Intrinsically Motivated Reinforcement Learning: A Promising Framework For Developmental Robot Learning

Andrew Stout, George D. Konidaris and **Andrew G. Barto**
Department of Computer Science
University of Massachusetts – Amherst
{stout, gdk, barto}@cs.umass.edu

Abstract

One of the primary challenges of developmental robotics is the question of how to learn and represent increasingly complex behavior in a self-motivated, open-ended way. Barto, Singh, and Chentanez (Barto, Singh, & Chentanez 2004; Singh, Barto, & Chentanez 2004) have recently presented an algorithm for *intrinsically motivated reinforcement learning* that strives to achieve broad competence in an environment in a task-nonspecific manner by incorporating internal reward to build a hierarchical collection of skills. This paper suggests that with its emphasis on task-general, self-motivated, and hierarchical learning, intrinsically motivated reinforcement learning is an obvious choice for organizing behavior in developmental robotics. We present additional preliminary results from a gridworld abstraction of a robot environment and advocate a layered learning architecture for applying the algorithm on a physically embodied system.

Introduction

One of the primary challenges of developmental robotics is the question of how to learn and represent increasingly complex behavior in a self-motivated, open-ended way. We argue in this paper that, equipped with recent advances pertaining to temporal abstraction and hierarchy, reinforcement learning (RL) provides a promising framework for learning and representing hierarchical skills. Indeed, we are presently engaged in ongoing research in *intrinsically motivated reinforcement learning*, an approach introduced by Barto, Singh, & Chentanez (2004) wherein the primary reinforcement signal is generated within the agent, allowing it develop broad competence in an environment in an open-ended fashion.

However, when applied naïvely to robotic tasks, RL methods often struggle with the continuous and high dimensional state and action spaces and insufficient learning experience. In some cases a simpler and more elegant solution is to *layer* learning, so that RL takes place not over the raw sensor space, for instance, but rather over a learned economical representation of that space which facilitates RL.

Thus, in this paper we:

- advocate a layered approach to learning architectures for developmental robotics;
- advocate intrinsically motivated RL (Barto, Singh, & Chentanez 2004; Singh, Barto, & Chentanez 2004) as an especially promising approach to developmental learning; and
- present preliminary results applying intrinsically motivated RL to a gridworld abstraction of a robot domain.

In the next section we briefly review RL and layered learning. Next we review a recent success integrating RL and behavior-based robotics, using a distributed topological map as an intermediary layer. We then review an algorithm for intrinsically motivated RL, and present a simple gridworld experiment illustrating its potential. Finally, we discuss the benefits of this approach and advocate a layered architecture for bringing this approach to bear on embodied systems, as well as other directions for future work.

Background

Reinforcement Learning

Reinforcement learning (Sutton & Barto 1998) aims to solve the problem of a behaving agent learning to approximate an optimal behavioral policy through interaction with an environment. This generally takes the form of learning to maximize a numerical reward signal over time in a given environment. This reward signal is the only learning feedback obtained from the environment, and thus RL falls somewhere between unsupervised learning (where no signal is given at all) and supervised learning (where a signal indicating the correct action is given), which makes it well suited to developmental robotics.

Most RL algorithms adapt dynamic programming methods to focus on the most relevant parts of the value space—behavioral trajectories. State or state-action values are estimated from experience and "backed up" to compute approximately optimal policies of actions—those that maximize expected long-term reward. The Markov decision process is a popular formalism in RL:

at each stage the agent, in one of the set of possible states, chooses from the set of available actions an action, which presumably (stochastically) influences the agent's subsequent transition to the next state, receiving a reward in the process. The *policy* maps from states and actions to probabilities of executing a given action in a given state.

Options *Options* (Precup 2000; Sutton, Precup, & Singh 1999) are a principled framework for temporal abstraction in RL. Briefly, an option is roughly analogous to a subroutine: it has an *initiation set* of states in which it can be invoked, an internal policy mapping states and actions to probabilities of execution, and a termination condition mapping states to the probability of the option terminating in that state. When an option is invoked it follows its internal policy until termination; this allows an option to be considered a temporally extended action, freeing the agent from needing to choose an action at each step. One option's policy may call another option, creating an elegant mechanism for behavioral hierarchy.

The options framework has a solid theoretical foundation, extending Markov decision processes to semi-Markov decision processes (Barto & Mahadevan 2003), and two components of the options framework are particularly important to the algorithm presented below:

Option Models are probabilistic descriptions of the effects of executing an option. They can be (approximately) learned from experience, and allow stochastic planning to be extended from primitive (one-step) actions to higher levels of abstraction.

