Solving Uncertain Markov Decision Processes

J. Andrew Bagnell, Andrew Y. Ng, Jeff G. Schneider*

Abstract

The authors consider the fundamental problem of finding good policies in uncertain models. It is demonstrated that although the general problem of finding the best policy with respect to the worst model is NP-hard, in the special case of a convex uncertainty set the problem is tractable. A stochastic dynamic game is proposed, and the security equilibrium solution of the game is shown to correspond to the value function under the worst model and the optimal controller. The authors demonstrate that the uncertain model approach can be used to solve a class of *nearly* Markovian Decision Problems, providing lower bounds on performance in stochastic models with higher-order interactions. The framework considered establishes connections between and generalizes paradigms of stochastic optimal, mini-max, and H_∞/robust control. Applications are considered, including robustness in reinforcement learning, planning in nearly Markovian decision processes, and bounding error due to sensor discretization in noisy, continuous state-spaces.

1 Introduction

A problem that is of fundamental interest in planning and control problems is determining the robustness of a plan or policy to modeling errors. It has long been recognized in the control community that controllers developed in the framework of stochastic optimal control may exhibit unsatisfactory online performance due to poor robustness to small modeling errors. More recently, this problem has been recognized in the reinforcement learning community as well, and algorithms have been suggested to deal with it. Minimizing the risk and impact of brittle controllers is one of some import in model-based reinforcement learning solutions, particularly ones where failure has significant consequences [Bagnell and Schneider, 2001], as invariably a learned model has certain inaccuracies, due both to under-modeling and insufficient training data.

One basic approach is to abandon the framework of stochastic control entirely. [Heger, 1994] and [Morimoto and Doya, 2001] adopt this approach, replacing a probabilistic model of interactions with a deterministic, worst case evaluation criterion. [Heger, 1994]'s technique (mini-max Q-learning) assumes that at each step the

*Drew Bagnell and Jeff Schneider are with Carnegie Mellon's Robotics Institute, E-mail:dbagnell@ieee.org, Jeff.Schneider@ri.cmu.edu. Andrew Ng is with the Department of Computer Science, University of California at Berkeley. Email: ang@cs.berkeley.edu

worst possible result happens to an agent. As might be imagined, this can lead to rather conservative estimates of the value of a policy, particularly in the instances where a stochastic model is an accurate representation. In the worst case- a truly stochastic model where there is some small probability of catastrophe at each state, this algorithm evaluates all policies as equally abysmal performers.

Much work in the control community has been directed towards this problem, and the formulation known as H_∞ robust control has been a research focus for decades and the efficient computational solution to the linear problem has attracted a great deal of interest among practitioners. H_∞, generalized to non-linear problems is closely related to mini-max criterion described above as in this framework the controller is pitted against a *disturber* that can inject an L_2 bounded disturbance into the feedback loop. The controller seeks to maintain the stability of the system by attenuating this disturbance. The relation between disturbances in the H_∞ formulation and model error can be attributed to results like the *small-gain theorem* [van der Schaft, 1999]. The relationship is particularly sharp in the linear case, where the disturber can be identified with a norm-bounded (in the Hardy-space, H_∞, hence the name) linear perturbation. In the context of finite-state machines (deterministic MDP's), H_∞ reduces to the mini-max framework described above with the addition of costs incurred against the disturber for the "energy" required for applying each disturbance. [Morimoto and Doya, 2001] shows how H_∞ might be phrased in terms of reinforcement learning.

