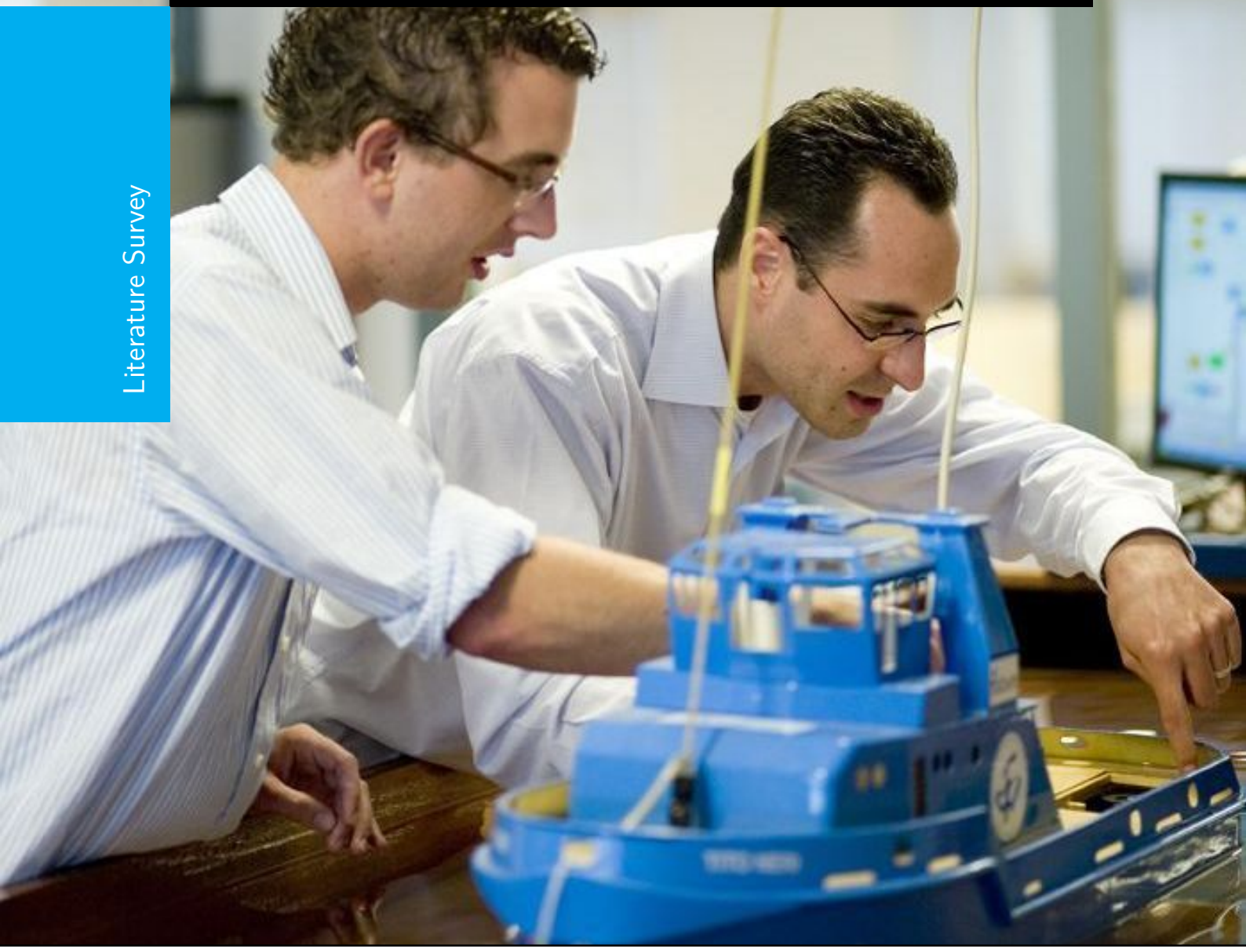
Reinforcement Learning for Tracking Control in Robotics

Yudha Prawira Pane

Literature Survey



TU Delft Delft University of Technology

Delft Center for Systems and Control

Reinforcement Learning for Tracking Control in Robotics

Literature Survey

Yudha Prawira Pane

January 26, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The implementation work in this thesis was done at DCSC's robotics lab.

