Planning for Closed-Loop Execution Using Partially Observable Markovian Decision Processes

Lonnie Chrisman[*]
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
chrisman@cs.cmu.edu

Abstract

A distinction is drawn between open-loop and closed-loop plans, where the latter explicitly specifies how run-time feedback is to be acquired and used. It is argued that some planning problems, especially those involving deliberation about information gathering activities, require planners to generate closed-loop plans. Partially Observable Markovian Decision Processes (POMDPs) are proposed as a general representation for physical and perceptual actions with uncertainty. This paper reports progress in using POMDPs in planning for situated, autonomous, closed-loop execution.

Closed-Loop Planning

After a planner derives a course of action for a situated agent, additional feedback from the environment typically becomes available to the agent during the execution of the plan. This feedback can, and in almost every situated system does, influence the intended future course of execution. However, the knowledge that feedback will be forthcoming may or may not be utilized by the planner at planning time. Planning systems that do not take advantage of this knowledge simply produce open-loop plans, where the expectation or probability is high that the plan will succeed if executed blindly (i.e., without sensing). The fact that the plan is not executed blindly is ignored at planning time and simply means that the execution will be somewhat more reliable than the planner might have predicted. The vast majority of existing planning systems produce open-loop plans (e.g., [Sacerdoti, 1975], [Chapman, 1987], [Dean *et al.*, 1990], [Kanazawa and Dean, 1989], [Wellman, 1990], [Drummond and Bresina, 1990]).

Instead of ignoring the fact that additional information will become available during execution, this knowledge can be leveraged *at planning time* to produce closed-loop plans. Closed-loop plans contain explicit contingencies, action choices that depend upon information obtained during execution, and even sequences of actions which serve to gather information upon which to base future decisions. The deliberative control of sensing, for example, mandates closed-loop planning when the justification for performing a sensing operation rests upon the impact the resulting information is expected to have on the agent's ability to choose appropriate actions ([Howard, 1966], [Hager, 1990], [Chrisman and Simmons, 1991], [Ogasawara, 1991], [Whitehead and Ballard, 1990]). There are also many additional examples of planning problems that require the consideration of expected feedback at planning time in order to choose reasonable future courses of action.

Planning for closed-loop execution requires reasoning about uncertainty and about what information will and will not be observed during future execution. It also involves evaluating costs (of actions and sensing) and the associated tradeoffs — tradeoffs commonly requiring consideration of very large or infinite time horizons. These considerations can commonly be represented as Partially Observable Markovian Decision Processes (POMDPs) [Koenig, 1991].

The next section begins by examining some planning problems that require reasoning about closed-loop plans. The remainder of the paper is devoted a review of POMDPs, and of approaches and concerns related to planning for closed-loop execution using POMDPs as probabilistic models for action.

Planning Examples

There are many instances of planning problems where finding even the optimal open-loop plan fails to provide an adequate solution. To generate adequate solutions for problems of this type, a planner must consider how information is to be obtained during execution and how such information may change future action choices. A few example problems are described here in order to demonstrate the importance of accounting for expected feedback at planning time.

Planned Selective Sensing: For most conventional planners, actions produce physical changes in the

[*]This research was sponsored by NASA under contract number NAGW-1175. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of NASA or the U.S. government.

state of the world. However, more generally an agent may have some amount of explicit control over its perceptual system with the ability to select between sensing operations. Pure sensing operations do not alter the world state — they only provide information to the agent about the world state.

The importance of task-directed selective sensing or the need to selectively focus perceptual attention is far from controversial ([Abramsom, 1991], [Chrisman and Simmons, 1991], [Dean *et al.*, 1990], [Reece and Shafer, 1991], [Simon, 1983], [Whitehead and Ballard, 1990]), yet open-loop planning is severely deficient on this account because it is never rational to include a pure sensing operation as part of an open-loop plan. Because, by definition, an open-loop plan makes no explicit consideration of run-time feedback, any information obtained during execution will not influence the future course of execution suggested by the plan. Since it is assumed that any selectable sensing operation comes with at least some cost, paying for an action that is believed to have no behavioral influence constitutes an irrational choice. The fundamental point is that a pure open-loop planner cannot rationally plan sensing operations.

