through the state update formula, the resulting belief state would be ill-defined, with zero probability for each of the states. In the remainder of the paper, we discuss the possible strategies for handling this *impossibility problem.*

Strategies for handling the impossibility problem can vary from reactive strategies, which try to recover from the inconsistent observation, to proactive strategies, which modify the update formula to avoid the impossibility problem altogether. We consider the following strategies:

Blind: When a 0-probability observation is encountered, the belief state update is performed ignoring the observation (just using the underlying state transition probabilities) [13]. This corresponds to believing the truncated model and not believing the observation, in some sense "flying blind" until the observations once again correspond to a possible observation according to the model. The potential problem is that if the model is really off track, it may never get back on.

The update formula is as in Equation 2, except when a 0-probability observation is encountered; in that case, the update formula is:

$$T_i(\pi(t)|j,a) = \sum_{l\in\mathcal{N}} p_{li}(a)\pi_l(t) \tag{3}$$

which is simply ignoring the observations.

Observation-based: When a 0-probability observation is encountered, the belief state is reconstructed from the observation (i.e., computing the probability of each state given the observation) [6]. This corresponds to believing the observation and not believing the model. The potential problem here is that the observation may skew the belief state to states that are in fact very unlikely or impossible in the full (untruncated) belief state.

The update formula is as in Equation 2, except when a 0-probability observation is encountered; in that case, the update formula is:

$$T_i(\pi(t)|j,a) = \frac{q_{ij}(a)}{\sum_{l\in\mathcal{N}} q_{lj}(a)}, \tag{4}$$

which is simply ignoring the previous state information and calculating the belief state from the observation received. This is equivalent to assuming a uniform prior belief state.

Average: This approach (and the following ones) tries to avoid the 0-probability observations altogether by taking into account the observations at each state without waiting for the problem to occur. Here the approach is to average the belief states using the formulas from the Blind and the Observation-based approaches (Equations 3 and 4), but using them all the time, proactively, rather than simply when there is an inconsistent observation.

The average of the belief states from the two formulas is weighted according to the amount of *accumulated truncation* that has been performed, i.e., the accumulated loss of information caused by truncating the belief state to k elements at each step. The accumulated truncation is a number in the range $[0, 1]$ representing the portion of the belief state truncated. The confidence in the belief state is inversely related to the accumulated truncation: the more truncation, the less confidence in its value.

However, the confidence is somewhat increased by the belief in the observation. This is represented by updating the accumulated truncation after each step: if the accumulated truncation is p_{trunc}, the updated truncation is $p_{trunc} \cdot (1 -$

p_{trunc}) (before the current belief state is truncated).

$$\begin{aligned} T_i(\pi(t)|j,a) &= (1-p_{trunc})\left[\sum_{l\in\mathcal{N}} p_{li}(a)\pi_l(t)\right] \\ &\quad +p_{trunc}\left[\frac{q_{ij}(a)}{\sum_{l\in\mathcal{N}} q_{lj}(a)}\right] \end{aligned}$$

The problem here is that you may have two inaccurate models mixed together, which may produce a distribution that looks nothing like the real distribution.

Mix: This approach also performs a weighted average, but in this case the average is done within the belief state formula, weighting the individual elements. The update formula is:

$$T_i(\pi(t)|j,a) = \frac{f(i,j,a)}{\sum_{l\in\mathcal{N}} f(l,j,a)} \tag{5}$$

where

$$\begin{aligned} f(i,j,a) &= \frac{1}{N}(1-p_{trunc})q_{ij}(a) \\ &\quad +q_{ij}(a)\sum_{k\in\mathcal{N}} p_{ki}(a)\pi_k(t) \\ &\quad +\frac{1}{N}p_{trunc}\sum_{k\in\mathcal{N}} p_{ki}(a)\pi_k(t) \end{aligned}$$

where p_{trunc} is again the accumulated truncation. Note that the middle term is the same as the original update equation (Equation 2), but the end terms are weighted to believe the observation or the state-based model. The $\frac{1}{N}$ terms reflect the uniform distribution over states (observations) when the observation (state) is believed exclusively. The accumulated truncation is updated as with the Average strategy.