Abstract

This is an abstract.

Table of Contents

Preface ix

Acknowledgements xi

1 Introduction 1
- 1-1 Problem Definition . . . 2
- 1-2 Goal of the Thesis . . . 2
- 1-3 Literature Study Approach . . . 3
- 1-4 Outline . . . 3

2 Reinforcement Learning Preliminaries 5
- 2-1 Goal as Cost Minimization . . . 5
- 2-2 Markov Decision Process . . . 6
- 2-3 Value Function . . . 6
- 2-4 Policy and value iteration . . . 7
- 2-5 Actor Critic Methods . . . 7

3 Reinforcement Learning for Tracking Problem: A Survey 9
- 3-1 Reinforcement Learning for Optimal Tracking Control . . . 9
- 3-2 Dynamic Tuning via Reinforcement Learning . . . 10
- 3-3 Nonlinear Compensation for Tracking via Reinforcement Learning . . . 10

4 Simulation & Verification 11
- 4-1 Simulated Setup . . . 11
- 4-2 Simulation Result and Analysis . . . 11
- 4-3 Discussion . . . 11

5 Future Work and Experiments Plan 13

6 Conclusion **15**

A Appendix **17**

A-1 Simulation Program 17

A-1-1 A MATLAB listing 17

Glossary **21**

List of Acronyms 21

List of Symbols 21

List of Figures

2-1 Actor critic structure 8

List of Tables

Preface

Acknowledgements

Delft, University of Technology
January 26, 2015

Yudha Prawira Pane

"It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience"

— *Albert Einstein*

Chapter 1

Introduction

Reference or trajectory tracking is one of the building blocks to perform a complex task in robotics. Given a desired path/trajectory, the robot must be able to follow it as quickly as possible with minimum error. Capability to perform this precise tracking is crucial for robots that are to be deployed at manufacturing industries such as semiconductor, automotive, and recently, the emerging application of 3D printing.

Statistics by International Federation of Robotics (IFR) [1] shows that the global sales of industrial robots continues to increase steadily. In 2014, it is expected that the total number of industrial robots installed reaches 205,000 units, a rise of approximately 15 % from previous year. The survey points out that the mature markets such as automotive, electronics, and metal are responsible for such growth.

Meanwhile, there is also a growing interests in applying robots to relatively new applications such as 3D printing, architecture, and art. For instance, research done by Gramazio et. al [2] [3] aims to push the capability of industrial robots to make direct fabrication based on CAD model a reality. The advantage of using robots over conventional CNC machines lies on their flexibility, easy-to-adapt feature, and high degrees of freedom (DoF) – enabling execution of difficult configuration in 3-dimension (3D) space. These aforementioned applications demand high precision since a minuscule of error could lead to a defect product or even worse, a disaster. Therefore a precise, accurate reference tracking capability is inevitable.

In order to achieve this, a reference tracking control is needed. However, a robot brings along non-linearities, noises, and external disturbances that are difficult to model, let alone compensate. This unknown properties often hinders the controller to perform optimally. A class of controllers which solely depends on the system's model will surely suffer a poor tracking accuracy. The natural answer to this problem is to introduce a controller capable of adjusting its parameter overtime by comparing the reference to the actual trajectory. By doing so, the controller will have an extra degree of freedom to compensate for the unknown properties hence improving the tracking quality. The controller of such characteristic belongs to the class of adaptive controller.

In this thesis, a method to improve the performance of nominal controller by using Reinforcement Learning (RL) is proposed. Despite decades of extensive research on RL, its application to

optimize tracking problem in robotics is still a relatively unexplored topic. Based on the literature, there are three potential approaches to address the tracking problem. The first one comes from the work of Lewis et. al. on RL for optimal control. Lewis and his group have been developing a comprehensive research on RL for solving the solution to adaptive optimal control. Their research has been extended for discrete [4] and continuous time [5], for linear [6] and non-linear system [7]. Furthermore, their technique could also be applied in Q-learning [4] and actor-critic structure [7]. The second approach is proposed by Bayiz et. al. in [8]. The paper discusses a slightly different approach by using RL to learn disturbance compensation for nonlinear system. This disturbance compensation acts as an additive input signal to the control signal. Finally, the third approach uses the notion of adaptive gain scheduling. Buchli et.al present an algorithm called Policy Improvement with Path Integral (PI^2) to vary the gain of a Proportional Derivative (PD) controller in order to achieve a desired terminal state [9] [10]. Having explained the motivation of this thesis, now we are ready to define the research problem.