Stochastic models were developed to model situations where state transitions happen due to events that are largely unobservable and uncontrollable. While H_∞ is doubtless a useful scheme for control, there are many applications of practical interest where a stochastic model remains an excellent approximation of real system behavior, and a worst case deterministic model has both very poor fidelity with the real system and leads to controllers with performance so conservative as to have little practical value. Yet we still wish to ensure that our policies remain robust to errors made in modeling the stochastic process. A fundamentally different approach has been taken in *risk-sensitive* optimal control. In the risk sensitive framework, the basic structure of a stochastic model is retained, but the risk-neutral additive cost function is replaced with a cost function that emphasizes variance in cost occurred during control– in particular, risk-sensitive controllers will prefer a deterministic reward r to a lottery l with $E[l] = r$. This seems a natural preference– at least for human decision makers. Risk sensitive criteria can also be related to a type of model uncertainty; [Fleming and Hernandez-Hernandez, 1997] connects risk-sensitive control, where the value is defined as the time-average expectation of an exponential sum of costs, to a dynamic game, similar to the H_∞ problem, where the disturber influences next state transitions distributions, but pays a penalty for the relative entropy incurred at each step between the nominal model and the disturber's preferred next state distribution. The risk sensitive criterion thus provides a link to our desiderata of robustness, but lacks strucuture: deviations in the transition probabilities at every state are considered equally and no bound is imposed to prevent possibly arbitrarily large deviations from a nominal model by the disturber.

2 Model Uncertainty Sets

We propose to address the stochastic robustness problem more directly. Consider the problem of finding a stationary, Markovian policy that performs optimally in cumulative discounted reward in an uncertain Markov Decision Problem. Uncertainty will be described in terms of a set (henceforth the *uncertainty set*) of possible transition matrices, and optimality will be defined as maximizing cumulative discounted

reward starting from a given state. (Equivalently a distribution over states.) As the only thing to vary between the processes will be the transition functions, we will abuse notation somewhat and denote each MDP and its transition matrix, $P^a_{ij} = Pr(j|i,a)$ with the same symbol.

We denote the (finite) state space by $\mathcal{X}$ and the (finite) space of controls by $\mathcal{A}$. The discount factor in the MDP will be indicated by γ, the reinforcement function by $R(x)$, the uncertainty set of transition probabilities by $\mathcal{P}$, and the set of next state distributions drawn from $\mathcal{P}$ corresponding to a state i and control a as $\mathcal{P}^a_i$.

To ensure robustness we wish to find controllers that perform well on *all* models in the uncertainty set. Specifically we envision a static game in which a stationary Markovian controller is chosen, and then a model is chosen from the uncertainty set so as to minimize the expected reinforcement recieved by the controller. That is, the games value (to the controller) is defined as:

$$\min_{p\in\mathcal{P}} \max_{\pi\in\Pi} E_{p,\pi}[\sum_t \gamma^t R(X_t)] \tag{1}$$

2.1 General Sets

It is unfortunately the case that finding policies that are guaranteed to be good across unrestricted sets of models is computationally challenging. A reduction similar to Littman's proof [Littman, 1994] of the hardness of finding optimal memoryless policies in POMDPs proves the following result:

Proposition 1 *Finding the stationary memoryless policy that maximizes the least expected reward over an uncertainty set of Markovian Decision Problems is NP-hard.* ■

2.2 Convex Sets

We now consider more restricted uncertainty sets that we will show are both broad enough to be quite interesting with regard to applications, yet restrictive enough to admit tractable solutions.

Definition 1 *We call an uncertainty set of transition functions* convex *if for every action a, every state i, and any two transitions kernels T^a_i and S^a_i, all distributions "on the line" between the two sets are included. That is, all transition functions of the form*

$$V^a_i = \alpha(i,a)T^a_i + (1-\alpha(i,a))S^a_i$$

for any function $\alpha(i,a)$ taking its values in the range $[0,1]$, are also in the uncertainty set.

To support the claim that *convex uncertainty sets* are interesting, we provide some examples.

Example 1 *The class of uncertainty sets known as* interval-MDPs *[?] where every element of the transition matrix has bound constraints such as $\zeta^a_{ij} \le P^a_{ij} \le \psi^a_{ij}$ is a convex uncertainty set.*

Example 2 *The class of uncertainty sets where each row of the the transition matrix is constrained to lie in some* relative-entropy *ball around a nominal distribution ζ^a_i, $\{P^a_i | \mathcal{I}(P^a_i|\zeta^a_i) \le \epsilon^a_i\}$, describes a convex uncertainty set.*

Example 3 *In the mini-max model next states are chosen independently and deterministically for each state-action pair. The uncertainty set consisting of for each state-action pair the convex hull of the possible next states in the probability simplex constitutes a convex uncertainty set.*

