Reinforcement Learning is Direct Adaptive Optimal Control

Richard S. Sutton, Andrew G. Barto, Ronald J. Williams *

Abstract

In this paper we present a control-systems perspective on one of the major neural-network approaches to learning control, *reinforcement learning.* Control problems can be divided into two classes: 1) regulation and tracking problems, in which the objective is to follow a reference trajectory, and 2) optimal control problems, in which the objective is to extremize a functional of the controlled system's behavior that is not necessarily defined in terms of a reference trajectory. Adaptive methods for problems of the first kind are well known, and include self-tuning regulators and model-reference methods, whereas adaptive methods for optimal-control problems have received relatively little attention. Moreover, the adaptive optimal-control methods that have been studied are almost all *indirect* methods, in which controls are recomputed from an estimated system model at each step. This computation is inherently complex, making adaptive methods in which the optimal controls are estimated directly more attractive. We view reinforcement learning methods as a computationally simple, direct approach to the adaptive optimal control of nonlinear systems. For concreteness, we focus on one reinforcement learning method (Q-learning) and on its analytically proven capabilities for one class of adaptive optimal control problems (markov decision problems with unknown transition probabilities).

INTRODUCTION

All control problems involve manipulating a dynamical system's input so that its behavior meets a collection of specifications constituting the control objective. In some problems, the control objective is defined in terms of a reference level or reference trajectory that the controlled system's output should match or track as closely as possible. Stability is the key issue in these regulation and tracking problems. In other problems, the control objective is to extremize a functional of the controlled system's behavior that is not necessarily defined in terms of a reference level or trajectory. The key issue in the latter problems is constrained optimization; here optimal-control methods based on the calculus of variations and dynamic programming have been extensively studied. In recent years, optimal control has received less attention than regulation and tracking, which have proven to be more tractable both analytically and computationally, and which pro-

*Presented at 1991 American Control Conference, Boston, MA, June 26-28, 1991. Richard Sutton is with GTE Laboratories Inc., Waltham, MA 02254. Andrew Barto is with University of Massachusetts, Amherst, MA 01003. Ronald Williams is with Northeastern University, Boston, MA 02115. The authors thank Erik Ydstie, Judy Franklin, and Oliver Selfridge for discussions contributing to this article. This research was supported in part by grants to A.G. Barto from the National Science Foundation (ECS-8912623) and the Air Force Office of Scientific Research, Bolling AFB (AFOSR-89-0526), and by a grant to R.J. Williams from the National Science Foundation (IRI-8921275).

duce more reliable controls for many applications.

When a detailed and accurate model of the system to be controlled is not available, adaptive control methods can be applied. The overwhelming majority of adaptive control methods address regulation and tracking problems. However, adaptive methods for optimal control problems would be widely applicable if methods could be developed that were computationally feasible and that could be applied robustly to nonlinear systems.

Tracking problems assume prior knowledge of a reference trajectory, but for many problems the determination of a reference trajectory is an important part—if not the *most* important part—of the overall problem. For example, trajectory planning is a key and difficult problem in robot navigation tasks, as it is in other robot control tasks. To design a robot capable of walking bipedally, one may not be able to specify a desired trajectory for the limbs a priori, but one can specify the objective of moving forward, maintaining equilibrium, not damaging the robot, etc. Process control tasks are typically specified in terms of overall objectives such as maximizing yield or minimizing energy consumption. It is generally not possible to meet these objectives by dividing the task into separate phases for trajectory planning and trajectory tracking. Ideally, one would like to have both the trajectories and the required controls determined so as to extremize the objective function.

For both tracking and optimal control, it is usual to distinguish between *indirect* and *direct* adaptive control methods. An indirect method relies on a system identification procedure to form an explicit model of the controlled system and determines then the control rule from the model. Direct methods determine the control rule without forming such a system model.

In this paper we briefly describe learning methods known as *reinforcement learning* methods, and present them as a direct approach to adaptive optimal control. These methods have their roots in studies of animal learning and in early learning control work (e.g., [22]), and are now an active area of research in neural networks and machine learning (e.g., see [1,41]). We summarize here an emerging deeper understanding of these methods that is being obtained by viewing them as a synthesis of dynamic programming and stochastic approximation methods.