1-1 Problem Definition

The fundamental problem in this literature study concerns the non-optimal performance of nominal controller with respect to reference tracking task. Hence the research question can be raised as follows.

"Is it possible to integrate Reinforcement Learning technique to a nominal controller in a certain structure such that reference tracking performance of the controlled system significantly improves?"

While conducting a research, it is often wise to restrict oneself to a simple context, but still captures the essential elements of the original problem [11]. Therefore, in answering this question, some simplifying assumptions are made.

1. The system to be controlled is fully actuated
2. The system to be controlled is observable. This assumption is necessary in order to satisfy Markov property [12].
3. Nominal, stabilizing controller is available
4. Identification reveals some information about the system, but alone is not adequate to design an accurate reference tracking controller.

1-2 Goal of the Thesis

The goal of this thesis are as follows:

1. To provide a general framework of improving tracking control using RL
2. To apply and compare existing method of RL for tracking application to the 3D printing robot setup
3. To come up with modification or improvement of previous methods

1-3 Literature Study Approach

In order to build a strong theoretical foundation for later implementation, the following literature approach is used. The order does not necessarily represent a sequential process.

1. To gather as many relevant papers as possible from reputable academic search engines. Relevant means papers which deal with RL and control system. Additional pointer to tracking problem is heavily considered. Examples of sources being used are Web of Science, IEEE Xplore and Google Scholar.
2. To discuss the detail of future experimental setup (UR5 3D printing robot) with Marco de Gier, who was working on the setup at the time this literature is written.
3. From the papers, extract existing methods which have the potential for application to the future experiments. So far, there are 3 different methods that are considered. These methods will be explained in detail in Chapter 3.
4. Create simple simulation programs showing how each method works

1-4 Outline

The structure of this literature review is arranged as follows. In the next chapter, an introductory materials of RL is presented. This covers the framework widely used in RL (Markov Decision Process), the principle of value and policy iteration, the formulation of RL for continuous space, and the actor-critic structure which suits the framework of control system. Chapter 3 provides the result of literature study being conducted. This includes the detailed explanation of methods found and their comparison. Furthermore, a new controller is proposed.

Chapter 2

Reinforcement Learning Preliminaries

This chapter is dedicated to present a concise theory of reinforcement learning. The first section will show how a certain goal can be formalized as a reward maximization – one of the ideas which serves as a basic foundation of Reinforcement Learning (RL). Section 2-2 explains the basics of Markov Decision Process (MDP), a general framework used in RL problem. The notion of value function will be discussed in Section 2-3. Subsequently, a method to solve RL, namely policy and value iteration will be developed in section 2-4. Finally, Section 2-5 will discuss the actor-critic structure which is an alternative solution to policy iteration.

2-1 Goal as Cost Minimization

The nature of RL is inspired by the way living organisms learn to reach their desired goals. Animals for instance, learn by first acting on the environment, observe the changes that occur, and improve their action iteratively. One example is a circus lion that is tasked to perform acrobatic show while its trainer observing the progress. If the lion successfully executes the task, it will be rewarded with foods. Conversely, punishment will be inflicted whenever it fails. The lion initially has no idea of how to perform the task. However through trial and error, it will follow its instinct to increase the frequency of receiving rewards while trying its best to avoid punishments. In a certain duration of training, the circus lion will be finally able to perform the task flawlessly.

Now we will formalize above illustration for robotics application. A robot can be described by its states x_k with subscript k denoting time instance. Applying an action u_k will bring the robot to state x_{k+1} with immediate reward r_{k+1}. Subsequently, at $k+1$ the robot applies u_{k+1} which yields state x_{k+2} and r_{k+2}. This action-state-update iteration is run for infinite time instances. The goal is defined as maximization of cumulative reward the robot receives. In control engineering, reward is usually replaced with cost. In that case the goal is defined as minimization problem. Starting from now, we will define goal as minimization of future cost J.

