Studies in Systems, Decision and Control 166

Ruizhuo Song
Qinglai Wei
Qing Li

Adaptive Dynamic Programming: Single and Multiple Controllers

Science Press
Beijing

Springer

Studies in Systems, Decision and Control

Volume 166

Series editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

The series “Studies in Systems, Decision and Control” (SSDC) covers both new developments and advances, as well as the state of the art, in the various areas of broadly perceived systems, decision making and control–quickly, up to date and with a high quality. The intent is to cover the theory, applications, and perspectives on the state of the art and future developments relevant to systems, decision making, control, complex processes and related areas, as embedded in the fields of engineering, computer science, physics, economics, social and life sciences, as well as the paradigms and methodologies behind them. The series contains monographs, textbooks, lecture notes and edited volumes in systems, decision making and control spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/13304

Ruizhuo Song · Qinglai Wei
Qing Li

Adaptive Dynamic Programming: Single and Multiple Controllers

Science Press
Beijing

Springer

Ruizhuo Song
University of Science and Technology Beijing
Beijing, China

Qinglai Wei
Institute of Automation
Chinese Academy of Sciences
Beijing, China

Qing Li
University of Science and Technology Beijing
Beijing, China

ISSN 2198-4182 ISSN 2198-4190 (electronic)
Studies in Systems, Decision and Control
ISBN 978-981-13-1711-8 ISBN 978-981-13-1712-5 (eBook)
https://doi.org/10.1007/978-981-13-1712-5

Jointly published with Science Press, Beijing, China

The print edition is not for sale in China Mainland. Customers from China Mainland please order the print book from: Science Press, Beijing.

Library of Congress Control Number: 2018949333

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

In the past few decades, neurobiological discoveries towards understanding learning mechanisms of the human brain have provided indications of how to design more effective and responsive decision and control systems for complex interconnected human-engineered systems. Advances in computational intelligence reinforcement learning (RL) methods, particularly those with an actor-critic structure in the family of approximate/adaptive dynamic programming (ADP) techniques, have made inroads in comprehending and mimicking brain functionality at the level of the brainstem, cerebellum, basal ganglia, and cerebral cortex. The works of Doya and others have shown that the basal ganglia acts as a critic in selecting action commands sent to the muscle motor control systems. The concentration of dopamine neurotransmitters in the basal ganglia is modified based on rewards received from previous actions, so that successful actions are more likely to be repeated in the future. The cerebellum learns a dynamics model of environments for control purposes. This motivates a three-level actor-critic structure with learning networks for critic, actor, and model dynamics. Based on this actor-critic structure, the ADP algorithms which are state feedback control methods have been developed as powerful tools for solving optimal control problems online.

ADP techniques are used to design new structures of adaptive feedback control systems that learn the solutions to optimal control problems by measuring data along the system trajectories. This book studies the optimal control based on ADP for two categories of dynamic feedback control systems: systems with single control input and systems with multiple control inputs.

This book is organized into 13 chapters. First, Chap. 1 gives a preparation of the book. After that, the book is divided into two parts. In Part I, we develop novel ADP methods for optimal control of the systems with one control input. Chapter 2 introduces a finite-time optimal control method for a class of unknown nonlinear systems. Chapter 3 proposes finite-horizon optimal control for a class of nonaffine time-delay nonlinear systems. Chapter 4 presents multi-objective optimal control for a class of nonlinear time-delay systems. Chapter 5 develops multiple actor-critic

structures for continuous-time optimal control using input–output data. In Chap. 6, complex-valued nonlinear system optimal control method is studied based on ADP. In Chap. 7, off-policy neuro-optimal control is developed for unknown complex-valued nonlinear systems based on policy iteration. In Chap. 8, optimal tracking control is proposed with the convergence proof. Part II concerns multi-player systems from Chaps. 9 to 13. In Chap. 9, off-policy actor-critic structure for optimal control of unknown systems with disturbances is presented. In Chap. 10, iterative ADP method for solving a class of nonlinear zero-sum differential games is introduced. In Chap. 11, neural network-based synchronous iteration learning method for multi-player zero-sum games is developed. In Chap. 12, off-policy integral reinforcement learning (IRL) method is also employed to solve nonlinear continuous-time multi-player non-zero-sum games. In Chap. 13, optimal distributed synchronization control for continuous-time heterogeneous multi-agent differential graphical games is established.

Dr. Ruizhuo Song completed Chaps. 1–12 and wrote 250,000 words. Dr. Qinglai Wei completed Chap. 13 and wrote 50,000 words. The authors would like to acknowledge the help and encouragement they have received from colleagues in the University of Science and Technology Beijing, and Institute of Automation, Chinese Academy of Sciences during the course of writing this book.

The authors are very grateful to the National Natural Science Foundation of China (NSFC) for providing necessary financial support to our research. The present book is the result of NSFC Grants 61374105, 61673054, 61722312, and in part by the Fundamental Research Funds for the Central Universities under Grant FRF-GF-17-B45.