Intra-option Learning Methods allow the internal policies of many options to be updated simultaneously, regardless of which option is actually executing.

In most of the work using options, the options must be hand-designed by the engineer in advance. It is clear that dynamically creating and learning options is a desirable ability, and several researchers have recently proposed methods for doing so, e.g. (Şimşek & Barto 2004), (McGovern 2002). This work falls into that category, and is unique in that rather than creating options tailored to a specific task, our algorithm creates options based on intrinsic motivation.

Layered Learning

Because it has so many attractive properties, several researchers have added RL capabilities to their robots. However, applying RL directly over the robot's (very large) sensor space often leads to convergence problems due to violations of the Markov assumption and the sheer enormity of the space. One solution to this is the use of *layered learning* to provide a suitably abstract problem space that RL can solve efficiently.

It is now widely accepted that a layered and incremental approach to designing robot control systems (Brooks 1986) works well in practice, and further that the interaction of layered, parallel control elements can produce interestingly complex adaptive behavior (Pfeifer & Scheier 1999).

By the same token, we argue that learning elements should be layered in a robot's control system in the same way that more static control elements are. There are several examples of layering RL on top of a behavioral basis, e.g. (Huber & Grupen 1997). The natural extension is to add additional learning layers in between. Layered learning (Stone 2000; Utgoff & Stracuzzi 2002) means that we can use lower-level learning elements to learn useful structures, discretizations, and behavior that can help make higher-level learning feasible, and allows for the interaction of several learning elements to generate complex adaptive behavior.

One example of the approach described above is the layered, distributed and asynchronous reinforcement learning model developed by Konidaris & Hayes (2005). RL was layered on top of a learned topological map, which itself was layered on top of a reactive behavioral substrate on a robot to perform puck foraging in an artificial arena. The use of a reactive behavioral substrate created conditions under which the topological map could be easily learned, while the topological map served to keep the state space small and task relevant. This, coupled with the use of asynchronous and parallelizable updates that took advantage of the fact that computation is very much faster than action in embodied domains, allowed the robot's RL element to converge in real time, between decisions.

This worked well, but the learning dynamics it displayed were those of traditional RL: a task-specific, externally imposed reward function was used to achieve a certain behavior, after which no additional learning took place. The elegance of a layered architecture is that these dynamics can be addressed at the level of the RL layer, taking the lower behavioral and topological levels for granted[1]. We thus turn our attention now to a RL system designed to display the task-general, open-ended learning dynamics emphasized in the developmental robotics approach.

Intrinsically Motivated Reinforcement Learning

Barto, Singh, & Chentanez (2004) introduce a model of *intrinsically motivated* reinforcement learning employing the options framework. The model is grounded in an elaboration of the traditional conception of RL, wherein the environment is "factored" into an external environment and an environment internal to the agent. It is this internal environment which provides the reward signal to the RL system. Note that this elaboration still allows for rewards from the external environment: these are simply "transduced" by the internal environment.

[1] In principle, at least. We recognize that in practice things are rarely quite that simple.

In the traditional approach to RL the reward function is tailored specifically to the task at hand (navigating a maze, or winning at backgammon, for example), and crafting this reward function can require significant ingenuity. The notion of intrinsically motivated RL is that the critic in the internal environment includes the agent's motivational system—and that this motivational system should be sophisticated and general, and should not need to be redesigned for each specific task the agent undertakes. Driven by this task-general intrinsic motivation, the agent builds up a hierarchical collection of skills—in effect achieving *broad competence* in its environment. These skills can then be applied to any specific task the agent finds itself called upon to learn.

There are many possibilities for the source of intrinsic motivation, including surprise, novelty (Huang & Weng 2002), or "learning progress" (Kaplan & Oudeyer 2004). Thus far the neuroscience of dopamine neurons (Horvitz, Stewart, & Jacobs 1997) has been the most direct inspiration for the implemented model of intrinsic motivation (Barto, Singh, & Chentanez 2004; Singh, Barto, & Chentanez 2004), and our experiments here also follow that path, although we plan to explore other sources of intrinsic motivation in future work.

