CHAPTER 1

INTRODUCTION

1.1 FROM RL TO RADP

1.1.1 Introduction to RL

Reinforcement learning (RL) is originally observed from the learning behavior in humans and other mammals. The definition of RL varies in different literature. Indeed, learning a certain task through trial-and-error can be considered as an example of RL. In general, an RL problem requires the existence of an *agent*, that can interact with some unknown *environment* by taking *actions*, and receiving a *reward* from it. Sutton and Barto referred to RL as *how to map situations to actions so as to maximize a numerical reward signal* [47]. Apparently, maximizing a reward is equivalent to minimizing a *cost*, which is used more frequently in the context of optimal control [32]. In this book, a mapping between situations and actions is called a *policy*, and the goal of RL is to learn an optimal policy such that a predefined cost is minimized.

As a unique learning approach, RL does not require a supervisor to teach an agent to take the optimal action. Instead, it focuses on how the agent, through interactions with the unknown environment, should modify its own actions toward the optimal one (Figure 1.1). An RL iteration generally contains two major steps. First, the agent evaluates the cost under the current policy, through interacting with the environment. This step is known as *policy evaluation*. Second, based on the evaluated cost, the agent adopts a new policy aiming at further reducing the cost. This is the step of *policy improvement*.

Robust Adaptive Dynamic Programming, First Edition. Yu Jiang and Zhong-Ping Jiang.

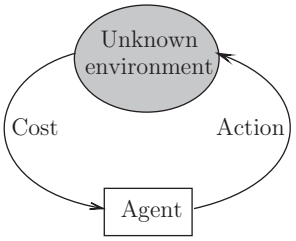

FIGURE 1.1 Illustration of RL. The agent takes an action to interact with the unknown environment, and evaluates the resulting cost, based on which the agent can further improve the action to reduce the cost.

As an important branch in machine learning theory, RL has been brought to the computer science and control science literature as a way to study artificial intelligence in the 1960s [37, 38, 54]. Since then, numerous contributions to RL, from a control perspective, have been made (see, e.g., [2, 29, 33, 34, 46, 53, 56]). Recently, AlphaGo, a computer program developed by Google DeepMind, is able to improve itself through reinforcement learning and has beaten professional human Go players [44]. It is believed that significant attention will continuously be paid to the study of reinforcement learning, since it is a promising tool for us to better understand the true intelligence in human brains.

1.1.2 Introduction to DP

On the other hand, dynamic programming (DP) [4] offers a theoretical way to solve multistage decision-making problems. However, it suffers from the inherent computational complexity, also known as the *curse of dimensionality* [41]. Therefore, the need for approximative methods has been recognized as early as in the late 1950s [3]. In [15], an iterative technique called policy iteration (PI) was devised by Howard for Markov decision processes (MDPs). Also, Howard referred to the iterative method developed by Bellman [3, 4] as value iteration (VI). Computing the optimal solution through successive approximations, PI is closely related to learning methods. In 1968, Werbos pointed out that PI can be employed to perform RL [58]. Starting from then, many real-time RL methods for finding online optimal control policies have emerged and they are broadly called approximate/adaptive dynamic programming (ADP) [31, 33, 41, 43, 55, 60–65, 68], or neurodynamic programming [5]. The main feature of ADP [59, 61] is that it employs ideas from RL to achieve online approximation of the value function, without using the knowledge of the system dynamics.

1.1.3 The Development of ADP

The development of ADP theory consists of three phases. In the first phase, ADP was extensively investigated within the communities of computer science and

