Studies on Intelligent Adaptive Control of Autonomous Systems with Applications to Longitudinal Vehicle Following

Submitted by

Dai Xiaohui

Department of Electronic and Information Engineering

the Hong Kong Polytechnic University

A thesis submitted in the requirements

for the Degree of Master of Philosophy

November 2003

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledge has been made in the text.

_____ (Signed)

Dai Xiaohui (Name of student)

Abstract

In the last two decades, the design of intelligent vehicles that either assist or replace the driver has attracted a lot of attention from both academic researchers and industrial entrepreneurs. This thesis addresses the important problem of autonomous vehicle control within the academic framework and provides new algorithms for solving longitudinal vehicle following based on adaptive and fuzzy control methodologies. Throughout this thesis, the author reports the design of intelligent controllers with more flexibility but less a priori knowledge about system models.

In the first stage, the author concentrates on the development of a novel adaptive fuzzy controller. The vehicle longitudinal control system falls into a class of partially known nonlinear systems. A fuzzy system is employed to approximate the ideal controller. A relationship between approximation error and parameters of the fuzzy controller is established first. Then, the adaptive laws of the fuzzy controller are obtained based on Lyapunov synthesis approach. All the parameters of fuzzy controller are adjustable. This is the major difference between my work and the others.

However, a weakness of the proposed adaptive fuzzy controller is that it requires some information about the system and it only aims at a specific nonlinear system. To this end, I investigate Q-learning, a model free reinforcement learning (RL) method, and its applicability as a controller design approach for real systems in a knowledge-poor environment. The focus is on two issues: (i) the structure of the Q

estimator network and fuzzy controller, and (ii) the development of learning algorithms for both of them. A Takagi-Sugeno type fuzzy inference system and a multiple-layer feed-forward neural network are employed as action producer and Q estimator respectively. The learning algorithms for the Q estimator network and the fuzzy controller are developed based on the temporal difference methods as well as the gradient descent algorithm.

The efficiency of applying RL directly may not always be appropriate. Therefore, the author proposes a controller based on dual heuristic programming (DHP) to enhance the controller performance. The structure and adaptation algorithms of the controller for vehicle following problems are presented. The proposed controller has two advantages compared with other controllers based on adaptive critic designs: (i) the system model is not required directly or indirectly, and (ii) it can take advantage of the TS type fuzzy controller to incorporate a priori knowledge. The simulation results of the controller based on RL and those of the controller based on DHP are compared and the advantages of the technique are also explored.

The application of these intelligent adaptive controllers to autonomous vehicle control systems has been described. Conclusions are drawn based on studies performed via theoretical analysis and computer simulations.

Publications

1. Dai, X., Li, C. K. and Rad, A. B. "Performance Comparison of Autonomous Vehicle Controllers". *Mechanical and Electrical Engineering Technology*, Vol.31, no.6, pp.117-121 (2002)

2. Dai, X., Li, C. K. and Rad, A. B. "A novel adaptive fuzzy controller for application in autonomous vehicles". *Mechanical and Electrical Engineering Technology*, Vol.31, no.6, pp.121-126 (2002)

3. Dai, X., Li, C. K. and Rad, A. B. "The model of the artificial immune response". *The 9th International Conference on Enhancement and Promotion of Computational Methods in Engineering and Science*, Macao, August 5-8, 2003. (This paper will also be collected in a book entitled "EPMESC IX - Computational Methods in Engineering and Science" to be published by A. A. Balkema of the Swets & Zeitlinger Publishers.)

4. Dai, X., Li, C. K. and Rad, A. B. "An Approach to Tune Fuzzy Controllers Based on Reinforcement Learning". *The 12th IEEE International Conference on Fuzzy Systems*, St. Louis, USA, May 25-28, 2003, pp.517-522 (2003)

5. Dai, X., Li, C. K. and Rad, A. B. "Adaptive Control of a Class of Nonlinear Systems with Fuzzy Approximators". *The 12th IEEE International Conference on Fuzzy Systems*, St. Louis, USA, May 25-28, 2003, pp.384-389 (2003)

6. Dai, X., Li, C. K. and Rad, A. B. "A Novel Adaptive Fuzzy Controller for Application in Autonomous Vehicles". *IEEE Transactions on Vehicular Technology*, submitted.

7. Dai, X., Li, C. K. and Rad, A. B. "An Approach to Tune Fuzzy Controllers Based on Reinforcement Learning for Autonomous Vehicle Control". *IEEE Transactions on Industrial Electronics*, submitted.

8. Dai, X., Li, C. K. and Rad, A. B. "Autonomous Vehicle Longitudinal Control Based on Adaptive Critic Designs". *IEEE Transactions on Systems, Man and Cybernetics Part B*, submitted.

9. Li, C.K., Tao, T. and Dai, X. "A dual adaptive model estimator for target tracking". *The 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Hong Kong, April 6-10, 2003, pp. VI_53-VI_56 (2003)

Acknowledgments

An enormous credit is due to Dr. C.K. Li and Prof. A.B. Rad. Dr. Li and Prof. Rad served as my chief supervisor and co-supervisor for two years. Without their invaluable advice, patient assistance and careful guidance, this thesis would not have come to fruition.

The completion of this thesis would not have been possible without the literature contributions cited herein. In particular, I wish to thank Dr. Li-Xin Wang, Prof. Mo Jamshidi and Prof. Richard S. Sutton for inspiring me through their work.

Special thanks is due to my colleagues for their help and companionship. They are Dr. K.F. Wan, Ah-Ho, Amanda, Bryan, James, Jason, Johnny, Linda, Matthew and Oscar.

From a financial standpoint, thanks is due to the following organizations. The work in this thesis is funded by grant G-W085. Financial aid was also provided by the Department of Electronic and Information Engineering and IEEE Neural Network Society to support me for the academic conference.

I also wish to thank my Hong Kong friends, that I got to know in Cantonese Class, for their kindness, encouragement and friendship.

Finally, I would like to thank my parents for their encouragement and support, and last but not least I would like to express a very special gratitude to my girlfriend, who provided endless love and support all along.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Autonomous systems are referred to as "intelligent machines" that continuously interact with their operating environment and respond appropriately to anomalies from status quo in a similar fashion to humans. These systems are designed with the state of art in sensor technology, hard and soft computing and are governed by advanced control algorithms and are complex and multidisciplinary. Due to their vast importance in diverse areas such as transportation, surveillance, inspection, cleaning and entertainment, etc. [Tzafestas 1999]; research and industrial groups from all over the world have taken the challenge of addressing numerous design and implementation issues associated with this fascinating and rapidly emerging filed of research and development. The research studies reported in thesis focuses on an important application area of autonomous systems, i.e. autonomous vehicles within the context of Automated Highway Systems (AHS). AHS is the infrastructure through which the road is shared with manually driven cars and individual or platoon of autonomous vehicles. Advocates of AHS suggest that full automation can greatly increase highway capacity while improving safety [Bender 1991, Shladover 1995]. This call is in response to unmanageable increase in highway congestion and the number of traffic accidents. Transportation experts from all over the world agree that majority of traffic accidents occur due to human fatigue and/or negligence. Another

important justification for AHS and autonomous vehicles has been a never ending increase in the number of cars in the roads in developed as well as developing countries. Autonomous vehicles operate in two modes: In mode 1, they operate at a supervisory level and assist or warn the human driver where appropriate (this mode has already been implemented in many new luxury cars). In mode 2, the vehicle is transformed to an intelligent driving agent that takes over the task of driving (this has only been tried with prototype vehicles). Problems associated with designing true autonomous vehicles are numerous and challenging. Among the difficulties are complex vehicle dynamics, nonlinearities and uncertainties, safety issues, control algorithms and weather patterns, etc.

The focus of this thesis is to address the problem of control algorithms appropriate for autonomous vehicles. Adaptive control is known to manage well the uncertainties of environment and the controlled object. Intuitively, an adaptive controller is a controller that can modify its behavior in response to changes in the dynamics of the process and the character of the disturbances. The idea of adaptive is appealing. Conventional control systems are designed using mathematical models of physical systems. When the uncertainties in the plant and environment are large, the fixed feedback controllers may not be adequate, and adaptive controllers are used. Note that adaptive control in conventional control theory has a specific and rather narrow meaning. In particular it typically refers to adapting to variations in the constant coefficients of mathematical model describing the unknown plant: these new coefficient values are identified and then used, directly or indirectly, to reassign the

values of the constant coefficients in the controller.

However, the controller should cope with significant un-modeled dynamics and unexpected changes in the plant, in the environment and in the control objectives. This will involve the use of advanced decision making processes to generate control actions such that a certain performance level is maintained even though there are little knowledge about the model and drastic changes in the operating conditions. The need to use intelligent methods in autonomous control stems from the need for an increased level of autonomous decision making abilities in achieving complex control tasks. Fuzzy control [Passino & Yurkovich 1998] is one such method that has been highly popular and rather straightforward to implement. It has been regarded as a practical alternative for a variety of challenging control applications since it provides a convenient method for constructing nonlinear controllers via the use of heuristic information. Such heuristic information may come from an experienced operator or existing knowledge. Therefore, fuzzy control provides a user-friendly formalism for representing and implementing the control objectives. In particular, it is believed that since it may not require mathematical model of the controlled object to achieve acceptable performance, it should be very suitable for autonomous vehicle systems.

This thesis describes three possible approaches to implementing advanced decision making processes on an autonomous vehicle, operating in an unknown or partially unknown environment. Before we proceed any further, let us explicitly set down the objectives of this work, and also the justification for these objectives.

1.1 Objectives and Research Direction

The major goal of this work is to provide a framework that makes the use of adaptive and fuzzy logic control or other soft computing techniques on autonomous vehicle systems. Throughout this thesis, we try to design the controller with more flexibility but less a priori knowledge about vehicle model or environment. In other words, we want to design a vehicle controller whose parameters may vary on line during operation, but the model constraints may not be strict. Therefore, it can be expected to accommodate for a higher degree of uncertainty. We may tune the controller based on desired trajectory or action signal. The former method is conventional adaptive control, and the latter method is supervised learning. However, the conventional adaptive control requires the model of the system directly or indirectly, and the structure of the model must be available if the exact parameters of the system are not known. The performance of supervised learning is related closely with the training data, which may have the problem of that the performance is not good if the training data lack of generalization.

It seems a dilemma between the performance achieved and the model information required. To achieve high performance of the controller, we require a model (or training data) as much as possible. But the cost to obtain models or training data may be high, or they are even impossible to get due to the unavoidable uncertainties. In this thesis, we want to achieve a trade-off between the above two aspects. We try to get acceptable performance with less priori model information. The performance may be improved by the interaction with environment.

In the first stage of our research, we will concentrate on the development of the vehicle controller using the idea of conventional adaptive control. The proposed controller is called adaptive fuzzy controller since it employs fuzzy logic systems and the parameters of the fuzzy systems are adjustable. The adaptive fuzzy controller is different from the non-adaptive fuzzy controller in the following two aspects: i) The fuzzy controller in the adaptive fuzzy control system is changing during real-time operation, whereas the fuzzy controller in the non-adaptive control system is fixed before real-time operation, and ii) the additional component, the adaptation law, is introduced to the adaptive fuzzy control system to adjust the fuzzy controller parameters.