With its emphasis on task-general, self-motivated, and hierarchical learning, intrinsically motivated RL is an obvious choice for developmental robotics. Experience with intrinsic motivation in hierarchical RL is still very preliminary, and the only experimental results to date are on an abstract gridworld. This paper presents an application of this approach to a domain intended to be a stepping stone from the abstract environment presented in (Barto, Singh, & Chentanez 2004) to a real robotic domain. While still discrete and deterministic, the domain is more "robot-like" than the prior work on intrinsically motivated RL, and we believe that given the appropriate support from other layers in a learning architecture (such as a learned topological map), the approach will be adaptable to real robotic applications.

We next describe the specifics of the intrinsically motivated RL algorithm used, which is based very closely on the algorithm presented by Singh, Barto, & Chentanez (2004), and then describe an experiment illustrating its behavior.

The Algorithm

The algorithm for intrinsically motivated RL departs from traditional RL mostly in its use of intrinsic reward to learn a collection of useful skills. In many other respects the algorithm is a combination of established algorithms for hierarchical RL. The description here, although organized differently, is similar that in (Singh, Barto, & Chentanez 2004), where further details can be found.

Saliency Present implementations depend on the hardwired salience of certain stimuli or events in the agent's environment (although it bears repeating that the larger idea of intrinsically motivated RL does not depend on this particular model of intrinsic motivation). For example, in the experiments described below, changes in light or sound intensity are considered salient. We consider such notions of saliency to be roughly analogous to the saliency of certain stimuli—such as the smell of food or movement of a potential threat—that are hardwired by evolution into the nervous systems of animals in nature. These stimuli are by necessity specific to the animal's ecological niche, but are general with respect to specific settings within that niche and with respect to specific tasks or skills the animal undertakes.

Skills The first time the agent experiences a given salient event it creates and initializes an option to bring about that event. An event option's initiation set is initialized to include the state just prior to the salient event, and the termination probability for the state in which the event occurred is initialized to one. In addition, an option model is initialized for the event option, estimating the probability of the option terminating in a given state with a given cumulative reward when executed from a given state. As the agent gains experience the option's initiation set grows to include states that lead to states in the current initiation set, and whenever the agent experiences a salient event in a novel state the termination probability of the option for that event is set to one. The algorithm updates the option policies and option models of all options simultaneously using intra-option learning. Once initialized, an option is available as an action to other options as well as to the behavioral (top-level) policy, which provides an elegant and natural way of building a hierarchy of skills.

Intrinsic Reward The implementation of intrinsic reward associated with these salient stimuli is inspired by the response of dopamine neurons to novelty (Horvitz, Stewart, & Jacobs 1997). The intrinsic reward for the occurrence of a salient event is proportional to the prediction error of that event in the learned option model for that event. Thus when an event first occurs, or occurs in a previously unexperienced context, its intrinsic reward will initially be high (it will be 'surprising' or 'interesting'). While the event option policies are updated only with respect to the extrinsic reward signal (if any) and a (hardwired) reward for successfully terminating an option, the behavioral policy incorporates the intrinsic reward in its update. Thus 'surprise' drives the agent to try to bring the event about. However, as the agent repeatedly does so it becomes better at both bringing about and predicting the occurrence of the event. As the event becomes more predictable it becomes less rewarding (the agent gets 'bored'). The algorithm also naturally handles extrinsic reward, but importantly it does not depend on it.

Behavior The agent behaves with an ϵ-greedy policy with respect to its behavioral action-value function. The behavioral action-value function is learned through a combination of Q-learning and SMDP planning, and maps states and actions (initially only the primitive actions; options are included as they become available) to their expected long-term reward.

An Experiment

We now present preliminary experimental results demonstrating the performance of the algorithm. In this work we assume the existence of lower layers of learning sufficient for supporting a high level representation of the state and action spaces. This assumption allows us to test our ideas in a simple gridworld, where we can focus on the high-level behavior we wish to demonstrate. While we believe the layered approach successfully demonstrated by Konidaris & Hayes (2005) and discussed above gives us cause to believe that this temporary abstraction is justified, we also recognize the danger of these assumptions. Future work integrating intrinsically motivated RL into a layered learning architecture on a physically embodied robot will inevitably present challenges not addressed in the present work, but we do not believe this detracts from the promise of intrinsically motivated RL as a layer driving open-ended learning on developing robots.

Experimental Setup

The gridworld used in this work (figure 1) is an abstraction of a 'playworld' environment which has actually been built on an Aibo-scale by some of our colleagues for future experimentation with these ideas on Aibo robots. The world consists of two rooms with a door between them. There are push-panels on the walls which, when pushed, turn the lights on or off or open or close the door. The second room contains a charger.

