Reactive and Automatic Behavior in Plan Execution

Pat Langley
Robotics Laboratory
Computer Science Dept.
Stanford University
Stanford, CA 94305
LANGLEY@CS.STANFORD.EDU

Wayne Iba
RECOM Technologies
Mail Stop 269-2
NASA Ames Research Center
Moffett Field, CA 94035
IBA@WIND.ARC.NASA.GOV

Jeff Shrager
Palo Alto Research Center
Xerox Corporation
3333 Coyote Hill Road
Palo Alto, CA 94304
SHRAGER@XEROX.COM

Abstract

Much of the work on execution assumes that the agent constantly senses the environment, which lets it respond immediately to errors or unexpected events. In this paper, we argue that this purely reactive strategy is only optimal if sensing is inexpensive, and we formulate a simple model of execution that incorporates the cost of sensing. We present an average-case analysis of this model, which shows that in domains with high sensing cost or low probability of error, a more 'automatic' strategy – one with long intervals between sensing – can lead to less expensive execution. The analysis also shows that the distance to the goal has no effect on the optimal sensing interval. These results run counter to the prevailing wisdom in the planning community, but they promise a more balanced approach to the interleaving of execution and sensing.

Reactive and Automatic Execution

Much of the recent research on plan execution and control has focused on *reactive* systems. One central characteristic of such approaches is that the agent senses the environment on each time step, thus ensuring that it can react promptly to any errors or other unexpected events. This holds whether the agent draws on large-scale knowledge structures, such as plans (Howe & Cohen, 1991) or cases (Hammond, Converse, & Marks, 1988), or bases its decisions on localized structures, such as control rules (Bresina, Drummond, & Kedar, 1993; Grefenstette, Ramsey, & Schultz, 1990) or neural networks (Sutton, 1988; Kaelbling, 1993).

However, human beings still provide the best examples of robust physical agents, and the psychological literature reveals that humans do not always behave in a reactive manner. People can certainly operate in reactive or 'closed-loop' mode, which closely couples execution with sensing (Adams, 1971). But at least in some domains, humans instead operate in *automatic* or 'open-loop' mode, in which execution proceeds in the absence of sensory feedback (Schmidt, 1982). Thus, at least in some contexts, successful agents appear to prefer nonreactive strategies to reactive ones.[1]

One explanation for this phenomenon is that there exists a tradeoff between the cost of sensing, which models of reactive agents typically ignore, and the cost of errors that occur during execution. For some situations, the optimal sensing strategy is completely reactive behavior, in which the agent observes the environment after each execution step. For other situations, the best strategy is completely automatic behavior, in which execution occurs without sensing. In most cases, the optimum will presumably fall somewhere between these two extremes, with the agent sensing the world during execution, but not after every step.

There exist other reasons for preferring automatic to reactive behavior in some situations. At least for humans, the former appears to require fewer attentional resources, which lets them execute multiple automatic procedures in parallel. Humans also exhibit a well-known tradeoff between speed and accuracy, and in some cases one may desire an automated, rapid response to a reactive, accurate one. However, our goal here is not to provide a detailed account of human execution strategies, but to better understand the range of such strategies and the conditions under which they are appropriate. Thus, we will focus on the first explanation above, which assigns an explicit cost to the sensing process.

In the following pages, we attempt to formalize the tradeoff between the cost of sensing and the cost of errors, and to identify the optimal position for an agent to take along the continuum from closed-loop, reactive behavior to open-loop, automatic behavior. In the next section, we present an idealized model of execution that takes both factors into account, followed by an analysis

[1] Note that the distinction between reactive and automatic behavior is entirely different from the more common distinction between reaction and deliberation. The former deals exclusively with sensing strategies during execution, whereas the latter contrasts execution with plan generation.

of this model. After this, we present some theoretical curves that illustrate the behavioral implications of the analysis. Finally, we discuss related work on sensing and execution, along with some prospects for future research.

