

# iCORE Research Grant Renewal Proposal

## Reinforcement Learning and Artificial Intelligence

chair: Richard S. Sutton

### 1 Executive summary

It is proposed to renew iCORE funding for the Reinforcement Learning and Artificial Intelligence (RLAI) laboratory at the University of Alberta created in connection with a chair establishment grant to Richard Sutton.

The RLAI research program pursues an approach to artificial intelligence and engineering problems in which they are formulated as large optimal control problems and approximately solved using reinforcement learning methods. Reinforcement learning is a new body of theory and techniques for optimal control that has been developed in the last twenty years primarily within the machine learning and operations research communities, and which have separately become important in psychology and neuroscience. Reinforcement learning researchers have developed novel methods to approximate solutions to optimal control problems that are too large or too ill-defined for classical solution methods such as dynamic programming. For example, reinforcement learning methods have obtained the best known solutions in such diverse automation applications as helicopter flying, elevator scheduling, playing backgammon, and resource-constrained scheduling. The objectives of the RLAI research program are to create new methods for reinforcement learning that remove some of the limitations on its widespread application and to develop reinforcement learning as a model of intelligence that could approach human abilities.

The RLAI laboratory was founded in 2003 with support from iCORE, AICML, NSERC, and the University of Alberta, with the iCORE contribution being about 45% of the total. Three full-time faculty were newly hired into the department of computing science at that time and became the laboratory's first three principal investigators: Richard Sutton, Dale Schuurmans, and Michael Bowling. A fourth principal investigator, Csaba Szepesvari, was recruited into the department as an associate professor to join the project in its third year. Otherwise the RLAI laboratory has grown steadily over its first four years; currently it includes approximately 30 graduate students, five postdocs, eight software developers, and two part-time administrative assistants.

A crude indication of the quality of these personnel comes from their recognition by other funding agencies. Dale Schuurmans has been awarded a Tier II Canada Research Chair, and both Michael Bowling and Csaba Szepesvari have been awarded Alberta Ingenuity Fund New Faculty Grants since the project started. Twelve of the project's current graduate students have been awarded and are currently supported by major scholarships.

The primary outputs of the project are highly qualified personnel and publications in the international peer-reviewed scientific literature. The RLAI laboratory published 77 papers in archival venues over the last two annual reporting periods. Six PhD and seven Masters students supervised by RLAI principal investigators have graduated. Five postdoctoral fellows have gone on to research jobs in academia or in major industrial research laboratories.

## 2 Research area

This section contains an overview of the proposed research of the RLAI project, dividing it into three main interrelated areas. The first is extensions of conventional reinforcement learning algorithms; there are many open problems in reinforcement learning, and the RLAI project seeks to solve them as opportunities arise. The second area is the extension of reinforcement learning ideas to address the more ambitious goals of artificial intelligence. We feel there is a natural transition from the more advanced reinforcement learning methods to mechanisms for knowledge representation, search, and human-level reasoning. A major goal for the project is to explore, implement, and illustrate these relationships. The third main area of RLAI research is a focus on applications—on designing algorithms and software particularly suited for applications, and on several specific application areas in which we are working or planning to work.

### 2.1 Reinforcement learning

The RLAI project is focused on basic research at the intersection of artificial intelligence and a relatively new field developed over the last twenty years known as reinforcement learning [1, 2]. Reinforcement learning arose out of ideas from the psychology of animal learning and is usually described as a class of methods for approximating dynamic programming. However, these historical perspectives do not apply to some important modern reinforcement learning methods (e.g., policy-gradient methods without value functions) and may obscure reinforcement learning’s novel and distinctive characteristics. It may be more accurate and revealing to define reinforcement learning as the study of *sampling-based* methods for solving *general* prediction and decision problems.

The generality of the problems addressed in reinforcement learning has three important aspects. First, they are optimal control problems, problems in which the objective is to maximize a quantity, rather than to track a target signal or to drive it to a desired state. Second, they are multi-stage problems, problems in which there is sequence of time steps at which observations or decisions are made, and the decisions made at one time step affect the decision choices available at a later steps. Third, the problems involve general nonlinear dynamics. These three kinds of generality are familiar from classical work on Markov decision processes (MDPs) and sequential decision making, on which many of reinforcement learning’s methods and much of its theory is based. Reinforcement learning methods differ from classical methods primarily in that they are sampling-based.

*Sampling-based methods* are computational methods that operate on and require only samples of the behavior of the system to be predicted or controlled. Classical Monte Carlo methods, such as the Metropolis algorithm [3], are examples of sampling methods. Reinforcement learning extends sampling methods to the general context of optimal sequential decision making. One advantage of sampling-based methods is that they tend to scale better to large state and action spaces than other methods. Another is that they can be applied when the system is not known explicitly (e.g., as a set of differential or difference equations, or as an MDP), but only in the form of a simulation, as is often the case in large applications. A major advantage of sampling methods is that in some cases they can be applied when the system is completely unknown—neither a model nor a simulation is available—by using samples obtained directly by interacting with the system. In this case, the sampling-based method gains information about the system while interacting with it; this is the rationale for the word “learning” in the name reinforcement learning.

A striking early application of reinforcement learning was that by IBM’s Gerald Tesauro to the game of backgammon [4]. Tesauro’s program, called TD-Gammon, used a learning algorithm that is a straightforward combination of the TD( $\lambda$ ) reinforcement learning algorithm [5], nonlinear function approximation using a multi-layer backpropagation neural network [6], and a shallow heuristic search. Starting with little backgammon knowledge beyond the rules of the game, the program learned from a large number of simulated games to play much better than any prior computer program. After adding some backgammon knowledge, TD-Gammon became as good or better than the world’s best human players, many of whom have changed the way they play opening positions to follow TD-Gammon’s example. Backgammon is an MDP with about  $10^{20}$  states and would be impossible to address with classical optimization methods such as dynamic programming.

A second example application is provided by the elevator dispatcher developed by Crites and Barto [7]. This is a dispatcher for a 10-story building with four elevator cars. The system was modeled as a semi-Markov decision process (SMDP) estimated to have more than  $10^{22}$  states. The reinforcement learning method used was Q-learning [8], an important extension of TD( $\lambda$ ) to control problems, together with backpropagation neural networks. Performance evaluated over 30 hours of simulated elevator time showed that a dispatching policy learned by the reinforcement learning system performed significantly better than a suite of other dispatching algorithms, including advanced research dispatchers and one closely modeled on a policy commonly used in the industry.