From the sequence of cost obtained over time, we can define a formalization of goal, called expected return. Return J_t is a function that maps the sequence of costs into real number. An example of return is the sum of the costs.

$$J_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T \tag{2-1}$$

2-2 Markov Decision Process

MDP is defined as a tuple $\langle X, U, f, \rho \rangle$ which satisfies Markov property [13]. The detailed explanation of Markov property can be found on [12] section 3.5 but the main idea is that to determine the probability of a state at certain time, it is sufficient to only know the state of previous time instance. The elements of the tuple are:

- X is the state space
- U is the action space
- $f : X \times U \to X$ is the state transition function (system dynamics)
- $\rho : X \times U \to \mathbb{R}$ is the reward function

In control engineering, f represents the system dynamics which is a transition function mapping a current state and action to the one-step ahead state up to a probability distribution. This probability distribution is mathematically denoted in Equation (2-2).

$$\Pr\{x_{t+1} = x', r_{t+1} = r | x_t, u_t\} \tag{2-2}$$

where x denotes state, u denotes action, and r denotes immediate reward obtained upon applying the input on the corresponding state.

2-3 Value Function

Value function describes how good a particular state or state-action pair under a certain policy. As previously explained, in this thesis we will stick to control engineering convention by seeing RL as cost minimization problem. Therefore, the smaller value function of a state x, the better it is. The optimum value function is denoted by $V^\pi(x)$ for state-value function and $Q^\pi(x, u)$ for action-value function. Furthermore, one can always find a policy which gives an optimal value function V^*. This optimal value function respects the Bellman optimality equation, which can be written as 2-3. The action-value function counterpart is denoted in Equation 2-4.

$$V^*(x) = \rho(x, u) + \gamma \min_u V^*(f(x, u)) \tag{2-3}$$

$$Q^*(x, u) = \rho(x, u) + \gamma \min_u Q^*(f(x, u), u') \tag{2-4}$$

Discount factor γ is introduced to avoid the value function goes to infinity. Once V^* is known, the optimal policy can be taken in a greedy way as in Equation 2-5. This concludes the formulation of RL problem. The subsequent sections will deal with two methods to solve for the solution.

$$\pi^* = \arg\max_{\pi} V^*(x) \tag{2-5}$$

2-4 Policy and value iteration

The optimal policy can be reached asymptotically by means of iteration. Let initial policy be given by $\pi_i(x, u)$. Then a new policy can be determined by first evaluating the value of π_i and recursively calculate the new policy π_{i+1}. This process can be casted as an iteration algorithm as follows.

Initialization: Start from admissible policy π_i
for *i = 1 to N* **do**
  Policy Evaluation:
  $V_{i+1}(x_t) = \rho(x_t, \pi_i(x_t)) + \gamma V_{i+1}(x_{t+1})$
  Policy Iteration:
  $\pi_{i+1}(x_t) = arg\min_{\pi}(\rho(x_t, \pi(x_t)) + \gamma V_{i+1}(x_{t+1}))$
end

Algorithm 1: Policy iteration algorithm

The prove of iteration above is provided in [14]. In order to increase computational efficiency, instead of evaluating value function V for all possible state x in every iteration, one can formulate a value function evaluation recursively. The policy iteration above is then modified as shown in Algorithm 2. It can be guaranteed that V_i will eventually converge to V^*.

Initialization: Start from admissible policy π_i
for *i = 1 to N* **do**
  Value Iteration:
  $V_{i+1}(x_t) = \rho(x_t, \pi_i(x_t)) + \gamma V_i(x_{t+1})$
  Policy Iteration:
  $\pi_{i+1}(x_t) = arg\min_{\pi}(\rho(x_t, \pi(x_t)) + \gamma V_{i+1}(x_{t+1}))$
end

Algorithm 2: Value iteration algorithm

This type of method to find solution to RL is called Dynamic Programming (DP). This method is closely related with a branch of control system, namely optimal control.