This book is dedicated to Tiantian, who makes every day exciting.

Beijing, China
October 2017

Ruizhuo Song
Qinglai Wei
Qing Li

Contents

About the Authors

Ruizhuo Song Ph.D., Associate Professor, School of Automation and Electrical Engineering, University of Science and Technology Beijing.

She received her Ph.D. in control theory and control engineering from Northeastern University, Shenyang, China, in 2012. She was a Postdoctoral Fellow at the University of Science and Technology Beijing, Beijing, China. She is currently an Associate Professor at the School of Automation and Electrical Engineering, University of Science and Technology Beijing. She was a Visiting Scholar in the Department of Electrical Engineering at the University of Texas at Arlington, Arlington, TX, USA, from 2013 to 2014. She was a Visiting Scholar in the Department of Electrical, Computer, and Biomedical Engineering at the University of Rhode Island, Kingston, RI, USA, from January 2018 to February 2018. Her current research interests include optimal control, multi-player games, neural network-based control, nonlinear control, wireless sensor networks, and adaptive dynamic programming and their industrial application. She has published over 40 journal and conference papers and co-authored two monographs.

Qinglai Wei Ph.D., Professor, The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences.

He received the B.S. degree in Automation, and the Ph.D. degree in control theory and control engineering, from the Northeastern University, Shenyang, China, in 2002 and 2009, respectively. From 2009–2011, he was a postdoctoral fellow with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently a professor of the institute. He has authored three books, and published over 80 international journal papers. His research interests include adaptive dynamic programming, neural network-based control, computational intelligence, optimal control, nonlinear systems and their industrial applications.

Dr. Wei is an Associate Editor of IEEE Transaction on Automation Science and Engineering since 2017, IEEE Transaction on Consumer Electronics since 2017, Control Engineering (in Chinese) since 2017, IEEE Transactions on Cognitive and

Developmental Systems since 2017, IEEE Transaction on Systems Man, and Cybernetics: Systems since 2016, Information Sciences since 2016, Neurocomputing since 2016, Optimal Control Applications and Methods since 2016, Acta Automatica Sinica since 2015, and has been holding the same position for IEEE Transactions on Neural Networks and Learning Systems during 2014–2015. He is the Secretary of IEEE Computational Intelligence Society (CIS) Beijing Chapter since 2015. He was the Program Chair of The 14th International Symposium on Neural Networks (ISNN 2017), Program Co-Chair of The 24th International Conference on Neural Information Processing (ICONIP 2017), Registration Chair of the 12th World Congress on Intelligent Control and Automation (WCICA 2016), 2014 IEEE World Congress on Computational Intelligence (WCCI 2014), the 2013 International Conference on Brain Inspired Cognitive Systems (BICS 2013), and the Eighth International Symposium on Neural Networks (ISNN 2011). He was the Publication Chair of 5th International Conference on Information Science and Technology (ICIST 2015) and the Ninth International Symposium on Neural Networks (ISNN 2012). He was the Finance Chair of the 4th International Conference on Intelligent Control and Information Processing (ICICIP 2013) and the Publicity Chair of the 2012 International Conference on Brain Inspired Cognitive Systems (BICS 2012). He was guest editors for Neual Computing and Applications and Neurocomuting in 2013 and 2014, respectively. He was a recipient of Shuang-Chuang Talents in Jiangsu Province, China, in 2014. He was a recipient of the Outstanding Paper Awards of IEEE Transactions on Neural Network and Learning Systems in 2018, Acta Automatica Sinica in 2011, Zhang Siying Outstanding Paper Award of Chinese Control and Decision Conference (CCDC) in 2015 and Best Paper Award of IEEE 6th Data Driven Control and Learning Systems Conference (DDCLS) in 2017. He was a recipient of Young Researcher Award of Asia Pacific Neural Network Society (APNNS) in 2016. He was recipient of Young Scientist Award, Yang Jiachi Science and Technology Awards (Second Class Prize), and Natural Science Award (First Prize) in Chinese Association of Automation (CAA) in 2017. He was the PI for 13 national and local government projects.

Qing Li Ph.D., received his B.E. from North China University of Science and Technology, Tangshan, China, in 1993, and the Ph.D. in control theory and its applications from University of Science and Technology Beijing, Beijing, China, in 2000. He is currently a Professor at the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. He has been a Visiting Scholar at Ryerson University, Toronto, Canada, from February 2006 to February 2007. His research interests include intelligent control and intelligent optimization.

Symbols

x	State vector
u	Control vector
F	System function
i	Index
$\mathbb{R}^n$	State space
Ω	State set
J, V	Performance index functions
U	Utility function
J^*	Optimal performance index function
u^*	Law of optimal control
N	Terminal time
W	Weight matrix between the hidden layer and output layer
Q, R	Positive definite matrices
α, β	Learning rate
H	Hamiltonian function
e_c	Estimation error
E_c	Squared residual error

Chapter 1
Introduction

1.1 Optimal Control

Optimal control is one particular branch of modern control. It deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A control problem includes a cost functional that is a function of state and control variables. An optimal control is a set of differential equations describing the paths of the control variables that minimize the cost function. The optimal control can be derived using Pontryagin's maximum principle (a necessary condition also known as Pontryagin's minimum principle or simply Pontryagin's Principle), or by solving the Hamilton–Jacobi–Bellman (HJB) equation (a sufficient condition). For linear systems with quadratic performance function, the HJB equation reduces to the algebraic Riccati equation (ARE) [1].