A fundamental requirement of Definition (1) is that the uncertainty *factor* by state/action pair– that is, for each state-action pair the next-state distribution can be chosen independently of the next state-distribution chosen elsewhere in the state space. It is this condition that suggests the introduction of a stochastic dynamic game with turns between a controller that chooses an action a at each time step, and a disturber who counters with the next step distribution $P(j|i',a)$ from a set of allowable next state distributions $\mathcal{P}_i^a$. The value of each state i for the controller in the game is defined, with $X_0 := i$, as the (lower) limit of the discounted cost averaged over trajectories.

$$\liminf_{n\to\infty} \frac{1}{n} \sum_{t=0}^{\infty} \max_{a\in\mathcal{A}} \min_{p\in\mathcal{P}} \gamma^t R(X_t) \tag{2}$$

We note that the dynamic game described above is a zero-sum one of complete information. It follows from well know results in game theory [Basar and Olsder, 1995] that the following holds:

Lemma 1 *If $\mathcal{P}_x^a$ defines a compact region of the probability simplex for every x and a, then there exists for both the controller and disturber Markovian, stationary, and deterministic optimal strategies.*
Proof: This follows from noting that in the one time-step game the reinforcement incurred is continuous over a compact region and hence its minimum is attained. An appeal to the contraction mapping theorem standard in discounted MDPs and stochastic games proves the result. ■

Note that value-iteration in the dynamic game implicitly gives a policy for both the disturber and the controller by look-ahead in the one-step game. Surprisingly, although the dynamic game was introduced as a relaxation of the game defined in equation (1), as it appears to give much more power to the controller and disturber, the above result implies that under the further assumption that $\mathcal{P}$ is a convex uncertainty set (which as noted factors by state-action pairs), the dynamic game solution is *also* a solution to the static game (1) of uncertain models described above:

Theorem 1 *If $\mathcal{P}$ is a compact and convex uncertainty set, then the optimal disturbance is a stationary, Markovian distribution over next states and can be identified with an element of $\mathcal{P}$. Value-iteration in a compact convex uncertainty converges to the value function of the optimal policy with respect to worst model in $\mathcal{P}$.*

We present the algorithm:

Algorithm 1 *Robust Value Iteration*

1. *Initialize V to the zero vector the size of the state space.*
2. *Repeat until V converges:*

3. *For all states i and controls a, assign to matrix Q_{min} the solution to the optimization problem:*

$$Q_{min(i,a)} = \min_{p \in \mathcal{P}_i^a} E_p[V + R(i)]$$

4. *Update V by maximizing over the control set:*

$$V(i) = \max_{a \in \mathcal{A}} Q_{min(i,u)}$$

This algorithm can also be easily extended to run asynchronously and online. It is not clear however, that Algorithm (1) leads to a computable solution, as Step 3 requires a minimization over an uncountable set of probability distributions. Perhaps surprisingly it is not only computable but tractable:

Corollary 1 *Finding the (near)-optimal policy to guarantee performance (within a given ϵ) in a compact and convex uncertainty set of MDPs is a polynomial time problem, and Algorithm (1) solves the problem in polynomial time.*
Proof: (Sketch) Note that the minimization required in the algorithm is a convex program: The expectation is linear in p and thus forms a linear objective function subject to convex constraints. It is known that convex programs can be solved in polynomial-time (in the number of states) by interior point (e.g. ellipsoidal algorithm) methods. Further, each epoch of value-iteration reduces the error geometrically, so that in time logarithmic in ϵ, the maximum allowed sub-optimality, an approximately optimal policy can be computed. ■

We see that in sharp contrast to the general problem, convex uncertainty sets lead to a tractable solution to the uncertain model control problem. The algorithm presented, as it requires convex programs to be solved at each step can be quite expensive in practice, despite the polynomial time guarantees. However, the algorithm naturally takes advantage of structure in the uncertainty set. In the case of variation bounds on the probabilities, the convex program reduces to a linear program, and so for small structured uncertainty sets, the LP can reduce to a simple minimization over a finite set of possible disturbances corresponding to the vertices of the constraint set.