Reinforcement learning methods have also been used in significant manufacturing [9], telecommunications [10, 11], and finance applications [12].

Despite such successes, there remain numerous limitations on the effective application of reinforcement learning to large or otherwise challenging prediction and control problems. Some of the main topics on which we propose to extend and improve current reinforcement learning techniques are described in the following subsections.

### 2.1.1 Function approximation

At the heart of all reinforcement learning methods is the updating of a data structure representing a decision-making policy or value function. In almost all real-world applications there are far too many states to store a representation that can be exact, and only an approximation of these functions can be stored; this is referred to as *function approximation*. Although many kinds of function approximation are possible, function approximators that are linear in their parameters are the most widely used and the best understood. By appropriate and generous choice of features, linearity of the function form is often not a significant limitation. For example, in our application to Computer Go, we use a linear function approximator with 1.5 million features to represent an extremely complex evaluation function with high precision. The theory and practice of linear function approximators will be the focus of our research in this area.

On the practical side, the choice of features for a linear function approximator for reinforcement learning methods is more an art than a technology. Tile coding [13, 14, 1] has been shown practical in many reinforcement learning applications and is a key software component of the RL-Toolkit previously developed by the RLAI project, but even here many parameters that have a large effect on performance must be chosen by hand. Our proposals for automating much of the process of parameter setting are discussed in section 2.3.1. Another way of exploring the practice of function approximation is through large-scale applications such as our case studies in Computer Go [15] and Hearts [16].

On the theoretical side, we propose to develop *adaptive* reinforcement learning algorithms, algorithms which choose their function approximator based on the data in such a way that, for large sample sets, their performance is competitive with that of the best possible methods, even those using extensive prior knowledge (such as of the set of relevant features, the order of smoothness, or the manifold where the data lies). Adaptive algorithms have been key to much of the success of learning methods in supervised learning tasks; most modern supervised learning algorithms, including SVMs, SVR, Lasso, and aggregation, build on adaptive ideas in some way. We expect that the success of adaptive algorithms in supervised learning can be repeated in reinforcement learning, resulting in methods that non-experts can use effectively.

Another focus of our research will be on second-order learning methods for reinforcement learning, such as least-squares methods [17, 18, 19]. We have introduced a new variant of linear temporal-difference (TD) learning, called incremental least-squares TD learning, or iLSTD [20]. This method is more data efficient than conventional TD algorithms such as TD( $\lambda$ ) and is more computationally efficient than non-incremental least-squares TD methods such as LSTD [17, 18]. In particular, the per-time-step complexities of iLSTD and TD( $\lambda$ ) are  $O(n)$ , where  $n$  is the number of features, whereas that of LSTD is  $O(n^2)$ . This difference can be decisive in modern applications of reinforcement learning where the use of a large number of features has proven to be an effective solution strategy. In computational experiments, we have shown that iLSTD converges faster than TD( $\lambda$ ) and almost as fast as LSTD. We propose to extend the theory of iLSTD to the policy iteration (control) case (as in [19]) and explore the seemingly close relationships to sampling-based planning methods (section 2.2.5).

### 2.1.2 Actor-critic and policy-gradient reinforcement learning methods

One class of reinforcement learning methods that have been a focus of RLAI research, and that we propose to continue to explore, is that of *actor-critic* methods. Actor-critic reinforcement learning methods are based on the simultaneous online estimation of the parameters of two structures, called the *actor* and the *critic*. The actor corresponds to a conventional action-selection policy or control law, mapping states to actions in a probabilistic manner. The critic corresponds to a conventional value function, mapping states to expected cumulative future reward. Thus, the critic addresses a problem of prediction, whereas the actor is concerned with control. These problems are separable, but are solved simultaneously to find an optimal policy, as in policy iteration. One reason actor-critic methods are appealing is that, by explicitly representing the policy class, they can more easily be biased toward a particular kinds of policy, which can be important in applications where safety is a concern, or in providing prior knowledge. Another reason is that the mapping onto biological implementations seems more straightforward for actor-critic architectures. The intuitive appeal and other advantages of actor-critic methods make them appealing in a variety of reinforcement learning contexts.

Actor-critic methods were among the earliest methods to be investigated in reinforcement learning [21, 22, 23]. They were largely supplanted in the 1990's by methods which estimated state-action value functions and used them directly to select actions without an explicit policy structure. This approach is appealing because of its simplicity, but when combined with function approximation has theoretical difficulties including in some cases a possibility of instability. These problems led to renewed interest in methods with an explicit representation of the policy, which came to be known as *policy gradient* methods [24, 25, 26, 27]. Policy gradient methods without TD learning can easily be proved convergent, but suffer from slow convergence caused by high variance in gradient estimation. Combining policy-gradient methods with TD learning of the value function is a promising avenue toward a more effective method. We have recently extended the theory of actor-critic methods to include convergence

for several new methods using TD learning [28]. Our results are the first to prove convergence when TD errors are used in both the actor and the critic.

Another approach to speeding up policy gradient algorithms was proposed by Kakade [29] and then refined and extended by Bagnell and Schneider [30] and by Peters et al. [31]. Their idea was to replace the policy gradient estimate with an estimate of the so-called *natural gradient* of the policy. This is motivated by the intuition that a change in the way the policy is parameterized should not influence the result of the policy update. In terms of the policy update rule, the move to the natural gradient amounts to linearly transforming the gradient using the inverse Fisher information matrix of the policy. Two of the algorithms for which we have obtained new convergence results utilize natural gradients. Our new theory is based on the two-time-scale techniques pioneered by Borkar and others [32, 33, 34, 35]. Our results are the first two-time-scale results to incorporate natural gradients.

A limitation of all two-time-scale theory is that it has not as yet led to practical actor-critic algorithms; convergence rates for parameter settings that match the theoretical convergence conditions are too slow to be practical. Finding a balance between the asymptotic guarantees and acceptable interim performance is a focus of our continuing research. We propose to conduct research to further extend the two-time-scale theory of actor-critic methods into a regime where step-size parameters can be set large enough to achieve fast learning.

### 2.1.3 Online learning

An online learning algorithm is one that improves its behavior during the normal operation of the system to be controlled or predicted. This is in contrast with the more common, off-line learning algorithms which are trained with a fixed amount of data and then used with learning turned off. In off-line learning the training data has to be fully representative of the system's behavior or performance may be poor. Online learning systems overcome this potential weakness by continually adapting their behavior to the current behavior of the system. Online learning algorithms are a focus of the proposed research.