2-5 Actor Critic Methods

The second method for solving RL is by using temporal-difference learning. It is favored due to its model-free nature. In this section, we will discuss a class of Temporal-Difference (TD) called actor-critic method. The idea of actor-critic structure is to separate policy and value function into called actor and critic entity respectively (see Figure 2-1). These actor ψ and critic θ function are parameterized by function approximators and updated using the temporal

difference signal δ in every iteration. The actor-critic method is presented in Algorithm 3 (adapted from [13]). Note that $\tilde{u}$ denotes random exploration term.

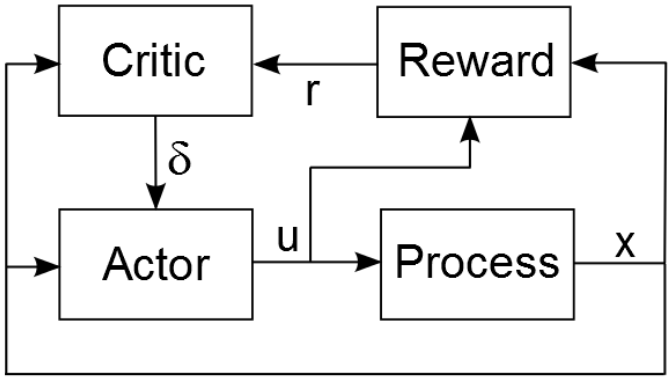


Figure 2-1: Actor critic structure

for *every trial* **do**
  Initialize x_0 and $u_0 = \tilde{u}_0$
  repeat
    apply u_k, measure x_{k+1}, receive r_{k+1}
    choose next action $u_{k+1} = \hat{\pi}(x_{k+1}, \psi_k) + \tilde{u}_{k+1}$
    $\delta_k = r_{k+1} + \hat{V}(x_{k+1}, \theta_k) - \hat{V}(x_k, \theta_k)$
    $\theta_{k+1} = \theta_k + \alpha_c \delta_k \frac{\partial \hat{V}(x,\theta)}{\partial \theta}\Big|_{x=x_k, \theta=\theta_k}$
    $\psi_{k+1} = \psi_k + \alpha_c \delta_k \frac{\partial \hat{V}(x,\psi)}{\partial \psi}\Big|_{x=x_k, \psi=\psi_k}$
  until *terminal state*;
end

Algorithm 3: Actor-critic algorithm

Chapter 3

Reinforcement Learning for Tracking Problem: A Survey

Despite the success of Reinforcement Learning (RL) in many robotics problem (e.g. learning to fly [15], walk [16] and navigate [17]), the application of RL for tracking control is not a widely explored topic. Over the spans of the literature survey, author finds several attempts to exploits RL for tracking problem, which can be categorized into 3 different approaches: dynamic tuning, RL for optimal control, and RL for nonlinear additive compensator.

This chapter covers the foundational theory of the 3 aforementioned approaches. The main idea, advantages, limitations and ease of implementation are the key issues which will be discussed. These issue will serve as the basis of the argument to choose one method for later implementation. The chapter starts in Section 3-1 by providing explanation about optimal tracking control using RL. Section 3-2 deals with the so called dynamic tuning – a class of gain scheduling which makes use of RL. The third method, presented in Section 3-3, is a relatively new approach which employs RL to learn additive input compensation.

3-1 Reinforcement Learning for Optimal Tracking Control

This method is initiated and developed by Lewis et. al. which aims to solve the tracking by RL problem from dynamic programming perspective. The method uses optimal control, a branch of control theory whose root is closely related to dynamic programming [18]. The method starts from the downside of optimal tracking control which requires the solution of non-causal differential equation. It turns out that by modifying the cost function and the state of the optimal control, a causal representation can be obtained, followed by RL to asymptotically solve for the solution .

To provide an easier comparison between the standard optimal control solution with RL-based one, this section starts by formulating the optimal tracking problem and deriving the solution. Next, the modified formulation of optimal control which allows the causal

formulation of infinite horizon optimal control problem is discussed. Following is the policy iteration algorithms to solve the optimal control. In this section, only discrete-time Linear Quadratic Tracking (LQT) problem is considered. Although the extension to non-linear and continuous time optimal control problem is not straightforward, the main idea is actually very similar.