REINFORCEMENT LEARNING

Reinforcement learning is based on the commonsense idea that if an action is followed by a satisfactory state of affairs, or by an improvement in the state of affairs (as determined in some clearly defined way), then the tendency to produce that action is strengthened, i.e., reinforced. This idea plays a fundamental role in theories of animal learning, in parameter-perturbation adaptive-control methods (e.g., [12]), and in the theory of learning automata and bandit problems [8,26]. Extending this idea to allow action selections to depend on state information introduces aspects of feedback control, pattern recognition, and associative learning (e.g., [2,6]). Further, it is possible to extend the idea of being "followed by a satisfactory state of affairs" to include the long-term consequences of actions. By combining methods for adjusting action-selections with methods for estimating the long-term consequences of actions, reinforcement learning methods can be devised that are applicable to control problems involving temporally extended behavior (e.g., [3,4,7,13,14,30,34,35,36,40]). Most formal results that have been obtained are for the control of Markov processes with unknown transition probabilities (e.g., [31,34]). Also relevant are formal results showing that optimal controls can] often be computed using more asynchronous or incremental forms of dynamic programming than are conventionally used (e.g., [9,39,42]). Empirical (simulation) results using reinforcement learning combined with neural networks or other associative memory structures have shown robust efficient learning on a variety of nonlinear control problems (e.g., [5,13,19,20,24,25,29,32,38,43]). An overview of the role of reinforcement learning within neural-network approaches is provided by [1]. For a readily accessible example of reinforcement learning using neural networks the reader is referred to Anderson's article on the inverted pendulum problem [43].

Studies of reinforcement-learning neural networks in nonlinear control problems have generally focused on one of two main types of algorithm: *actor-critic learning* or *Q-learning.* An actor-critic learning system contains two distinct subsystems, one to estimate the long-term utility for each state and another to learn to choose the optimal action in each state. A Q-learning system maintains estimates of utilities for all state-action pairs and makes use of these estimates to select actions. Either of these techniques qualifies as an example of a direct adaptive optimal control algorithm, but because Q-learning is conceptually simpler, has a better-developed theory, and has been found empirically to converge faster in many cases, we elaborate on this particular technique here and omit further discussion of actor-critic learning.

Q-LEARNING

One of the simplest and most promising reinforcement learning methods is called *Q-learning* [34]. Consider the following finite-state, finite-action Markov decision problem. At each discrete time step, $k = 1, 2, \ldots$, the controller observes the state x_k of the Markov process, selects action a_k, receives resulant reward r_k, and observes the resultant next state x_{k+1}. The probability distributions for r_k and x_{k+1} depend only on x_k and a_k, and r_k has finite expected value. The objective is to find a control rule (here a stationary control rule suffices, which is a mapping from states to actions) that maximizes at each time step the expected discounted sum of future reward. That is, at any time step k, the control rule should specify action a_k so as to maximize

$$E\Big\{\sum_{j=0}^{\infty} \gamma^j r_{k+j}\Big\},$$

where γ, $0 \leq \gamma < 1$, is a discount factor.

Given a complete and accurate model of the Markov decision problem in the form of the state transition probabilities for each action and the probabilities specifying the reward process, it is possible to find an optimal control rule by applying one of several dynamic programming (DP) algorithms. If such a model is not available a priori, it could be estimated from observed rewards and state transitions, and DP could be applied using the estimated model. That would constitute an indirect adaptive control method. Most of the methods for the adaptive control of Markov processes described in the engineering literature are indirect (e.g., [10,18,21,28]).

Reinforcement learning methods such as Q-learning, on the other hand, do not estimate a system model. The basic idea in Q-learning is to estimate a real-valued function, Q, of states and actions, where $Q(x, a)$ is the expected discounted sum of future reward for performing action a in state x and performing optimally thereafter. (The name "Q-learning" comes purely from Watkins's notation.) This function satisfies the following recursive relationship (or "functional equation"):

$$Q(x, a) = E\{r_k + \gamma \max_b Q(x_{k+1}, b) \mid x_k = x, a_k = a\}.$$

An optimal control rule can be expressed in terms of Q by noting that an optimal action for state x is any action a that maximizes $Q(x, a)$.