Dynamic programming is based on Bellman's principle of optimality: an optimal (control) policy has the property that no matter what previous decisions have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions. Dynamic programming is a very useful tool in solving optimization and optimal control problems.

In the next, we will introduce optimal control problems for continuous-time and discrete-time linear systems.

1.1.1 Continuous-Time LQR

A special case of the general optimal control problem given is the linear quadratic regulator (LQR) optimal control problem [2]. The LQR considers the linear time invariant dynamical system described by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1.1}$$

R. Song et al., *Adaptive Dynamic Programming: Single and Multiple Controllers*, Studies in Systems, Decision and Control 166,
https://doi.org/10.1007/978-981-13-1712-5_1

with state $x(t) \in \mathbb{R}^n$ and control input $u(t) \in \mathbb{R}^m$. To this system is associated the infinite-horizon quadratic cost function or performance index

$$V(x(t_0), t_0) = \frac{1}{2}\int_{t_0}^{\infty} \{x^{\mathrm{T}}(t)Qx(t) + u^{\mathrm{T}}(t)Ru(t)\}\mathrm{d}\tau \tag{1.2}$$

with weighting matrices $Q > 0$ and $R > 0$. It is assumed that (A, B) is stabilizable, $(A, \sqrt{Q})$ is detectable.

The LQR optimal control problem requires finding the control policy that minimizes the cost (1.2):

$$u^*(t) = \arg\min_{u(t)} V(t_0, x(t_0), u(t)). \tag{1.3}$$

The solution of this optimal control problem is given by the state-feedback

$$u(t) = -Kx(t) \tag{1.4}$$

where $K = R^{-1}B^{\mathrm{T}}P$, and matrix P is a positive definite solution of the algebraic Riccati equation (ARE)

$$A^{\mathrm{T}}P + PA + Q - PBR^{-1}B^{\mathrm{T}}P = 0. \tag{1.5}$$

Under the stabilizability and detectability conditions, there is an unique positive semidefinite solution of the ARE that yields a stabilizing closed-loop controller given by (1.4). That is, the closed-loop system is asymptotically stable.

1.1.2 Discrete-Time LQR

Consider the discrete-time LQR problem where the dynamical system described by

$$x(k+1) = Ax(k) + Bu(k) \tag{1.6}$$

with k the discrete time index. The associated infinite horizon performance index has deterministic stage costs and is

$$V(k) = \frac{1}{2}\sum_{i=k}^{\infty} \left\{x^{\mathrm{T}}(i)Qx(i) + u^{\mathrm{T}}(i)Ru(i)\right\}. \tag{1.7}$$

The value function for a fixed policy depends only on the initial state $x(k)$. A difference equation equivalent to this infinite sum is given by

$$V(x(k)) = \frac{1}{2}(x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k)) + V(x(k+1)). \tag{1.8}$$

Assuming the value is quadratic in the state so that

$$V(x(k)) = \frac{1}{2}x^{\mathrm{T}}(k)Px(k) \tag{1.9}$$

for some kernel matrix P yields the Bellman equation form

$$\begin{aligned} x^{\mathrm{T}}(k)Px(k) =& x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k) \\ &+ (Ax(k) + Bu(k))^{\mathrm{T}}P(Ax(k) + Bu(k)). \end{aligned} \tag{1.10}$$

Assuming a constant state feedback policy $u(k) = -Kx(k)$ for some stabilizing gain K, we write

$$(A - BK)^{\mathrm{T}}P(A - BK) - P + Q + K^{\mathrm{T}}RK = 0. \tag{1.11}$$

This is a Lyapunov equation.

In the above, the solution methods for linear system optimal control problems are given. However, it is often computationally untenable to run true dynamic programming due to the backward numerical process required for its solutions, i.e., as a result of the well-known "curse of dimensionality".

1.2 Adaptive Dynamic Programming

Reinforcement learning (RL) is a type of machine learning developed in the computational intelligence community in computer science and engineering. It has been extensively used to solve optimal control problems and it is a computational approach to learning from interactions with the surrounding environment and concerned with how an agent or actor ought to take actions so as to optimize a cost of its long-term interactions with the environment. In the context of control, the environment is the dynamic system, the agent corresponds to the controller, and actions correspond to control signals. The RL objective is to find a strategy that minimizes an expected long-term cost.