Fixmix: This approach is like Mix, except that the sensors have a prior confidence p_{obs}, and this is used along with the accumulated truncation to weight the state-based and observation-based elements. In this case the accumulated truncation update is $p_{trunc} \cdot (1-p_{obs})$. The update formula is Equation 5, but where

$$\begin{aligned} f(i,j,a) &= \frac{1}{N}p_{frac}q_{ij}(a) \\ &\quad +q_{ij}(a)\sum_{k\in\mathcal{N}} p_{ki}(a)\pi_k(t) \\ &\quad +\frac{1}{N}(1-p_{frac})\sum_{k\in\mathcal{N}} p_{ki}(a)\pi_k(t), \end{aligned}$$

where

$$p_{frac} = \frac{1-p_{trunc}}{1-p_{trunc}+p_{obs}}.$$

4 Experiments

The On-Line Markov Tracking approach can be applied to a number of problem domains. Sequential POMDPs are appropriate for domains that follow a trajectory over time. For example, the computer could have the job of following a spoken text and producing a subtitled text to accompany it. Or the computer could have the job of following a musical score and playing an accompaniment. The generalization of sequential POMDPs to POMDPs with a constant bound on the number of transitions per state leads to application domains that include telerobotics and space vehicle status tracking.

We are investigating these and other "real" applications, but for initial results, we constructed a number of randomly-generated scenarios to illustrate our ideas.

We chose a sequential POMDP of length 200 plus a distinguished start and end state. The system state at all but the start and end was chosen from a set of 10 states. Possible observations corresponded to the states, with the most probable being the correct state, with decreasing probabilities for surrounding states: for state S3, the probabilities of the observations were: S0 = 0.02, S1 = 0.1, S2 = 0.2, S3 = 0.345, S4 = 0.2, S5 = 0.1, S6 = 0.02, S7 = 0.01, S8 = 0.005, S9 = 0.

The underlying model was executed by stochastically transitioning from state to state, producing an observation, and updating each of the models corresponding to the strategies. The truncation was relatively severe - 2 states were preserved at each step. This was so that 0-probability states would be observed. The model was executed repeatedly until a 0-probability state was observed by the Blind strategy, then the output of that run was stored. In all, 50 runs were stored.

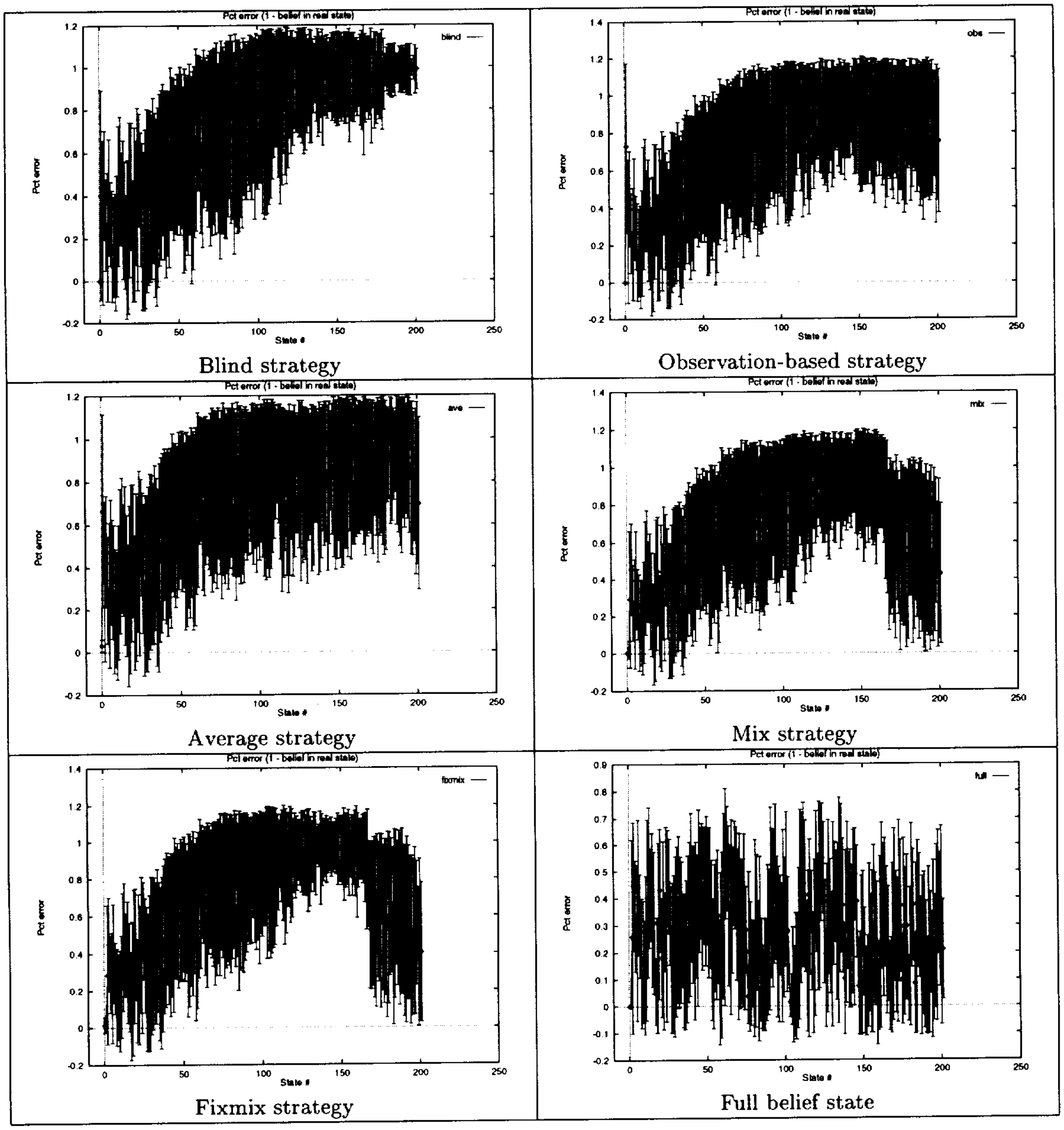
A few measures were collected for evaluation. They were collected per state and per iteration (since the underlying model sometimes stays at a single state for multiple iterations). The two are similar, but with tighter error bounds on the per-state measures, so they will be shown here.