This work is to extend the works of other researchers. For the development of the proposed adaptive fuzzy controller, the objective can be summarized as follows:

1) For a vehicle longitudinal system, it can fall into a class of specific continuous time SISO nonlinear system with some unknown parameters. A fuzzy system is employed to approximate the ideal controller. A relationship between approximation error and parameters of the fuzzy controller should be established first.

2) Design the adaptive laws of the parameters of the fuzzy controller for the specific nonlinear system based on Lyapunov synthesis approach. We intend to tune all the parameters of fuzzy controller to achieve better performance. This is different from the most of the current research on adaptive fuzzy control which only tunes the parameters of the consequences of fuzzy rules.

The weakness of the adaptive fuzzy control is that we should have some model knowledge about the controlled object, such as model structure, parameters range etc. This model information as well as the training data may be unavailable or difficult to obtain. Moreover, the proposed design approach only aims at the specific nonlinear system. Therefore, we want to design a vehicle controller which need as less model information and training data as possible.

Reinforcement learning has been introduced to solve the problems of lack of a priori knowledge. For the reinforcement learning, it only needs the critic information, i.e., rewards and punishments (evaluative signal), and it is based on the common sense idea that if an action is followed by a satisfactory state, or by an improvement, then the tendency to produce that action is strengthened, i.e., reinforced. So, the actual correct actions or their trajectories are not required. Since the evaluative signal contains much less information, the reinforcement learning is appropriate for system operating in a knowledge-poor environment [Chiang et al. 1997, Sutton & Barto 1998].

In the next stage of our research, we would make reinforcement learning more applicable as controller design approach for finding the optimal controller (in other words, tune the parameters of the controller). The objective is to derive methods that:

1) Are able to deal with continuous state space problems. Most reinforcement learning approaches are based on systems with discrete state and action space configurations. However, controllers for real systems often have continuous state values as input and continuous actions as output.

2) Are able to deal with nonlinear system. This is necessary for vehicle longitudinal control.

3) Do not need or need as less model information and training data as possible. As we said before, it is unavailable or difficult to obtain the desired training data, and also difficult to get the accurate vehicle model.

4) Some qualitative experiences may be incorporated into the design procedure if possible. This can speed up the design procedure and avoid the unstable.

But the efficiency of application of reinforcement learning directly may not be good. We may consider adaptive critic designs and make the output of the critic network as the approximation of derivative of overall cost, instead of the approximation of overall cost. To enhance the performance of the controller, the farther objective may be expressed as follows:

1) Are able to design the vehicle longitudinal controller based on adaptive critic designs, but without the model directly or indirectly. This is not easy especially if we want the critic network to output the derivative of the overall cost.

2) Are able to expedite the learning procedure. This is especially important for practical problems.

It should be mentioned that the objective of employing adaptive critic designs does not preclude the objective of employing reinforcement learning, on the contrary, the former is the extension and improvement of latter.

1.2 Motivation and Rationale

Our research work is significant both for theory development and for practical applications. As we have mentioned in the beginning of this chapter, the design of autonomous vehicle control system is an important part of AHS. Traffic congestion is a big problem today, the principal motivation for an AHS is to increase capacity. One can also argue that an AHS will be safer, since data suggest that human error accounts for 90% of accidents. Estimates of the actual increase in capacity that an AHS would provide range from factors of 2 to 6 over current peak capacities (about 2000 vehicles/lane/hour) [Hedrick 1994]. Meanwhile, AHS would reduce emissions and fuel consumption. While full automation is the long-term goal, AHS deployment is likely to proceed in incremental stages, utilizing available results as early as possible [Yanakiev 2001]. Our research work is specific for autonomous vehicle longitudinal control system and can been thought as the first stage of this direction.

However, our research work does not only simply apply the existing algorithms to autonomous vehicle control problems. We have proposed the theory behind the applications. In fact, the theory development is a more important part of our research.

For known model structure but unknown parameters, we propose the adaptive fuzzy control design. Since the adaptive fuzzy controller can adjust all its parameters to the changing environment, better performance is usually achieved compared with fuzzy controller with fixed membership function and consequences of fuzzy rules.

If we lack of model information, we can design a fuzzy controller based on reinforcement learning. The proposed controller can be adaptive and improve its

performance purely based on its interaction with environment.

To enhance the learning efficiency of the controller based on reinforcement learning, we can improve it based on adaptive critic designs.

The theory contributions of our research work are that it provides several applicable design methods to design a controller which is adaptive and requires as little priori knowledge as possible.

1.3 Outline of the Thesis

We begin this thesis in chapter 2 with an introduction of intelligent control and autonomous vehicle control. We emphasize the concepts of "adaptive" and "fuzzy" because they will guide the controller design in the following chapters. For autonomous vehicle control, we present the background and define our research direction. We focus on vehicle longitudinal control, so we also review the existing algorithms which include conventional approaches and intelligent control approaches in this area.

We then go on to address the adaptive fuzzy controller in chapter 3. In this chapter, we mainly aim at how to approximate the ideal controller with fuzzy systems for a special class of nonlinear systems. The adaptation of the parameters of fuzzy rules is developed based on Lyapunov synthesis approach. We verify the proposed method by simulation studies.

Chapter 4 discusses how to combine the fuzzy controller with reinforcement learning to alleviate the requirement of the system model in chapter 3. We can expect

that the proposed controller can tune the parameters on-line and acquire some experience of the world. Consequently, the performance of the controller is improved step and step. We illustrate the effectiveness of the proposed controller for the vehicle longitudinal control problem.

Chapter 5 shows the improvement of the controller performance based on dynamic heuristic programming. We give the detailed design procedure for vehicle longitudinal controller without vehicle models. Special consideration is dedicated to reduce the computational complexity and speed up the training procedure. We also compare the performance of the controller of this chapter with that of chapter 4.

Finally, chapter 6 summarizes the contributions made by this thesis, discusses their relevance and suggests fruitful directions for further work.

1.4 Statements of Originality

The main contributions made by the author in this thesis are given in the following statements:

- A novel adaptive fuzzy controller is proposed based on some model information of vehicle model. It can adjust the parameters of the consequences of fuzzy rules as well as those of the membership functions of fuzzy systems. Consequently, a stable and more flexible controller is achieved, compared with fuzzy controller with fixed fuzzy rules.
- A new approach is suggested for tuning parameters of fuzzy controllers. The parameters of fuzzy controllers are tuned based on reinforcement learning with only the "evaluative signal". The adaptation of the parameters of fuzzy controllers does not require any model information. Unlike some existing approaches that

select an optimal action based on finite discrete actions, the proposed controller obtains the continuous control output directly.

- A vehicle longitudinal controller is presented based on adaptive critic designs. It can tune its parameters through the interaction with environment. It has the advantages of adaptive, no need of "teacher" signal. And some priori knowledge can be incorporated into the controller due to fuzzy systems employed. In addition, unlike some existing controllers based on ACD, the system model, which may be obtained before hand or by identification, is not required for the proposed controller.
- In this thesis, we want to achieve a trade-off between the controller performance and model information required.

1.5 Publications

At the time of writing this thesis, 9 conference papers have been published / accepted as below. Also, there are 3 papers that have been submitted to international journals. The full list is as follows:

1. Dai, X., Li, C. K. and Rad, A. B. "Performance Comparison of Autonomous Vehicle Controllers". *Mechanical and Electrical Engineering Technology*, Vol.31, no.6, pp.117-121 (2002)
2. Dai, X., Li, C. K. and Rad, A. B. "A novel adaptive fuzzy controller for application in autonomous vehicles". *Mechanical and Electrical Engineering Technology*, Vol.31, no.6, pp.121-126 (2002)
3. Dai, X., Li, C. K. and Rad, A. B. "The model of the artificial immune response". *The 9th International Conference on Enhancement and Promotion of*

Computational Methods in Engineering and Science, Macao, August 5-8, 2003. (This paper will also be collected in a book entitled "EPMESC IX - Computational Methods in Engineering and Science" to be published by A. A. Balkema of the Swets & Zeitlinger Publishers.

4. Dai, X., Li, C. K. and Rad, A. B. "Adaptive Control of a Class of Nonlinear Systems with Fuzzy Approximators". *The 12th IEEE International Conference on Fuzzy Systems*, St. Louis, USA, May 25-28, 2003, pp.384-389 (2003)
5. Dai, X., Li, C. K. and Rad, A. B. "An Approach to Tune Fuzzy Controllers Based on Reinforcement Learning". *The 12th IEEE International Conference on Fuzzy Systems*, St. Louis, USA, May 25-28, 2003, pp.517-522 (2003)
6. Dai, X., Li, C. K. and Rad, A. B. "A Novel Adaptive Fuzzy Controller for Application in Autonomous Vehicles". *IEEE Transactions on Vehicular Technology*, submitted.
7. Dai, X., Li, C. K. and Rad, A. B. "An Approach to Tune Fuzzy Controllers Based on Reinforcement Learning for Autonomous Vehicle Control". *IEEE Transactions on Industrial Electronics*, submitted.
8. Dai, X., Li, C. K. and Rad, A. B. "Autonomous Vehicle Longitudinal Control Based on Adaptive Critic Designs". *IEEE Transactions on Systems, Man and Cybernetics Part B*, submitted.
9. Li, C.K., Tao, T. and Dai, X. "A dual adaptive model estimator for target tracking". *The 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Hong Kong, April 6-10, 2003, pp. VI_53-VI_56 (2003).

Chapter 2

Literature Review

The purpose of this chapter is to set the scene for the rest of the thesis. The intention is to give a very selective yet sufficient overview of the general literature directly related to the work reported in the thesis. Those citations closely related to a particular topic of interest in this thesis are cited in the literature survey of the relevant chapter. Although, the focus is the autonomous vehicle control, we also briefly explain the area of intelligent control. In this research, the main element of intelligent control is "fuzzy system". To increase the flexibility of controller, we also consider the concept of "adaptive". The two basic principles in this thesis are "fuzzy" and "adaptive". These two concepts will guide the controller design in the following chapters.

For autonomous vehicle control, we present the background and define our research direction. We focus on vehicle longitudinal control and review the existing algorithms which include conventional approaches and intelligent control approaches in this area.

2.1 Intelligent and Autonomous Control

2.1.1 Intelligent Control

The literature of control theory is abundant of references to intelligent control especially in the last two decades. Academic researchers and industrial experts have shown great interest in intelligent control and have provided solutions to problems

that can not be well addressed by mathematical model-based techniques. The concept of intelligent control is based on a joint understanding of the notions of "control systems" and "intelligent systems". It emerged as an interdisciplinary field of computer controlled systems and artificial intelligence in the late seventies or early eighties [Hangos et al. 2001].

A control system, that employs the techniques of fuzzy systems, neural networks, expert and planning systems, and genetic algorithms, may be considered as intelligent control since these techniques are derived from human reasoning or biological intelligence.

The traditional control has encountered many difficulties in its applications [Cai 1997]. First of all, the design and analysis for the traditional control systems are based on their precise models that are usually difficult to achieve owing to complexity, nonlinearity, uncertainty, time-varying, and incomplete characteristic of the existing practical systems. Secondly, some critical hypotheses have to be put forward in studying and modeling the control systems; however, these hypotheses are hard to match in practice. Thirdly, in order to increase the control performances, the complexity of systems has to be increased too. As a result, the reliability of the control systems would be decreased. One of more effective ways to solve the above problems is to use intelligent control, since intelligent control normally may not rely on the development and use of a mathematical model of the process to be controlled.