The robot perceives its location and orientation (we assume these are provided by the topological map, for instance), a list of visible objects, light intensity, and various sounds. It can move forward, rotate clockwise or counterclockwise, approach any object it can see, push a push-panel, or charge itself. Changes in light and sound are hardwired as salient events.

The robot starts at a random location in the left room, in the dark. It can see the glowing push-panel, but it can not see the other push-panel or the door until it has turned on the light by pushing the glowing push-panel. Pushing the other push-panel will open the door, causing an alarm to ring. When it is facing the charger it may charge itself, which causes a bell to ring and earns an extrinsic reward. A small extrinsic punishment is given at each time step as a 'cost of living'. Every 250 steps the robot is 'kidnapped', and the experiment is reset to initial conditions with the robot placed in a random location in the left room.

The world was designed to include objects engaged with varying levels of difficulty. Engaging the light switch is easy, but engaging the charger requires a number of intermediate steps. Clearly, if the robot has already learned skills to turn the light on and open the door, these will be of use in learning to engage the charger.

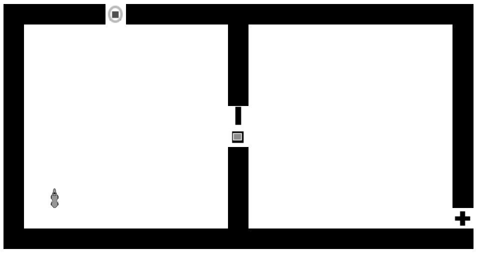

Figure 1: The gridworld environment.

Results

Barto, Singh, & Chentanez (2004) present results from applying the intrinsically motivated RL algorithm in a smaller and more abstract gridworld. They show that as expected (and, indeed, designed), the agent gains competence by first learning to achieve 'easy' salient stimuli, and then building on these skills to achieve more 'difficult' stimuli. When the agent first encounters a salient stimulus it receives high intrinsic reward, but as it learns to predict that event the level of intrinsic reward drops, until it encounters that event again unexpectedly.

This work is ongoing and the results we report here are similar but still very preliminary in nature. Figure 2 shows a record of salient event occurrences over the course of the experiment. At the beginning the robot quickly discovers and learns to predict turning the light on and off. Later it learns to open and close the door, and soon after learns to charge itself (ringing a bell), a behavior which persists because it is extrinsically rewarding.

Figure 3 plots the number of steps (from the initiation of a testing period) to the achievement of each of the salient events. An initial period of exploration is visible in the first few thousand steps (due to optimistic initial values). At about 2500 steps the robot starts to consistently achieve the light on and off events, and at roughly 12500 steps it discovers how to open the door and ring the bell. A second period of exploration ensues, and at the end of the experiment we can see that the robot has learned to consistently turn on the light, open the door, and ring the bell efficiently. Note that it also learns that turning the light *off* and *closing* the door are not worth the effort.

Discussion and Future Work

While we believe intrinsically motivated reinforcement learning has many features that make it an appealing approach to developmental robotics, it is clear that much remains to be done to demonstrate the viability of the approach. A first step is to more thoroughly demonstrate the algorithm's performance on the gridworld presented above. Once that has been accom-