The Q-learning procedure maintains an estimate $\hat{Q}$ of the function Q. At each transtion from step k to $k+1$, the learning system can observe x_k, a_k, r_k, and x_{k+1}. Based on these observations, $\hat{Q}$ is updated at time step $k+1$ as follows: $\hat{Q}(x, a)$ remains unchanged for all pairs $(x, a) \neq (x_k, a_k)$ and

$$\hat{Q}(x_k, a_k) := \hat{Q}(x_k, a_k) + \qquad (1)$$

$$\beta_k[r_k+\gamma \max_b \hat{Q}(x_{k+1}, b)-\hat{Q}(x_k, a_k)],$$

where β_k is a gain sequence such that $0 < \beta_k < 1$, $\sum_{k=1}^{\infty} \beta_k = \infty$, and $\sum_{k=1}^{\infty} \beta_k^2 < \infty$. Watkins [34] has shown that $\hat{Q}$ converges to Q with probability one if all actions continue to be tried from all states. This is a weak condition in the sense that it would have to be met by any algorithm capable of solving this problem. The simplest way to satisfy this condition while also attempting to follow the current estimate for the optimal control rule is to use a stochastic control rule that "prefers," for state x, the action a that maximizes $\hat{Q}(x, a)$, but that occassionally selects an action at random. The probability of taking a random action can be reduced with time according to a fixed schedule. Stochastic automata methods or exploration methods such as that suggested by Sato et al. [28] can also be employed [4].

Because it does not rely on an explicit model of the Markov process, Q-learning is a direct adaptive method. It differs from the direct method of Wheeler and Narendra [37] in that their method does not estimate a value function, but constructs the control rule directly. Q-learning and other reinforcement learning methods are most closely related to—but were developed independently of—the adaptive Markov control method of Jalali and Ferguson [16], which they call "asynchronous transient programming." Asynchronous DP methods described by Bertsekas and Tsitsiklis [9] perform local updates of the value function asynchronously instead of using the systematic updating sweeps of conventional DP algorithms. Like Q-learning, asynchronous transient programming performs local updates on the state currently being visited. Unlike Q-learning, however, asynchronous transient programming is an indirect adaptive control method because it requires an explicit model of the Markov process. The Q function, on the other hand, combines information about state transitions and estimates of future reward without relying on explicit estimates of state transition probablities. The advantage of all of these methods is that they require enormously less computation at each time step than do indirect adaptive optimal control methods using conventional DP algorithms.

REPRESENTING THE Q FUNCTION

Like conventional DP methods, the Q-learning method given by (1) requires memory and overall computation proportional to the number of state-action pairs. In large problems, or in problems with continuous state and action spaces which must be quantized, these methods becomes extremely complex (Bellman's "curse of dimensionality"). One approach to reducing the severity of this problem is to represent $\hat{Q}$ not as a look-up table, but as a parameterized structure such as a low-order polynomial, k-d tree, decision tree, or neural network. In general, the local update rule for $\hat{Q}$ given by (1) can be adapted for use with any method for adjusting parameters of function representations via *supervised learning* methods (e.g., see [11]). One can define a general way of moving from a unit of experience (x_k, a_k, r_k, and x_{k+1}, as in (1)) to a training example for $\hat{Q}$:

$$\hat{Q}(x_k, a_k) \quad \text{should be} \quad r_k + \gamma \max_b \hat{Q}(x_{k+1}, b).$$

This training example can then be input to any supervised learning method, such as a parameter estimation procedure based on stochastic approximation. The choice of learning method will have a strong effect on generalization, the speed of learning, and the quality of the final result. This approach has been used successfully with supervised learning methods based on error backpropagation [19], CMACs [34], and nearest-neighbor methods [25]. Unfortunately, it is not currently known how theoretical guarantees of convergence extend to various function representations, even representations in which the estimated function values are linear in the representation's parameters. This is an important area of current research.