A Model of Execution Cost

We would like some model of execution that lets us understand the tradeoff between the cost of sensing and the cost of errors. Of course, any model is necessarily an idealization of some actual situation, and from the many possible models, we must select one that is simultaneously plausible and analytically tractable. Thus, we will begin with a realistic scenario and introduce some simplifying assumptions that we hope retain the essential characteristics.

One common problem that involves physical agents is robot navigation. In some approaches, the agent retrieves or generates a plan for moving through an environment, then executes this plan in the physical world. Unfortunately, the plan does not always operate as desired. One source of divergence from the planned path comes from actuator error: a command to turn 35 degrees or to move 5.2 meters ahead may not execute exactly as specified. Another source of divergence comes from external forces: another agent may bump into the robot or an unexpected slope may alter its direction. Similar issues arise in the control of planes and boats, where malfunctioning effectors or unpredictable forces like wind can take the craft off the planned course.

In the standard reactive control regimen, the agent senses the environment on every time step, detects errors or divergences as soon as they occur, and takes action to put the agent back on the desired path.[2] The quality of the resulting plan execution takes into account the number of steps required to reach the goal or some similar measure. In a more general framework, execution quality also takes into account the cost c of sensing, which discourages a rational agent from unnecessary sensing and leads it to sample the environment only after every s time steps. This sensing cost may actually add to the execution time, or it may draw on other resources; here we care only that it somehow contributes to the overall cost of execution.

For such navigation contexts, we would like to determine the optimal sensing interval s for the execution of a given plan; in other words, we would like the sensing interval s that produces the most desirable compromise between purely reactive and purely automatic control. Naturally, the best value for s will be partly determined by the sensing cost c, which we assume includes the cost of matching the result against the expected situation. Two characteristics of the plan's interaction with the environment also come into play: the probability p that, on a given time step, an error will occur that takes the agent away from the desired path, and the distance d (or the number of time steps) to the goal if no such errors occur during execution.

There exist a variety of ways to ground these terms. We might model the situation geometrically, assuming that each error introduces some angular divergence from the current path, and that error recovery involves changing the angle of motion back toward the goal. We could then determine the expected distance added to the executed path as a function of the error-free distance, the probability and amount of error, and the sensing interval. Combining this with the sensing cost, we could derive the expected cost of execution. Note that this model assumes that errors have *persistence*; that is, the cost of an error increases with the time it goes undetected.

Although such a geometric model has attractions, a considerable amount of AI research (including much work on navigation) has relied instead on a state-space formalism. For example, one can divide any two-dimensional region into a grid, with the typical location having four neighbors (up, down, left, and right). One can represent these locations as states in a problem space, and the actions connecting them as operators for moving from one state to another. A particular path through this space corresponds to a navigation plan that the agent intends to carry out, and if the operators along the path are uncertain, then the agent may diverge from the planned path during execution. Such models of agent behavior are commonly used in work on reinforcement learning (e.g., Sutton, 1988; Kaelbling, 1993).

We could directly analyze the tradeoff between sensing and execution costs within this two-dimensional framework. However, Shrager, Hogg, and Huberman (1988) have noted that, for sparsely connected state spaces and accurate control decisions, one can approximate search through the space as a one-dimensional random walk, with each step taking one either closer to the goal (with probability $1 - p$) or further away (with probability p). We can adapt this idea to the execution of a state-space plan, giving the model that we will use in the remainder of the paper.

Figure 1 depicts this idealized situation. The agent, located at the current state, is moving toward the goal state, which is d steps to the right. With probability $1 - p$, the agent's next action will take it one step closer to the goal. However, with probability p, the action will instead introduce an error that not only takes it one step further from the goal, but reverses the agent's direction, so that future steps (unless again reversed) will lead away from the goal. (This persistent effect of errors distinguishes the model from a simple random walk, in which the probability of moving in a given direction is independent of past events.) The agent can correct this situation, but it can only detect the problem through sensing, which has cost c, leading it to sample the environment only after every s time steps.