Online learning is challenging because learning and control are interleaved. Current online learning solutions for MDPs are limited to small finite domains [36, 37, 38] or make strong assumptions (e.g., that the uncertainty is parametric [39, 40]) that seriously limit their applicability because complex environments do not satisfy these conditions. The only non-parametric and non-asymptotic result for learning is very recent [38], although this result still assumes that the the state space of the MDP is finite.

Basic research is required to develop and study online learning algorithms that will work in large, complex environments. In particular the following issues would be addressed in the proposed research: (1) understanding what makes efficient online learning possible; (2) characterizing the behavior of online learning algorithms; and (3) developing online learning algorithms that are efficient in terms of both data and computation.

We propose to study online learning using a non-parametric approach. If the environment is complex, it is rarely realistic to assume that the uncertainty takes a certain, identifiable parametric form. Non-parametric algorithms require weaker prior information, such as bounds on the signals or bounds on the moments of the signals. Hence, non-parametric methods can be applied to much larger problems and to a larger range of problems. In complex environments the challenges are compounded because of the large size of the underlying state and action spaces.

## 2.2 Grounded artificial intelligence

It is conventional to divide reinforcement learning methods into those that are model free and those that are model based. Model-free methods compute their value function and policy directly from experience, whereas model-based methods form a model of the system to be controlled and use it as an intermediate step toward computing their value functions and policies. Model-free methods are conceptually simpler and generally require less computation and memory, whereas model-based methods can be more efficient in terms of the amount of experience needed to achieve a given performance level. By converting their experience into a model, model-based methods are able to retain it and make use of it later, whereas model-free methods use each experience once, when it occurs, and then discard it. Thus, model-free methods can be more computationally efficient whereas model-based methods are more data efficient.

The usual notion of a model in reinforcement learning is an estimate of the system’s transition probabilities and expected rewards—the conventional structures defining a specific Markov decision process. The model specifies the probability of going from each state to each other state as a function of the action taken. This knowledge about the learning agent’s environment can then be used by simple planning processes, such as the policy iteration method of dynamic programming, to compute the value function and policy.

Model-based methods are where reinforcement learning starts to touch the larger concerns and ambitions of artificial intelligence. The model corresponds roughly to knowledge about the world, and computing the value function for the Markov decision process corresponds to heuristic search, planning and reasoning. The reinforcement learning approach holds promise because the planning methods are more general and systematic; they handle stochastic systems and are domain independent. The reinforcement learning approach is usually treated as more low-level, closer to data; this suggests that it may be more amenable to using modern machine learning methods to learn about the world’s transition structure. The artificial intelligence approach is often higher level, closer to human-level reasoning. A goal of the current and proposed research is to begin to bridge the gap between lower-level reinforcement learning and higher-level artificial intelligence to gain the advantages of both.

The primary focus of this part of the research program is on how intelligent machines represent their knowledge of the world. Knowledge representation is widely recognized as critical to the performance of all artificial intelligence systems, and it has long been a goal for artificial intelligence systems to be able to express their knowledge in terms of their interaction with the world. Connecting knowledge to experience in such a way could enable it to be maintained, verified and even learned without human intervention. This ability may be essential to the practical development of large artificial intelligence systems; attaining it is a central objective of the RLAI research project.

Grounding knowledge in experience is challenging because knowledge is high-level and conceptual whereas experience is low-level and sensory-motor. Several levels and types of abstraction are required to bridge the gulf. In particular, it is necessary to abstract over time, particularly over temporally abstract actions, and over state. We propose to further explore the technology of *options* [41, 42] for temporal abstraction, and the technology of *predictive state representations* [43, 44, 45] for state abstraction. the overall framework we use is an extension of predictive state representations that is known as *temporal-difference networks* [46, 47] (see also [49]).

### 2.2.1 Representing temporally abstract knowledge with options

Fundamental to artificial intelligence, as well as to the theory of systems and control, is the problem of representing knowledge about the system and about possible courses of action at a multiplicity of interrelated temporal scales. For example, a human traveler must decide which cities to go to, whether to fly, drive, or walk, and the individual muscle contractions involved in each step. Options are a generic concept of “courses of action” which includes both primitive actions such as muscle contractions and temporally extended actions such as traveling to a distant city [41, 42] (see also [50, 51, 52]). The theory of options is based on the theories of Markov and semi-Markov decision processes (SMDPs), but extends these in significant ways. Options can be used in place of actions in all of the planning and learning methods conventionally used in reinforcement learning. Options and models of options can be learned for a wide variety of different subtasks, and then rapidly combined to solve new tasks. Options provide a bridge between the two most important existing theoretical frameworks used in reinforcement learning: MDPs and SMDPs. Options permit planning and learning simultaneously at a multiplicity of times scales, and toward a wide variety of subtasks, which substantially increases the efficiency and abilities of reinforcement learning methods.

Temporally extended knowledge is represented as *option models*. The model of an option gives the expected outcome of executing the option in each state where it can be applied. The outcome is in the form of a “next” state, or distribution of states, at the time the option finishes, together with the expected cumulative reward along the way. This form for the abstract knowledge is exactly as required for it to be used just like low-level knowledge (state-transition probabilities) in conventional dynamic-programming planning methods.

We use a terminology of *questions* and *answers*. The questions corresponding to an option model are “what will the state become if the option is option is executed now?” and “what will the reward be along the way?”. The answers to these questions are provided by the option model. Option models are predictive and temporally abstract representations of the *dynamics* of the reinforcement learning agent’s world, and in this sense they are knowledge about the cause and effect relationships in that world. We now turn to how we can abstract over states in terms of questions and answers, and in terms of option models.

### 2.2.2 Predictive representations of state

The usual assumption in reinforcement learning and in much of control theory is that the state of the system is readily available even if all of its implications for the future are not. This assumption is reasonable for many control, gaming, and operations research problems, but not for many other problems, including the target problems in artificial intelligence. In vision problems, for example, occluded objects are not perceptually available. In everyday activities we have to remember where objects are, what steps we have already done, and what we plan to do next, none of which may be immediately apparent from sensory input. The concept of state is so essential that we generally do not abandon it even if states are not readily available. Instead we use the history of what *is* available to construct an estimate of the state.