Autonomous systems have the capability to independently (and successfully) perform complex tasks without human interactions. Consumer and governmental

demands for such systems are frequently forcing engineers to push many functions normally performed by humans into machines. The general trend has been for engineers to incrementally "add more intelligence" in response to consumer, industrial, and government demands and thereby create systems with increased levels of autonomy. In this process of enhancing autonomy by adding intelligence, engineers often study how humans solve problems, and then try to directly automate their knowledge and techniques to achieve high levels of automation.

From above, we can see that it is a good way to use intelligent control methods to design autonomous control systems. In our research, we need to deal with autonomous vehicle control problems. Naturally, we think of intelligent control approaches. We partially adopt fuzzy systems, neural networks and reinforcement learning for autonomous vehicle control. We hope to take the advantages of intelligent control to handle the un-modeled dynamics of the vehicle and the uncertainty of the environment in a comparatively simple way.

2.1.2 Adaptive Control

Although, intelligent control can handle incomplete information about the controlled object and environment to some extent, it would be better if we consider the idea of "adaptive". We may improve the controller performance further if we can make the controller adaptive.

In English language, "to adapt" means to change a behavior to conform to new circumstances. Although a meaningful definition of adaptive control is still lacking.

We take the definition in the book of Astrom, K.J. and Wittenmark, B. [Astrom & Wittenmark 1995]:

An adaptive controller is a controller with adjustable parameters and a mechanism for adjusting the parameters.

The key element in the adaptive controller is the parameter adjustment mechanisms. There are several ways to tune parameters, such as gain scheduling, model-reference adaptive control, self-tuning control etc.

"Adaptation" is attractive because an adaptive controller can be expected to accommodate for a higher degree of uncertainty than a fixed control structure. So we consider adaptation of the controller as the basic requirement during the controller design procedure. Here, we should note that "intelligent control" is the upper level approach, while "adaptive" is the lower level method. We hope to combine the strengths of these two approaches to achieve better performance.

2.2 A Review of Techniques Relevant to This Thesis

The main techniques, which we use for intelligent control of autonomous vehicles, are fuzzy control, reinforcement learning and adaptive critic designs.

2.2.1 Fuzzy Control

Fuzzy control [Passino 2001] is a methodology to represent and implement a (smart) human's knowledge about how to control a system. The main advantage of fuzzy control is that it provides a heuristic (not necessarily model-based) approach to nonlinear controller construction. In this thesis, we employ a Takagi-Sugeno type

fuzzy controller since it has a mathematical expression as its consequence [Takagi & Sugeno 1985]. This makes it easy to analyze theoretically.

In the earlier stage of fuzzy controller design, the fuzzy rules and their membership functions were designed by the controller designers through translating the operator's manual control. This information, through a process of trial and error, generated a rule table that could be implemented in the form of if-then rules. This heuristic-based design is time consuming and has been the subject of some criticism to design of fuzzy systems. Although the achievements of heuristic-based fuzzy control has been significant, the design process has been viewed to be not rigorous due to its lack of formal synthesis techniques, which guarantee the basic requirements for control systems such as global stability and acceptable performance. The emergence of the model-based design of fuzzy controllers [Driankov & Palm 1998] has provided alternative solution. The model-based design combines the conventional/modern control theory with the fuzzy logic control. The major objective of the model-based fuzzy control is to use the full available knowledge of existing linear and nonlinear design and analysis methods to achieve better performances than either fuzzy control or conventional control acting alone. The model-based fuzzy control shares the advantages of analysis of stability, performance and robustness with classical methods. It also has the advantages of fuzzy control such as incorporating the knowledge of human experts or operators.

One of the model-based fuzzy control is adaptive fuzzy control. Fuzzy controllers are supposed to handle incomplete information. Adaptive control is to maintain

consistent performances of a system in the presence of uncertainties. Therefore, adaptive fuzzy control may have the both advantages of fuzzy control and adaptive control.

The basic idea of adaptive fuzzy control is as follows: For a plant with unknown components, a fuzzy system is used to approximate the ideal controller directly, or construct the controller indirectly by approximating the plant model. The parameters of the fuzzy system are adjustable online such that the plant output tracks the reference model output.

Adaptive fuzzy control and conventional adaptive control have similarities and differences. They are similar in: i) the basic configuration and principles are more or less the same, and ii) the mathematical tools used in the analysis and design are very similar. The main differences are: i) the fuzzy controller has a special nonlinear structure that is universal for different plants, whereas the structure of a conventional adaptive controller changes from plant to plant, and ii) human knowledge about the plant dynamics and control strategies can be incorporated into adaptive fuzzy controllers, whereas such knowledge is not considered in conventional adaptive control systems. This second difference identifies the main advantage of adaptive fuzzy control over conventional adaptive control.

The main advantages of adaptive fuzzy control over nonadaptive fuzzy control are: i) better performance is usually achieved because the adaptive fuzzy controller can adjust itself to the changing environment, and ii) less information about the plant is required because the adaptation law can help to learn the dynamics of the plant during

real-time operation.

The main disadvantages of the adaptive fuzzy control over nonadaptive fuzzy control are: i) the resulting control system is more difficult to analyze because it is not only nonlinear but also time varying, and ii) implementation is more costly.

2.2.2 Reinforcement Learning

Reinforcement learning (RL) has attracted considerable attention in the past because it provides an effective approach to control and decision problems for which optimal solutions are analytically unavailable or difficult to obtain. Reinforcement learning is based on the common sense idea that if an action is followed by a satisfactory state, or by an improvement, then the tendency to produce that action is strengthened, i.e. reinforced. In essence, reinforcement learning is a direct adaptive optimal control [Sutton et al. 1992].

In reinforcement learning, the system is told indirectly about the performance of the current control action through evaluation signal. The study of reinforcement learning relates to the credit assignment where, given the performance of a process, one has to assign the reward or blame attribute to the individual elements contributing to that performance. Temporal difference (TD) methods can be used to solve the temporal credit assignment problem [Sutton & Barto 1998].

From a historical perspective, Sutton and Barto [Sutton & Barto 1998] identified two key research trends that led to the development of reinforcement learning: the trial and error learning from psychology and the dynamic programming methods from

mathematics.

It is no surprise that the early researchers in reinforcement learning were motivated by observing animals (and people) learning to solve complicated tasks. Notably, Roger Thorndike's work in operant conditioning identified an animal's ability to form associations between an action and a positive/negative reward that follows [Thorndike 1911].

The other historical trend in reinforcement learning arises from the "optimal control" work performed in the early 1950s. By "optimal control", we refer to the mathematical optimization of reinforcement signals. Today, this work falls into the category of dynamic programming and should not be confused with the optimal control techniques of modern control theory. Mathematician Richard Bellman is deservedly credited with developing the techniques of dynamic programming to solve a class of deterministic "control problems" via a search procedure [Bellman 1957]. By extending the work in dynamic programming to stochastic problems, Bellman and others formulated the early work in Markov decision processes.

Barto and others combined these two historical approaches in the field of reinforcement learning. The reinforcement learning agent interacts with an environment by observing states, x, and selecting actions, a. After each moment of interaction (observing x and choosing a), the agent receives a feedback signal, or reinforcement signal, r, from the environment. This is much like the trial and error approach from animal learning and psychology. The goal of reinforcement learning is to devise a control algorithm, called a policy, that selects optimal actions (a) for each

observed state (x). By optimal we mean those actions which produce the highest reinforcements (r) not only for the immediate action, but also for future actions not yet selected. The mathematical optimization techniques of Bellman are integrated into the reinforcement learning algorithm to arrive at a policy with optimal actions.

Reinforcement learning can solve some difficult and diverse control problems and has some successful applications. Crites and Barto successfully applied reinforcement learning to control elevator dispatching in large scale office buildings [Crites & Barto 1996]. Their controller demonstrates better service performance than state-of-the-art, elevator-dispatching controllers. To further emphasize the wide range of reinforcement learning control, Singh and Bertsekas have out-competed commercial controllers for cellular telephone channel assignment [Singh & Bertsekas, 1996]. There has also been extensive application to HVAC control with promising results [Anderson et al. 1996]. Early applications of reinforcement learning include world-class checker players [Samuel 1959] and backgammon players [Tesauro 1994]. Inverted pendulum [Si & Wang 2001], mountain car [Jouffe 1998], and robot navigation [Zalama et al. 2002] etc. have emerged as benchmarks for reinforcement learning studies.

2.2.3 Adaptive Critic Designs

Adaptive Critic Designs are new optimization techniques based on the concepts of reinforcement learning and approximate dynamic programming [Werbos 1990, Prokhorov et al. 1995, Prokhorov & Wunsch 1997, Prokhorov 1997, Eaton et al. 2000,

Venayangamoorthy et al. 2002]. Adaptive Critic Designs are suitable for a given series of control actions which must be taken sequentially, and not knowing the effect of these actions until the end of the sequence, which may be difficult for supervised learning [Venayangamoorthy et al. 2002]. More important, adaptive critic designs may not need the training data or system models. The only requirements for ACD are to know the final cost and the one-step cost.

Dynamic programming (DP) methods use the principle of optimality [Bellman 1957] to find a strategy of action that optimizes a desired performance metric or cost subject to nonlinear dynamical constraints. Although the backward DP approach reduces the space of admissible solutions, it remains computationally too expensive for higher dimensional systems, with a large number of stages. The required multiple generation and expansion of the state and the storage of all optimal costs lead to a number of computations that grows exponentially with the number of state variables, commonly referred to as the "curse of dimensionality" or "expanding grid" [Kirk, 1970]. Approximate dynamic programming (ADP) and temporal difference methods use incremental optimization combined with a parametric structure to reduce the computational complexity associated with evaluating the cost [Bellman 1973]. Unlike discrete DP, ADP algorithms progress forward in time, and approximate both the optimal policy and the cost in real time by considering only the present value of the state.

Adaptive critic designs (ACD) reproduce the most general solution of ADP by deriving recurrence relations for the optimal policy, the cost, and, possibly, their

derivatives. The goal is to overcome the curse of dimensionality, while ensuring convergence to a near-optimal solution over time [Howard 1960, Bertsekas & Tsitsiklis 1996]. Adaptive critics offer a unified approach to dealing with the controller's nonlinearity, robustness, and reconfiguration for a system whose dynamics can be modeled by a general ordinary differential equation.

The simplest form of adaptive critic design, Heuristic Dynamic Programming (HDP), uses a parametric structure called an action network to provide control output $u(t)$ which approximates the control policy and another parametric structure called a critic network to approximate the overall cost $J(t)$. In practice, since the parameters of this architecture adapt only by means of the scalar cost, HDP has been shown to converge very slowly [Werbos 1990].