operations research. PI and VI are usually employed as two basic algorithms. In [46], Sutton introduced the temporal difference method. In 1989, Watkins proposed the well-known Q-learning method in his PhD thesis [56]. Q-learning shares similar features with the action-dependent heuristic dynamic programming (ADHDP) scheme proposed by Werbos in [62]. Other related research work under a discrete time and discrete state-space Markov decision process framework can be found in [5, 6, 8, 9, 41, 42, 48, 47] and references therein. In the second phase, stability is brought into the context of ADP while real-time control problems are studied for dynamic systems. To the best of our knowledge, Lewis and his co-workers are the first who contributed to the integration of stability theory and ADP theory [33]. An essential advantage of ADP theory is that an optimal control policy can be obtained via a recursive numerical algorithm using online information without solving the Hamilton-Jacobi-Bellman (HJB) equation (for nonlinear systems) and the algebraic Riccati equation (ARE) (for linear systems), even when the system dynamics are not precisely known. Related optimal feedback control designs for linear and nonlinear dynamic systems have been proposed by several researchers over the past few years; see, for example, [7, 10, 39, 40, 50, 52, 66, 69]. While most of the previous work on ADP theory was devoted to discrete-time (DT) systems (see [31] and references therein), there has been relatively less research for the continuous-time (CT) counterpart. This is mainly because ADP is considerably more difficult for CT systems than for DT systems. Indeed, many results developed for DT systems [35] cannot be extended straightforwardly to CT systems. As a result, early attempts were made to apply Q-learning for CT systems via discretization technique [1, 11]. However, the convergence and stability analysis of these schemes are challenging. In [40], Murray et. al proposed an implementation method which requires the measurements of the derivatives of the state variables. As said previously, Lewis and his co-workers proposed the first solution to stability analysis and convergence proofs for ADP-based control systems by means of linear quadratic regulator (LQR) theory [52]. A synchronous policy iteration scheme was also presented in [49]. For CT linear systems, the partial knowledge of the system dynamics (i.e., the input matrix) must be precisely known. This restriction has been completely removed in [18]. A nonlinear variant of this method can be found in [22] and [23].

The third phase in the development of ADP theory is related to extensions of previous ADP results to nonlinear uncertain systems. Neural networks and game theory are utilized to address the presence of uncertainty and nonlinearity in control systems. See, for example, [14, 31, 50, 51, 57, 67, 69, 70]. An implicit assumption in these papers is that the system order is known and that the uncertainty is static, not dynamic. The presence of dynamic uncertainty has not been systematically addressed in the literature of ADP. By dynamic uncertainty, we refer to the mismatch between the nominal model (also referred to as the *reduced-order system*) and the real plant when the order of the nominal model is lower than the order of the real system. A closely related topic of research is how to account for the effect of unseen variables [60]. It is quite common that the full-state information is often missing in many engineering applications and only the output measurement or partial-state measurements are available. Adaptation of the existing ADP theory to this practical scenario is important yet non-trivial. Neural networks are sought for addressing the state estimation problem

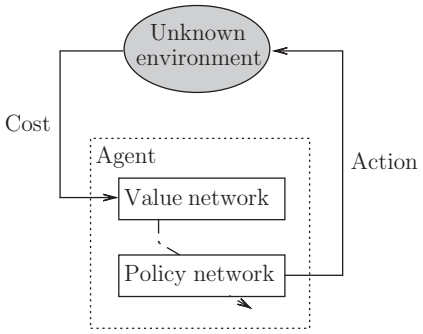

FIGURE 1.2 Illustration of the ADP scheme.

[12, 28]. However, the stability analysis of the estimator/controller augmented system is by no means easy, because the total system is highly interconnected and often strongly nonlinear. The configuration of a standard ADP-based control system is shown in Figure 1.2.

Our recent work [17, 19, 20, 21] on the development of robust ADP (for short, RADP) theory is exactly targeted at addressing these challenges.

1.1.4 What Is RADP?

RADP is developed to address the presence of dynamic uncertainty in linear and nonlinear dynamical systems. See Figure 1.3 for an illustration. There are several reasons for which we pursue a new framework for RADP. First and foremost, it is well known that building an exact mathematical model for physical systems often is

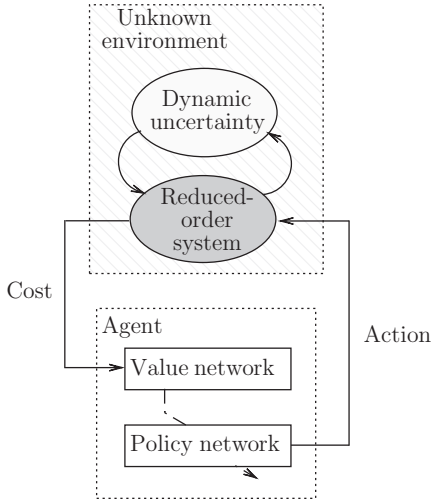

FIGURE 1.3 In the RADP learning scheme, a new component, known as dynamic uncertainty, is taken into consideration.

a hard task. Also, even if the exact mathematical model can be obtained for some particular engineering and biological applications, simplified nominal models are often more preferable for system analysis and control synthesis than the original complex system model. While we refer to the mismatch between the simplified nominal model and the original system as dynamic uncertainty here, the engineering literature often uses the term of *unmodeled dynamics* instead. Second, the observation errors may often be captured by dynamic uncertainty. From the literature of modern nonlinear control [25, 26, 30], it is known that the presence of dynamic uncertainty makes the feedback control problem extremely challenging in the context of nonlinear systems. In order to broaden the application scope of ADP theory in the presence of dynamic uncertainty, our strategy is to integrate tools from nonlinear control theory, such as Lyapunov designs, input-to-state stability theory [45], and nonlinear small-gain techniques [27]. This way RADP becomes applicable to wide classes of uncertain dynamic systems with incomplete state information and unknown system order/dynamics.