In the real-world, open-loop plans do not always perform as predicted; therefore, it is necessary to perform sensing during execution in order to detect failures. Execution monitoring subsystems are usually included in systems with open-loop plans ([Fikes and Nilsson, 1971], [Wilkins, 1985], [Doyle *et al.*, 1986]). Unlike the open-loop plan itself, the execution monitor can alter the course of action as the result of run-time feedback (either by replanning or repairing), and thus it may be rational for an execution monitor to perform sensing operations. However, as the next example shows, the separation between the gathering of information and the planning process may be undesirable.

Contingency Costs: The cost of recovering from failures, where the actual action outcome differs from the anticipated outcome, varies wildly in the real-world depending upon the nature and context of the failure. For example, the expected cost of recovering from someone suddenly appearing on the sidewalk in front of an agent is considerably less than the expected cost of recovering from the loss of footing while walking a tightrope over Niagara Falls. Recovery costs (or more generally, contingency costs) constitute a very important consideration in many problems, but unlike the preceding two extreme examples, expected recovery costs are seldom immediately available. To estimate the cost of a recovery, it is generally necessary to consider how to recover, which is a new planning problem in and of itself [Abramsom, 1991]. In short, properly estimating recovery costs requires contingency planning.

Open-loop planners do not account for contingency costs. Maximizing the probability of success can produce poor and non-intuitive plans when contingency costs vary greatly. Maximizing expected utility for a single action sequence is also insufficient since the sequence used for a recovery will differ from the anticipated action sequence. In addition, recovery costs affect choices that occur before unanticipated failures, so this issue is separated from execution monitoring and must be address earlier — at planning time.

From the perspective of closed-loop planning, it is rather unnatural to use terms like "failure" and "recovery" since the planner may anticipate and plan for many outcomes. It is more natural to view all cases as possible outcomes where some outcomes are more likely than others. Although the ability to produce closed-loop plans is necessary for problems like these, the consideration of contingencies by agents with limited computational resources ([Russell and Wefald, 1991], [Simon, 1983]) introduces additional tradeoffs ([Doyle and Wellman, 1990]) between obtaining accurate cost estimates versus deferring the planning of unlikely contingencies ([Olawsky and Gini, 1990], [Gervasio, 1990]).

POMDPs

The theory of Partially Observable Markovian Decision Processes (POMDPs) provides a very general representational model for actions and closed-loop planning. The most distinct characteristic of POMDP models is that the representation of perceptual input is clearly separated from the representation of world state. Action models specify not only how the action changes the world state, but also how the world state and the action identity determine what is sensed. This separation is crucial for reasoning about what information needs to be sensed and how it can best be gathered.

In a discrete-time, finite POMDP, the world is at any given moment in one of a finite number of states $S = \{s_1, s_2, ..., s_n\}$. At discrete time intervals the agent chooses actions from a finite set A. The world is assumed to evolve stochastically according to the Markov assumption, modeled by the transition probabilities $p_{ij}^a = Pr\{S_{t+1} = j | S_t = i, A_t = a\}$. The agent, however, cannot directly access the true state of the world, and must obtain all information about the state through (possibly noisy) observations. Let θ_t denote the observation at time t. The relationship between the state and observations is modeled by $r_{jk}^a = Pr\{\theta_t = k | S_{t+1} = j, A_t = a\}$. A real-valued expected reward, $g(S_t, A_t)$, is received after each action and is used to specify desired world states (eg. goals) as well as action and sensing costs. Planning with a POMDP involves maximizing some measure of long-term utility, usually "average expected reward per action" or "cumulative discounted reward."

POMDPs have been considered for a wide variety of applications, including machine maintenance, quality control, internal auditing, economics, searches, military encounters, and data communications [Monahan, 1982]. My interests are in using POMDPs to model physical situated action in the context of planning under uncertainty and selective perception. Purely physical actions can be easily modeled by setting $r_{j0}^a = 1$ and $r_{jk}^a = 0$

for $k > 0$. Pure sensory operations (even with sensor noise) can be modeled as actions that do not change the world state ($p_{ii}^a = 1$), but which do return information. And more generally, physical actions with state-dependent feedback are naturally represented [Koenig, 1991]. POMDPs naturally support reasoning about optimality and about the value of obtaining information (either perfect or noisy) [Howard, 1966].

Drawbacks

There are two major deficiencies that come along with POMDP's generality. The first is related to the frame problem and involves a lack of representational conciseness. The second, more serious problem is the large computational complexities involved in generalized closed-loop planning.