An alternative approach referred to as Dual Heuristic Programming (DHP) has been proposed [Werbos 1990, 1997]. Here, the critic network approximates the derivatives of $J(t)$ with respect to the state, thereby correlating the adjustable parameters in the architecture to a larger number of dependent variables. Although the advantages of DHP over HDP have been discussed extensively in the literature from a theoretical point of view, few successful implementations have been reported. Due to the use of derivative information, the recurrence relations that can be obtained for DHP are more involved and may require an accurate model of the system to be controlled. The critic network approximates a nonlinear mapping characterized by a much larger-dimensional output space. Therefore, practical aspects such as function approximation are more challenging in DHP than they are in HDP.

Many other methodologies have been proposed over the years to alleviate some of the difficulties mentioned above, producing more advanced designs. Global Dual Heuristic Programming (GDHP), for example, has been developed with the purpose of combining the advantages of both HDP and DHP architectures. In this case, the critic network approximates both the overall cost $J(t)$ and its derivatives.

Action-dependent (AD) versions of all these approaches are obtained by designing a critic network that has direct knowledge of the control policy (produced by the action network) through its inputs, as opposed to only having knowledge of its derivatives through its adaptation (as in the action-independent ACD designs). The motivation behind this is convenient implementation for action-dependent ACD designs. To adapt the action network, we ultimately need the derivative $\partial J(t)/\partial u(t)$, rather than $J(t)$ itself. This problem becomes simple if the input of critic contains $u(t)$, because we can get $\partial J(t)/\partial u(t)$ directly from critic network based on backpropagation.

Action-dependent versions of HDP, DHP and GDHP can be named as ADHDP (action-dependent heuristic dynamic programming), ADDHP (action-dependent dual heuristic programming) and ADGDHP (action-dependent global dual heuristic programming) respectively. ADHDP is similar with Q learning.

From above, we can see that adaptive critic designs can be divided into three categories, and each of the categories may also have the action dependent version.

2.3 Autonomous Vehicle Control

2.3.1 Background

Traffic congestion problems and driving safety issues on highways have motivated an increased amount of research on highway automation and it is being investigated worldwide in several programs, such as ITS in the US and PVS, SSVS, and ARTS under ITS Japan. Readers who are interested can refer the comprehensive overviews of highway automation which are given by Bender [Bender 1991] and Shladover [Shladover 1995]. The field of AHS is very broad. From the control architecture, it can consist of network layer, link layer, coordination layer, regulation layer and physical layer. From the functions of AHS, it can be divided into sensors, signal processing, control computation, control actuation and vehicle dynamics, etc [Varaiya 1993, Sheikholeslam & Desoer 1993, Raza & Ioannou 1996, Ioannou & Bose 1999, Horowitz & Varaiya 2000].

There are three AHS control tasks: 1) To assign a path to each vehicle. 2) To carry out safely the maneuvers of platoon formation, stabilization and dissolution, lane change, and entry and exit. 3) To implement those maneuvers via feedback laws (algorithms) that control each vehicle's throttle, braking and steering actuators [Hedrick 1994].

The design of intelligent vehicle control system is an important part of AHS, and the vehicle control may consider the maneuvers such as lead vehicle tracking, follower, join, split, lane change, entry and exit.

While full automation is the long-term goal, AHS deployment is likely to proceed in incremental stages, utilizing available results as early as possible [Yanakiev & Kanellakopoulos 2001]. In the first stage, for example, vehicles would have only longitudinal control capabilities for vehicle following without inter-vehicle communication, with the driver assuming responsibility for steering and emergency situations. In that respect, systems currently in various stages of research and development can be classified into three categories:

• Autonomous systems: depend only on information obtained by the sensors located on the vehicle itself, usually relative distance and velocity to stationary objects and moving vehicles. They are, therefore, implementable in the immediate future and, in fact, have started to appear as commercial products (collision warning, adaptive cruise control).

• Cooperative systems: add information transmitted by neighboring vehicles, usually acceleration and steering inputs. Hence, they can perform more demanding tasks than autonomous systems such as coordinated driving in a group, but their time to commercialization is likely to be longer.

• Automated highway systems: add information obtained from the roadway infrastructure such as messages regarding traffic conditions and road geometry and lateral information from magnetic nails or reflective guardrails installed on the highway. Such systems can perform even more demanding tasks, like fully automated driving in a platoon, but must face many more obstacles (standardization, liability issues, public acceptance) on their way to implementation.

Important research programmes like PROMETHEUS (PROgraM for Europe Traffic with Highest Efficiency and Unprecedented Safety) in Europe, PATH (Program on Advanced Technology for Highway) in the USA or PVS (Personal Vehicle System) in Japan enabled great cooperation between car manufactures and sensor or actuator suppliers, and have contributed much to this community [Vlacic et al. 2001]. However, the largest and richest body of research has been in vehicle-following longitudinal control system, especially in more recent years. From the late 1970s, most relevant work has included nonlinear, time varying and adaptive control considerations [Shladover 1995].

2.3.2 Scope and Direction of Research

Autonomous vehicle control is a complex control task. The vehicle model is a highly nonlinear system with dynamic characteristics that vary with the changed operating conditions, such as velocity, road conditions etc. Also, there are time delays with the vehicle engine. The vehicle longitudinal model, which is developed by the longitudinal control group of University of California at Berkeley, is represented by twelve state variables: four for the engine, two for the transmission, and six for the drive wheel, plus two time delays associated with the engine [Guldner et al. 1997]. Although we may use a linearized model for vehicle lateral control, this is only valid under the assumption of small angles during normal highway driving conditions (i.e. non-emergency situations) within the physical limits of tires [Hedrick et al. 1993]. As a result, effective vehicle control is not a trivial task.

Automated vehicle longitudinal control and lateral control are two important parts of Autonomous vehicle control. Lateral control maintains the vehicle in the center of the lane (lane-keeping maneuver) and steers the vehicle to an adjacent lane (lane-change maneuver), while maintaining good passenger comfort all times. Longitudinal control assumes cars are driven along a line on the highway and consists of the desired following distance and the design of a control system that regulates the speed of the vehicle in accordance with the given spacing policy.

In this thesis, we emphasize automated vehicle longitudinal control when a constant spacing policy is employed by an autonomous vehicle. The task of vehicle longitudinal control is to regulate both the relative velocity v_r and the spacing deviation Δx of the preceding car and following car to zero. And this task can be combined into the control objective $v_r + k \cdot \Delta x = 0$, where k is a positive design constant. This control objective makes sense intuitively: if two vehicles are closer than desired ($\Delta x < 0$) and the control objective is satisfied ($v_r = -\Delta x > 0$), then the following car is moving slower than the preceding car, which is what we expect. And the situation is also what we want if $\Delta x > 0$.

If we think the position and velocity of the preceding car is v_l and x_l respectively, those of the following car are x and v respectively, then we define

$$v_r(t) = v_l(t) - v(t), \quad \Delta x(t) = x_l(t) - x(t) \tag{2-1}$$

2.4 A Brief Review of Vehicle Longitudinal Control

A number of publications have reported on the autonomous vehicle control problem

and there are many existing control algorithms to deal with the vehicle longitudinal (car-following) problems in the literatures. In the following subsections, we will introduce the conventional approaches and intelligent control approaches respectively.

2.4.1 Conventional Approaches

The simplest controller is PD controller, which comes from the decision making of human drivers. The human driver senses the distance between his car and the front car, and also estimates the possible change of the distance based on the velocity of the controlled car and the front car. The PD controller is very simple, however, the performance of the PD controller is very limited. Ioannou et al. linearized the vehicle model around the operating point and employed the PID controller with fixed gain and gain scheduling scheme [Ioannou et al. 1993]. They also adopted Lyapunov approach to design an adaptive controller which eliminated the weakness of the PID controller, because the latter is based on look-up tables that are developed a priori by performing certain experiments.

Yanakiev and Kanellakopoulos also used the linearized vehicle model, but they added a signed Quadratic (Q) term in the PI controller to make the controller more aggressive at large errors, but it does not have the undesirable side effect of overshoot [Yanakiev & Kanellakopoulos 1996, 2001].

Ioannou and Chien also proposed an autonomous intelligent cruise control (AICC), which followed directly from the theory of feedback linearization where one part of the control action is used to cancel the nonlinearities, and the other part is used to

assign the eigenvalues of the resulting linear system [Ioannou & Chien 1993]. Hedrick, Swaroop et al. design the longitudinal controller using "Multiple Sliding Surface" methodology, which is closely related to the sliding mode control [Hedrick 1998, Swaroop et al. 2001]. The dynamic surface controller can achieve very good performance. However, it requires communication with the leading vehicle.

So far, many longitudinal controllers have been proposed. Each controller is designed for different a purpose, therefore the control laws and performances can not be compared against each other. However, all studies above depended upon a precise vehicle model to some extent. Deriving a precise model is difficult not only for the complexity of the vehicle system, but also for the uncertainties and disturbances of the vehicle due to numerous parameters including the vehicle, road, weather conditions, etc. Many researchers believe that intelligent control methods could be used to deal with vehicle longitudinal control problems.

2.4.2 Intelligent Control Approaches

To circumvent the complexity of vehicle models, intelligent control approaches, such as fuzzy and neuro-controllers have been applied to automated vehicle control system. Kehtarnavaz et al. [1994] generated the fuzzy rules of vehicle following controller by a self-organizing neural network. Kim et al. [1996] adopted neural systems to learn the fuzzy rules with both unsupervised and supervised learning, and implemented fuzzy throttle and brake control. Huang and Ren [1999] constructed a neural fuzzy network (NFN) for automated vehicle guidance control and used training data to

identify the weights of NFN by mixed genetic/gradient algorithm. Mar and Lin [2001] presented an ANFIS controller for car-following collision prevention system, which used the reference signal to on-line update the parameters of ANFIS.

The main advantages of these methods are that they don not require the exact model of vehicles and may not be sensible to imprecise data from sensor. But the tradeoff is that the performance of the controller depends much on training data and priori knowledge or experiences of human operators. Therefore, the simple design of fuzzy or neuro-controller may not be adequate. It would be better if fuzzy or neuro-controllers could tune their parameters to achieve better performance according to current situation or performance of the controller.

Chapter 3

Autonomous Vehicle Controller based on Adaptive Fuzzy Technique

The important design consideration of adaptive fuzzy control is how to tune the parameters of fuzzy controller, i.e., how to construct the adaptive laws. Recently, some researchers have proposed to construct adaptive fuzzy controllers by Lyapunov synthesis approach [Wang 1993, Su & Stepanenko 1994, Spooner & Passino 1996, Chen et al. 1996, Tsay et al. 1999, Han et al. 2001, Feng 2002]. Using this approach, we can adjust the parameters to guarantee the stability and also achieve better performance. The success of this approach owes to the combination of robust adaptive systems theory and fuzzy approximation theory, where the fuzzy controller is used to approximate the unknown system model or the controller.

However, most of the current research on adaptive fuzzy control only tunes the parameters of the consequences of fuzzy rules. This may cause the approximation property of fuzzy systems not to be adequate. It also affects the performance of the controller. Aiming at this problem, we intend to tune all the parameters of fuzzy controller. In order to tune these parameters, a linear relationship between approximation error and all parameters of fuzzy rules is established first. Then we design the adaptive laws of these parameters based on Lyapunov synthesis approach. The advantage of our method is that we can tune not only the parameters of the

consequences of fuzzy rules, but also the parameters of the membership functions. As a result, a stable and more flexible controller is achieved.