As we discuss below, a variety of methods have been proposed for state estimation and representation when the states are not readily available, including POMDPs, history methods such as generalizations of *k*th-order Markov models, and *predictive state representations*, a newer predictive approach that relates in many ways to the temporal abstraction ideas discussed above, and which we propose to investigate further.

The standard way of addressing the hidden state problem is to study *partially observable Markov decision processes* (POMDPs) [53], in which the state of the underlying MDP is not visible but must be inferred from a (typically) smaller set of observations stochastically related to the states. (POMDPs are the generalization of hidden Markov models (HMMs) to decision processes.) The problem with this approach is that it is strongly dependent on a good model of the system dynamics. Most uses of POMDPs assume a perfect dynamics model and attempt only to estimate state. There are algorithms for simultaneously estimating state and dynamics (e.g., [54]), analogous to the Baum-Welch algorithm for HMMs [55], but these are only effective at tuning parameters that are already approximately correct. If important aspects of the dynamics are genuinely unknown, then these methods are rarely effective (e.g., [56]).

In practice, history-based approaches, such as variable-order Markov models (e.g., [57, 58]), are often much more effective. Here, the state representation is a relatively simple record of the stream of past actions and observations. It might record the occurrence of a specific subsequence, or that one event has occurred more recently than another. Such representations are far more closely linked to the data than are POMDP representations. One way of saying this is that POMDP learning algorithms encounter many local minima and saddle points because all their states are equipotential. History-based systems immediately break symmetry, and their direct learning procedure makes them relatively simple. McCallum [59] and others have shown in a number of examples that sophisticated history-based methods can be effective in large problems, and are often more practical than POMDP methods even in small ones.

The predictive approach to state representation [43, 44, 45, 60] (which we propose to investigate further) is like the POMDP approach in that it updates the state representation recursively rather than directly computing it from data. This enables it to attain generality and compactness at least equal to that of the POMDP approach [43]. However, the predictive approach is also like the history-based approach in that its representations are grounded in data. Whereas a history-based representation looks to the past and records what did happen, a predictive representation looks to the future and represents what *will* happen.

A predictive state representation is based on a set of specific questions about future experience that can be compared with what actually happens. In conventional predictive state representations the questions concern the probability of specific concrete action and observation sequences. The questions are “if these actions are taken, will these observations occur?” It is natural to generalize this idea to questions about the outcomes of options as suggested in the preceding section. In either case, the main conceptual advance is putting the predictive questions in machine readable form (and not just their answers); prior work with prediction learning has taken the questions to be implicit, in the human designer’s mind but not in the machine’s. Representing them explicitly enables the set of questions being asked to be large and subject to autonomous elaboration. Predictive state representations are an active area of research (e.g., [61, 62, 63, 64, 65, 49, 66, 67]).

Predictive representations of state in which the questions are about the outcomes of options enable the representation of concepts about the long-term outcome of extended ways of behaving—a substantial generalization of prior work in reinforcement learning and in some ways even going beyond conventional artificial intelligence representations. For example, it is easy using option models to represent the abstract concept of “the room with the battery charger” as the expected outcome of the `dock-with-the-charger` option. Similarly, the abstract concept “chair present” could be the outcome of the `try-to-sit` option.

### 2.2.3 Temporal difference networks

Temporal-difference networks are perhaps the final structural step in combining temporally and spatially abstract knowledge that is grounded in experience. This is the step of making knowledge *compositional*. This is achieved in a predictive representation by allowing the predictive questions to be about outcomes that are themselves the answers to other predictive questions.

The difference is best understood with an example. Using option-based predictive representations with concrete outcomes it is easy to represent “the room with the battery charger” in terms of the outcome of the `dock-with-the-charger` option, as discussed above, because the outcome is concrete; the robot can directly sense whether or not power is trickling into its battery. But we also want to be able to represent knowledge such as “at the end of the hall is a room with a battery charger in it”. It is natural to represent this in terms of the options `follow-the-hall-to-its-end` and `dock-with-the-charger`, one composed with the other. That is, the knowledge is that at the end of `follow-the-hall-to-its-end` the robot will reach a state in which the prediction for `dock-with-the-charger` is high. The outcome is the answer to another question rather than a concrete observation.

The idea of temporal-difference networks with option-based predictive questions is a combination of several steps that are each conceptually simple, but together constitute significant complexity. The ramifications are far from clear and the extent and generality of the mechanisms designed so far has yet to be determined. One focus is on designing a learning algorithm for the general case. We have completed successful experiments with one such algorithm [47, 48], but the algorithm is slow to learn, largely because of the special challenges of off-policy learning, which we discuss next. Much more foundational work is needed to increase the efficiency of this learning algorithm and to explore different ways of using it so that variance does not become an issue. Creating compelling examples and expositions of our experience-based approach to artificial intelligence is also a priority for the proposed research (e.g., [68]).

### 2.2.4 Off-policy learning

Off-policy learning is learning about one way of behaving (the target policy) while actually behaving in some other way (the behavior policy). The problem of off-policy learning is that classical TD reinforcement learning methods such as Q-learning, TD( $\lambda$ ), and dynamic programming can become unstable during off-policy learning if function approximation is used. As both TD and function approximation are thought to be essential for large-scale applications, and off-policy learning is currently seen as necessary for learning temporally abstract system models, this instability is a key stumbling block to extending reinforcement learning abilities. Moreover, if all three of these (off-policy, TD, and function approximation) could be combined it would maximize the potential power of temporal-difference networks.

There are methods for off-policy learning using importance sampling that are known to be stable, but they are also known to have very high variance, causing learning to be very slow [69]. We have empirically tested and mathematically analyzed a variety of off-policy learning algorithms, including least-squares methods [20], importance-sampling methods [69], and dual-representation methods [70]. As part of the work with temporal-difference networks, we have developed a new concept, that of a “recognizer” [71], which achieves faster off-policy learning. A recognizer observes behavior and accepts it, or not, as something that it is learning about. It recognizes a portion of the behavior as, in effect, corresponding to the target policy. Recognizers are used to condition the predictions made by temporal-difference networks on

the options taken. Recent experiments suggest that importance sampling using recognizers has lower variance and is much better behaved than previous importance-sampling methods for off-policy learning.