3 Solving Nearly Markovian Decision Problems

Under-modeling is a certainty; compromise is an inevitable consequence of the need for computationally tractable models– both learned and hand-designed. Ensuring that control algorithms aren't brittle with respect to these unmodeled dynamics is of paramount importance in model-based reinforcement learning and and planning problems. We briefly consider some notions of stochastic processes that behave as if they were "nearly" Markovian, and then discuss algorithms designed to behave robustly to this form of perturbation.

Definition 2 *Two controlled stochastic processes, P and Q are ϵ-close in variation if $P(x'|x,u) - \epsilon \leq Q(x'|x,u) \leq P(x'|x,u) + \epsilon$ holds for all x, x', u and all possible histories of the process.*

Definition 3 *Controlled stationary stochastic process P is ϵ-close in relative entropy to processes Q if for all possible histories of each process*

$$sup_{x,x',u} \mathcal{I}(P(x'|x,u)|Q(x'|x,u)) \leq \epsilon$$

Definition 4 *A controlled stochastic processes X is* boundedly non-markovian in variation*, with bound ϵ if there exists a Markov process with transition matrix T such that T and X are ϵ close in variation. T is denoted the nominal model for X.*

Definition 5 *A controlled stochastic process X is* boundedly non-markovian in relative-entropy *if there exists a Markov process with transition matrix T such that X is ϵ-close in relative entropy to T. T is denoted the nominal model for X.*

We first observe that MDP solutions possess some inherent robustness to bounded non-markovian behavior of a process:

Theorem 2 *Solving for the optimal policy in a nominal Markovian decision problem T, corresponding to a boundedly non-markovian decision problem (with bound ϵ in either variation or relative entropy) induces error over the optimal policy that is polynomially bounded in ϵ.* *Proof:*

Note that bounded relative entropy implies polynomially bounded variation and thus w.l.o.g. we consider only the variation case. At each time step the loss is bounded by $2\epsilon R_{\max}$, so that the total error is bounded by $2\epsilon R_{\max}/(1-\gamma)$. ■

As discussed in the introduction, methods other than stochastic optimal control with respect to the nominal model exist for mitigating the impact of non-markovian behavior. Of particular interest is the H_∞ paradigm, where energy costs are associated with disturbances injected at each time step.

The simple modification of step (3) of Algorithm (1) to include energy costs as follows:

$$Q_{min(i,a)} = \min_{p \in \mathcal{P}_i^a} E_p[V + R(i) + \lambda S(i,j)]$$

where $S(i,j)$ is a non-negative cost associated with each transition (from i to j), representing energy injected by the disturber to induce unfavorable transitions and λ is the scalar H_∞ coefficient, generalizes the standard discrete state, discounted H_∞ problem. It is easy to recover the standard formulation by relaxing all of the next-state distribution constraints to include the entire simplex, and ensuring that for every state i there exists a j so $S(i,j) = 0$.

Noting that the dynamic game introduced in Section (3) does not require the Markov property, and recognizing that both definitions of boundedly non-markovian processes form convex and compact uncertainty sets, we immediately have an algorithm that computes the strongest possible performance bounds on this very general class of controlled stochastic processes, and computes stationary, Markovian policies that achieve that bound.

4 Experiments

4.1 Path Planning

Robot path planning has been the subject of a great deal of research effort, but the vast majority of it concentrates on static obstacles. Recent research on mobile robots operating in crowded domains [Burgard *et al.*, 1999] has revealed some of the limitations of this approach, and further recent research [Montemerlo and Thrun, 2001] has demonstrated that tracking dynamic objects is feasible using sequential monte-carlo filtering applied to the laser-range finder data collected by the robot. A