We have addressed several existing vehicle longitudinal controllers in Chapter 2. However, for the controllers which employ soft computing technology, the parameters of the controller may be fixed or tuned off-line. It would be attractive if the controller can be tuned on-line to improve its performance. This motivates adaptive fuzzy control for vehicle longitudinal control. We employ a direct adaptive fuzzy controller that approximates an ideal optimal controller. All parameters of the fuzzy controller are tuned on-line. In order to tune these parameters, a linear relationship between approximation error and parameters is first established. The corresponding adaptive laws are designed next based on Lyapunov synthesis approach. The advantage of the proposed method is that parameters of the consequences of fuzzy rules as well as those of the membership functions are tuned. As a result, a stable and more flexible controller is achieved.

The rest of this chapter is organized as follows. In section 3.1, the nonlinear model and control objective are presented. Since we aim to approximate the ideal controller using a fuzzy system, the fuzzy logic system is discussed in section 3.2, which is composed of two parts: the structure of fuzzy systems and the error of fuzzy approximators. Afterwards, section 3.3 explains our proposed adaptive fuzzy control based on Lyapunov synthesis approach. In section 3.4, vehicle longitudinal control is used to verify the theoretical analysis. Section 3.5 concludes this chapter.

3.1 Problems Formulation

We consider a specific n th-order nonlinear system of the form:

$$x^{(n)} = f(x, \dot{x}, \cdots, x^{(n-1)}) + bu_p$$

$$y = x \tag{3-1}$$

where, f is unknown continuous function and b is unknown constant, $u_p \in \Re$ and $y \in \Re$ are the input and output of the system respectively. We assume that the state vector $\underline{x} = (x_1, x_2, \cdots, x_n)^T = (x, \dot{x}, \cdots, x^{(n-1)})^T \in \Re^n$ is available for measurement. The control objective is to force the output of the system y to follow a given bounded reference signal $y_m(t)$ under the constraints that all signals involved are bounded.

Using feedback linearization, we know that there exists some ideal controller if f and g are known.

$$u^* = \frac{1}{b}[-f(\underline{x}) + v(t)] \tag{3-2}$$

where, $v(t) = y_m^{(n)} + \underline{k}^T \underline{e}$, $\underline{e} = (e, \dot{e}, \cdots, e^{(n-1)})^T$, $\underline{k} = (k_n, \cdots, k_1)^T$, $e = y_m - y$, let $\underline{k} = (k_n, \cdots, k_1)^T$ be such that all roots of the polynomial $h(s) = s^r + k_1 s^{r-1} + \cdots + k_r$ are in the open left half plane.

If we apply the above ideal controller (equation (3-2)) to equation 1, then we obtain:

$$e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0 \tag{3-3}$$

The above equation implies that $\lim_{t \to \infty} e(t) = 0$, because we choose $\underline{k} = (k_n, \cdots, k_1)^T$ as all roots of the polynomial $h(s) = s^r + k_1 s^{r-1} + \cdots + k_r$ being in the left half plane.

However, we cannot obtain the ideal controller u^* directly because f and b

are unknown. Since a fuzzy system can be considered as a universal approximator, we use a fuzzy controller to approximate the above unknown but existent ideal controller u^*. In this chapter, we construct the controller using a fuzzy system and directly adjust all parameters of the fuzzy controller, i.e., we concentrate on a direct fuzzy adaptive control. It is worth noting that our approach can tune all parameters (including the parameters of membership functions) of fuzzy rules.

3.2 Description of Fuzzy Logic Systems

Before the fuzzy adaptive controller is proposed, we discuss the structure and the approximation error of fuzzy logic systems we adopted.

3.2.1 Structure of Fuzzy Logic Systems

Consider a multiple-input single-output (MISO) fuzzy controller which performs a mapping from an state vector $\underline{x} = (x_1, x_2, ..., x_n)^T \in \Re^n$ to a control input $u \in \Re$. Using the Takagi-Sugeno model, the IF-THEN rules of the fuzzy controller may be expressed as:

$$R_l: \text{ IF } \quad x_1 \text{ is } F_1^l \text{ and} \cdots \text{and } x_n \text{ is } F_n^l$$

$$\text{THEN } u = K_1^l g_1(\underline{x}) + K_2^l g_2(\underline{x}) + \cdots + K_m^l g_m(\underline{x}) \tag{3-4}$$

where F_i^l is the label of the fuzzy set in x_i, for $l = 1,2,...,M$. $g_1(\underline{x}), g_2(\underline{x}), \ldots$, and $g_m(\underline{x})$ are any known function of the state vector. K_1^l, K_2^l ... and K_m^l are the constant coefficients of the consequent part of the fuzzy rule.

In this chapter, we would use product inference for the fuzzy implication and t norm, singleton fuzzifier and center average defuzzifier, consequently, the final output

value is:

$$u(\underline{x}) = \frac{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\mu^{F_i^l}(x_i)\right)\cdot\left(K_1^l g_1(\underline{x}) + \cdots + K_m^l g_m(\underline{x})\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\mu^{F_i^l}(x_i)\right)} \tag{3-5}$$

Here, we adopt Gaussian function as the membership function of the fuzzy system because its excellent approximation properties [Liu & Si 1994], i.e.

$$\mu^{F_i^l}(x_i) = \exp\left(-\left(\frac{x_i - c_i^l}{\sigma_i^l}\right)^2\right) \tag{3-6}$$

for $i = 1,2,...,n$ and $l = 1,2,...,M$.

And we can rewrite the (3-5) as:

$$u(\underline{x}) = \theta^T \xi(\underline{x}) = \theta^T \xi(\underline{x} \mid c, \sigma) \tag{3-7}$$

where $\theta = (K_1^1, \cdots, K_m^1, K_1^2, \cdots, K_m^2, \cdots, K_m^M)^T$ is a parameter vector, c, σ are vectors with the elements of c_i^l and σ_i^l in equation (3-6) respectively, and $\xi(\underline{x}) = (\xi_1^1(\underline{x}), \cdots, \xi_m^1(\underline{x}), \xi_1^2(\underline{x}), \cdots, \xi_m^2(\underline{x}), \cdots, \xi_m^M(\underline{x}))^T$ is a regressive vector with the regressor $\xi_j^l(\underline{x})$ defined as

$$\xi_j^l(\underline{x}) = \frac{\left(\prod_{i=1}^{n}\mu^{F_i^l}(x_i)\right)\cdot g_j(\underline{x})}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\mu^{F_i^l}(x_i)\right)} \tag{3-8}$$

3.2.2 Error of Fuzzy Approximators

We have mentioned that we use fuzzy systems to approximate the ideal controller. The fuzzy systems approximate a function by covering the whole input space with fuzzy rule patches and averaging patches that overlap. The approximation properties of fuzzy systems are related to the number of fuzzy rules, the shape of membership

functions (such as triangle, trapezoid, Gaussian, etc.) and parameters of each fuzzy rule (including parameters of the membership functions and parameters of the consequences).

Theoretically, the fuzzy system discussed in section 3.3.1, that is the fuzzy systems with product inference engine, singleton fuzzifier, center average defuzzifier and Gaussian membership functions, can approximate any square-integrable function on the compact set $U \subset \Re^n$ to arbitrary accuracy [Wang & Mendel 1992].

However, we may or may not find the above ideal fuzzy system which can approximate any function to arbitrary accuracy due to some constraints such as the number of fuzzy rules and the parameters of the membership functions. We will discuss the error of fuzzy approximators next within this subsection.

Unlike some existing papers [Wang 1993, Spooner & Passino 1996, Tsay et al. 1999, Feng 2002], in which only adjusting the parameters of the consequences (θ of section 3.3.1) is considered, we also consider how to adjust the parameters of the membership functions (c, σ of section 3.3.1), such that the approximation properties of the fuzzy system may be improved. However, tuning the parameters of the Gaussian membership function is not a trivial work because the nonlinear relations among the parameters. Some methods have been proposed to tune the parameters of the membership functions, however, most of these methods require the training data [Homaifar & McCormick 1995, Kim 1997].

We define the approximation error between our controller and the ideal controller which is mentioned in section 3.2 as:

$$\varepsilon(\underline{x}) = u^*(\underline{x}) - u(\underline{x}) \tag{3-9}$$

Theorem 3-1: The function approximation error $\varepsilon(\underline{x})$ can be expressed as:

$$\varepsilon(\underline{x}) = \phi_\theta^T \xi(\underline{x} \mid c, \sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + d_u(\underline{x}) \tag{3-10}$$

where,

$$\phi_\theta = \theta^* - \theta,\ \phi_c = c^* - c,\ \phi_\sigma = \sigma^* - \sigma,\ u_c = \frac{\partial u}{\partial c},\ u_\sigma = \frac{\partial u}{\partial \sigma} \tag{3-11}$$

$$|d_u| \le w^{*T} \cdot Y \tag{3-12}$$

(where θ^*, c^*, σ^* represent the optimal parameter vectors of fuzzy approximator, $w^* \in \Re^4$ is an unknown constant vector which is related to some bounded constants and $Y = [1, \|\theta\|, \|c\| \cdot \|\theta\|, \|\sigma\| \cdot \|\theta\|]^T$)

Proof: We denote $\varepsilon_u(\underline{x})$ as the approximation error between the optimal fuzzy approximator and ideal controller, i.e.,

$$\varepsilon_u(\underline{x}) = u^*(\underline{x}) - u(\underline{x} \mid \theta^*, c^*, \sigma^*) \tag{3-13}$$

where, $u(\underline{x} \mid \theta^*, c^*, \sigma^*)$ represents the optimal fuzzy approximator with the change of parameters θ, c, σ, and θ^*, c^*, σ^* represent the optimal parameter vectors accordingly, i.e.,

$$(\theta^*, c^*, \sigma^*) = \arg\min_{\theta, c, \sigma} [\sup |u(\underline{x} \mid \theta, c, \sigma) - u^*(\underline{x})|] \tag{3-14}$$

We assume $\varepsilon_u(\underline{x})$ is bounded by a constant ε^*, i.e.,

$$|\varepsilon_u| \le \varepsilon^* \tag{3-15}$$

This is reasonable because the fuzzy system with Gaussian membership has good approximation properties, especially when all the parameters of fuzzy rules can be adjusted.