Additionally, RADP can be applied to large-scale dynamic systems as shown in our recent paper [20]. By integrating a simple version of the cyclic-small-gain theorem [36], asymptotic stability can be achieved by assigning appropriate weighting matrices for each subsystem. Further, certain suboptimality property can be obtained. Because of several emerging applications of practical importance such as smart electric grid, intelligent transportation systems, and groups of mobile autonomous agents, this topic deserves further investigations from an RADP point of view. The existence of unknown parameters and/or dynamic uncertainties and the limited information of state variables give rise to challenges for the decentralized or distributed controller design of large-scale systems.

1.2 SUMMARY OF EACH CHAPTER

This book is organized as follows. Chapter 2 studies ADP for uncertain linear systems, of which the only a priori knowledge is an initial, stabilizing static state-feedback control policy. Then, via policy iteration, the optimal control policy is approximated. Two ADP methods, on-policy learning and off-policy learning, are introduced to achieve online implementation of conventional policy iteration. As a result, the optimal control policy can be approximated using online measurements, instead of the knowledge of the system dynamics.

Chapter 3 further extends the ADP methods for uncertain affine nonlinear systems. To guarantee proper approximation of the value function and the control policy, neural networks are applied. Convergence and stability properties of the nonlinear ADP method are rigorously proved. It is shown that semi-global stabilization is attainable for a general class of continuous-time nonlinear systems, under the approximate optimal control policy.

Chapter 4 focuses on the theory of global adaptive dynamic programming (GADP). It aims at simultaneously improving the closed-loop system performance and achieving global asymptotic stability of the overall system at the origin. It is shown that the equality constraint used in policy evaluation can be relaxed to a sum-of-squares

(SOS) constraint. Hence, an SOS-based policy iteration is formulated, by relaxing the conventional policy iteration. In this new policy iteration algorithm, the control policy obtained at each iteration step is globally stabilizing. Similarly, the SOS-based policy iteration can be implemented online, without the need to identify the exact system dynamics.

Chapter 5 presents the new framework of RADP. In contrast to the ADP theory introduced in Chapters 2–4, RADP does not require all the state variables to be available, nor the system order assumed known. Instead, it incorporates a subsystem known as the dynamic uncertainty that interacts with a simplified reduced-order model. While ADP methods are performed on the reduced model, the interactions between the dynamic uncertainty and the simplified model are studied using tools borrowed from modern nonlinear system analysis and controller design. The learning objective in RADP is to achieve optimal performance of the reduced-order model in the absence of dynamic uncertainties, and maintain robustness of stability in the presence of the dynamic uncertainty.

Chapter 6 applies the RADP framework to solve the decentralized optimal control problem for a class of large-scale uncertain systems. In recent years, considerable attention has been paid to the stabilization of large-scale complex systems, as well as related consensus and synchronization problems. Examples of large-scale systems arise from ecosystems, transportation networks, and power systems. Often, in real-world applications, precise mathematical models are hard to build, and the model mismatch, caused by parametric and dynamic uncertainties, is thus unavoidable. This, together with the exchange of only local system information, makes the design problem challenging in the context of complex networks. In this chapter, the controller design for each subsystem only needs to utilize local state measurements without knowing the system dynamics. By integrating a simple version of the cyclic-small-gain theorem, asymptotic stability can be achieved by assigning appropriate nonlinear gains for each subsystem.

Chapter 7 studies sensorimotor control with static and dynamic uncertainties under the framework of RADP [18, 19, 21, 24]. The linear version of RADP is extended for stochastic systems by taking into account signal-dependent noise [13], and the proposed method is applied to study the sensorimotor control problem with both static and dynamic uncertainties. Results presented in this chapter suggest that the central nervous system (CNS) may use RADP-like learning strategy to coordinate movements and to achieve successful adaptation in the presence of static and/or dynamic uncertainties. In the absence of the dynamic uncertainties, the learning strategy reduces to an ADP-like mechanism.

All the numerical simulations in this book are developed using MATLAB® R2015a. Source code is available on the webpage of the book [16].