### 2.2.5 Sampling-based planning

Planning refers to the use of state representations and corresponding models of the world to anticipate the consequences of alternative courses of action and to pick among them. Planning is a core topic for the entire research program, corresponding roughly to reasoning in people and to optimal decision making in control theory. Much of the proposed research program has been directed toward constructing flexible, expressive, multi-scale approximate models of the world, but it is not enough to know how the world will behave; that knowledge must be used flexibly to support decision making. Planning with the kind of function approximation needed for large state spaces is largely an open problem. There are a number of possible approaches arising out of our previous work with iLSTD [20] and with Dyna-style planning systems [72, 73, 74]. We propose to develop these theoretically and experimentally. The initial objective for this work will be a general and sound planning algorithm for worlds with linearly approximated dynamics at a single time scale. If this is achieved then extensions to multiple time scales and predictive state representations will be attempted.

## 2.3 Applications-oriented research

Although the proposed research is primarily curiosity directed, it includes work that is applications oriented and that is in support of possible applications.

### 2.3.1 Standardized reinforcement learning software

One way the proposed research would support applications (as well as further research) would be in the production and support of software and software protocols for reinforcement learning systems. In the past four years we have created and released three sets of software. This software has been placed in the public domain and can be downloaded from <http://RLAI.net>:

- The **RL-Toolkit** is a collection of software and guidelines to facilitate reinforcement learning research and the development of applications. The RL-Toolkit includes software for function approximation, graphics, and demonstrations of reinforcement learning. Most of the RL-Toolkit is written in Python, a popular open-source programming language; some is written in the C++ to maximize efficiency.
- **RL-Glue** is a standard protocol and software interface for inter-connecting reinforcement learning agents and environments. The software enables agents and environments written in different languages to be interfaced. RL-Glue has been a success in that it is now used throughout the world for research and is beginning to be used for education. RL-Glue was used in three international benchmark and competition events at the Neural Information Processing Systems Conference and the International Conference on Machine Learning. These two conferences are the premier forums for reinforcement learning research. RL-Glue appears to have become a de-facto international standard for comparing reinforcement learning algorithms.
- The **RL-Library** is a central site for storing and organizing reinforcement learning code based on standards such as RL-Glue. Such sites exist for other branches of machine

learning, and the need for one in reinforcement learning has been felt for some time, but it has not been possible to create it without a standard such as RL-Glue.

The proposed project would further develop RL-Glue and the RL-Library by adding visualization software, software explicitly in support of reinforcement learning competitions, and additional domains, algorithms and benchmarks. Significant ongoing support will also be required for RL-Glue and the RL-Library in order for them to have maximal impact on the field and to take full advantage of their ability to enhance Alberta’s international reputation.

In concert with our software effort we propose to develop a new focus on “black box” reinforcement learning algorithms. These are algorithms that have no parameters or settings of any kind and that can be used without knowledge of what is going on inside. This is a long-term goal that will probably be the topic of several graduate student’s theses. Our goal for the near future is to develop reinforcement learning algorithms that require only meta-parameters—parameters for setting the other parameters. Such systems will inevitably pay a penalty in initial learning rate, but they would be far easier to use, which could be critical for practical commercial applications.

### **2.3.2 Gas-electric hybrid vehicles**

In collaboration with Toyota Motor Corporation, we have been exploring the use of reinforcement learning technology for improving the fuel efficiency of gas-electric hybrid cars. In this application we seek to minimize fuel consumption without interfering with the car’s performance. In fact, we would like the driver to be unable to tell that the learning system is operating. The car should drive just the same whether or not the learning system is engaged; the gas mileage should just be better if it is. Whenever developing a major application, new challenges arise that feed into, inform, and ultimately direct ongoing research. The special challenge in this application is that the driver’s choices will impact performance in ways outside of the learning algorithm’s control. The driver may ask for accelerations that can be delivered only with poor fuel efficiency. The driver may take the kind of trips that are not fuel efficient (e.g., very short trips or fast highway driving). The challenge for us is to design a learning agent that can separate the effects of its decisions from the effects of the driver’s decisions in order to learn efficiently.

We have formulated this challenge as an instance of the more general problem of learning control with disturbances. Disturbances are defined as aspects of the world that impact performance but which cannot be affected by the choices of the learning agent. The driver is one such disturbance (assuming the controller is achieving its goal of unobtrusiveness). For another example, consider controlling the heating system for a building to maximize comfort while minimizing costs. A major disturbance here is the weather; it will have a large impact on heating costs, but is outside the control of the learning agent.

We have developed a new algorithm for reinforcement learning with disturbances and have shown in simplified cases that it can significantly improve learning performance. This work is in its initial stages and we expect to explore several different algorithms before determining which are most effective in general and in the hybrid-car domain in particular. The proposed research would further develop this application and the relationship with Toyota.

### 2.3.3 Computer Go

The proposed research would involve continuing to develop our application of reinforcement learning methods to Computer Go. This work has been extremely useful in testing research ideas and directing further research. The ancient oriental game of Go has long been a challenge to artificial intelligence. The techniques that have worked so well in chess and so many real-world applications had seemed to have no traction in Go because of the large branching factor, making traditional search impossible. After decades of research, the best Computer Go programs were still no challenge to weak amateur players. A year and a half ago it looked likely that computers might never play a strong game of Go. Since then there has been a revolution in the world of Computer Go.

The revolution began with the introduction of a new algorithm by Kocsis and (RLAI principal investigator) Szepesvari for Monte Carlo tree-search [75]. Their algorithm, known as UCT, is based on a bandit algorithm called UCB [76]. UCT is a sampling-based search algorithm for large, discrete action spaces with a hierarchical structure. UCT was applied to Computer Go by many researchers, leading to a huge jump in performance. In competitions on the standard Computer Go Server, all of the top ten programs are now based on UCT, and the best non-UCT program is 300 rating points worse than the tenth best UCT program, a difference corresponding to about a ten-to-one chance of winning. The Computer Go revolution and RLAI team member's role in it was recognized in an article in *Scientific American* ("Silicon Smackdown," by Karen A. Frenkel, June 2007, Vol. 296, Issue 6, p. 32).

The best Computer Go program in the world is now MoGo, developed by Sylvain Gelly (University of Paris South) with contributions by RLAI team member David Silver. MoGo reached a major landmark in Computer Go this summer by becoming the first ever program to win a game against a professional Go player under tournament conditions. The win was against Guo Juan, a 5 dan professional player and probably the strongest Go player in Europe. A year ago no one would have imagined that this would happen for decades.