Although this model is remarkably simple, we believe that it provides a viable approximation for much of the

[2] In some work (e.g., Simmons, 1990), the results of sensing instead determine the path followed in a conditional plan, but here we assume the aim is to follow a single desired path.

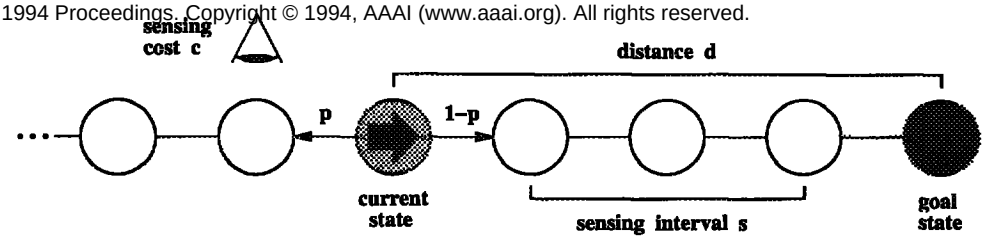

Figure 1. Modeling execution as a 'persistent' random walk in which the agent is distance d from the goal state, an execution error that inverts the direction of movement occurs with probability p, and sensing occurs every s time steps with cost c.

work on reactive behavior. The framework seems especially appropriate for navigation in a discretized two-dimensional space, as assumed by Sutton (1988) and Kaelbling (1993). Here each node in the state space has no more than four neighbors, provided one allows only moves to adjacent locations or changes in orientation. A large set of states produces a sparsely connected space; thus, unless errors are very likely, the situation can be approximated by the random walk in our model.

Analysis of the Execution Model

Within the above framework, we would like to determine $O(c, s, p, d)$, the expected overall cost of executing a plan given agent parameters for the sensing cost c and sensing frequency s, and given domain parameters for the error rate p and distance d. This quantity is

$$O(c, s, p, d) = \left(1 + \frac{c}{s}\right) E(s, p, d) \quad ,$$

where the first term in the product corresponds to the cost per execution step (assuming unitary cost for each movement) and the second gives the expected number of steps during execution. We can further decompose the latter term into

$$E(s, p, d) = s \cdot \frac{d}{G(s, p)}$$

where the ratio on the right expresses the expected number of times the agent will have to sense during the execution process. This is simply the initial distance to the goal, d, divided by the expected number of steps taken toward the goal during the intra-sensing interval s, denoted as $G(s, p)$. We can expand this last term to

$$G(s, p) = \sum_{j=0}^{s} R(j, s, p) \cdot Q(j, s) \quad ,$$

where $R(j, s, p)$ is the probability that exactly j errors will occur during the sensing interval s and $Q(j, s)$ is the expected number of steps taken toward the goal during that interval given that exactly j errors have occurred within it.

The first of these terms follows a simple binomial distribution, giving the expression

$$R(j, s, p) = \binom{s}{j} p^j (1-p)^{s-j} \quad .$$

The second term is more complex, but the basic idea is that the j errors can occur during the s time steps in any one of $\binom{s}{j}$ ways, with equal probability. We can determine the expected progress toward (or away from) the goal resulting from any one of the errors, then compute a weighted average. As a special case, if the agent makes zero errors, we have $Q(0, s) = s$, since each of the s steps takes it toward the goal. Another special case occurs when the agent makes exactly one error. We start with the expression

$$Q(1, s) = \frac{1}{s} \sum_{i=0}^{s-1} i - (s - i) \quad ,$$

since the error can occur before any one of the s time steps, and the resulting progress in each case is the number of time steps prior to the error (i up to $s - 1$), minus the negative progress occurring during the remaining time steps $(s - i)$. If we simplify this expression, we see that $Q(1, s) = -1$.