HYBRID DIRECT/INDIRECT METHODS

Q-learning and other reinforcement learning methods are incremental methods for performing DP using actual experience with the controlled system in place of a model of that system [7,34,36]. It is also possible to use these methods with a system model, for example, by using the model to generate hypothetical experience which is then processed by Q-learning just as if it were experience with the actual system. Further, there is nothing to prevent using reinforcement learning methods on both actual and simulated experience simultaneously. Sutton [32] has proposed a learning architecture called "Dyna" that simultaneously 1) performs reinforcement learning using actual experiences, 2) applies the same reinforcement learning method to model-generated experiences, and 3) updates the system model based on actual experiences. This is a simple and effective way to combine learning and incremental planning capabilities, an issue of increasing significance in artificial intelligence (e.g., see [15]).

CONCLUSIONS

Although its roots are in theories of animal learning developed by experimental psychologists, reinforcement learning has strong connections to theoretically justified methods for direct adaptive optimal control. When procedures for designing controls from a system model are computationally simple, as they are in linear regulation and tracking tasks, the distinction between indirect and direct adaptive methods has minor impact on the feasibility of an adaptive control method. However, when the design procedure is very costly, as it is in nonlinear optimal control, the distinction between indirect and direct methods becomes much more important. In this paper we presented reinforcement learning as an on-line DP method and a computationally inexpensive approach to direct adaptive optimal control. Methods of this kind are helping to integrate insights from animal learning [7,33], artificial intelligence [17,27,31,32], and perhaps—as we have argued here—optimal control theory.

References

[1] Barto, A.G. (1990) Connectionist learning for control: An overview. In: *Neural Networks for Control*, edited by W.T. Miller, R.S. Sutton, and P.J. Werbos, pp. 5–58, MIT Press.

[2] Barto, A.G., Anandan, P. (1985) Pattern recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:360–375.

[3] Barto, A.G., Singh, S.P. (1990) Reinforcement Learning and Dynamic Programming, *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems*, 83–88, Yale University, New Haven, CT.

[4] Barto, A.G., Singh, S.P. (1991) On the computational economics of reinforcement learning, In *Connectionist Models: Proceedings of the 1990 Summer School*, edited by D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, San Mateo, CA: Morgan Kaufmann.

[5] Barto, A.G., Sutton R.S., Anderson, C.W. (1983) Neuronlike elements that can solve difficult learning control problems. *IEEE Trans. on Systems, Man, and Cybernetics, SMC-13*, No. 5, pp. 834–846.

[6] Barto, A.G., Sutton, R.S., Brouwer, P.S. (1981) Associative search network: A reinforcement learning associative memory. *Biological Cybernetics*, *40*, 201–211.

[7] Barto, A.G., Sutton, R.S., Watkins, C.J.C.H. (1990) Learning and sequential decision making. In *Learning and Computational Neuroscience,* M. Gabriel and J.W. Moore (Eds.), MIT Press.

[8] Berry, D.A., Fristedt, B. (1985) *Bandit Problems: Sequential Allocation of Experiments.* London: Chapman and Hall.

[9] Bertsekas, D.P. Tsitsiklis, J.N. (1989) *Parallel Distributed Processing: Numerical Methods*, Prentice-Hall.

[10] Borkar, V., Varaiya, P. (1979) Adaptive control of markov chains I: Finite parameter set. *IEEE Transactions on Automatic Control*, 24:953–957, 1979.

[11] Duda, R.O., Hart, P.E. (1973) *Pattern Classification and Scene Analysis.* New York: Wiley.

[12] Eveleigh, V.W. (1967) *Adaptive Control and Optimization Techniques*, New York: McGraw-Hill.

[13] Grefenstette, J. J., Ramsey, C. L., & Schultz, A. C. (1990) Learning sequential decision rules using simulation models and competition. *Machine Learning 5*, 355–382.

[14] Hampson, S.E. (1989) *Connectionist Problem Solving: Computational Aspects of Biological Learning.* Boston: Birkhauser.

[15] Hendler, J. (1990) *Planning in Uncertain, Unpredictable, or Changing Environments* (Working notes of the 1990 AAAI Spring Symposium), Edited by J. Hendler, Technical Report SRC-TR-90-45, University of Maryland, College Park, MD.

[16] Jalali, A., Ferguson, M. (1989) Computationally efficient adaptive control algorithms for markov chains, *Proceedings of the 28th Conference on Decision and Control*, 1283–1288.