- The percent error compared to the actual underlying state. This is 1 - belief in the real state, thus giving a range of [0,1]. In the figures this is referred to as "percent error."
- The 1/0 measure of whether the actual underlying state is the most likely state. Averaged over multiple trials gives a range of values [0,1]. In the figures this is referred to as "percent of time state ID correct."
- The difference between the full belief state and the truncated belief state (this is the sum of the absolute value of the difference for each state). The range of possible values is [0,2]. This we deemed less interesting, since the full belief state itself has a fair amount of error with respect to the actual underlying state.

The results can be seen in Figures 2–3, where for reference purposes the performance of the complete belief state is shown as well. In Figure 2 the mean and standard deviation are shown for each strategy; in Figure 3 the means are shown superimposed for comparison. All the strategies start to diverge from the correct track over time, but at a particular point in the sequence, the Mix and Fixmix strategies are brought back on track by a sequence of identifiable states. Note that neither the Observation-based nor the Average strategy recovers very well.

The experimental data are noisy, and the standard deviations large. The large differences between either Mix or Fixmix and any other strategy are significant after Mix and Fixmix recover; otherwise the differences are statistically insignificant.

The sudden change in the graphs raises some concern, since this indicates that something in the problem structure makes it particularly identifiable for the Mix/Fixmix strategies at



Blind strategy

Observation-based strategy

Average strategy

Mix strategy

Fixmix strategy

Full belief state

Figure 2. Percent error for each strategy.