Then we can rewrite (3-9) and (3-13) as:

$$\varepsilon(\underline{x}) = u(\underline{x}\,|\,\theta^*, c^*, \sigma^*) + \varepsilon_u(\underline{x}) - u(\underline{x}\,|\,\theta, c, \sigma) \tag{3-16}$$

If we deal with $u(\underline{x}\,|\,\theta^*, c^*, \sigma^*)$ using Taylor's expansion at (θ, c, σ) based on $u(\underline{x}\,|\,\theta^*, c^*, \sigma^*) = \theta^{*T}\xi(\underline{x}\,|\,c^*, \sigma^*)$ and $u(\underline{x}) = \theta^T\xi(\underline{x}\,|\,c, \sigma)$, then

$$\begin{aligned} u(\underline{x}\,|\,\theta^*, c^*, \sigma^*) = u(\underline{x}\,|\,\theta, c, \sigma) + (\theta^* - \theta)^T \cdot \frac{\partial u}{\partial \theta} + (c^* - c)^T \cdot \frac{\partial u}{\partial c} + (\sigma^* - \sigma)^T \cdot \frac{\partial u}{\partial \sigma} \\ + o(\underline{x}\,|\,(\theta^* - \theta), (c^* - c), (\sigma^* - \sigma)) \end{aligned} \tag{3-17}$$

Let

$$\phi_\theta = \theta^* - \theta,\ \phi_c = c^* - c,\ \phi_\sigma = \sigma^* - \sigma,\ u_c = \frac{\partial u}{\partial c},\ u_\sigma = \frac{\partial u}{\partial \sigma} \tag{3-18}$$

Then based on (3-7), we have

$$\frac{\partial u}{\partial \theta} = \xi(\underline{x}\,|\,c, \sigma),\ u_c = \xi_c^{T'}\theta,\ u_\sigma = \xi_\sigma^{T'}\theta \tag{3-19}$$

where, $\xi_c^{T'}$ and $\xi_\sigma^{T'}$ is the derivative of vector $\xi(\underline{x}\,|\,c, \sigma)$ with respect to c and σ respectively.

Then from (3-17) we obtain:

$$u(\underline{x}\,|\,\theta^*, c^*, \sigma^*) = u(\underline{x}\,|\,\theta, c, \sigma) + \phi_\theta^T\xi(\underline{x}\,|\,c, \sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + o(\underline{x}\,|\,\phi_\theta, \phi_c, \phi_\sigma) \tag{3-20}$$

Let

$$d_u(x) = o(x\,|\,\phi_\theta, \phi_c, \phi_\sigma) + \varepsilon_u(x) \tag{3-21}$$

Using $u(\underline{x}) = \theta^T\xi(\underline{x}\,|\,c, \sigma)$, and combining (3-16), (3-18), (3-19), (3-20), (3-21) we have:

$$\begin{aligned} \varepsilon(\underline{x}) &= \phi_\theta^T\xi(\underline{x}\,|\,c, \sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + d_u(\underline{x}) \\ &= \phi_\theta^T\xi(\underline{x}\,|\,c, \sigma) + \phi_c^T\xi_c^{T'}\theta + \phi_\sigma^T\xi_\sigma^{T'}\theta + d_u(\underline{x}) \end{aligned} \tag{3-22}$$

From (3-20), we can express $\|o(\underline{x}\,|\,\phi_\theta, \phi_c, \phi_\sigma)\|$ as follow:

$$\begin{aligned} \|o(\underline{x}\,|\,\phi_\theta, \phi_c, \phi_\sigma)\| &= \left\|\theta^{*T}\xi(\underline{x}\,|\,c^*, \sigma^*) - \theta^T\xi(\underline{x}\,|\,c, \sigma) - \phi_\theta^T\xi(\underline{x}\,|\,c, \sigma) - \phi_c^T\xi_c^{T'}\theta - \phi_\sigma^T\xi_\sigma^{T'}\theta\right\| \\ &= \left\|\theta^{*T}[\xi(\underline{x}\,|\,c^*, \sigma^*) - \xi(\underline{x}\,|\,c, \sigma)] - (c^* - c)^T\xi_c^{T'}\theta - (\sigma^* - \sigma)^T\xi_\sigma^{T'}\theta\right\| \end{aligned} \tag{3-23}$$

Obviously, there should be constant $\bar{\theta}$, $\bar{c}$ and $\bar{\sigma}$ that satisfy:

$$\|\theta^*\| \le \bar{\theta}, \ \|c^*\| \le \bar{c} \text{ and } \|\sigma^*\| \le \bar{\sigma} \tag{3-24}$$

We assume $\xi(\underline{x} \mid c, \sigma)$ and the derivative of $\xi(\underline{x} \mid c, \sigma)$ with respect to c and σ are bounded.

$$\|\xi(\underline{x} \mid c, \sigma)\| \le c_1, \ \left\|\xi_c^{T'}\right\| \le c_2 \text{ and } \left\|\xi_\sigma^{T'}\right\| \le c_3 \tag{3-25}$$

So, we have:

$$|o(\underline{x} \mid \phi_\theta, \phi_c, \phi_\sigma)| \le 2\bar{\theta}c_1 + (\bar{c} + \|c\|) \cdot c_2 \cdot \|\theta\| + (\bar{\sigma} + \|\sigma\|) \cdot c_3 \cdot \|\theta\| \tag{3-26}$$

and

$$\begin{aligned} |d_u| &= |o(x \mid \phi_\theta, \phi_c, \phi_\sigma) + \varepsilon_u(\underline{x})| \\ &\le 2\bar{\theta}c_1 + (\bar{c} + \|c\|) \cdot c_2 \cdot \|\theta\| + (\bar{\sigma} + \|\sigma\|) \cdot c_3 \cdot \|\theta\| + \varepsilon^* \\ &= [w_1^*, w_2^*, w_3^*, w_4^*] \cdot [1, \|\theta\|, \|c\| \cdot \|\theta\|, \|\sigma\| \cdot \|\theta\|]^T = w^{*T} \cdot Y \end{aligned} \tag{3-27}$$

where, the facts $|\varepsilon_u| \le \varepsilon^*$, and ε^* is a constant, has been used, and

$$\begin{aligned} w_1^* &= 2\bar{\theta}c_1 + \varepsilon^* \\ w_2^* &= \bar{c}c_2 + \bar{\sigma}c_3 \\ w_3^* &= \bar{c}c_2 \\ w_4^* &= \bar{\sigma}c_3 \end{aligned} \tag{3-28}$$

Q.E.D.

Remarks:

1) In some previous works only tuning parameter θ is considered, the approximation error is expressed as $\varepsilon(\underline{x}) = \phi_\theta^T \xi(\underline{x}) + d_u(\underline{x})$ accordingly.

2) In order to tune parameters c, σ, we need to express the approximation error $\varepsilon(\underline{x}) = u^*(\underline{x}) - u(\underline{x})$ with the term $\phi_c = c^* - c$, $\phi_\sigma = \sigma^* - \sigma$.

3) We have made some assumptions here, i)$|\varepsilon_u| \le \varepsilon^*$, this is reasonable because we

can tune all the parameters of the fuzzy system so that the approximation error between the optimal fuzzy approximator and the ideal controller could be small. ii) the derivative of $\xi(\underline{x}\,|\,c,\sigma)$ with respect to c and σ are bounded, i.e. $\|\xi(\underline{x}\,|\,c,\sigma)\| \le c_1$, $\left\|\xi_c^T\right\| \le c_2$ and $\left\|\xi_\sigma^T\right\| \le c_3$, this may also be reasonable to the bounded x, c, σ.

4) $d_u(\underline{x})$ is a residual term of the approximation error $\varepsilon(\underline{x})$, $|d_u| \le w^{*T} \cdot Y$ is an important property for Lyapunov synthesis approach (see it in section 3.4). Although we do not know the value of w^* clearly, we can estimate it by adaptive laws (see it also in section 3.4).

Now, we discuss how to calculate u_c and u_σ in (3-10). We could obtain vectors u_c and u_σ if we could get the vector components of them.

If we let

$$z^l = \prod_{i=1}^{n} \exp\left(-\left(\frac{x_i - c_i^l}{\sigma_i^l}\right)^2\right)$$

$$\bar{y}^l = K_1^l g_1(\underline{x}) + \cdots + K_m^l g_m(\underline{x}) \tag{3-29}$$

$$a = \sum_{i=1}^{M} (\bar{y}^l z^l),\quad b = \sum_{l=1}^{M} z^l,\quad u = \frac{a}{b}$$

Then we can calculate the components of u_c and u_σ by the following equations:

$$\frac{\partial u}{\partial c_i^l} = \frac{\partial u}{\partial z^l} \cdot \frac{\partial z^l}{\partial c_i^l} = \frac{\bar{y}^l - u}{b} z^l \frac{2(x_i - c_i^l)}{(\sigma_i^l)^2} \tag{3-30}$$

$$\frac{\partial u}{\partial \sigma_i^l} = \frac{\partial u}{\partial z^l} \cdot \frac{\partial z^l}{\partial \sigma_i^l} = \frac{\bar{y}^l - u}{b} z^l \frac{2(x_i - c_i^l)^2}{(\sigma_i^l)^3} \tag{3-31}$$

3.3 Development of a Direct Adaptive Fuzzy Controller

Inspired from [Spooner & Passino 1996], we construct the control law as:

$$u_p = u + u_{co} + u_{bd} \tag{3-32}$$

The above equation means that the direct adaptive control law is comprised of a bounding control term, u_{bd}, a compensating control term u_{co}, and an adaptive fuzzy control term, u, which is mentioned in section 3.3 and is used to approximate the ideal controller. The compensating controller is used to compensate for approximation errors in representing the actual nonlinear dynamics by fuzzy systems with ideal parameter values. The bounding control is used to restrict the output trajectory of the system so that fuzzy systems may be defined for a small range of states. The bounding controller in this manner is similar to the supervisory control (described in the following sections). The structure of the proposed control is shown in Figure 3.1.

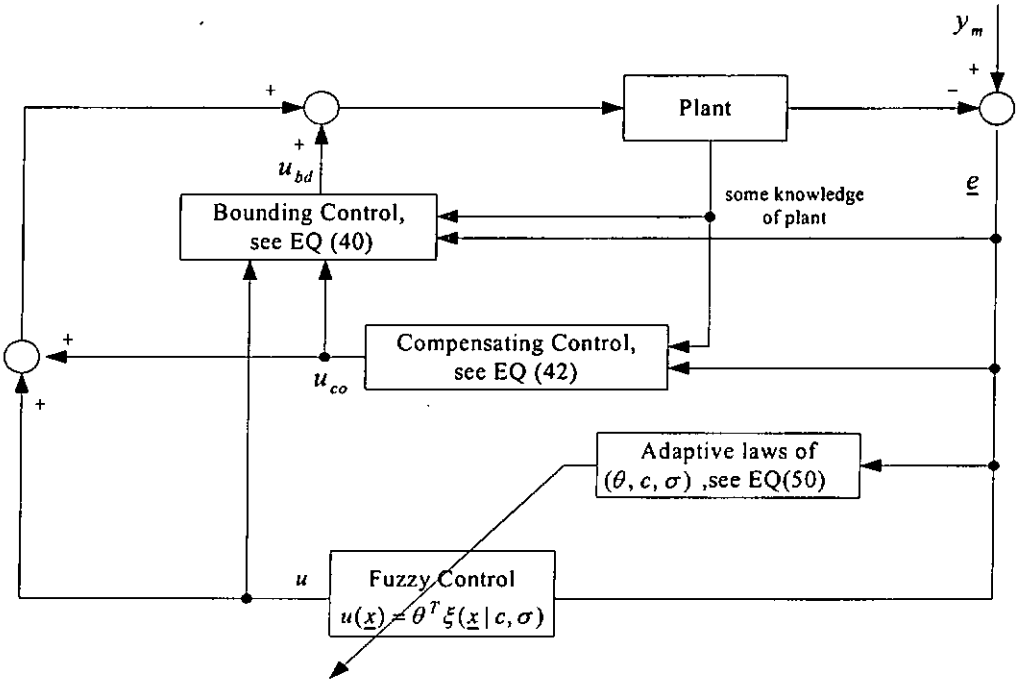


Figure 3.1: The structure of our proposed adaptive fuzzy control

After substituting the control law into the system, we will have

$$x^{(n)} = f(\underline{x}) + b[u + u_{co} + u_{bd}] \tag{3-33}$$

After some straightforward manipulation, we can obtain the error equation of the closed-loop system

$$\begin{aligned} \dot{\underline{e}} &= A\underline{e} + B[u^* - u - u_{co} - u_{bd}] \\ &= A\underline{e} + B[\phi_\theta^T \xi(x \mid c,\sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + d_u(x) - u_{co} - u_{bd}] \end{aligned} \tag{3-34}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & \cdots & \cdots & -k_1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ b \end{bmatrix} \tag{3-35}$$