When the agent makes two or more errors, the expected progress between the two errors (subsequent to the first) are additive inverses and therefore tend to cancel each other. Thus, whenever the number of errors is even, they cancel completely and the overall expected progress is simply the expected progress prior to the first error. We can express this quantity as

$$Q(j, s) = \frac{1}{\binom{s}{j}} \sum_{i=0}^{s-j} i \cdot \binom{s - (i+1)}{j - 1} \quad ,$$

which averages, with respect to the total number of ways the j errors can occur, the progress prior to the first error weighted by the number of ways that subsequent errors can occur.

For the case in which the number of errors is odd, all but one of the subsequent errors cancel; thus, we must include the expected progress between the first and second errors. Since we assume the agent starts in the right direction, the first error results in negative progress and we must subtract this term from the previous expression. Thus, for this situation $Q(j, s)$ becomes

$$\frac{1}{\binom{s}{j}} \left[\sum_{i=0}^{s-j} i \binom{s - (i+1)}{j - 1} - \sum_{i=0}^{s-j} \sum_{k=1}^{s-j-i+1} k \binom{s - (i+1) - k}{j - 2} \right] .$$

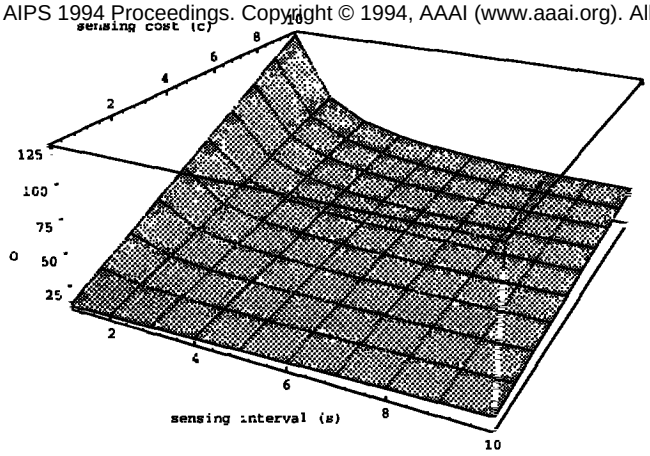

Figure 2. Effect of the sensing interval s and the sensing cost c on the overall execution cost O, when the error probability p is 0.06 and the distance d to the goal is ten.

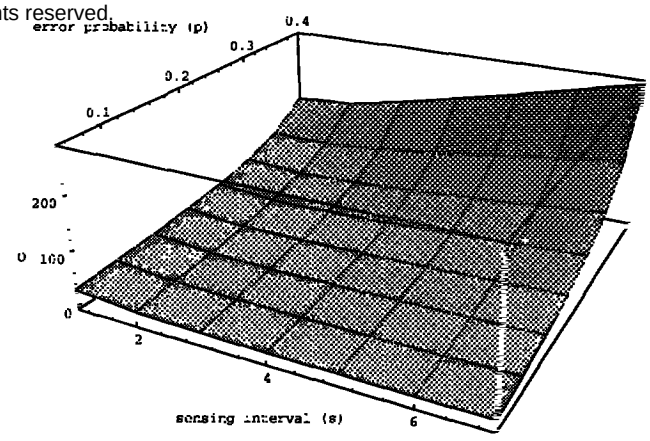

Figure 3. Effect of the sensing interval s and the error probability p on the overall execution cost O, when the sensing cost c is three and the distance d to the goal is seven.

Given the above expressions and settings for the four parameters, we can predict the expected overall cost $O(c, s, p, d)$ of executing a plan. Clearly, we would like to select a sensing interval s that minimizes this cost. For particular values of sensing cost c, error probability p, and distance d, we could in principle determine this optimal value by taking the partial derivative of O with respect to s, setting this expression to zero, and solving for s. We are currently working on this step of the analysis, but we have not yet obtained a closed-form solution.

Behavioral Implications of the Analysis

Although the equations in the previous section provide an analytic model of execution and sensing, their implications are not obvious. In order to better understand the interactions among the agent and environment parameters, and their effects on the overall execution cost, we carried out three 'thought experiments'.[3] In each case, we varied the sensing interval s and one other parameter, with the aim of illuminating the tradeoff between the costs of sensing and action. Our intent was not to show that one method is universally better than another, but to show exactly the opposite: that there are some situations in which reactive behavior outperforms automatic processing, but that there also exist cases in which automatic behavior is superior.