In their pure form as stated above, POMDP models require explicit enumeration of all states and state transitions. In A.I., where the number of states is typically astronomically large, this is clearly infeasible. However, POMDPs are not any different than other probabilistic models in this respect, and statistical independence may be leveraged to obtain concise representations (see [Wellman, 1990], [Dean and Wellman, 1991]) in the form of Influence Diagrams ([Howard and Matheson, 1984], [Shachter, 1986]). Influence Diagrams have already become common place in the A.I. community, and can usually be considered to be a special case of POMDPs evaluated over a finite-time horizon.

Unfortunately, Influence Diagrams do not help with the very severe computational complexity associated with POMDP-based planning. Exact algorithms exist that compute optimal policies. Most of these are based on Sondik's seminal work ([Smallwood and Sondik, 1973], [Sondik, 1978]) but also include algorithms for evaluating influence diagrams [Shachter, 1986]; however, these formulations can all be shown to be at least P-space hard[1] in general. Approximation techniques have also been developed [Lovejoy, 1991], but these are still far too computationally complex for the applications being considered here. This intractability and the complex intricacies of these algorithms have also impeded the widespread application of POMDPs in other disciplines [Lovejoy, 1991].

It is very clear that general purpose, exact techniques will always be infeasible; furthermore, it is likely that general purpose approximation techniques will remain inapplicable to the sorts of very large state spaces involved in planning for situated action. However, I have considerable hope that reasonable approximation techniques can be obtained for the specific types of problems encountered by situated agents in real-world situations. These techniques must harness inherent properties of situated action and sensing, and the regularities of the real-world[2]. I have made some progress toward these ends and report two reasonable approaches below.

Exploiting Reactive Execution

Reactive systems, where the choice of action is (almost) a function of current percepts, have achieved considerable success in the past few years at controlling situated robots (e.g., [Brooks, 1991], [Connell, 1989]). Behaviors can potentially be generated by a planner ([Rosenschein and Kaelbling, 1986], [McDermott, 1990]), and these recent successes suggest that reactivity may be a good property to exploit in order to obtain computationally feasible closed-loop planners.

The basic idea is for the planner to consider only reactive behaviors, eliminating all other options from consideration. In [Chrisman and Simmons, 1991] we found that this corresponds to the use of "static sensing policies," and can result in very substantial computational advantages. This class of purely reactive or statically sensed plans can also be viewed as the class of plans whose execution requires no internal state in a plan interpretor [Gat, 1991]. Especially with problems involving incomplete or selective attention, this turns out to be a critical limitation [Chrisman *et al.*, 1991], but the benefits of the general approach are not lost. Techniques such as hierarchical task decomposition [Chrisman and Simmons, 1991] can be used to introduce very small amounts of internal execution state, resulting in nearly reactive closed-loop plans, which may have considerable potential for effective application.

Exploiting Predictability and Observability

In many cases, an agent's action models may be quite good, the effects of actions predictable and almost deterministic, and the observability of the world state high. During execution, the agent continually applies Bayesian conditioning to maintain an updated belief (state mass distribution) about the current state. The result is that an agent will usually have a very good idea (with little uncertainty) what the current world state is. Some very preliminary experiments (using POMDP models learned autonomously [Chrisman, 1992]) have indicated that this may often be more the rule than the exception for various non-exploration tasks that require

[1] The P-space hardness result for influence diagrams follows directly from the P-space hardness result for POMDPs given in Theorem 6 in [Papadimitriou and Tsitsiklis, 1987].

[2] The best known such property is that the vast majority of state variables remain unaffected by any single action execution. This is the basis for the frame problem ([McCarthy and Hayes, 1969]). [Wellman, 1990] has shown that the ability to include only the direct effects in the representation of an action is a result of the conditional independence between the action and all current and future state variables, given the direct effects of the action. The ability to harness conditional independence for computational advantage is, therefore, extremely important. Due to the pervasiveness of this topic in the literature on Bayesian reasoning, and because I have nothing new to contribute here, this aspect is not considered further in this paper.

tracking partially observable aspects of the world. This property can be harnessed in a straightforward fashion to obtain an approximate planner.