REFERENCES

[1] L. C. Baird. Reinforcement learning in continuous time: Advantage updating. In: Proceedings of the IEEE World Congress on Computational Intelligence, Vol. 4, pp. 2448–2453, Orlando, FL, 1994.

[2] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics*, 13(5):834–846, 1983.

[3] R. Bellman and S. Dreyfus. Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, 13(68):247–251, 1959.

[4] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[5] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, 4th ed. Athena Scientific, Belmont, MA, 2007.

[6] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Nashua, NH, 1996.

[7] S. Bhasin, N. Sharma, P. Patre, and W. Dixon. Asymptotic tracking by a reinforcement learning-based adaptive critic controller. *Journal of Control Theory and Applications*, 9(3):400–409, 2011.

[8] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, Cambridge, 2008.

[9] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst. *Reinforcement Learning and Dynamic Programming using Function Approximators*. CRC Press, 2010.

[10] T. Dierks and S. Jagannathan. Output feedback control of a quadrotor UAV using neural networks. *IEEE Transactions on Neural Networks*, 21(1):50–66, 2010.

[11] K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

[12] L. A. Feldkamp and D. V. Prokhorov. Recurrent neural networks for state estimation. In: Proceedings of the Twelfth Yale Workshop on Adaptive and Learning Systems, pp. 17–22, New Haven, CT, 2003.

[13] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.

[14] H. He, Z. Ni, and J. Fu. A three-network architecture for on-line learning and optimization based on adaptive dynamic programming. *Neurocomputing*, 78(1):3–13, 2012.

[15] R. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

[16] Y. Jiang and Z. P. Jiang. *RADPBook*. http://yu-jiang.github.io/radpbook/. Accessed May 1, 2015.

[17] Y. Jiang and Z. P. Jiang. Robust approximate dynamic programming and global stabilization with nonlinear dynamic uncertainties. In: Proceedings of the 50th IEEE Conference on Joint Decision and Control Conference and European Control Conference (CDC-ECC), pp. 115–120, Orlando, FL, 2011.

[18] Y. Jiang and Z. P. Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, 2012.

[19] Y. Jiang and Z. P. Jiang. Robust adaptive dynamic programming. In: D. Liu and F. Lewis, editors, *Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control*, Chapter 13, pp. 281–302. John Wiley & Sons, 2012.

[20] Y. Jiang and Z. P. Jiang. Robust adaptive dynamic programming for large-scale systems with an application to multimachine power systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 59(10):693–697, 2012.

[21] Y. Jiang and Z. P. Jiang. Robust adaptive dynamic programming with an application to power systems. *IEEE Transactions on Neural Networks and Learning Systems*, 24(7):1150–1156, 2013.

[22] Y. Jiang and Z. P. Jiang. Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):882–893, 2014.

[23] Y. Jiang and Z. P. Jiang. Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 60(11):2917–2929, November 2015.

[24] Z. P. Jiang and Y. Jiang. Robust adaptive dynamic programming for linear and nonlinear systems: An overview. *European Journal of Control*, 19(5):417–425, 2013.

[25] Z. P. Jiang and I. Mareels. A small-gain control method for nonlinear cascaded systems with dynamic uncertainties. *IEEE Transactions on Automatic Control*, 42(3):292–308, 1997.

[26] Z. P. Jiang and L. Praly. Design of robust adaptive controllers for nonlinear systems with dynamic uncertainties. *Automatica*, 34(7):825–840, 1998.

[27] Z. P. Jiang, A. R. Teel, and L. Praly. Small-gain theorem for ISS systems and applications. *Mathematics of Control, Signals and Systems*, 7(2):95–120, 1994.

[28] Y. H. Kim and F. L. Lewis. *High-Level Feedback Control with Neural Networks*. World Scientific, 1998.

[29] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica*, 50(4):1167–1175, 2014.

[30] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, New York, 1995.

[31] F. L. Lewis and D. Liu. *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. John Wiley & Sons, 2012.

[32] F. L. Lewis and K. G. Vamvoudakis. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(1):14–25, 2011.

[33] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, 9(3):32–50, 2009.

[34] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal Control*, 3rd ed. John Wiley & Sons, New York, 2012.

[35] D. Liu and D. Wang. Optimal control of unknown nonlinear discrete-time systems using iterative globalized dual heuristic programming algorithm. In: F. Lewis and L. Derong, editors, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, pp. 52–77. John Wiley & Sons, 2012.

[36] T. Liu, D. J. Hill, and Z. P. Jiang. Lyapunov formulation of ISS cyclic-small-gain in continuous-time dynamical networks. *Automatica*, 47(9):2088–2093, 2011.