3.3.1 Bounding Control

Define a function:

$$V_{bd} = \frac{1}{2}\underline{e}^T P \underline{e} \tag{3-36}$$

And because all roots of the polynomial $h(s) = s^n + k_1 s^{n-1} + \cdots + k_n$ are in the open left half plane ($\underline{k} = (k_n, \cdots, k_1)^T$ is user defined, which has been mentioned in section 3.2), we can find P which is a symmetric positive definite matrix satisfying the Lyapunov equation

$$A^T P + PA = -Q \tag{3-37}$$

where $Q > 0$. Differentiate the V_{bd} with respect to t, we have

$$\begin{aligned} \dot{V}_{bd} &= -\frac{1}{2}\underline{e}^T Q \underline{e} + \underline{e}^T PB[u^* - u - u_{co} - u_{bd}] \\ &\le -\frac{1}{2}\underline{e}^T Q \underline{e} + \left|\underline{e}^T PB\right|[|u^*| + |u| + |u_{co}|] - \underline{e}^T PBu_{bd} \end{aligned} \tag{3-38}$$

Assumption: We can determine a function $f^U(\underline{x})$ and constant b_L such that

$$|f(\underline{x})| \le f^U(\underline{x}) \text{ and } 0 < b_L \le b \tag{3-39}$$

This means we should have some knowledge of the system, but this is not very difficult to get.

Under the above assumption and (3-2), we could construct the bounding control u_{bd} as:

$$u_{bd} = I\,\mathrm{sgn}(\underline{e}^T PB)[|u| + |u_{co}| + \frac{1}{b_L}(f^U(\underline{x}) + |y_m^{(n)}| + |\underline{k}^T \underline{e}|)] \tag{3-40}$$

where $I = 1$ if $V_{bd} > \overline{V}$ ($\overline{V}$ is a constant specified by the designer) and $I = 0$ if $V_{bd} \le \overline{V}$: And due to $b > 0$ (equation (3-39)), we can evaluate the value of $\mathrm{sgn}(\underline{e}^T PB)$. So, when $V_{bd} > \overline{V}$, we have

$$\dot{V}_{bd} \le -\frac{1}{2}\underline{e}Q\underline{e} \le 0 \tag{3-41}$$

So, using the bounding control u_{bd}, we always have $V_{bd} \le \overline{V}$. This means we can restrict the state of the system in a desired range using the bounding control.

3.3.2 Compensating Control

We use the compensating control to compensate for the approximation error in modelling u^* by a fuzzy system.

From the equation (3-10), we know that $d_u(\underline{x})$ is a residual term of the approximation error $\varepsilon(\underline{x})$, and $d_u(\underline{x})$ can not be expressed by linear combination of parameter (θ, c, σ). To reduce the negative effect of $d_u(\underline{x})$ to our defined Lyapunov functions, we consider the compensating control as:

$$u_{co} = \mathrm{sgn}(\underline{e}^T PB) w^T Y \tag{3-42}$$

And also, due to $b>0$, we can evaluate the value of $\text{sgn}(\underline{e}^T PB)$. However, in order to avoid chattering of the system response around the equilibrium point where the system error is zero, we can simply modify the equation of u_{co} to:

$$u_{co} = w^T Ysat((\underline{e}^T P_n)/\varepsilon) \tag{3-43}$$

where

$$sat(x) = \begin{cases} 1, & x \geq 1 \\ x, & 1 > x > -1 \\ -1, & x \leq -1 \end{cases} \tag{3-44}$$

ε is a constant specified by the designer and $\varepsilon > 0$.

3.3.3 Adaptive Laws

Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}\underline{e}^T P\underline{e} + \frac{b}{2\gamma_1}\phi_\theta^T\phi_\theta + \frac{b}{2\gamma_2}\phi_c^T\phi_c + \frac{b}{2\gamma_3}\phi_\sigma^T\phi_\sigma + \frac{b}{2\gamma_4}\phi_w^T\phi_w \tag{3-45}$$

Based on (3-34), taking the derivative with respect to t yields:

$$\begin{aligned} \dot{V} = &-\frac{1}{2}\underline{e}^T Q\underline{e} + \underline{e}^T PB[\phi_\theta^T\xi(x\,|\,c,\sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + d_u(x) \\ &- \text{sgn}(\underline{e}^T PB)w^T Y - u_{bd}] + \frac{b}{\gamma_1}\phi_\theta^T\dot{\phi}_\theta + \frac{b}{2\gamma_2}\phi_c^T\dot{\phi}_c + \frac{b}{2\gamma_3}\phi_\sigma^T\dot{\phi}_\sigma + \frac{b}{2\gamma_4}\phi_w^T\dot{\phi}_w \end{aligned} \tag{3-46}$$

From (3-12), we obtain:

$$\begin{aligned} \dot{V} &\leq -\frac{1}{2}\underline{e}^T Q\underline{e} + \underline{e}^T PB[\phi_\theta^T\xi(x\,|\,c,\sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + \text{sgn}(\underline{e}^T PB)w^{*T}Y - \text{sgn}(\underline{e}^T PB)w^T Y - u_{bd}] \\ &\quad + \frac{b}{\gamma_1}\phi_\theta^T\dot{\phi}_\theta + \frac{b}{2\gamma_2}\phi_c^T\dot{\phi}_c + \frac{b}{2\gamma_3}\phi_\sigma^T\dot{\phi}_\sigma + \frac{b}{2\gamma_4}\phi_w^T\dot{\phi}_w \\ &= -\frac{1}{2}\underline{e}^T Q\underline{e} + \underline{e}^T PB[\phi_\theta^T\xi(x\,|\,c,\sigma) + \phi_c^T u_c + \phi_\sigma^T u_\sigma + \text{sgn}(\underline{e}^T PB)\phi_w^T Y - u_{bd}] \\ &\quad + \frac{b}{\gamma_1}\phi_\theta^T\dot{\phi}_\theta + \frac{b}{2\gamma_2}\phi_c^T\dot{\phi}_c + \frac{b}{2\gamma_3}\phi_\sigma^T\dot{\phi}_\sigma + \frac{b}{2\gamma_4}\phi_w^T\dot{\phi}_w \end{aligned} \tag{3-47}$$

From (3-40), we have

$$\underline{e}^T PBu_{bd} = I(\underline{e}^T PB)\mathrm{sgn}(\underline{e}^T PB)[|u| + |u_{co}| + \frac{1}{b_L}(f^U(\underline{x}) + |y_m^{(n)}| + |\underline{k}^T \underline{e}|)] \geq 0 \quad (3\text{-}48)$$

then we obtain:

$$\begin{aligned} \dot{V} \leq & -\frac{1}{2}\underline{e}^T Q\underline{e} + \frac{b}{\gamma_1}\phi_\theta^T[\gamma_1 \underline{e}^T P_n \xi(\underline{x}) + \dot{\phi}_\theta] + \frac{b}{\gamma_2}\phi_c^T[\gamma_2 \underline{e}^T P_n u_c + \dot{\phi}_c] \\ & + \frac{b}{\gamma_3}\phi_\sigma^T[\gamma_3 \underline{e}^T P_n u_\sigma + \dot{\phi}_\sigma] + \frac{b}{\gamma_4}\phi_w^T[\gamma_4 \,\mathrm{sgn}(\underline{e}^T PB)\underline{e}^T P_n Y + \dot{\phi}_w] \end{aligned} \quad (3\text{-}49)$$

where P_n is the last column of P.

We could choose the adaptive laws as:

$$\begin{aligned} \dot{\theta} &= \gamma_1 \underline{e}^T P_n \xi(\underline{x}) \\ \dot{c} &= \gamma_2 \underline{e}^T P_n u_c \\ \dot{\sigma} &= \gamma_3 \underline{e}^T P_n u_\sigma \\ \dot{w} &= \gamma_4 \,\mathrm{sgn}(e^T PB)\underline{e}^T P_n Y \end{aligned} \quad (3\text{-}50)$$

Using the facts $\dot{\phi}_\theta = -\dot{\theta}$, $\dot{\phi}_c = -\dot{c}$, $\dot{\phi}_\sigma = -\dot{\sigma}$ and $\dot{\phi}_w = -\dot{w}$, we obtain

$$\dot{V} \leq -\frac{1}{2}\underline{e}^T Q\underline{e} \quad (3\text{-}51)$$

3.4 Example: Vehicle Longitudinal Controller

Within this section, we will apply our proposed adaptive fuzzy controller to vehicle longitudinal control. The objective of the adaptive fuzzy controller is to maintain a safe distance between the preceding car and the following car. The strength of our approach is that we do not require the training data, and fuzzy rules can be updated on-line according to the performance of the controller. And our approach needs little knowledge about the car. As a result, it can be transported to any vehicles regardless of the nonlinear and often unobservable dynamics.

3.4.1 Vehicle Longitudinal Dynamics

Several vehicle models have been proposed for different purposes. For vehicle longitudinal control design, we only consider throttle and brake control for longitudinal control, and do not consider the steering wheel. The vehicle dynamics may be expressed as the following mathematical model [Swaroop et al. 2001]:

$$\ddot{x} = \frac{F - c\dot{x}^2 - d}{M} \tag{3-52}$$

$$\dot{F} = \frac{1}{\tau}(-F + u_p) \tag{3-53}$$

where, in the first equation, x, F, c, d, M are the position, the engine traction force, effective aerodynamic drag coefficient, rolling resistance friction, and effective inertia respectively. If we consider the engine dynamics, we have the additional equation (3-53), where the engine traction force F can be modeled as a first order system, and u_p is the control input.

We should say here that, although we present the exact vehicle longitudinal model, we use only some knowledge of this model to design our controller, and we may not know the exact values of all parameters in equations (3-52) and (3-53), instead, we should only know the bound of the parameters. In other words, the controller design does not require a complete model.

3.4.2 Simulation Results

The main objective of vehicle longitudinal control is to maintain a constant safe spacing between the preceding car and following car.