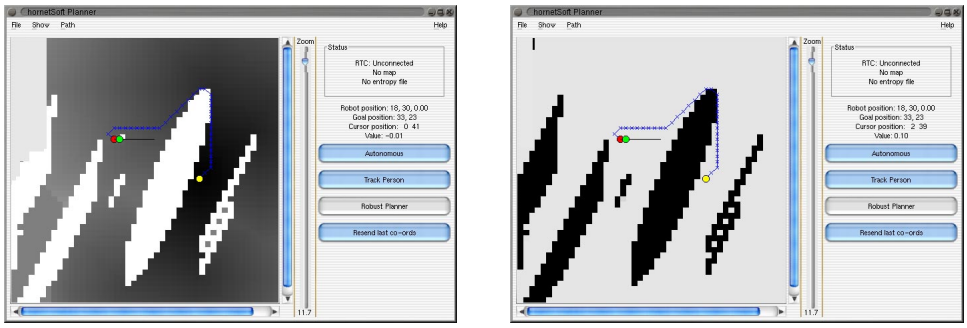

Figure 1: These figures illustrate a path planned by the robot (dark leftmost circle) giving a dynamic obstacle (lighter circle adjacent to the robot) a wide berth on it's route to the goal (light circle on far right). The figure on right illustrates the value function compute by the robust dynamic programming algorithm while the figure on the right illustrates the path taken.

significant difficulty in planning with dynamic obstacles is the difficulty of modeling such objects– the dynamics of a human obstructing the robot's path are stochastic and possibly very high dimensional.

In this work, we developed a new planner for the Nursebot project, modeling a dynamic obstacle in the robot's path in the uncertain MDP framework. Specifically, the dynamic obstacle is modeled as a "lazy" random walk with unknown and time-varying drift. In the eight connected grid used for planning, the dynamic obstacle is subject only to the linear constraints:

$$Pr(\textit{Obstacle doesn't move}) \geq .25$$

This gives a vast space of possible dynamics (transition matrices) for the object whose details we leave unspecified. In this model the dynamic obstacle is thought of as inadvertently adversarial (perhaps not an unrealistic assumption for some humans interacting with the robot), and we require path planning to remain robust to all such obstacles.

In many situations paths resulting from the uncertain MDP solutions result in paths that show greater deference to the impedance caused by the dynamic obstacle than solutions that treat the object as static. Figure (1) illustrates the robot circuiting the obstacle, and the resulting value function. Lighter regions indicated regions of higher cost. As the resulting plans are feedback strategies, the illustrations depict the plan resulting from the object staying still. (The computed strategy, of course, does not.) A conventional planner skirts the obstacle to minimize cost.

In the Figure (2) we compare solutions derived by the standard planner and the robust planner, noting that to avoid interference from the dynamic obstacle, the robust controller takes a longer and very different path to the goal, despite the existence of a feasible (although potentially quite difficult) path between the dynamic obstacle and the other impediments in the environment.

The behaviors engendered by this robust planner have some very desirable characteristics for planning with dynamic obstacles.

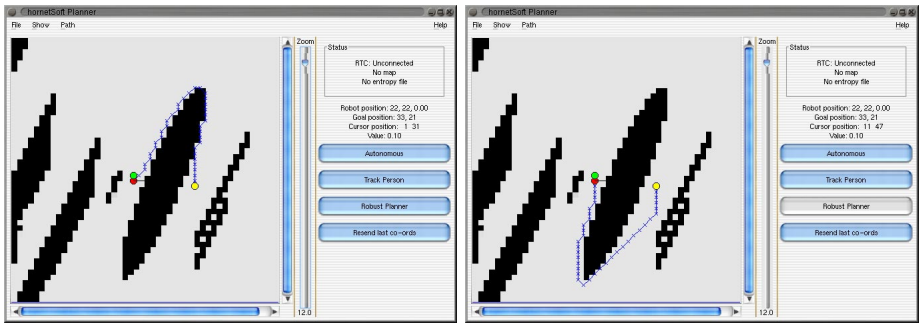

Figure 2: The left figure shows the plan generated by the conventional dynamic programmer. On the right the uncertain MDP solution leads to a path (from the dark circle robot to the light circle goal) that makes a greater effort to avoid the dynamic obstacle.

4.2 The Mountain-Car POMDP

We briefly illustrate how the algorithm described in the section above can be used to solve certain non-Markovian problems. In particular, we consider a variation on the well-studied "Mountain-Car" problem, where unlike the standard formulation, only an approximate, discrete representation of the state is available for control.