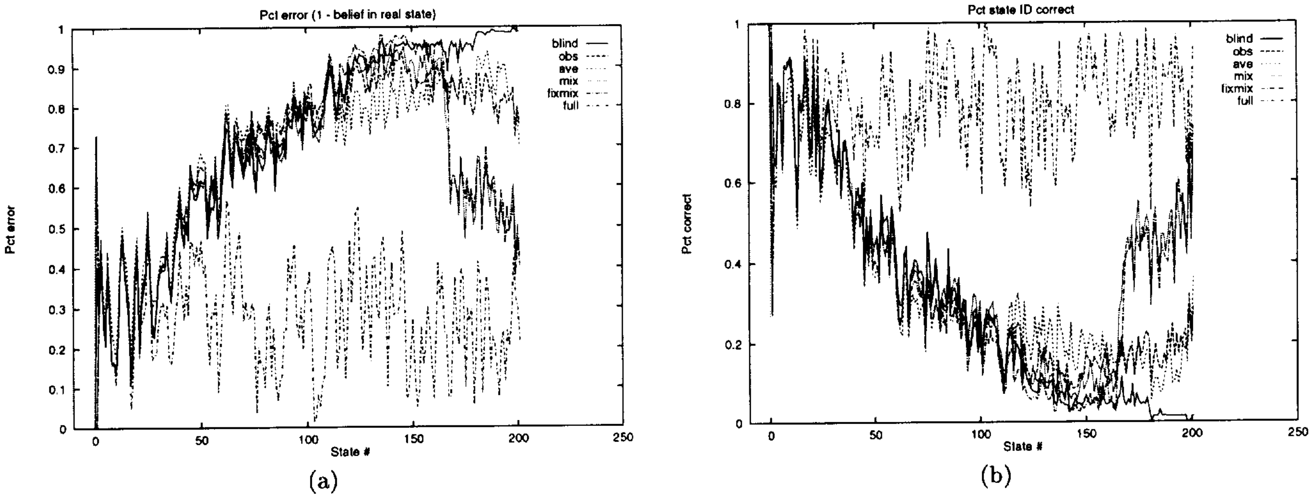


Figure 3. Performance of strategies for size 2 belief state. (a) Percent error for all strategies. (b) Percent of time state ID correct, all strategies.

a specific area of the problem. In fact, a second randomly-generated sequence of states exhibited a similar behavior, albeit at a different area of the problem. In general, the behavior appears to be linked to a sequence of states which collectively provide a landmark for the tracking procedure. The combination of prediction and observation focuses on the particular part of the problem space in which this landmark must fall. Further tests are needed on other problems to understand better the behavior.

5 Time/Accuracy Tradeoff

To explore the tradeoff between computation time and accuracy, we re-ran the experiment with belief states of size 1–4. A belief state of size 1 corresponds to the extreme case of keeping only the most probable state at each step. A belief state of size 4 in this case corresponds to a relatively large percentage of the maximum possible size (10), and as such, would be expected to be relatively accurate.

The comparison of the Mix strategy over these belief state sizes can be seen in Figure 4. The accuracy can be seen to increase over this small range of test sizes. It is interesting to note that even in the extreme case of a belief state of size 1, the strategy pulls itself back on track after drifting away.

A confounding factor in the experiments is the experimental noise based on the randomness of the observations and transitions. At belief state size 4, the accuracy advantage of the Mix and Fixmix strategies over the other strategies begins to be lost in the not inconsequential experimental noise.

This result is a confirmation of what we would expect: the more severely restricted the belief state, the greater the error in the resulting tracking procedure. More complex domains and problems should help shed light on when each strategy is appropriate and how constrained the belief state can be.

6 Discussion

In this paper we have presented preliminary results evaluating and illustrating methods that allow fast tracking of systems represented as POMDPs. The tracking method runs in constant time and linear space with respect to the state space size. We have shown that in preliminary cases, a mixture of state-based and observation-based tracking shows the most promise for recovery from tracking errors.

The ultimate goal is to understand the limitations and uses of this tracking method and of the strategies for recovering from and avoiding impossible observations. More experimental and theoretical work is needed to move towards that goal.

Compact representations, such as in [2], help reduce the state-space size needed for updates. To get constant-time algorithms, the transitions still need to be restricted to a constant size, but this would allow a larger constant bound to be used for the same performance.

The relation between the problem structure and the tracking strategy is intriguing. For example, a complete explanation of the effectiveness of the Mix and Fixmix strategies on the example problem likely hinges on characteristics of the problem formulation; a better understanding of that dependence will allow a better understanding of the fundamental properties of the strategies.

The averaging strategies (Average, Mix, Fixmix) bear a resemblance to the Kalman filter updates with confidence-based weighting of state-based and observation-based updates. The theoretical basis of the Kalman filtering approach is lacking in the strategies for Markov Tracking. This connection deserves further examination to find whether that relationship could be exploited.

REFERENCES

[1] R. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[2] C. Boutilier and D. Poole, 'Computing optimal policies for partially observable decision processes using compact representations', in *Proceedings of AAAI-96*, (1996).