MoGo's victory against Guo Juan was in a smaller version of Go played on a 9x9 board. This form of Go is commonly played by people and is the standard in Computer Go research. MoGo has been recognized as the world's best program in 9x9 Go for some time. Extending this, this summer MoGo won the Gold medal at the 2007 Computer Olympiad for full 19x19 Go.

Current versions of UCT are Monte Carlo methods, meaning they use sample trajectories that go all the way to the end of the game. This has been the most effective approach in Computer Go so far, but at least in other applications the use of learned value functions and of temporal-difference learning is probably necessary. RLAI research has also pursued this direction, producing a system called RLGO that learns a linear value function with roughly 1.5 features corresponding to the shapes in small regions of the board. After self-play training with the TD(0) algorithm, RLGO learned a better static evaluation function than any other system that did not use very extensive domain knowledge.

Computer Go has come of age as a tool for artificial intelligence and reinforcement learning research. We are now able to run many learning trials and get statistically significant comparisons of different learning algorithms in Computer Go (e.g., [77]). In particular, we have begun to use it as a testbed for a variety of different strategies and combinations of strategies for sampling-based search. This has become a very active and productive area of research for us and for the international research community, and we will certainly pursue it in the proposed research.

### 2.3.4 Computational models of animal learning

As noted earlier, reinforcement learning is studied in psychology and neuroscience as well as engineering and artificial intelligence. In psychology, reinforcement learning methods are important models of elemental learning processes in animals, such as classical conditioning [78]. In neuroscience, reinforcement learning methods are the dominant models of reward systems in the brain, in particular of the dopamine system [79]. Although the overwhelming preponderance of RLAI research is in artificial intelligence and the engineering domain, we propose a small experimental component to the research program by taking advantage of a relationship which we have forged with Professor E. James Kehoe of the University of New South Wales. Dr. Kehoe is one of the world’s foremost experts in animal learning psychology. His area of specialization is in the temporal structure of prediction processes in animal learning (classical conditioning), which makes his work particularly relevant to this project. There are striking relationships between animal learning models and computational reinforcement learning algorithms; we are exploring ways in which animal learning behavior may yield new insights leading to better algorithms. This has happened several times in the past, leading to some of the most effective modern algorithms such as TD( $\lambda$ ) and Q-learning. There are animal experiments that can significantly help the computational work and that Kehoe can do quickly and inexpensively in Australia.

The animal experiments and models are meant to shed light on a fundamental problem in reinforcement learning as it might be used in real-time applications: how is the time series of inputs (sensor data or stimuli) represented to the learning algorithm. For any punctate external signal there must be a temporally extended internal representation, roughly corresponding to a history-based representation as discussed in section 2.2.2. We are modeling this representation as a sequence of temporally extended internal “micro-stimuli” with a range of delays and proportional dispersions. Micro-stimuli turn a simple sensory event into a temporally extended and multi-component representation which enables precise timing of responses. We expect to use similar sensory representations in our robotics testbeds.

### 2.3.5 Sensor-rich robotics

The RLAI project has included a significant effort in robotics. For example, we have used a robotic Segway to demonstrate robot geo-caching—the locating of a hidden cache through knowledge only of its GPS coordinates. Currently we are exploring extending the idea of sensor bootstrapping—defining the meaning of one sensor in terms of its dynamic, causal relationship to another sensor. Our focus over the next year or more will be on route following. We plan to integrate ideas from ongoing research into a robot system capable of completing the Turkey Trot, a four-kilometer walk for charity that happens every fall at the University of Alberta. The system will require robust positioning, local and global navigation, and obstacle avoidance, all over a long challenging route. It will be further complicated by crowds of other walkers. We hope to enter a preliminary system in the fall of 2008, with the aim to successfully complete the course at an average walking pace in the fall of 2009.

We propose to add a further robotics effort to challenge, direct, and inspire the research on grounded artificial intelligence discussed earlier in this proposal. This robot will be small and mobile and outfitted with an unusually rich set of sensors, including sensors for touch, acceleration, motion, sound, vision, and several kinds of proximity. The initial objective will be for the robot to form an extended multi-level model of the relationships among its sensors and between its sensors and its actuators. We have proposed that higher-level knowledge can be grounded in raw data of sensations and actions; this robotic platform will challenge and

inspire us to see if it can really be done. We also plan to use this platform as a test case for rapid learning and for the use of reinforcement learning by non-experts. We would like a person whose has no training to be able to teach the system new ways of behaving in an intuitive manner much as one might train a particularly cooperative dog.

### **3 Research team**

The proposed research team is headed by four principal investigators: Richard Sutton (team leader), Csaba Szepesvari, Dale Schuurmans, and Michael Bowling, all faculty members in the Computing Science department at the University of Alberta. These researchers have all published extensively in reinforcement learning and in related artificial intelligence fields, and are well-known internationally. As noted earlier, Schuurmans has been awarded a Tier II Canada Research Chair, and both Bowling and Szepesvari have been awarded Alberta Ingenuity Fund (AIF) New Faculty Grants. We anticipate that Barnabas Póczos and Eric Wiewiora will remain as postdoctoral members of the team, and that Mike Sokolsky will remain as robot technician. Póczos was recruited from Hungary, Wiewiora from UCSD, and Sokolsky from CMU; we expect to be able to continue to recruit postdoctoral fellows and other team members from international research centers. Many of the graduate students came to Alberta specifically to work on the RLAI project or with RLAI principal investigators. Of the 31 students in the 2006 reporting period, 12 held major scholarships, including 6 NSERC scholarships, 4 AIF fellowships, and 9 iCORE scholarships. The CVs of all named team members are attached.

### **4 Connection to current Alberta research**

The proposed research is closely tied to other research ongoing at the University of Alberta, including the world reknown research on electronic entertainment and gaming conducted by professors Schaeffer, Holte, Müller, and Buro. Jointly authored papers are common. The four PIs are also four of the eight PIs of the Alberta Ingenuity Centre for Machine Learning (AICML), a major centre funded by the AIF at \$2,000,000/year. The other four PIs are University of Alberta professors Schaeffer, Holte, Goebel, and Greiner. RLAI PIs also interact with Stuart Kauffman at the University of Calgary.

