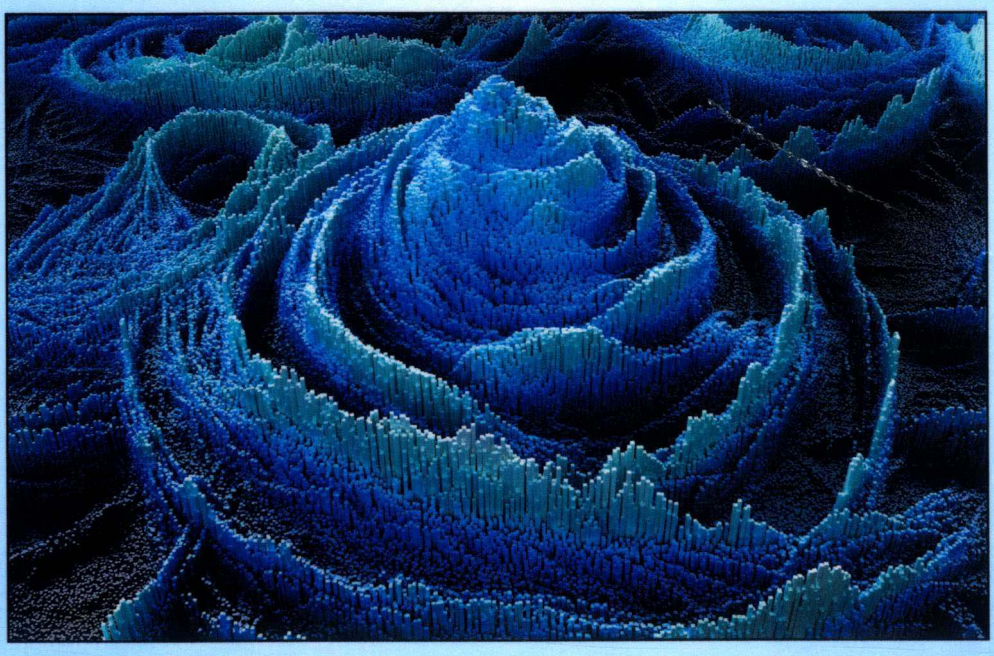
IEEE Press Series on Systems Science and Engineering
MengChu Zhou, Series Editor

Robust Adaptive Dynamic Programming



Yu Jiang • Zhong-Ping Jiang

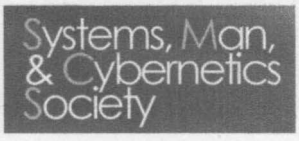
IEEE PRESS

WILEY

ROBUST ADAPTIVE DYNAMIC PROGRAMMING

YU JIANG
The MathWorks, Inc.

ZHONG-PING JIANG
New York University



IEEE PRESS

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data is available.

ISBN: 978-1-119-13264-6

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

ROBUST ADAPTIVE DYNAMIC PROGRAMMING

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

To my mother, Misi, and Xiaofeng
—Yu Jiang

To my family
—Zhong-Ping Jiang

ABOUT THE AUTHORS

Yu Jiang is a Software Engineer with the Control Systems Toolbox Team at The MathWorks, Inc. He received a B.Sc. degree in Applied Mathematics from Sun Yat-sen University, Guangzhou, China, a M.Sc. degree in Automation Science and Engineering from South China University of Technology, Guangzhou, China, and a Ph.D. degree in Electrical Engineering from New York University. His research interests include adaptive dynamic programming and other numerical methods in control and optimization. He was the recipient of the Shimemura Young Author Prize (with Prof. Z.P. Jiang) at the 9th Asian Control Conference in Istanbul, Turkey, 2013.

Zhong-Ping Jiang is a Professor of Electrical and Computer Engineering at the Tandon School of Engineering, New York University. His main research interests include stability theory, robust/adaptive/distributed nonlinear control, adaptive dynamic programming and their applications to information, mechanical and biological systems. In these areas, he has written 3 books, 14 book chapters and is the (co-)author of over 182 journal papers and numerous conference papers. His work has received 15,800 citations with an h-index of 63 according to Google Scholar. Professor Jiang is a Deputy co-Editor-in-Chief of the *Journal of Control and Decision*, a Senior Editor for the IEEE Transactions on Control Systems Letters, and has served as an editor, a guest editor and an associate editor for several journals in Systems and Control. Prof. Jiang is a Fellow of the IEEE and a Fellow of the IFAC.