One type of reinforcement learning algorithms employs the actor-critic structure shown in Fig. 1.1, which is also used in [3]. This structure produces forward-in-time algorithms that are implemented in real time wherein an actor component applies an action, or control policy, to the environment, and a critic component assesses the value of that action. The learning mechanism supported by the actor-critic structure has two steps: policy evaluation and executing by the critic. The policy evaluation step is performed by observing from the environment the results of applying current actions. Performance or value can be defined in terms of optimality objectives such as

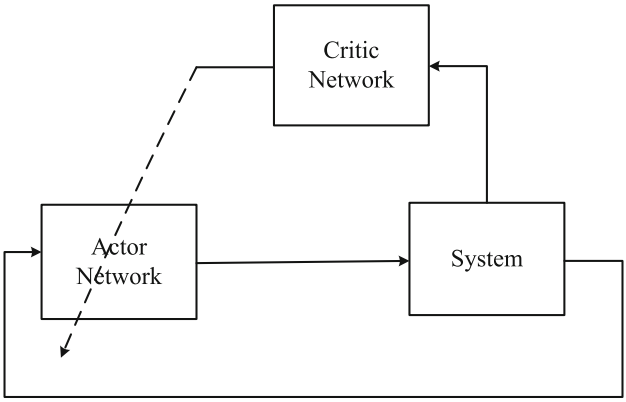

Fig. 1.1 Reinforcement learning with an actor/critic structure

minimum fuel, minimum energy, minimum risk, or maximum reward. Based on the assessment of the performance, one of several schemes can then be used to modify or improve the control policy in the sense that the new policy yields a value that is improved relative to the previous value. In this scheme, reinforcement learning is a means of learning optimal behaviors by observing the real-time responses from the environment to nonoptimal control policies.

Werbos developed actor-critic techniques for feedback control of discrete-time dynamical systems that learn optimal policies online in real time using data measured along the system trajectories [4–7]. These methods, known as approximate dynamic programming or adaptive dynamic programming (ADP), comprise a family of the basic learning methods: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. It builds critic to approximate the cost function and actor to approximate the optimal control in dynamic programming using a function approximation structure such as neural networks (NNs) [8].

Based on the Bellman equation for solving optimal decision problems in real time forward in time according to data measured along the system trajectories, ADP algorithm as an effective intelligent control method has played an important role in seeking solutions for the optimal control problem [9]. Now, ADP has two main iteration forms, namely policy iteration (PI) and value iteration (VI), respectively [10]. PI algorithms contain policy evaluation and policy improvement. An initial stabilizing control law is required, which is often difficult to obtain. Comparing to VI algorithms, in most applications, PI would require fewer iterations as a Newtons method, but every iteration is more computationally demanding. VI algorithms solve the optimal control problem without requirement of an initial stabilizing control law, which is easy to implement. For system (1.6) and the cost function (1.7), the detailed procedures of PI and VI are given as follows.

1. PI Algorithm

(1) Initialize. Select any admissible (i.e. stabilizing) control policy $h^{[0]}(x(k))$

(2) Policy Evaluation Step. Determine the value of the current policy using the Bellman Equation

$$V^{[i+1]}(x(k)) = r(x(k), h^{[i]}(x(k))) + V^{[i+1]}(x(k+1)). \tag{1.12}$$

(3) Policy Improvement Step. Determine an improved policy using

$$h^{[i+1]}(x(k)) = \arg\min_{h(\cdot)} \left\{ r(x(k), h(x(k))) + V^{[i+1]}(x(k+1)) \right\}. \tag{1.13}$$

2. VI Algorithm

(1) Initialize. Select any control policy $h^{[0]}(x(k))$, not necessarily admissible or stabilizing.

(2) Value Update Step. Update the value using

$$V^{[i+1]}(x(k)) = r(x(k), h^{[i]}(x(k))) + V^{[i]}(x(k+1)). \tag{1.14}$$

(3) Policy Improvement Step. Determine an improved policy using

$$h^{[i+1]}(x(k)) = \arg\min_{h(\cdot)} \left\{ r(x(k), h(x(k))) + V^{[i+1]}(x(k+1)) \right\}. \tag{1.15}$$

1.3 Review of Matrix Algebra

In this book, some matrix manipulations are the basic mathematical vehicle and, for those whose memory needs refreshing, we provides a short review.

1. For any $n \times n$ matrices A and B, $(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}$.

2. For any $n \times n$ matrices A and B, if A and B are nonsingular, then $(AB)^{-1} = B^{-1}A^{-1}$.

3. The Kronecker product of two matrices $A = [a_{ij}] \in \mathbb{C}^{m\times n}$ and $B = [b_{ij}] \in \mathbb{C}^{p\times q}$ is $A \otimes B = [a_{ij}B] \in \mathbb{C}^{mp\times nq}$.