3-2 Dynamic Tuning via Reinforcement Learning

This is the first section .

This is the subsection of the first section.

3-3 Nonlinear Compensation for Tracking via Reinforcement Learning

This is third section.

Chapter 4

Simulation & Verification

4-1 Simulated Setup

This chapter will cover figures and math.

4-2 Simulation Result and Analysis

4-3 Discussion

Chapter 5

Future Work and Experiments Plan

Chapter 6

Conclusion

Appendix A

Appendix

Appendices are found in the back.

A-1 Simulation Program

A-1-1 A MATLAB listing

```
%
% Comment
%
n=10;
for i=1:n
    disp('Ok');
end
```

Bibliography

[1] I. F. of Robotics (IFR), "Industrial robot statistics," *World Robotics 2014 Industrial Robots*, 2014.

[2] V. Helm, J. Willmann, F. Gramazio, and M. Kohler, "In-situ robotic fabrication: Advanced digital manufacturing beyond the laboratory," *Springer Tracts in Advanced Robotics 2014*, 2014.

[3] E. Lloret, A. R. Shahabb, M. Linus, R. J. Flatt, F. Gramazio, M. Kohler, and S. Langenberg, "Complex concrete structures: Merging existing casting techniques with digital fabrication," *Computer-Aided Design*, 2014.

[4] 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*, vol. 50, no. 4, pp. 1167 – 1175, 2014.

[5] H. Modares and F. Lewis, "Online solution to the linear quadratic tracking problem of continuous-time systems using reinforcement learning," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 3851–3856, Dec 2013.

[6] B. Kiumarsi-Khomartash, F. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control for linear discrete-time systems using reinforcement learning," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 3845–3850, Dec 2013.

[7] B. Kiumarsi and F. Lewis, "Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[8] Y. E. Bayiz and R. Babuska, "Nonlinear disturbance compensation and. reference tracking via reinforcement. learning with fuzzy approximators," *19th IFAC World Congress*, 2010.

[9] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, "Variable impedance control-a reinforcement learning approach," *Robotics: Science and Systems*, 2010.

[10] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E. Theodorou, and S. Schaal, "Reinforcement learning of impedance control in stochastic force fields," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2, pp. 1–6, Aug 2011.

[11] A. Einstein, "On the method of theoretical physics," vol. 1, pp. 163–169, Philosophy of Science, 1934.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 28. MIT press, 1998.

[13] R. Babuska, "Sc4081 knowledge-based control systems lecture slide,"

[14] D. Bertsekas, "Neuro-dynamic programming: An overview and recent results," in *Operations Research Proceedings 2006* (K.-H. Waldmann and U. Stocker, eds.), vol. 2006 of *Operations Research Proceedings*, pp. 71–72, Springer Berlin Heidelberg, 2007.

[15] A. Coates, P. Abbeel, and A. Ng, "Autonomous helicopter flight using reinforcement learning," in *Encyclopedia of Machine Learning* (C. Sammut and G. Webb, eds.), pp. 53–61, Springer US, 2010.

[16] J. Z. Kolter, P. Abbeel, and A. Y. Ng, "Hierarchical apprenticeship learning with application to quadruped locomotion," in *Advances in Neural Information Processing Systems 20* (J. Platt, D. Koller, Y. Singer, and S. Roweis, eds.), pp. 769–776, Curran Associates, Inc., 2008.

[17] S. Ross, B. Chaib-draa, and J. Pineau, "Bayesian reinforcement learning in continuous pomdps with application to robot navigation," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2845–2851, May 2008.

[18] R. Sutton, A. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *Control Systems, IEEE*, vol. 12, pp. 19–22, April 1992.

Glossary

List of Acronyms

RL	Reinforcement Learning
MDP	Markov Decision Process
DoF	degrees of freedom
PI^2	Policy Improvement with Path Integral
PD	Proportional Derivative
3D	3-dimension
DP	Dynamic Programming
TD	Temporal-Difference
LQT	Linear Quadratic Tracking