[3] J. de Kleer and B. C. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, **32**, 100–117, (1987).

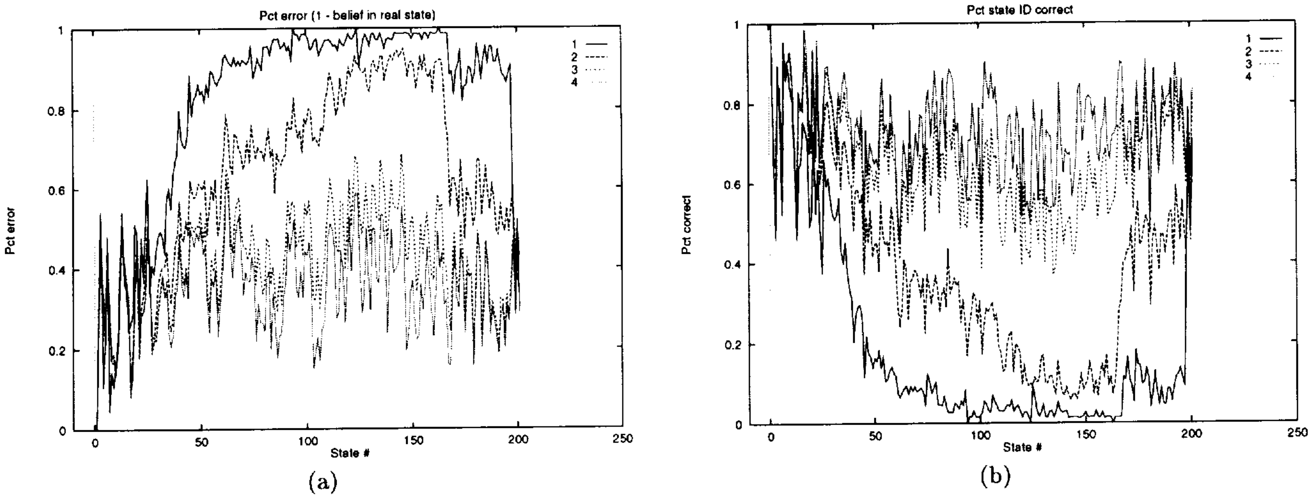


Figure 4. Results for different sizes of belief state, Mix strategy. (a) Percent error. (b) Percent of time state ID correct.

[4] T. Dean, L. P. Kaelbling, J. Kirman, and A. Nicholson, 'Planning under time constraints in stochastic domains', *Artificial Intelligence*, **76**, 35–74, (1995).

[5] E. Hansen, 'Solving POMDPs by searching in policy space', in *Proceedings of UAI '98*, (1998).

[6] J. Kurien, 'Personal communication'. 1997.

[7] M. L. Littman, A. Cassandra, and L. P. Kaelbling, 'Learning policies for partially observable environments: Scaling up', in *Proceedings of the Twelfth International Conference on Machine Learning*, eds., A. Prieditis and S. Russell, pp. 362–370, San Francisco, CA, (1995). Morgan Kaufmann.

[8] W. S. Lovejoy, 'A survey of algorithmic methods for partially observed markov decision processes', *Annals of Operations Research*, **28**, 47–65, (1991).

[9] R. Parr and S. Russell, 'Approximating optimal policies for partially observable stochastic domains', in *Proceedings of IJCAI-95*, (1995).

[10] R. D. Smallwood and E. J. Sondik, 'The optimal control of partially observable Markov processes over a finite horizon', *Operations Research*, **21**, 1071–1088, (1973).

[11] R. Washington, 'Incremental Markov-model planning', in *Proceedings of TAI-96, Eighth IEEE International Conference on Tools With Artificial Intelligence*, (1996).

[12] R. Washington, 'BI-POMDP: Bounded, incremental partially-observable markov-model planning', in *Proceedings of ECP'97, the Fourth European Conference on Planning*, (1997).

[13] R. Washington, 'Markov tracking for agent coordination', in *Proceedings of Agents '98, Second International Conference on Autonomous Agents*, (1998).

[14] C. C. White, III, 'A survey of solution techinques for the partially observed Markov decision process', *Annals of Operations Research*, **32**, 215–230, (1991).