4. If $A = [a_1, a_2, \ldots, a_n] \in \mathbb{C}^{m\times n}$, where a_i are the columns of A, the stacking operator is defined by $s(A) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$. It converts $A \in \mathbb{C}^{m\times n}$ into a vector $s(A) \in \mathbb{C}^{mn}$. Then for matrices A, B and D we have

$$s(ABD) = (D^{\mathrm{T}} \otimes A)s(B). \tag{1.16}$$

5. If $x \in \mathbb{R}^n$ is a vector, then the square of the Euclidean norm is $||x||^2 = x^{\mathrm{T}}x$.

6. If Q is symmetric, then $\dfrac{\partial}{\partial x}(x^{\mathrm{T}}Qx) = 2Qx$.

References

1. Zhang, H., Liu, D., Luo, Y., Wang, D.: Adaptive Dynamic Programming for Control-Algorithms and Stability. Springer, London (2013)
2. Vrabie, D., Vamvoudakis, K., Lewis, F.: Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles. The Institution of Engineering and Technology, London (2013)
3. Barto, A., Sutton, R., Anderson, C.: Neuron-like adaptive elements that can solve difficult learning control problems. IEEE Trans. Syst. Man Cybern. Part B, Cybern. **SMC–13**(5), 834–846 (1983)
4. Werbos, P.: A menu of designs for reinforcement learning over time. In: Miller, W.T., Sutton, R.S., Werbos, P.J. (eds.) Neural Networks for Control, pp. 67–95. MIT Press, Cambridge (1991)
5. Werbos, P.: Approximate dynamic programming for real-time control and neural modeling. In: White, D.A., Sofge, D.A. (eds.) Handbook of Intelligent Control. Van Nostrand Reinhold, New York (1992)
6. Werbos, P.: Neural networks for control and system identification. In: proceedings of IEEE Conference Decision Control, Tampa, FL, pp. 260–265 (1989)
7. Werbos, P.: Advanced forecasting methods for global crisis warning and models of intelligence. General Syst. Yearbook **22**, 25–38 (1977)
8. Liu, D., Wei, Q., Yang, X., Li, H., Wang, D.: Adaptive Dynamic Programming with Applications in Optimal Control. Springer International Publishing, Berlin (2017)
9. Werbos, P.: ADP: the key direction for future research in intelligent control and understanding brain intelligence. IEEE Trans. Syst. Man Cybern. Part B: Cybern. **38**(4), 898–900 (2008)
10. Lewis, F., Vrabie, D.: Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst. Mag. **9**(3), 32–50 (2009)

Chapter 2
Neural-Network-Based Approach for Finite-Time Optimal Control

This chapter proposes a novel finite-time optimal control method based on input-output data for unknown nonlinear systems using ADP algorithm. In this method, the single-hidden layer feed-forward network (SLFN) with extreme learning machine (ELM) is used to construct the data-based identifier of the unknown system dynamics. Based on the data-based identifier, the finite-time optimal control method is established by ADP algorithm. Two other SLFNs with ELM are used in ADP method to facilitate the implementation of the iterative algorithm, which aim to approximate the performance index function and the optimal control law at each iteration, respectively. A simulation example is provided to demonstrate the effectiveness of the proposed control scheme.

2.1 Introduction

The linear optimal control problem with a quadratic cost function is probably the most well-known control problem [1, 2], and it can be translated into Riccati equation. While the optimal control of nonlinear systems is usually a challenging and difficult problem [3, 4]. Furthermore, comparing with the known system dynamics case, it is more intractable to solve the optimal control problem of the unknown system dynamics. Generally speaking, most actual systems are nearly far too complex to present the perfect mathematical models. Whenever no model is available to design the system controller nor is easy to produce, a standard way is resorting to data-based techniques [5]: (1) on the basis of input-output data, the model of the unknown system dynamics is identified; (2) on the basis of the estimated model of the system dynamics, the controller is designed by model-based design techniques.

It is well known that neural network is an effective tool to implement intelligent identification based on input-output data, due to the properties of nonlinearity, adaptivity, self-learning and fault tolerance [6–10]. In which, SLFN is one of the most

R. Song et al., *Adaptive Dynamic Programming: Single and Multiple Controllers*, Studies in Systems, Decision and Control 166,
https://doi.org/10.1007/978-981-13-1712-5_2

useful types [11]. In [12], Hornik proved that if the activation function is continuous, bounded, and non-constant, then continuous mappings can be approximated by SLFN with additive hidden nodes over compact input sets. In [13], Leshno improved the results of [12] and proved that SLFN with additive hidden nodes and with a non-polynomial activation function can approximate any continuous target functions. In [11], it is proven in theory that SLFN with randomly generated additive and a broad type of activation functions can universally approximate any continuous target functions in any compact subset of the Euclidean space. For SLFN training, there are three main approaches: (1) gradient-descent based, for example back-propagation (BP) method [14]; (2) least square based, for example ELM method in this chapter; (3) standard optimization method based, for example support vector machine (SVM). While, the learning speed of feed-forward neural networks is in general far slower than required and it has been a major bottleneck in their applications for past decades [15]. Two key reasons are: (1) the slow gradient-based learning algorithms are extensively used to train neural networks, (2) all the parameters of the networks are tuned iteratively by using such learning algorithms. Unlike conventional neural network theories, in this chapter, the ELM method is used to train SLFN. Such SLFN can be as the universal approximator, one may simply randomly choose hidden nodes and then only need to adjust the output weights linking the hidden layer and the output layer. For given network architectures, ELM does not require human-intervened parameters, so ELM has fast convergence and can be easily used.