Figure 2 shows the joint effects of s and the sensing cost c on the overall cost $O(c, s, p, d)$, when the probability of error p is held constant at 0.06 and the distance d to the goal is ten. Here we see that, when the sensing cost c is zero, a purely reactive strategy produces the least expensive execution, with the cost monotonically increasing with higher values of s. This is the situation that most papers on reactive behavior assume. However, as the sensing cost increases, an intermediate strategy becomes optimal; for instance, when c is three, the best setting for s is six. Moreover, when c reaches ten, we find that the original effect is completely reversed, with an almost completely automatic strategy being most desirable, and with the cost monotonically increasing as s decreases.

A similar interaction emerges in Figure 3, which plots O as a function of the sensing interval s and the probability of error p when the sensing cost c is three and the distance d is seven. In this case, we see that, when the chances of an error during an execution step is very high (0.4), a purely reactive strategy proves least expensive. Yet when the planned course of action becomes more reliable (i.e., when p decreases), sensing on every time step ceases to be optimal and an intermediate strategy emerges as most desirable. For example, when $p = 0.2$, the optimal value for s is three. Finally, when the execution of a plan is sufficiently reliable (when $p = 0.05$), a completely automatic execution scheme becomes the strategy of choice.

However, the sensing interval does not interact with the final parameter, the distance d to the goal. Figure 4 maps the values of O against these two variables when the sensing cost is five and the error probability is 0.1. The graph reveals that the optimal setting for s is five, independent of the distance d, and similar results hold for other values of c and p. Thus, in our model an agent need not be concerned about its distance from the goal when selecting a sensing interval.

The behavioral implications of the model should now be clear. As the graphs show, there exist situations in which a purely reactive strategy leads to the lowest execution costs, but when one takes the sensing cost into

[3] We also ran actual experiments, using the same assumptions, as a check on our mathematical analysis. Since the results of these studies agreed closely with the predicted ones, we have not reported them here.

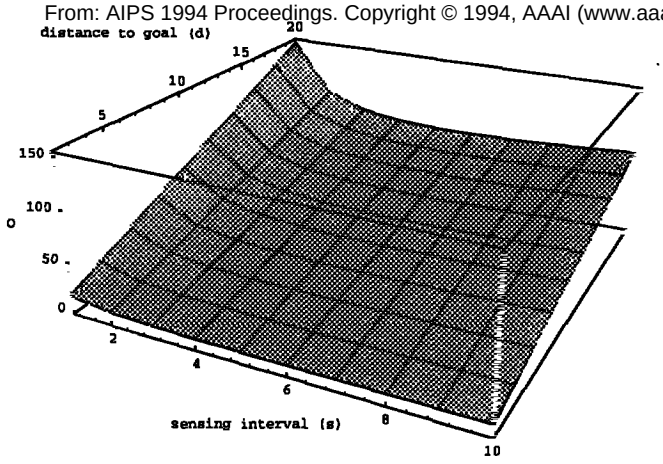

Figure 4. Effect of the sensing interval s and the distance d to the goal on the overall execution cost O, when the sensing cost c is five and there exists a 0.1 probability of error.

account, there also exist many cases in which a rational agent should sense only occasionally or even use a purely automatic strategy that involves no sensing at all. The literature's emphasis on reactive behavior has come from an oversimplified model that assumes the cost of sensing is negligible. This assumption is clearly violated for humans, and we believe that robust artificial agents will also be forced to take sensing costs into account.

Related and Future Work

As we noted earlier, the majority of research on execution has assumed a purely reactive strategy. Most examples of this approach rely on local decision knowledge (e.g., Sutton, 1988; Bresina et al., 1993), but even work that involves the generation or retrieval of entire plans (e.g., Howe & Cohen, 1991; Hammond et al., 1988) has often assumed that the agent senses the environment on every time step, giving it the ability to react immediately when behavior diverges from the desired path. Research in this tradition almost invariably ignores the cost of sensing.