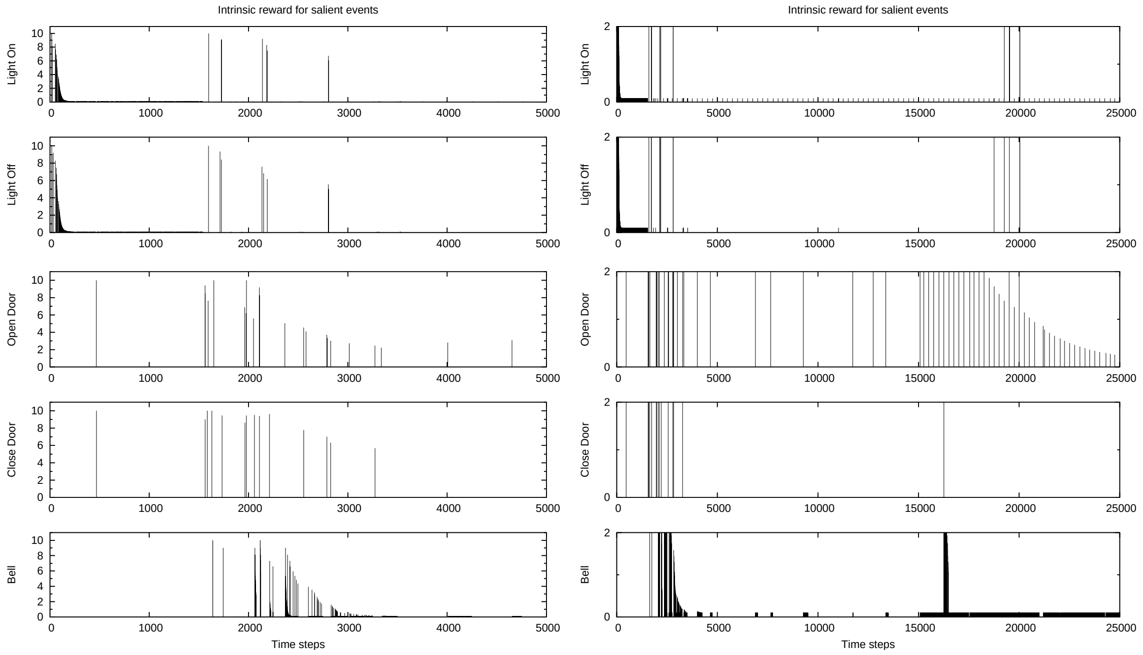

Figure 2: Records of intrinsic rewards for the occurrence of salient events. The left plot shows the first 5000 steps of the experiment, illustrating the exponential drop in intrinsic reward as salient events become predictable. The right plot shows the full 25000 steps of the experiment in detail, with small intrinsic rewards for predicted salient events indicating sustained charging behavior. The regular occurrence of the Light On event is due to the periodic 'reset' of the experiment.

plished, we hope to adapt the algorithm to one suitable for application on a real robot. To do this we propose adopting the layered approach discussed above in order to provide the intrinsically motivated RL layer with a tractable problem space.

One of the challenges of applying RL in the real world is the issue of efficiency. No consideration of efficiency has been made so far with respect to intrinsically motivated RL, and there are several obvious improvements that could be made (e.g. eligibility traces). As discussed above, the mismatch between computation time for a RL update and the time it generally takes a robot to take an action changes the efficiency dynamic dramatically, as it may be possible to perform dynamic programming to the point of convergence between decisions (Konidaris & Hayes 2005).

Other directions for future work are more specific to the particular challenges of developmental robotics. One shortcoming of the current model of intrinsic motivation is that intrinsic reward is based on a failure to predict a (salient) event. However, as Kaplan & Oudeyer (2003) have demonstrated, this can lead to undesirable behavior in environments involving areas with dynamics that are difficult or impossible to predict. As they and others have argued, a better approach is to note that learning is most fruitful in a "zone of proximal development"—areas that are learnable, neither too predictable (already learned) nor too unpredictable (impossible to learn). As mentioned briefly above, this is just one of several sources of intrinsic motivation we hope to explore.

It is also worth considering what level of primitive actions are to be engineered and considered 'innate'. The present work assumes a relatively high-level behavioral basis, while much work in the developmental robotics community concentrates on developing lower-level sensorimotor coordination. While it is clear that much is built-in in nature (e.g. some mammals can walk within hours after being born), it is also clear that learning takes place to refine sensorimotor coordination (e.g. Berthier, Rosenstein, & Barto 2005), and moving learning 'down the hierarchy' removes engineer bias (Blank, Kumar, & Meeden 2002) and leaves more room for online adaptation. To what extent intrinsic motivation is important for such low-level learning is an open and important question we hope to address in the future.

Conclusion

This paper has discussed an algorithm for intrinsically motivated reinforcement learning and argued that it has many characteristics that make it appealing for developmental robotics. Intrinsic motivation drives the system to learn and gain broad competence in a task-general manner, and the hierarchical RL framework provides an elegant means of building on previous learning in an open-ended fashion. However, while intrinsically motivated RL is well suited to situated learning, as currently formulated it is not well suited to learning directly in