[37] J. Mendel and R. McLaren. Reinforcement-learning control and pattern recognition systems. In: *A Prelude to Neural Networks*, pp. 287–318. Prentice Hall Press, 1994.

[38] M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[39] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani. Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 24(10):1513–1525, 2013.

[40] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks. Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 32(2):140–153, 2002.

[41] W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, New York, 2007.

[42] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Vol. 414. John Wiley & Sons, 2009.

[43] J. Si, A. G. Barto, W. B. Powell, and D. C. Wunsch (editors). *Handbook of Learning and Approximate Dynamic Programming*. John Wiley & Sons, Inc., Hoboken, NJ, 2004.

[44] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[45] E. D. Sontag. Input to state stability: Basic concepts and results. In: *Nonlinear and Optimal Control Theory*, pp. 163–220. Springer, 2008.

[46] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[47] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Cambridge University Press, 1998.

[48] C. Szepesvari. Reinforcement learning algorithms for MDPs. Technical Report TR09-13, Department of Computing Science, University of Alberta, Edmonton, CA, 2009.

[49] K. G. Vamvoudakis and F. L. Lewis. Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

[50] K. G. Vamvoudakis and F. L. Lewis. Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton–Jacobi equations. *Automatica*, 47(8):1556–1569, 2011.

[51] K. G. Vamvoudakis and F. L. Lewis. Online solution of nonlinear two-player zero-sum games using synchronous policy iteration. *International Journal of Robust and Nonlinear Control*, 22(13):1460–1483, 2012.

[52] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 45(2):477–484, 2009.

[53] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis. *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. IET, London, 2013.

[54] M. Waltz and K. Fu. A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control*, 10(4):390–398, 1965.

[55] F.-Y. Wang, H. Zhang, and D. Liu. Adaptive dynamic programming: An introduction. *IEEE Computational Intelligence Magazine*, 4(2):39–47, 2009.

[56] C. Watkins. Learning from delayed rewards. PhD Thesis, King's College of Cambridge, 1989.

[57] Q. Wei and D. Liu. Data-driven neuro-optimal temperature control of water gas shift reaction using stable iterative adaptive dynamic programming. *IEEE Transactions on Industrial Electronics*, 61(11):6399–6408, November 2014.

[58] P. Werbos. The elements of intelligence. *Cybernetica (Namur)*, (3), 1968.

[59] P. Werbos. Advanced forecasting methods for global crisis warning and models of intelligence. *General Systems Yearbook*, 22:25–38, 1977.

[60] P. Werbos. Reinforcement learning and approximate dynamic programming (RLADP) – Foundations, common misconceptions and the challenges ahead. In: F. L. Lewis and D. Liu, editors, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, pp. 3–30. John Wiley & Sons, Hoboken, NJ, 2013.

[61] P. J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD Thesis, Harvard University, 1974.

[62] P. J. Werbos. Neural networks for control and system identification. In: Proceedings of the 28th IEEE Conference on Decision and Control, pp. 260–265, Tampa, FL, 1989.

[63] P. J. Werbos. A menu of designs for reinforcement learning over time. In: W. Miller, R. Sutton, and P. Werbos, editors, *Neural Networks for Control*, pp. 67–95. MIT Press, Cambridge, MA, 1990.

[64] P. J. Werbos. Approximate dynamic programming for real-time control and neural modeling. In: D. White and D. Sofge, editors, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, pp. 493–525. Van Nostrand Reinhold, New York, 1992.

[65] P. J. Werbos. From ADP to the brain: Foundations, roadmap, challenges and research priorities. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 107–111, Beijing, 2014. doi: 10.1109/IJCNN.2014.6889359

[66] H. Xu, S. Jagannathan, and F. L. Lewis. Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses. *Automatica*, 48(6):1017–1030, 2012.

[67] X. Xu, C. Wang, and F. L. Lewis. Some recent advances in learning and adaptation for uncertain feedback control systems. *International Journal of Adaptive Control and Signal Processing*, 28(3–5):201–204, 2014.

[68] H. Zhang, D. Liu, Y. Luo, and D. Wang. *Adaptive Dynamic Programming for Control*. Springer, London, 2013.

[69] H. Zhang, Q. Wei, and D. Liu. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. *Automatica*, 47(1):207–214, 2011.

[70] X. Zhang, H. He, H. Zhang, and Z. Wang. Optimal control for unknown discrete-time nonlinear Markov jump systems using adaptive dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems*, 25(12):2141–2155, 2014.