Based on the SLFN identifier, the finite-time optimal control method is presented in this chapter. For finite-time control problems, the system must be stabilized to zero within finite time. The controller design of finite-time problems still presents a challenge to control engineers as the lack of methodology and the control step is difficult to determine. Few results relate to the finite-time optimal control based on ADP algorithm. As we know that [16] solved the finite-horizon optimal control problem for a class of discrete-time nonlinear systems using ADP algorithm. But the method in [16] adopts the BP networks to obtain the optimal control, which has slow convergence speed.

In this chapter, we will design the finite-time optimal controller based on SLFN with ELM for unknown nonlinear systems. First, the identifier is established by the input-output data. It is proven that the identification error converges to zero. Upon the data-based identifier, the optimal control method is proposed. We prove that the iterative performance index function converges to optimum, and the optimal control is also obtained. Compared to other popular implementation methods such as BP, the SLFN with ELM has the fast response speed and is fully automatic. It means that except for target errors and the allowed maximum number of hidden nodes, no control parameters need to be manually tuned by users.

The rest of this chapter is organized as follows. In Sect. 2.2, the problem formulation is presented. In Sect. 2.3, the identifier is developed based on the input-output data. In Sect. 2.4, the iterative ADP algorithm and the convergence proof are given. In Sect. 2.6, an example is given to demonstrate the effectiveness of the proposed control scheme. In Sect. 2.7, the conclusion is drawn.

2.2 Problem Formulation and Motivation

Consider the following unknown discrete-time nonlinear systems

$$x(k+1) = F(x(k), u(k)), \tag{2.1}$$

where the state $x(k) \in \mathbb{R}^n$ and the control $u(k) \in \mathbb{R}^m$. $F(x(k), u(k))$ is unknown continuous function. Assume that the state is completely controllable and bounded on Ω, and $F(0, 0) = 0$. The finite-time performance index function is defined as follows:

$$J(x(k), U(k, K)) = \sum_{i=k}^{K} \{x^{\mathrm{T}}(i) Q x(i) + u^{\mathrm{T}}(i) R u(i)\} \tag{2.2}$$

where Q and R are positive definite matrices, K is the finite positive integer, the control sequence $U(k, K) = (u(k), u(k+1), \ldots, u(K))$ is finite-time admissible [16]. The length of $U(k, K)$ is defined as $(K - k + 1)$.

This chapter is desired to find the optimal control for system (2.1) based on performance index function (2.2). Since the system dynamics is completely unknown, the optimal problem cannot be solved directly. Therefore, it is desirable to propose a novel method that does not need the exact system dynamics but only the input-output data, which can be obtained during the operation of the system. In this chapter, we propose a data-based optimal control scheme using SLFN with ELM and ADP method for general unknown nonlinear systems. The design of proposed controller is divided into two steps:

(1) The unknown nonlinear system dynamics is identified by SLFN identification scheme with convergence proof.
(2) The optimal controller is designed based on the data-based identifier.

In the following sections, we will discuss the establishment of the data-based identifier and the controller design in details.

2.3 The Data-Based Identifier

In this section, the ELM method is introduced and the data-based identifier is established with convergence proof. The structure of SLFN is in Fig. 2.1.

For N_1 arbitrary distinct samples $(\bar{x}(i), \bar{y}(i))$, where $\bar{x}(i) \in \mathbb{R}^{n_1}$, $\bar{y}(i) \in \mathbb{R}^{m_1}$, $i = 1, 2, \ldots, N_1$. The weight vectors between the input neurons and the jth hidden neuron are $w_j \in \mathbb{R}^{n_2}$. The weight vectors between the output neurons and the jth hidden neuron are $\bar{\beta}_j \in \mathbb{R}^{n_3}$, which will be designed by ELM method [17]. The number of hidden neurons is L. The threshold of the jth hidden neuron is b_j. The hidden layer activation function $g_L(\bar{x})$ is infinitely differentiable, then the

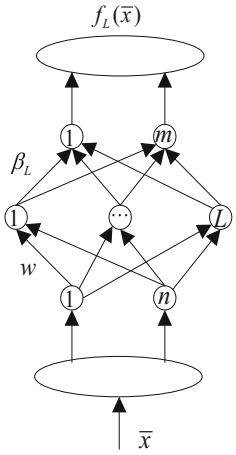

Fig. 2.1 The basic SLFN architecture

mathematically model of SLFN is [15]

$$f_L(\bar{x}(i)) = \sum_{j=1}^{L} \bar{\beta}_j g_L(w_j^{\mathrm{T}} \bar{x}(i) + b_j), i = 1, 2, \ldots, N_1. \tag{2.3}$$