The dynamics used are the standard ones given in [Sutton and Barto, 1998], but complete state observability is replaced by discrete valued sensors that output the car position to within .085 position units ($\frac{1}{20}$ of the cars position range) and .007 velocity units ($\frac{1}{20}$ of the maximum speed). Two approaches are presented to solving the problem. First the state uncertainty is treated by approximating the problem with a Markovian one. Assuming that for a given sensor measurements all states that result in this sensor measurement are equally likely, we compute transition probabilities by sampling one step forward from a uniform distribution over each sensor grid cell. Standard dynamic program is run on the resulting discrete MDP. Next, we use the robust value iteration described above to model the possible error introduced by assuming the problem is a discrete, fully-observed MDP. Sampling is again used to compute possible next state distributions, and the minimization step of Algorithm (1) is approximate with a simple minimization over a finite sample set. To ensure that the mountain car doesn't get "stuck" in certain states, additional time-extended actions are available to the controller that simply perform the accelerate, decelerate, or do nothing for 2,4, and 8 periods consecutively. These simply correspond to short, open-loop strategies the controller can choose to use.

Both methods are able to find satisficing controllers, however, the robust algorithm's value function serves as a lower bound on the actual cost when the controller is run on a simulation of the mountain car dynamics. The MDP solution often significantly under-estimates the actual cost of a state. (Figure 2) The controller developed using the robust value iteration algorithm is quite conservative, preferring multiple swing-ups to ensure success. In a number of places it still out-performs the standard controller. (Figure 3).

It is interesting to observe that the sharp under-estimates of the true cost of a policy in the MDP do *not* vanish with increased resolution of the discretization. (See Figure(4.2)). This is a very general phenomenon that occurs along switches in the optimal policy, so that given a δ-fine discretization, one can expect these kind of inaccurcate value estimates to occur on $O(1/\delta)$ of the state space. MDP

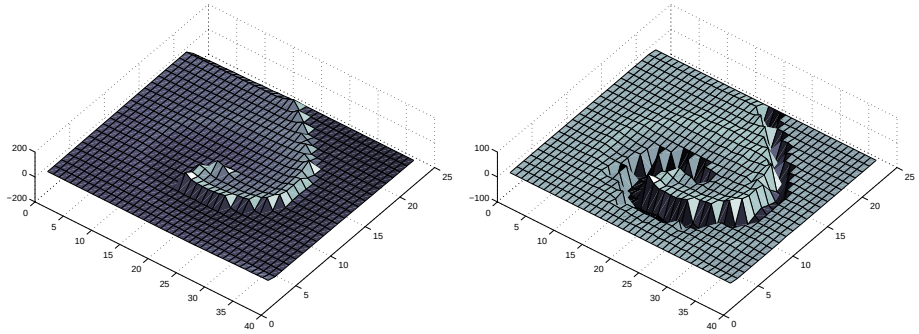

Figure 3: A graphical depiction over the coarsely discretized state-space of the relative performances of the robust and standard controllers. The left graph illustrates the degree to which the standard control underestimates the true value of each state, while the right graph illustrates the relative performance of the two controllers in simulations. Positive values indicated the robust solution out-performed the standard one.

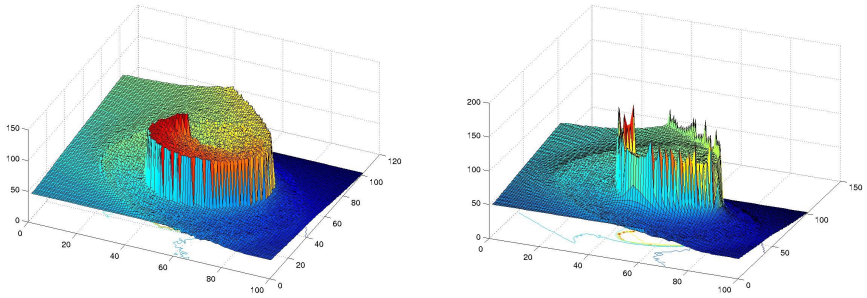

Figure 4: The left figure illustrates the mountain-car value-function computed by Algorithm (1) on a fine discretization of state-space. On the right the we see the simulated performance of the MDP solution. Note the difference in vertical scale.

solutions are helped by noise in this case as it smoothes the dynamics, making the uncertainty in the transitions due to discretization relatively small.