PREFACE AND ACKNOWLEDGMENTS

This book covers the topic of adaptive optimal control (AOC) for continuous-time systems. An adaptive optimal controller can gradually modify itself to adapt to the controlled system, and the adaptation is measured by some performance index of the closed-loop system. The study of AOC can be traced back to the 1970s, when researchers at the Los Alamos Scientific Laboratory (LASL) started to investigate the use of adaptive and optimal control techniques in buildings with solar-based temperature control. Compared with conventional adaptive control, AOC has the important ability to improve energy conservation and system performance. However, even though there are various ways in AOC to compute the optimal controller, most of the previously known approaches are model-based, in the sense that a model with a fixed structure is assumed before designing the controller. In addition, these approaches do not generalize to nonlinear models.

On the other hand, quite a few model-free, data-driven approaches for AOC have emerged in recent years. In particular, adaptive/approximate dynamic programming (ADP) is a powerful methodology that integrates the idea of reinforcement learning (RL) observed from mammalian brain with decision theory so that controllers for man-made systems can learn to achieve optimal performance in spite of uncertainty about the environment and the lack of detailed system models. Since the 1960s, RL has been brought to the computer science and control science literature as a way to study artificial intelligence, and has been successfully applied to many discrete-time systems, or Markov Decision Processes (MDPs). However, it has always been challenging to generalize those results to the controller design of physical systems. This is mainly because the state space of a physical control system is generally continuous and unbounded, and the states are continuous in time. Therefore, the convergence and the stability properties have to be carefully studied for ADP-based

approaches. The main purpose of this book is to introduce the recently developed framework, known as robust adaptive dynamic programming (RADP), for data-driven, non-model based adaptive optimal control design for both linear and nonlinear continuous-time systems.

In addition, this book is intended to address in a systematic way the presence of dynamic uncertainty. Dynamic uncertainty exists ubiquitously in control engineering. It is primarily caused by the dynamics which are part of the physical system but are either difficult to be mathematically modeled or ignored for the sake of controller design and system analysis. Without addressing the dynamic uncertainty, controller designs based on the simplified model will most likely fail when applied to the physical system. In most of the previously developed ADP or other RL methods, it is assumed that the full-state information is always available, and therefore the system order must be known. Although this assumption excludes the existence of any dynamic uncertainty, it is apparently too strong to be realistic. For a physical model on a relatively large scale, knowing the exact number of state variables can be difficult, not to mention that not all state variables can be measured precisely. For example, consider a power grid with a main generator controlled by the utility company and small distributed generators (DGs) installed by customers. The utility company should not neglect the dynamics of the DGs, but should treat them as dynamic uncertainties when controlling the grid, such that stability, performance, and power security can be always maintained as expected.

The book is organized in four parts. First, an overview of RL, ADP, and RADP is contained in Chapter 1. Second, a few recently developed continuous-time ADP methods are introduced in Chapters 2, 3, and 4. Chapter 2 covers the topic of ADP for uncertain linear systems. Chapters 3 and 4 provide neural network-based and sum-of-squares (SOS)-based ADP methodologies to achieve semi-global and global stabilization for uncertain nonlinear continuous-time systems, respectively. Third, Chapters 5 and 6 focus on RADP for linear and nonlinear systems, with dynamic uncertainties rigorously addressed. In Chapter 5, different robustification schemes are introduced to achieve RADP. Chapter 6 further extends the RADP framework for large-scale systems and illustrates its applicability to industrial power systems. Finally, Chapter 7 applies ADP and RADP to study the sensorimotor control of humans, and the results suggest that humans may be using very similar approaches to learn to coordinate movements to handle uncertainties in our daily lives.

This book makes a major departure from most existing texts covering the same topics by providing many practical examples such as power systems and human sensorimotor control systems to illustrate the effectiveness of our results. The book uses MATLAB in each chapter to conduct numerical simulations. MATLAB is used as a computational tool, a programming tool, and a graphical tool. Simulink, a graphical programming environment for modeling, simulating, and analyzing multidomain dynamic systems, is used in Chapter 2. The third-party MATLAB-based software SOSTOOLS and CVX are used in Chapters 4 and 5 to solve SOS programs and semidefinite programs (SDP). All MATLAB programs and the Simulink model developed in this book as well as extension of these programs are available at http://yu-jiang.github.io/radpbook/