Unlike the traditional popular implementations of SLFN, in this chapter, ELM is used to adjust the output weights. In theory, Refs. [18, 19] show that the input weights and hidden neurons biases of SLFN do not need be adjusted during training and one may simply randomly assign values to them. To be convenient for explanation, let $\beta_L = [\bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_L]^{\mathrm{T}}_{L \times m_1}$, $\bar{Y} = [\bar{y}(1), \bar{y}(2), \ldots, \bar{y}(N_1)]^{\mathrm{T}}_{N_1 \times m_1}$, and

$$\begin{aligned} H &= [h(\bar{x}(1)), h(\bar{x}(2)), \ldots, h(\bar{x}(N_1))]^{\mathrm{T}} \\ &= \begin{bmatrix} G(w_1, b_1, \bar{x}(1)) \ldots G(w_L, b_L, \bar{x}(1)) \\ \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \\ G(w_1, b_1, \bar{x}(N_1)) \ldots G(w_L, b_L, \bar{x}(N_1)) \end{bmatrix}_{N_1 \times L}, \end{aligned}$$

where $G(w_j, b_j, \bar{x}(i)) = g_L(w_j^{\mathrm{T}} \bar{x}(i) + b_j)$. So we have

$$H\beta_L = \bar{Y}. \tag{2.4}$$

Based on least-square method, it can be obtained that

$$\beta_L = H^{+} \bar{Y}, \tag{2.5}$$

where $H^{+} = (H^{\mathrm{T}} H)^{-1} H^{\mathrm{T}}$.

For SLEN in (2.3), the output weight β_L is the only value we want to obtain. In the following, a theorem is given to prove that β_L exists, which means that H is invertible.

Theorem 2.1 *If SLFN is defined as in (2.3), let the hidden neurons number is L. For N_1 arbitrary distinct input samples $\bar{x}(i)$ and any given w_j and b_j, we have H in (2.4) is invertible.*

Proof As $\bar{x}(i)$ are distinct, for any vector w_j according to any continuous probability distribution, then with probability one, $w_j^{\mathrm{T}}x(1)$, $w_j^{\mathrm{T}}x(2)$, ..., $w_j^{\mathrm{T}}x(N_1)$ are different from each other. Define the jth column of H is $c(j) = [g_L(w_j^{\mathrm{T}}\bar{x}(1) + b_j), g_L(w_j^{\mathrm{T}}\bar{x}(2) + b_j), \ldots, g_L(w_j^{\mathrm{T}}\bar{x}(N_1) + b_j)]^{\mathrm{T}}$, we can have $c(j)$ does not belong to any subspace whose dimension is less than N_1 [19]. It means that for any given w_j and b_j, according to any continuous probability distribution, H in (2.4) can be made full-rank, i.e., H is invertible.

Therefore, the SLFN with ELM method is summarized as follows [20]:

Step 1. Given a training set $(\bar{x}(i), \bar{y}(i))$, $i = 1, 2, \ldots, N_1$, hidden node output function $G(w_j, b_j, \bar{x}(i))$ and hidden node number L.

Step 2. Given arbitrary hidden node parameters (w_j, b_j), $j = 1, 2, \ldots, L$.

Step 3. Calculate the hidden layer output matrix H.

Step 4. According to (2.5) to calculate β_L.

Remark 2.1 ELM algorithm can work with wide type of activation functions, such as sigmoidal functions, radial basis, sine, cosine and exponential functions et.al.. The feed-forward networks with arbitrarily assigned input weights and hidden layer biases can universally approximate any continuous functions on any compact input sets [21].

Remark 2.2 It is important to point out that β_L in (2.5) has the smallest norm among all the least-squares solutions of $H\beta_L = \bar{Y}$. As the input weights and hidden neurons biases of SLFN are simply randomly assigned values. So training an SLFN is simply equivalent to finding a least-squares solution of the linear system $H\beta_L = \bar{Y}$. Although almost all learning algorithms wish to reach the minimum training error, however, most of them cannot reach it because of local minimum or infinite training iteration is usually not allowed in applications [21]. Fortunately, the special unique solution β_L in (2.5) has the smallest norm among all the least squares solutions.

2.4 Derivation of the Iterative ADP Algorithm with Convergence Analysis

For the unknown nonlinear system (2.1), the data-based identifier is established. Then we can design the iterative ADP algorithm to get the solution of the finite-time optimal control problem.

First, the derivations of the optimal control $u^*(k)$ and $J^*(x(k))$ are given in details. It is known that, for the case of finite horizon optimization, the optimal performance index function $J^*(x(k))$ satisfies [16]

$$J^*(x(k)) = \inf_{U(k)}\{J(x(k), U(k, K))\}, \tag{2.6}$$

where $U(k, K)$ stands for a finite-time control sequence. The length of the control sequence is not assigned.

According to Bellman's optimality principle, the following Hamilton–Jocabi–Bellman (HJB) equation

$$J^*(x(k)) = \inf_{u(k)}\{x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k) + J^*(x(k+1))\} \tag{2.7}$$

holds.