### **5 Sources of other funding**

Other than iCORE, the proposed research project would be funded by 1) the AIF via the AICML, providing for Szepesvari's salary; 2) the University of Alberta, providing for Bowling's salary; 3) the government of Canada via the CRC program, providing for Schuurman's salary and teaching relief; 4) the AIF via AICML, providing support for the research of all four RLAI PIs, who are also PIs of the AICML (the RLAI portion of AICML funding is approximately \$940,000/year); 5) the AIF through New Faculty Grants to Szepesvari and to Bowling; 6) NSERC through discovery grants to the four PIs; and 7) NSERC through a Collaborative Research Grant of which Sutton is one of the PIs. The total expected funding for RLAI research from identified sources other than iCORE, over the five years, is approximately \$7,570,000. A budget is attached that details these contributions.

## 6 Transition plan

The support for curiosity driven research provided and leveraged by iCORE funds is not easily replaced by funding from other sources. The RLAI research project is fortunate in receiving major support from AICML, and that is expected to continue after the proposed granting period. In addition, we believe that iCORE funding has enabled us to establish a record of research excellence, and that this will continue to be reflected in NSERC discovery grants and in our ability to attract and produce scholarship-holding students. Similarly, we hope to have success as we apply for greater funding from sources that we have so far only lightly tapped. In particular, we will apply for NSERC strategic and collaborative grants in reinforcement learning, in conjunction with our colleagues at McGill and Waterloo. We will also attempt to take full advantage of the Killam scholarships for supporting and attracting strong postdoctoral fellows from the US and other countries. The AIF is likely to announce a new program for supporting postdoctoral fellows in Alberta and we will attempt to take advantage of that. There are also possibilities for CRC chairs, and for international, national and provincial programs that support collaborations. There is the possibility of obtaining some funding from international sources, as we have already done with DARPA (the Defense Advanced Research Projects Agency) in the US. Finally, there is the possibility of creating financial support from interactions with industry, such as we have already done in a very small way through our NSERC collaborative grant with Bell Canada and Nortel, and as we could perhaps do through our collaboration with Toyota. With all of these sources in combination, we are confident that we can maintain a vibrant research program in reinforcement learning and artificial intelligence for the long term.

## References

- [1] Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
- [2] Bertsekas, D. P., and Tsitsiklis, J. N. (1996). *Neural Dynamic Programming*. Athena Scientific, Belmont, MA.
- [3] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.
- [4] Tesauro, G. J. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38:58–68.
- [5] Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning*, 3(1):9–44.
- [6] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. I, *Foundations*. Bradford/MIT Press, Cambridge, MA.
- [7] Crites, R. H., and Barto, A. G. (1996) Improving elevator performance using reinforcement learning. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA. MIT Press.

- [8] Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England.
- [9] Mahadevan, S., Marchalleck, N., Das, T., and Gosavi, A. (1997). Self-improving factory simulation using continuous-time average-reward reinforcement learning. In *Machine Learning: Proceedings of the Fourteenth International Conference*.
- [10] Singh, S., and Bertsekas, D. (1997). Reinforcement learning for dynamic channel allocation in cellular telephone systems. In *Advances in Neural Information Processing Systems 9*. MIT Press.
- [11] Nie, J., and Haykin, S. (1999). A Q-learning based dynamic channel assignment technique for mobile communication systems. *IEEE Transactions on Vehicular Technology*, 48(5):1676–1687.
- [12] Neuneier, R. (1997). Enhancing Q-learning for optimal asset allocation. In *Advances in Neural Information Processing Systems 10*.
- [13] Albus, J. S. (1981). *Brain, Behavior, and Robotics*. Byte Books.
- [14] Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (eds.), *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference*, pp. 1038–1044. MIT Press, Cambridge, MA.
- [15] Silver, D., Sutton, R. S., and Muller, M. (2007). Reinforcement learning of local shape in the game of Go. In *20th International Joint Conference on Artificial Intelligence*, pages 1053–1058.
- [16] Sturtevant, N., and White, A. (2006). Feature construction for reinforcement learning in hearts. In *5th International Conference on Computers and Games*.
- [17] Bradtke, S., and Barto, A. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning* 22:33–57.
- [18] Boyan, J. A. (2002). Technical update: Least-squares temporal difference learning. *Machine Learning* 49:233–246.
- [19] Lagoudakis, M., and Parr, R. (2003). Least-squares policy iteration. *Journal of Machine Learning Research* 4, pp. 1107–1149.
- [20] Geramifard, A., Bowling, M., and Sutton, R. S. (2006). Incremental least-squares temporal difference learning. In *Proceedings of The 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference*, pages 356–361.
- [21] Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846.
- [22] Sutton, R. S. (1984). *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts, Amherst, MA.
- [23] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

- [24] Marbach, P. (1998). *Simulation-Based Optimization of Markov Decision Processes*. PhD thesis, Dept. of EECS, MIT, Cambridge, MA.
- [25] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *Neural Information Processing Systems. MIT Press*, pages 1057–1063.
- [26] Konda, V. R., Tsitsiklis, J. N. (2000). Actor-critic algorithms. *NIPS-12*, pages 1008–1014, MIT Press.
- [27] Baxter, J., and Bartlett, P. L. (2001). Infinite-horizon gradient-based policy search. *Journal of Artificial Intelligence Research*, 15:319–350.
- [28] Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. (2008). Incremental natural actor-critic algorithms. In *Advances in Neural Information Processing Systems 20*.
- [29] Kakade, S. (2002). A natural policy gradient. In *Advances in Neural Information Processing Systems 14*, pages 1531–1538.
- [30] Bagnell, J. A., and Schneider, J. G. (2003). Covariant policy search. In *IJCAI-03*, pages 1019–1024.
- [31] Peters, J., Vijayakumar, S., and Schaal, S. (2005). Natural actor-critic. In *16th European Conference on Machine Learning (ECML 2005)*, pages 280–291.
- [32] Abdulla, M. S., and Bhatnagar, S. (2007). Reinforcement learning based algorithms for average cost Markov decision processes. *Discrete Event Dynamic Systems*, 17(1):23–52.
- [33] Bhatnagar, S., and Kumar, S. (2004). A simultaneous perturbation stochastic approximation based actor-critic algorithm for Markov decision processes. *IEEE Transactions on Automatic Control*, 49(4):592–598.
- [34] Konda, V. R., and Borkar, V. S. (1999). Actor-critic-type learning algorithms for Markov decision processes. *SIAM Journal of Control and Optimization*, 38(1):94–123.
- [35] Konda, V. R., and Tsitsiklis, J. N. (1999). Actor-critic algorithms. In *Advances in Neural Information Processing Systems 12*, pages 1008–1014.
- [36] Lai, T. L., and Yakowitz, S. (1995). Machine learning and nonparametric bandit theory. *IEEE Transactions on Automatic Control*, 40:1199–1209.
- [37] Burnetas, A. N., and Katehakis, M. N. (1997). Optimal adaptive policies for Markov Decision Processes. *Mathematics of Operations Research*, 22(1):222–255.
- [38] Auer, P., and Ortner, R. (2006). Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems 19 (NIPS)*.
- [39] Agrawal, R., Teneketzis, D., and Anantharam, V. (1989). Asymptotically efficient adaptive allocation schemes for controlled Markov chains: Finite parameter space. *IEEE Transaction on Automatic Control*, 34:1249–1259.
- [40] Graves, T. L., and Lai, T. L. (1997). Asymptotically efficient adaptive choice of control laws in controlled Markov chains. *SIAM J. of Control and Optimization*, 35:715–743.