A few researchers have incorporated decisions about sensing into the process of plan generation, viewing these as a form of action (Simmons, 1990; Kresbach et al., 1991; Schoppers, 1991). However, this work has focused on the construction of conditional plans that are guaranteed to achieve the goal, rather than explicitly taking into account the sensing cost. Tan's (1991) work on the induction of sensing strategies does address the cost of sensing, but here the goal is to determine the most efficient sequence of sensing actions, not the tradeoff between sensing costs and error costs. Chrisman and Simmons (1991) assign clear costs to both sensing and action, but they emphasize selection of the best sensing decision on each time step, rather than the decision of whether to sense at all.

Our work does have much in common with the approach taken by Iba (1991), who studied the continuum from closed-loop to open-loop behavior in the execution of retrieved motor schemas. However, Iba's aim was to minimize divergence of the execution trace from the desired behavior, and he ignored the cost of sensing. Yang (1992) has also dealt with the continuum in his work on macro-operators, showing that some environments support long sequences of actions without sensing, whereas others do not. But again, his work assumed sensing should be minimized but did not model its cost.

The research most closely related to our own comes from Hansen and Cohen (1993), who have reformulated the standard Markov decision framework to incorporate the cost of sensing. Their analysis shows that, in some domains, selective sensing can produce less expensive execution strategies than purely reactive schemes, and that one can use dynamic programming to determine the optimal monitoring interval. Moreover, they show that the desired interval decreases as the agent approaches the final state, an intuitively plausible result that our analysis does not provide. However, their framework differs from our own in assuming a cyclical domain, so that we cannot directly carry over their results.

Although our model moves beyond most others in its explicit treatment of sensing costs, considerable work remains to be done. We should connect our theoretical ideas to more traditional decision-theoretic analyses, and we should test the framework's ability to model the behavior of physical robots using actual navigation plans and using reasonable estimates of sensing cost. We have also made the implausible assumption that one sensing interval is optimal for an entire plan, and we should extend the model to handle plans with a number of sequential components, each with their own values for the p, d, c, and s parameters.

Moreover, we should generalize the framework to handle situations that involve many different sensors, each with its own characteristics. On every time step, the decision about whether to invoke a given sensor should take into account not only that sensor's expense, but also the information it is expected to provide, which is directly related to the probability that the sensor will reveal an execution error. In this framework, the optimal strategy would sample inexpensive, informative sensors frequently but sample costly, uninformative ones seldom or not at all, giving behavior that is reactive with respect to some sensors and automatic with respect to others.

Finally, we would like the agent to determine the optimal sensing interval for a given situation by taking the derivative of $O(c, s, p, d)$ with respect to s. However, for this it must know the values for the sensing cost c, the error probability p, and distance d, and we cannot expect an oracle to provide this information on request. Fortunately, the agent should be able to estimate these parameter settings from execution of the given plan, simply by collecting statistics over a number of trials. Such a learning agent would begin with a purely reactive strat-

egy but, if the parameter estimates recommend it, would move toward a more automatic execution scheme as it gains experience.

An extended model of this sort makes clear predictions about human learning. In particular, it indicates that, in the early stages of skill acquisition, people will operate in reactive mode to the extent their attentional resources allow, letting them estimate the domain characteristics. In uncertain domains where sensing is expensive, they will continue to execute skills in this manner even after long practice. But in domains where sensing costs less and errors are unlikely, they will gradually move to an automatic execution strategy. Unfortunately, most existing studies of human motor behavior have forced subjects into either closed-loop processing (e.g., Adams, 1971) or open-loop mode (e.g., Schmidt, 1982), and they have not systematically varied the domain characteristics with an eye toward the *transition* from the former to the latter. A clear test of our model would include different types of motor tasks and would give subjects the ability to range from automatic processing to purely reactive behavior.