The development of this book would not have been possible without the support and help of many people. The authors wish to thank Prof. Frank Lewis and Dr. Paul Werbos whose seminal work on adaptive/approximate dynamic programming has laid down the foundation of the book. The first-named author (YJ) would like to thank his Master's Thesis adviser Prof. Jie Huang for guiding him into the area of nonlinear control, and Dr. Yebin Wang for offering him a summer research internship position at Mitsubishi Electric Research Laboratories, where parts of the ideas in Chapters 4 and 5 were originally inspired. The second-named author (ZPJ) would like to acknowledge his colleagues—specially Drs. Alessandro Astolfi, Lei Guo, Iven Mareels, and Frank Lewis—for many useful comments and constructive criticism on some of the research summarized in the book. He is grateful to his students for the boldness in entering the interesting yet still unpopular field of data-driven adaptive optimal control. The authors wish to thank the editors and editorial staff, in particular, Mengchu Zhou, Mary Hatcher, Brady Chin, Suresh Srinivasan, and Divya Narayanan, for their efforts in publishing the book. We thank Tao Bian and Weinan Gao for collaboration on generalizations and applications of ADP based on the framework of RADP presented in this book. Finally, we thank our families for their sacrifice in adapting to our hard-to-predict working schedules that often involve dynamic uncertainties. From our family members, we have learned the importance of exploration noise in achieving the desired trade-off between robustness and optimality. The bulk of this research was accomplished while the first-named author was working toward his Ph.D. degree in the Control and Networks Lab at New York University Tandon School of Engineering. The authors wish to acknowledge the research funding support by the National Science Foundation.

YU JIANG
Wellesley, Massachusetts
ZHONG-PING JIANG
Brooklyn, New York

ACRONYMS

ADP	Adaptive/approximate dynamic programming
AOC	Adaptive optimal control
ARE	Algebraic Riccati equation
DF	Divergent force field
DG	Distributed generator/generation
DP	Dynamic programming
GAS	Global asymptotic stability
HJB	Hamilton-Jacobi-Bellman (equation)
IOS	Input-to-output stability
ISS	Input-to-state stability
LQR	Linear quadratic regulator
MDP	Markov decision process
NF	Null-field
PE	Persistent excitation
PI	Policy iteration
RADP	Robust adaptive dynamic programming
RL	Reinforcement learning
SDP	Semidefinite programming
SOS	Sum-of-squares
SUO	Strong unboundedness observability
VF	Velocity-dependent force field
VI	Value iteration

GLOSSARY

$\lvert\cdot\rvert$	The Euclidean norm for vectors, or the induced matrix norm for matrices
$\lVert\cdot\rVert$	For any piecewise continuous function $u : \mathbb{R}_+ \to \mathbb{R}^m$, $\lVert u\rVert = \sup\{\lvert u(t)\rvert, t \geq 0\}$
$\otimes$	Kronecker product
$\mathcal{C}^1$	The set of all continuously differentiable functions
$J_D^{\oplus}$	The cost for the coupled large-scale system
$J_D^{\odot}$	The cost for the decoupled large-scale system
$\mathcal{P}$	The set of all functions in $\mathcal{C}^1$ that are also positive definite and radially unbounded
$\mathcal{L}(\cdot)$	Infinitesimal generator
$\mathbb{R}$	The set of all real numbers
$\mathbb{R}_+$	The set of all non-negative real numbers
$\mathbb{R}[x]_{d_1,d_2}$	The set of all polynomials in $x \in \mathbb{R}^n$ with degree no less than $d_1 > 0$ and no greater than d_2
$\text{vec}(\cdot)$	$\text{vec}(A)$ is defined to be the mn-vector formed by stacking the columns of $A \in \mathbb{R}^{n\times m}$ on top of another, that is, $\text{vec}(A) = [a_1^T a_2^T \cdots a_m^T]^T$, where $a_i \in \mathbb{R}^n$, with $i = 1, 2, \ldots, m$, are the columns of A
$\mathbb{Z}_+$	The set of all non-negative integers
$[x]_{d_1,d_2}$	The vector of all $\binom{n+d_2}{d_2} - \binom{n+d_1-1}{d_1-1}$ distinct monic monomials in $x \in \mathbb{R}^n$ with degree no less than $d_1 > 0$ and no greater than d_2
∇	∇V refers to the gradient of a differentiable function $V : \mathbb{R}^n \to \mathbb{R}$

CONTENTS

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.

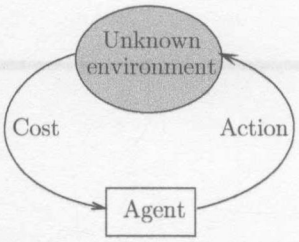


FIGURE 1.1 Illustration of RL. The agent takes an action to interact with the unknown environment, and evaluates the resulting cost, based on which the agent can further improve the action to reduce the cost.

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