Define the law of optimal control sequence starting at k by

$$U^*(k, K) = \arg\inf_{U(k,K)}\{J(x(k), U(k, K))\}, \tag{2.8}$$

and the law of optimal control vector by

$$u^*(k) = \arg\inf_{u(k)}\{x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k) + J^*(x(k+1))\}. \tag{2.9}$$

Therefore, we can have

$$J^*(x(k)) = x^{\mathrm{T}}(k)Qx(k) + u^{*\mathrm{T}}(k)Ru^*(k) + J^*(x(k+1)). \tag{2.10}$$

Based on the above preparation, the finite-time ADP method for unknown system is proposed. The iterative procedure is as follows.

For the iterative step $i = 1$, the performance index function is computed as

$$\begin{aligned} V^{[1]}(x(k)) &= \inf_{u(k)}\{x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k) + V^{[0]}(x(k+1))\} \\ &= x^{\mathrm{T}}(k)Qx(k) + u^{[1]\mathrm{T}}(k)Ru^{[1]}(k) + V^{[0]}(x(k+1)) \end{aligned} \tag{2.11}$$

where

$$u^{[1]}(x(k)) = \arg\inf_{u(k)}\{x^{\mathrm{T}}(k)Qx(k) + u^{\mathrm{T}}(k)Ru(k) + V^{[0]}(x(k+1))\}, \tag{2.12}$$

and $V^{[0]}(x(k+1))$ has two expression forms according to two different cases.

If for $x(k)$, there exists $U(k, K) = (u(k))$, s.t. $F(x(k), u(k)) = 0$, then $V^{[0]}(x(k+1))$ is

$$V^{[0]}(x(k+1)) = J(x(k+1), U^*(k+1, k+1)) = 0, \quad \forall x(k+1) \tag{2.13}$$

where $U^*(k+1,k+1)=(0)$. In this situation, the restrict term $F(x(k),u^{[1]}(k))=0$ for (2.11) is necessary.

If for $x(k)$, there exists $U(k,\bar{K})=(u(k),u(k+1),\ldots,u(\bar{K}))$, s.t. $F(x(k),U(k,\bar{K}))=0$, then $V^{[0]}(x(k+1))$ is

$$V^{[0]}(x(k+1))=J(x(k+1),U^*(k+1,K)), \tag{2.14}$$

where $U^*(k+1,K)=(u^*(k+1),u^*(k+2),\ldots,u^*(K))$.

For the iterative step $i>1$, the performance index function is updated as

$$\begin{aligned}V^{[i+1]}(x(k))&=\inf_{u(k)}\{x^{\mathrm{T}}(k)Qx(k)+u^{\mathrm{T}}(k)Ru(k)+V^{[i]}(x(k+1))\}\\&=x^{\mathrm{T}}(k)Qx(k)+u^{[i+1]\mathrm{T}}(k)Ru^{[i+1]}(k)+V^{[i]}(x(k+1)),\end{aligned} \tag{2.15}$$

where

$$u^{[i+1]}(x(k))=\arg\inf_{u(k)}\{x^{\mathrm{T}}(k)Qx(k)+u^{\mathrm{T}}(k)Ru(k)+V^{[i]}(x(k+1))\}. \tag{2.16}$$

In the above recurrent iterative procedure, the index i is the iterative step and k is the time step. The optimal control and optimal performance index function can be obtained by the iterative ADP algorithm (2.11)–(2.16).

In the following part, we will present the convergence analysis of the iterative ADP algorithm (2.11)–(2.16).

Theorem 2.2 *For an arbitrary state vector $x(k)$, the performance index function $V^{[i+1]}(x(k))$ is obtained by the iterative ADP algorithm (2.11)–(2.16), then $\{V^{[i+1]}(x(k))\}$ is a monotonically nonincreasing sequence for $i\geq 1$, i.e., $V^{[i+1]}(x(k))\leq V^{[i]}(x(k))$, $\forall i\geq 1$.*

Proof The mathematical induction is used to prove the theorem.

First, for $i=1$, we can have $V^{[1]}(x(k))$ in (2.11), $V^{[0]}(x(k+1))$ in (2.13), and the finite-time admissible control sequence $U^*(k,k+1)=(u^{[1]}(k),U^*(k+1,k+1))=(u^{[1]}(k),0)$. For $i=2$, we have

$$V^{[2]}(x(k))=x^{\mathrm{T}}(k)Qx(k)+u^{[2]\mathrm{T}}(k)Ru^{[2]}(k)+V^{[1]}(x(k+1)). \tag{2.17}$$

From (2.11), we have

$$\begin{aligned}V^{[1]}(x(k+1))=\inf_{u(k+1)}\{&x^{\mathrm{T}}(k+1)Qx(k+1)+u^{\mathrm{T}}(k+1)Ru(k+1)\\&+V^{[0]}(x(k+2))\}.\end{aligned} \tag{2.18}$$

So (2.17) can be expressed as