[17] Korf, R. E. (1990) Real-Time Heuristic Search. *Artificial Intelligence 42*: 189–211.

[18] Kumar, P.R., Lin, W. (1982) Optimal adaptive controllers for unknown markov chains. *IEEE Transactions on Automatic Control*, 25:765–774.

[19] Lin, Long-Ji. (1991) Self-improving reactive agents: Case studies of reinforcement learning frameworks. In: *Proceedings of the International Conference on the Simulation of Adaptive Behavior*, MIT Press.

[20] Mahadevan, S. Connell, J. (1990) Automatic programming of behavior-based robots using reinforcement learning. IBM technical report.

[21] Mandl, P. (1974) Estimation and control in markov chains. *Advances in Applied Probability*, 6:40–60.

[22] Mendel, J.M., McLaren, R.W. (1970) Reinforcement learning control and pattern recognition systems. In: *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications*, edited by J.M. Mendel & K.S. Fu, pp. 287–318. New York: Academic Press.

[24] Millington, P.J. (1991) *Associative Reinforcement Learning for Optimal Control*, M.S. Thesis, Massachusetts Institute of Technology, Technical Report CSDL-T-1070.

[25] Moore, A.W. (1990) *Efficient Memory-Based Learning for Robot Control.* PhD thesis, Cambridge University Computer Science Department.

[26] Narendra, K.S., Thathachar, M.A.L. (1989) *Learning Automata: An Introduction.* Prentice Hall, Englewood Cliffs, NJ.

[27] Samuel, A.L. (1959) Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development, 3*, 210–229. (Reprinted in *Computers and Thought*, edited by E.A. Feigenbaum & J. Feldman, pp. 71–105. New York: McGraw-Hill, 1963.)

[28] Sato, M., Abe, K., Takeda, H. (1988) Learning control of finite markov chains with explicit trade-off between estimation and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:677–684.

[29] Selfridge, O.G., Sutton,R.S., Barto, A.G. (1985) Training and Tracking in Robotics, *Proceedings of IJCAI-85*, International Joint Conference on Artificial Intelligence, pp. 670–672.

[30] Sutton, R.S. (1984) Temporal credit assignment in reinforcement learning. Doctoral dissertation, Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.

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

[32] Sutton, R.S. (1990) First results with Dyna: An integrated architecture for learning, planning, and reacting. *Proceedings of the 1990 AAAI Spring Symposium on Planning in Uncertain, Unpredictable, or Changing Environments.*

[33] Sutton, R.S., Barto, A.G. (1990) Time-derivative models of Pavlovian reinforcement. In *Learning and Computational Neuroscience,* M. Gabriel and J.W. Moore (Eds.), MIT Press.

[34] Watkins, C.J.C.H. (1989) *Learning with Delayed Rewards.* PhD thesis, Cambridge University Psychology Department.

[35] Werbos, P.J. (1987) Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. *IEEE Transactions on Systems, Man, and Cybernetics*, Jan-Feb.

[36] Werbos, P.J. (1989) Neural networks for control and system identification. In *Proceedings of the 28th Conference on Decision and Control*, pages 260–265.

[37] Wheeler, R.M., Narendra, K.S. (1986) Decentralized learning in finite markov chains. *IEEE Transactions on Automatic Control*, 31:519–526.

[38] Whitehead, S.D., Ballard, D.H. (in press) Learning to perceive and act by trial and error. *Machine Learning.*

[39] Williams, R.J., Baird, L.C. (1990) A mathematical analysis of actor-critic architectures for learning optimal controls through incremental dynamic programming, *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems*, 96–101, Yale University, New Haven, CT.

[40] Witten, I.H. (1977) An adaptive optimal controller for discrete-time Markov environments. *Information and Control*, 34:286–295.

[41] Sutton, R.S., Ed. (1992) *Machine Learning 8*, No. 3/4 (special issue on Reinforcement Learning), Kluwer Academic.

[42] Barto, A.G., Bradtke, S.J., Singh, S.P. (1991) Real-time learning and control using asynchronous dynamic programming. University of Massachusetts at Amherst Technical Report 91-57.

[43] Anderson, C.W. (1989) Learning to control an inverted pendulum using neural networks. *Control Systems Magazine*, 31–37, April.