At any moment in time, the current belief about the world can be summarized by a vector $\vec{\pi} = \langle \pi_1, \pi_2, ..., \pi_n \rangle$, such that $\vec{\pi} \in \mathcal{R}^n$ is a point in the (n-1)-dimensional unit simplex. π_i represents the probability that the world is in state i. While it is well known that the optimal utility is a complicated function $\delta : \pi \rightarrow \mathcal{R}$, in this case we can approximate the long-term utility of choosing action a by

$$V(a, \vec{\pi}) \approx \sum_{i=1}^{n} \pi_i V_i(a) \qquad (1)$$

where $V_i(a)$ is a recorded measure of the expected utility if the agent is in state i and executes action a. $V_i(a)$ is not the utility that would result in a totally observed MDP. Instead, $V_i(a)$ depends upon the typical uncertainty distributions experienced by the agent, and therefore depends upon the particular agent, environment, current policy, and tasks that the agent performs. The implication is that $V_i(a)$ must be learned through experience [Chrisman, 1992] rather than calculated. The optimal action is chosen according to

$$a^* = \arg\max_a V(a, \vec{\pi}) \qquad (2)$$

Equation (1) becomes increasingly accurate as the entropy in $\vec{\pi}$ decreases, corresponding to more certainty about the current world state. It also is reasonable when the uncertainty is distributed over a set of states, but where all these states agree on the optimal action. This latter case is important when addressing the frame problem and the importance of physically local state information for a situated agent. Ignorance about what is in the next room will generally not impact the optimality of the current action.

While (1) and (2) can be used to select physical actions, they cannot be directly applied to evaluating the effects of gaining information in the future. However, information gathering actions (such as with selective sensing) can be accommodated by searching sequences of immediate sensor feedback, and then using (1) and (2) to evaluate the utility improvement from the expected information gain. In general, even for physical action planning alone, this search improves the accuracy of the approximation in (1). Consider the application of action a to the situation described by belief $\vec{\pi}$. If Θ is the set of possible observations, then the resulting utility can be approximated by

$$V(a, \vec{\pi}) \approx \sum_{i=1}^{n} \pi_i g(s_i, a) + \gamma \sum_{\theta \in \Theta} Pr\{\theta | a, \vec{\pi}\} \max_{a'} V(a', T(\vec{\pi}, \theta, a)) \qquad (3)$$

where γ is a discount factor, and $T(\vec{\pi}, \theta, a) = \vec{\pi} P^a R^a(\theta) / \vec{\pi} P^a R^a(\theta) \vec{\mathbf{1}}$ is the projected state mass distribution the agent will believe if it perceives θ after executing the action a. Using (3), the search can be expanded to arbitrary depth with increasing accuracy, but since this must be carried out at every time point, it is expected that the search will generally stop at a depth of one or two.

Conclusion

POMDPs provide a powerful representational mechanism for actions and sensory operations. However, in general they are highly intractable. Generalized solution techniques, and even generalized approximation techniques, remain inapplicable to planning for situated closed-loop execution. I am optimistic that efficient approximation techniques designed specifically for situated interaction with the real-world, such as for autonomous agents, can be developed and utilized effectively. Two approaches were presented that I have been developing for such application. While these approaches appear promising, empirical support for their effectiveness remains a topic for future research.

References

[Abramsom, 1991] Bruse Abramsom. An analysis of error recovery and sensory integration for dynamic planners. In *Proc. of Ninth National Conference on Artificial Intelligence*, 1991.

[Brooks, 1991] Rodney A. Brooks. Intelligence without reason. In *Proc. IJCAI*, 1991.

[Chapman, 1987] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32, 1987.

[Chrisman and Simmons, 1991] Lonnie Chrisman and Reid Simmons. Sensible planning: focusing perceptual attention. In *Proceedings of Ninth National Conference on Artificial Intelligence*, 1991.

[Chrisman *et al.*, 1991] Lonnie Chrisman, Rich Caruana, and Wayne Carriker. Intelligent agent design issues: internal agent state and incomplete perception. In *AAAI Fall Symposium Series: Sensory Aspects of Robotic Intelligence*, Monterey, CA, November 1991.

[Chrisman, 1992] Lonnie Chrisman. Reinforcement learning with perceptual aliasing: the predictive distinctions approach. Submitted to AAAI, 1992.

[Connell, 1989] Jonathan Connell. A colony architecture for an artificial creature. Technical Report AI TR-1151, MIT, June 1989.

[Dean and Wellman, 1991] Thomas L. Dean and Michael P. Wellman. *Planning and control.* Morgan Kaufmann Publishers, 1991.

[Dean *et al.*, 1990] Thomas Dean, Kenneth Bayse, and Moises Lejter. Planning and active perception. In *Workshop on Innovative Approaches to Planning, Scheduling, and Control*, 1990.