Although space does not permit details of experiments, it is worth noting that the framework of uncertain MDPs is very natural for model-based reinforcement learning, and enables the learning of controllers that can guarrantee (with high probability) performance while learning, ensuring robustness that the certainty-equivalent approach cannot. This provides an efficient dynamic programming alternative to the policy search methods for robustness in reinforcement learning presented in [Bagnell and Schneider, 2001].

5 Conclusions and Further Work

The authors in this work have considered diverse application of the stochastic robustness framework including bounding error due to discretization, guaranteeing performance in reinforcement learning problems, and robust planning for a difficult to model dynamic obstacle, but by no means have exhausted the potential of the

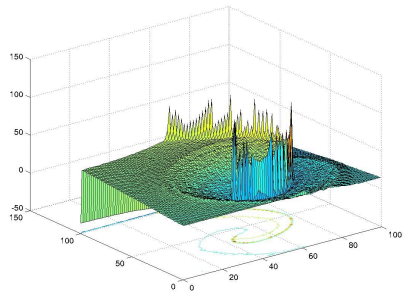

Figure 5: Error in the MDP cost estimate at a fine discretization.

approach presented. Future work may consider the application to variable resolution discretization in optimal control, as the technique naturally provides error bounds on the discretization. It would also be interesting to apply the technique to the kind of "assumed-density planning" demonstrated as practical POMDP solution algorithm by [Roy and Thrun, 1999] and [Rodriquez *et al.*, 1999] to achieve a measure of robustness to the errors introduced by the reduced belief-state.

Acknowledgements

The authors would like to thank Laurent el Ghaoui, Geoff Gordon, Sebastian Thrun and Remi Munos for enlightening conversation on the topics presented. The authors are indebted to Nicholas Roy, whom presented the opportunity and the means to develop planner for the Nursebot Pearl. Drew Bagnell is supported by a National Science Foundation Fellowship and Andrew Ng is supported by a Berkeley Fellowship.

References

[Bagnell and Schneider, 2001] J. Bagnell and J. Schneider. Robustness and exploration in policy-search based reinforcement learning. In *Proceedings of the 2001 IEEE Int. Conference on Robotics and Automation*. IEEE, 2001.

[Basar and Olsder, 1995] T. Basar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1995.

[Burgard *et al.*, 1999] W. Burgard, A.B. Cremers, D. Fox, D. Haehnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. In *To appear in Artificial Intelligence*, 1999.

[Fleming and Hernandez-Hernandez, 1997] W. H. Fleming and D. Hernandez-Hernandez. Risk-sensitive control of finite state machines on an infinte horizon i. *SIAM Journal of Control and Optimization*, 35, 1997.

[Heger, 1994] M. Heger. Consideration of risk in reinforcement learning. In *International Conference on Machine Learing*, 1994.

[Littman, 1994] M. Littman. Memoryless policies: Theoretical limitations and practical results. In *From Animal to Animats 3: Proceedings of the 3rd International Conference on Simulation and Adaptive Behavior*, 1994.

[Montemerlo and Thrun, 2001] M. Montemerlo and S. Thrun. Personal communications, 2001. regarding obstacle tracking in the Nursebot project.

[Morimoto and Doya, 2001] Jun Morimoto and Kenji Doya. Robust reinforcement learning. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 1061–1067. MIT Press, 2001.

[Rodriquez *et al.*, 1999] A.C. Rodriquez, R. Parr, and D. Koller. Reinforcement learning using approximate belief states. In *Advances in Neural Information Processing Systems 11*, 1999.

[Roy and Thrun, 1999] N. Roy and S. Thrun. Coastal navigation with mobile robots. In *Advances in Neural Information Processing Systems 11*, 1999.

[Sutton and Barto, 1998] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[van der Schaft, 1999] A. J. van der Schaft. *L_2-gain and Passivity techniques in Non-linear Control*. Springer-Verlag, 1999.