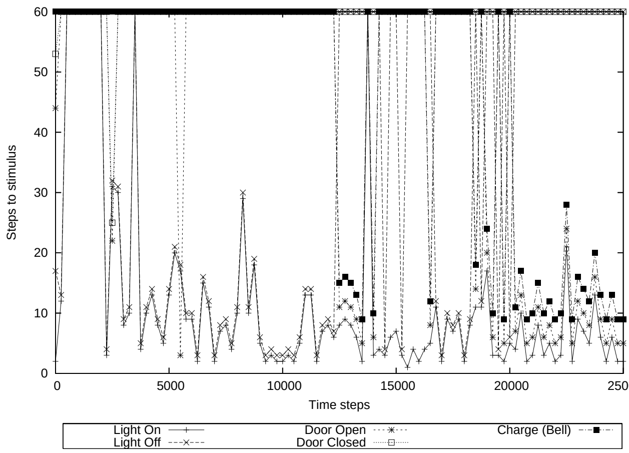

Figure 3: Steps to achievement of each salient event, tested every 250 steps. Achievement either occurred within 60 steps or not within 150, hence the graphs' upper limit. (The vertical-oriented lines show transitions between periods of achievement, lower, and non-achievement, higher.) The progressive learning of increasingly complex skills can be seen.

the high-dimensional, continuous problem space that physical embodiment involves. We thus advocated a layered approach to learning architectures for developmental robotics, wherein the lower layers of learning provide a tractable space for the upper layers. Much remains to be done, but we believe these ingredients hold great promise for developmental robot learning.

Acknowledgments

This work was supported in part by a subcontract from Rutgers University, Computer Science Department, under Award number HR0011-04-1-0050 from DARPA. A. Stout also gratefully acknowledges a Swarthmore College Lockwood Fellowship.

References

Barto, A. G., and Mahadevan, S. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems.*

Barto, A. G.; Singh, S.; and Chentanez, N. 2004. Intrinsically motivated learning of hierarchical collections of skills. In *Proc. of Inter. Conf. on Developmental Learning (ICDL).*

Berthier, N.; Rosenstein, M.; and Barto, A. 2005. Approximate optimal control as a model for motor learning. *Psychological Review* 112.

Blank, D.; Kumar, D.; and Meeden, L. 2002. Bringing up robot: Fundamental mechanisms for creating a self-motivating, self-organizing architecture. In *Proc. of the Growing Up Artifacts that Live workshop at Simulation of Adaptive Behavior 2002.*

Brooks, R. 1986. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2(1).

Şimşek, O., and Barto, A. G. 2004. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proc. of the 21st Inter. Conf. on Machine Learning.*

Horvitz, J.; Stewart, T.; and Jacobs, B. 1997. Burst activity of ventral tegmental dopamine neurons is elicited by sensory stimuli in the awake cat. *Brain Research* 759:251–258.

Huang, X., and Weng, J. 2002. Novelty and reinforcement learning in the value system of developmental robots. In *Proc. 2nd Inter. Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems.*

Huber, M., and Grupen, R. A. 1997. Learning to coordinate controllers – reinforcement learning on a control basis. In *Proc. of the 15th Int. Joint Conf. on Artificial Intelligence.*

Kaplan, F., and Oudeyer, P.-Y. 2003. Motivational principles for visual know-how development. In Prince, C.; Berthouze, L.; Kozima, H.; Bullock, D.; Stojanov, G.; and Balkenius, C., eds., *Proc. of the 3rd Inter. workshop on Epigenetic Robotics : Modeling cognitive development in robotic systems*, 73–80.

Kaplan, F., and Oudeyer, P.-Y. 2004. Maximizing learning progress: an internal reward system for development. In Lida, F.; Pfeifer, R.; Steels, L.; and Kuniyoshi, Y., eds., *Embodied Artificial Intelligence*, 259–270. Springer-Verlag.

Konidaris, G., and Hayes, G. 2005. An Architecture for Behavior-Based Reinforcement Learning. To appear, *Adaptive Behavior.*

McGovern, A. 2002. *Autonomous Discovery of Temporal Abstractions From Interactions With An Environment.* Ph.D. Dissertation, U. of Massachusetts, Amherst.

Pfeifer, R., and Scheier, C. 1999. *Understanding Intelligence.* MIT Press.

Precup, D. 2000. *Temporal Abstraction in Reinforcement Learning.* Ph.D. Dissertation, U. of Massachusetts, Amherst.

Singh, S.; Barto, A. G.; and Chentanez, N. 2004. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing 18.*

Stone, P. 2000. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer.* MIT Press.

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

Sutton, R.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1-2):181–211.

Utgoff, P. E., and Stracuzzi, D. J. 2002. Many layered learning. *Neural Computation* 14:2497–2539.