[Doyle and Wellman, 1990] John Doyle and Michael Wellman. Rational distributed reason maintenance for planning and replanning of large-scale activities. In *Workshop on Innovative Approaches to Planning, Scheduling, and Control*, 1990.

[Doyle *et al.*, 1986] Richard J. Doyle, David J. Atkinson, and Rajkumar S. Doshi. Generating perception requests and expectations to verify the execution of plans. In *Proc. of Fifth National Conference on Artificial Intelligence*, 1986.

[Drummond and Bresina, 1990] Mark Drummond and John Bresina. Anytime synthetic projection: Maximizing the probability of goal satisfaction. In *Proc. of Eighth National Conference on Artificial Intelligence*, 1990.

[Fikes and Nilsson, 1971] Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[Gat, 1991] Erran Gat. On the role of internal state in the control of autonomous mobile robots. In *AAAI Fall Symposium Series: Sensory Aspects of Robotic Intelligence*, Monterey, CA, November 1991.

[Gervasio, 1990] Melinda T. Gervasio. Learning general completable reactive plans. In *Proc. of Eighth National Conference on Artificial Intelligence*, 1990.

[Hager, 1990] Gregory D. Hager. *Task-directed sensor fusion and planning: a computational approach.* Kluwer Academic Publishers, 1990.

[Howard and Matheson, 1984] Ronald A. Howard and James E. Matheson. Influence diagrams. In *The principles and applications of decision analysis.* Strategic Decisions Group, Menlo Park, CA, 1984.

[Howard, 1966] Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, SSC-2(1):22–26, August 1966.

[Kanazawa and Dean, 1989] Keiji Kanazawa and Thomas Dean. A model for projection and action. In *Proc. IJCAI*, 1989.

[Koenig, 1991] Sven Koenig. Probabilistic and decision-theoretic planning using Markov decision theory. Master's thesis, U.C. Berkeley, 1991.

[Lovejoy, 1991] William S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991.

[McCarthy and Hayes, 1969] John McCarthy and Pat J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[McDermott, 1990] Drew McDermott. Planning reactive behavior: A progress report. In *Workshop on Innovative Approaches to Planning, Scheduling, and Control*, 1990.

[Monahan, 1982] George E. Monahan. A survey of partially observable Markov decision processes. *Management Science*, 28:1–16, 1982.

[Ogasawara, 1991] Gary H. Ogasawara. Control of sensing and planning using distributed processing and decision analysis. In *AAAI Fall Symposium Series: Sensory Aspects of Robotic Intelligence*, Monterey, CA, November 1991.

[Olawsky and Gini, 1990] Duane Olawsky and Maria Gini. Deferred planning and sensor use. In *Proc. DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, November 1990.

[Papadimitriou and Tsitsiklis, 1987] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, August 1987.

[Reece and Shafer, 1991] Douglas A. Reece and Steven Shafer. Using active vision to simplify perception for robot driving. Technical Report CMU-CS-91-199, Carnegie Mellon University, November 1991.

[Rosenschein and Kaelbling, 1986] Stanley J. Rosenschein and Leslie Pack Kaelbling. The synthesis of machines with provable epistemic properties. In *Proceedings of Conference on Theoretical Aspects of Reasoning about Knowledge*, 1986.

[Russell and Wefald, 1991] Stuart Russell and Eric Wefald. *Do the right thing: studies in limited rationality.* MIT Press, 1991.

[Sacerdoti, 1975] Earl D. Sacerdoti. The nonlinear nature of plans. In *Proc. IJCAI*, 1975.

[Shachter, 1986] Ross D. Shachter. Evaluating influence diagrams. *Operations Research*, 36, 1986.

[Simon, 1983] Herbert A. Simon. *Reason in Human Affairs.* Stanford University Press, 1983.

[Smallwood and Sondik, 1973] Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable Markov decision processes over a finite horizon. *Operations Research*, 21, 1973.

[Sondik, 1978] Edward J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: disounted case. *Operations Research*, 26:282–304, 1978.

[Wellman, 1990] Michael P. Wellman. The STRIPS assumption for planning under uncertainty. In *Proc. of Eighth National Conference on Artificial Intelligence*, 1990.

[Whitehead and Ballard, 1990] Steven D. Whitehead and Dana H. Ballard. Active perception and reinforcement learning. In *Proc. of Seventh International Machine Learning Conference*, 1990.

[Wilkins, 1985] David E. Wilkins. Recovering from execution errors in SIPE. *Computational Intelligence*, 1:33–45, 1985.