We began this paper with an intuition that ran counter to the prevailing wisdom that purely reactive strategies are always preferable. In order to formalize our ideas, we presented an idealized model of plan execution, followed by a tractable analysis that predicts the overall cost of execution in terms of the sensing interval s, the sensing cost c, the probability of error p, and the distance d to the goal. Our analysis, which we argued is applicable to much of the work in the literature, revealed that it is not always desirable to sense on every time step, and that the parameters c and p (but not d) influence the optimal setting for s. In future work, we plan to elaborate on the start we have made here, and we hope that other researchers will join us by taking a more balanced view of the spectrum from reactive to automatic execution.

Acknowledgements

This research was supported in part by Grant F49620-94-1-0118 from the Computer Science Division of the Air Force Office of Scientific Research.

References

Adams, J. A. (1971). A closed-loop theory of motor learning. *Journal of Motor Behavior*, *3*, 111–149.

Bresina, J., Drummond, M., & Kedar, S. (1993). Reactive, integrated systems pose new problems for machine learning. In S. Minton (Ed.), *Machine learning methods for planning*. San Mateo, CA: Morgan Kaufmann.

Chrisman, L., & Simmons, R. (1991). Sensible planning: Focusing perceptual attention. *Proceeding of the Ninth National Conference on Artificial Intelligence* (pp. 756–761). Anaheim, CA: AAAI Press.

Greenstette, J. J., Ramsey, C. L., & Schultz, A. C. (1990). Learning sequential decision rules using simulation models and competition. *Machine Learning*, *5*, 355–381.

Hammond, K. J., Converse, T., & Marks, M. (1988). Learning from opportunities: Storing and reusing execution-time optimization. *Proceeding of the Seventh National Conference on Artificial Intelligence* (pp. 536–540). St. Paul, MN: AAAI Press.

Hansen, E. A., & Cohen, P. R. (1993). Learning monitoring strategies to compensate for model uncertainty. *Working Notes of the AAAI-93 Workshop on Learning Action Models* (pp. 33–35). Washington, D.C.: AAAI Press.

Howe, A. E., & Cohen, P. R. (1991). Failure recovery: A model and experiments. *Proceeding of the Ninth National Conference on Artificial Intelligence* (pp. 801–808). Anaheim, CA: AAAI Press.

Iba, W. (1991). *Acquisition and improvement of human motor skills: Learning through observation and practice*. Doctoral dissertation, Department of Information & Computer Science, University of California, Irvine.

Kaelbling, L. P. (1993). Hierarchical learning in stochastic domains: Preliminary results. *Proceedings of the Tenth International Conference on Machine Learning* (pp. 167–173). Amherst, MA.

Kresbach, K., Olawsky, D., & Gini, M. (1992). An empirical study of sensing and defaulting in planning. *Proceedings of First International Conference on Artificial Intelligence Planning Systems* (pp. 136–144). College Park, MD: Morgan Kaufmann.

Schmidt, R. A. (1982). More on motor programs. In J. A. S. Kelso (Ed.), *Human motor behavior: An introduction*. Hillsdale, NJ: Lawrence Erlbaum.

Shrager, J., Hogg, T., & Huberman, B. A. (1988). A graph-dynamic model of the power law of practice and the problem-solving fan effect. *Science*, *242*, 414–416.

Schoppers, M. (1992). Building plans to monitor and exploit open-loop and closed-loop dynamics. *Proceedings of First International Conference on Artificial Intelligence Planning Systems* (pp. 204–213). College Park, MD: Morgan Kaufmann.

Simmons, R. (1990). Robust behavior with limited resources. *Proceedings of the AAAI Spring Symposium on Planning in Uncertain, Unpredictable, or Changing Environments*. Stanford, CA.

Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*, 9–44.

Tan, M. (1991). *Cost-sensitive robot learning*. Doctoral dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Yang, H. (1992). *Learning and using macro-operators for AI planning*. Doctoral dissertation, Department of Computer Science, Vanderbilt University, Nashville.