- [41] Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112:181–211.
- [42] Precup, D. (2000). Temporal Abstraction in Reinforcement Learning. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst.
- [43] Littman, M. L., Sutton, R. S., Singh, S. (2002). Predictive representations of state. *Advances in Neural Information Processing Systems 14*. MIT Press.
- [44] Rivest, R. L., and Schapire, R. E. (1994). Diversity-based inference of finite automata. *Journal of the ACM*, 41, 555–589.
- [45] Jaeger, H. (2000). Observable operator models for discrete stochastic time series. *Neural Computation*, 12, 1371–1398.
- [46] Sutton, R. S., and Tanner, B. (2004). Temporal-difference networks. In *Advances in Neural Information Processing Systems 17*, pages 1377–1384.
- [47] Sutton, R. S., Rafols, E. J., and Koop, A. (2005). Temporal abstraction in temporal-difference networks. In *Advances in Neural Information Processing Systems 18*, pages 1313–1320.
- [48] Raffols, E. (2006). *Temporal Abstraction in Temporal-difference Networks*. PhD thesis, Dept. of Computer Science, University of Alberta.
- [49] Wolfe, B., and Singh, S. (2006). Predictive state representations with options. In *Proceedings of the 23rd international conference on Machine learning*, pages 1025–1032.
- [50] Dietterich, T. G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* 13:227–303.
- [51] Parr, R. (1998). Hierarchical Control and Learning for Markov Decision Processes. Ph.D. Dissertation, Department of Computer Science, University of California at Berkeley.
- [52] Parr, R., Russell, S. (1998). Reinforcement learning with hierarchies of machines. *Advances in Neural Information Processing Systems 10*, pp. 1043–1049. MIT Press, Cambridge, MA.
- [53] Astrom, K. J. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.
- [54] Chrisman, L. (1992). Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 183–188). San Jose, California: AAAI Press.
- [55] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41, 164–171.
- [56] Shatkay, H., and Kaelbling, L. P. (1997). Learning topological maps with weak local odometric information. *Proceedings of Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)* (pp. 920–929).
- [57] Rissanen, J. (1983). A universal data compression system. *IEEE Transactions on Information Theory* 29 (5): 656–664.

- [58] Begleiter, R., El-Yaniv, R., and Yona, G. (2004). On prediction using variable order Markov models. *Journal of Artificial Intelligence Research* 22: 385–421.
- [59] McCallum, A. K. (1995). *Reinforcement learning with selective perception and hidden state*. Doctoral dissertation, Department of Computer Science, University of Rochester.
- [60] James, M. R., and Singh, S. (2004). Learning and discovery of predictive state representations in dynamical systems with reset. *Proceedings of the International Conference on Machine Learning*, 417-424.
- [61] Kolling, A., Jaeger, H., and Zhao, M. (2005). Efficient training of OOMs. In *Advances in Neural Information Processing Systems 18*, pages 555–562.
- [62] McCracken, P., and Bowling, M., James, M., Neufeld, J., Wilkinson, D. (2006). Learning predictive state representations using non-blind policies. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, 129–136.
- [63] Wiewiora, E. (2005). Learning predictive representations from a history. In *Proceedings of the Twenty-Second International Conference on Machine Learning (ICML 2005)*, pages 964–971.
- [64] Wingate, D., and Singh, S. (2006). Kernel predictive linear Gaussian models for nonlinear stochastic dynamical systems. In *Proceedings of the 23rd international conference on Machine learning*, pages 1017–1024.
- [65] Rudary, M., and Singh, S. (2006). Predictive linear-Gaussian models of controlled stochastic dynamical systems. In *Proceedings of the 23rd international conference on Machine learning*, pages 777–784.
- [66] Wolfe, B., Wingate, D., Soni, V., and Singh, S. (2007). Relational knowledge with predictive state representations. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2035–2040.
- [67] Wingate, D., and Singh, S. (2007). On discovery and learning of models with predictive state representations of state for agents with continuous actions and observations. In *Proceedings of the 2007 International Conference on Autonomous Agents and Multiagent Systems*.
- [68] Koop, A. (2007). Understanding experience: Temporal coherence and empirical knowledge representation. Master’s thesis, Dept. of Computer Science, University of Alberta, 2007.
- [69] Precup, D., Sutton, R. S., Dasgupta, S. (2001). Off-policy temporal-difference learning with function approximation. *Proceedings of the 18th International Conference on Machine Learning*.
- [70] Wang, T., Bowling, M., and Schuurmans, D. (2007). Dual representations for dynamic programming and reinforcement learning. In *Proceedings of the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 44-51.
- [71] Precup, D., Sutton, R. S., Paduraru, C., Koop, A., and Singh, S. (2006). Off-policy learning with options and recognizers. In *Advances in Neural Information Processing Systems 18*, pages 1097–1104.

- [72] Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224. San Mateo, CA.
- [73] Peng, J., and Williams, R. J. (1993). Efficient learning and planning within the Dyna framework. *Adaptive Behavior*, 1(4):437–454.
- [74] Paduraru, C. (2006). Planning with approximate and learned MDP models. Master’s thesis, Department of Computer Science, University of Alberta.
- [75] Kocsis, L., and Szepesvári, Cs. (2006). Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning (ECML-2006)*, pages 282–293.
- [76] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite time analysis of the multi-armed bandit problem. *Machine Learning*, 47(2-3):235–256.
- [77] Sutton, R. S., Koop, A., and Silver, D. (2007). On the role of tracking in stationary environments. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML 2007)*, pages 871–878.
- [78] Sutton, R. S., and Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88:135–170.
- [79] Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275:1593–1598.