We consider the output of the vehicle as

$$y = x - x_l \tag{3-54}$$

We simply select $y_m = 0$, then we obtain:

$$e = y_m - y = x_l - x \tag{3-55}$$

If we neglect the time lag of u, i.e., $\tau = 0$ in vehicle model, then

$$y^{(2)} = -(\dot{v}_l - \dot{v}) = -\dot{v}_l - \frac{1}{m}cv^2 - \frac{1}{m}d + \frac{1}{m}u$$

So, we have:

$$f(\underline{x}) = -\frac{1}{m}cv^2 - \frac{1}{m}d - \dot{v}_p, \quad b = \frac{1}{m} > 0 \tag{3-56}$$

And we obtain:

$$\underline{e} = [e, \dot{e}]^T = [x_l - x, v_l - v_f]^T = [\Delta x, v_r]^T \tag{3-57}$$

Next, from the (3-56), we may determine the upper bound of $f(\underline{x})$ and the lower bound of b.

$$f(\underline{x}) \le \frac{1}{m}c|v|^2 + \frac{1}{m}|d| + |a_p|$$

Based on the information provided in [Kim et al. 1996], $c = 0.44\,kg/m$, $d = 352\,kgm/s^2$, and we assume the minimum mass of vehicle $1000 \le m \le 2000kg$, the acceleration of vehicle $-3 \le a \le 1.5\,m/s^2$. We get

$$f^U(\underline{x}) = \frac{1}{m_{\min}}c \cdot |v|^2 + \frac{1}{m_{\min}}|d| + 3$$

$$g(\underline{x}) \ge \frac{1}{m_{\max}} = 0.0005 = b_L \tag{3-58}$$

Here, we consider the following fuzzy rules of the adaptive fuzzy controller.

$$R_l: \text{ IF } \quad \Delta x \text{ is } F_1^l \text{ and } v_r \text{ is } F_2^l$$

$$\text{THEN } \quad u = K_0^l + K_1^l \Delta x + K_2^l v_r \tag{3-59}$$

The detailed implementation procedure of the car longitudinal controller is as

follows:

1) Initialize the parameters (θ, c, σ) of fuzzy rules.

2) Obtain the relative speed and relative distance through the sensors of the car, and then we can get system error $\underline{e} = [\Delta x, v_r]^T$.

3) Calculate the membership of $\Delta x, v_r$ based on equation (3-6).

4) Calculate $\xi(\underline{x})$ based on equation (3-8) and fuzzy rule (3-59), and obtain the fuzzy controller output u based on θ and $\xi(\underline{x})$ (see equation (3-7)).

5) Obtain the compensating controller output u_{co} based on equation (3-43).

6) Calculate the bounding controller output u_{bd} by equation (3-40) based on u and u_{co}, which have been obtained in step 4 and 5, respectively.

7) Calculate the total control input u_p of vehicle based on equation (3-32).

8) Update the parameter (θ, c, σ) based on the adaptive laws (equation (3-50)).

9) Back to step 2.

We select $\underline{k} = (k_2, k_1)^T = (2,1)^T$ (so that $s^2 + k_1 s^1 + k_2$ are in the open left half plane, i.e. stable), $Q = diag(10,10)$, and we get symmetric positive definite matrix $P = \begin{bmatrix} 15 & 5 \\ 5 & 5 \end{bmatrix}$.

We adopt the fuzzy rules like equation (3-59) and totally have 9 rules in our simulation. Initially we define three fuzzy sets over the interval [-1,1] for δ, three fuzzy sets over the interval [-0.5,0.5] for v_r, which are shown in Figure 3.2.

In our simulations, the velocity profile of the preceding vehicle is shown in Figure 3.3. We simply choose all the parameters of θ to be zero, c and σ are chosen as Figure 3.2.

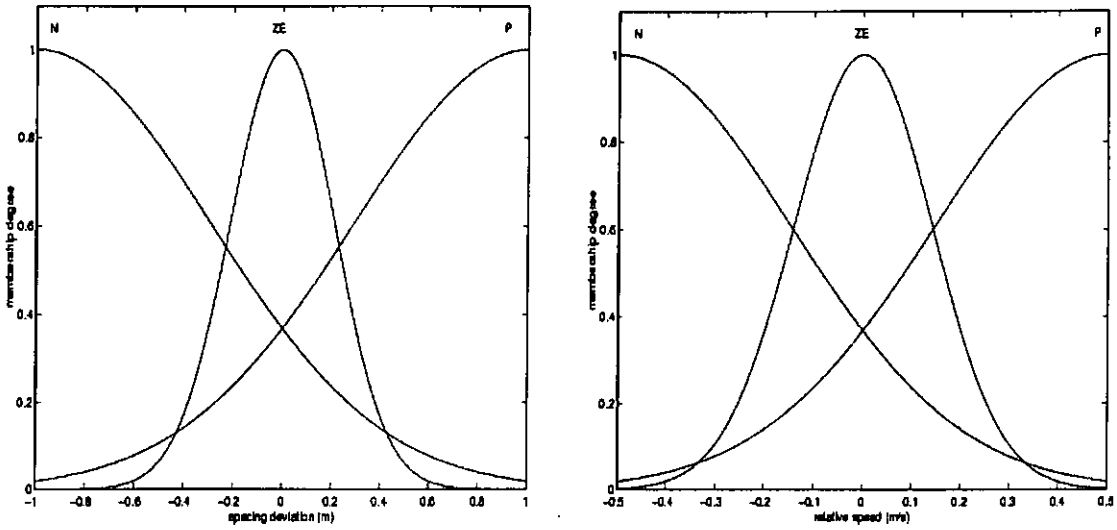


Figure 3.2: Fuzzy membership function of Δx (left) and v_r (right)

We should notice here, there is no acceleration of the preceding car to provide and we can only obtain the information of the relative speed and relative distance between the preceding car and the following car.

The simulation results of our proposed adaptive fuzzy control are shown in the Figure 3.3 and Figure 3.4.

We can see from the simulations that: 1) the following car follows the preceding car very quickly (the velocity of the following car is almost same as the preceding car), although the velocity of preceding car changes frequently, 2) the spacing deviation between the leading car and the following car converges to zero quickly and there is no oscillation, 3) there are some large spacing deviations in the beginning due to the initial parameters which are given randomly, and after some time, the parameters are tuned better so that good performance is achieved.

The advantage of our proposed method is that the controller can tune all the parameters (including parameters of Gaussian membership functions) automatically.

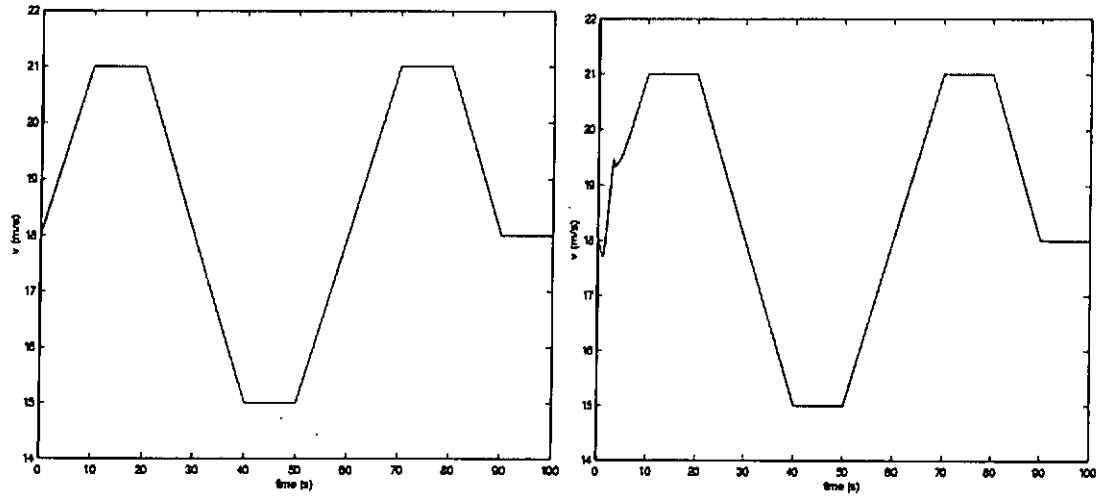


Figure 3.3: Velocity responses of the preceding car (left) and the following car (right) with the proposed adaptive fuzzy control

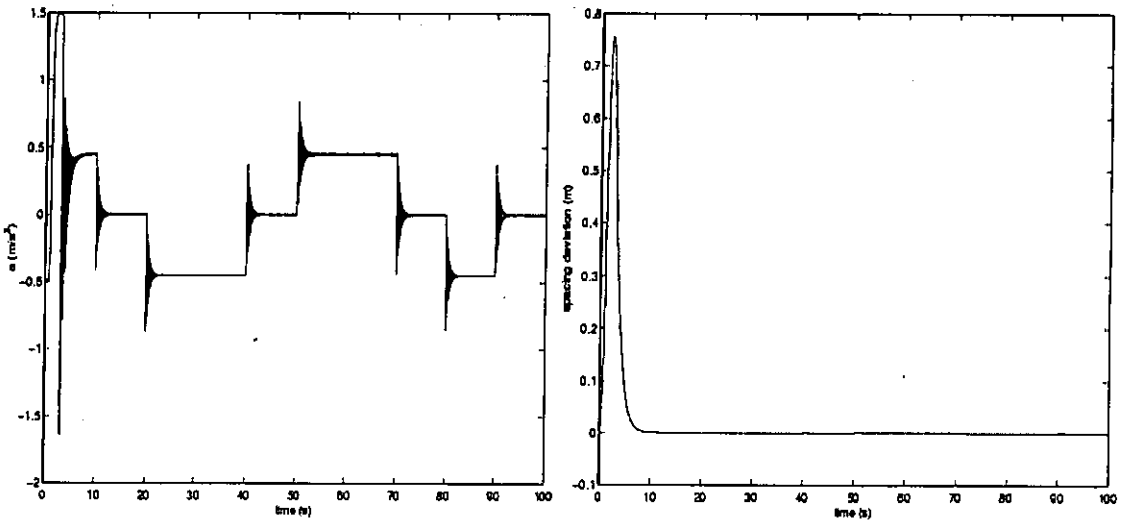
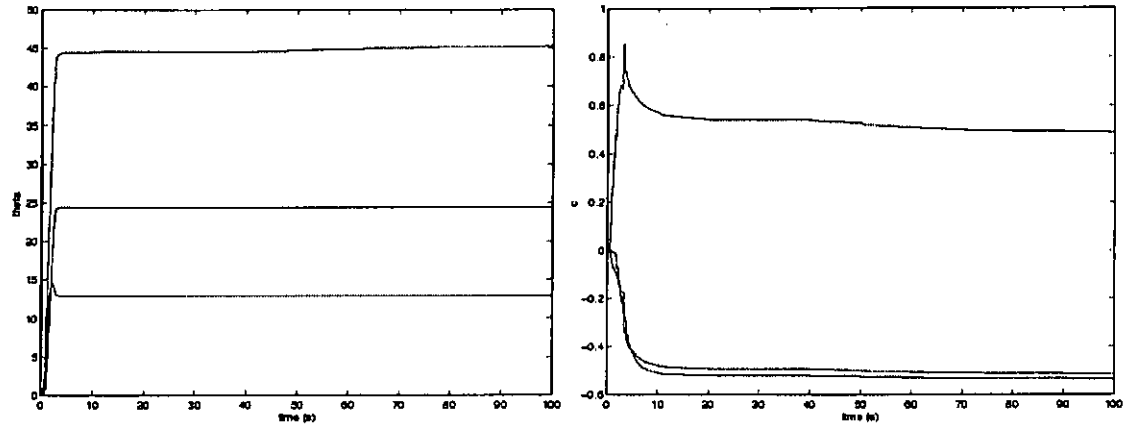


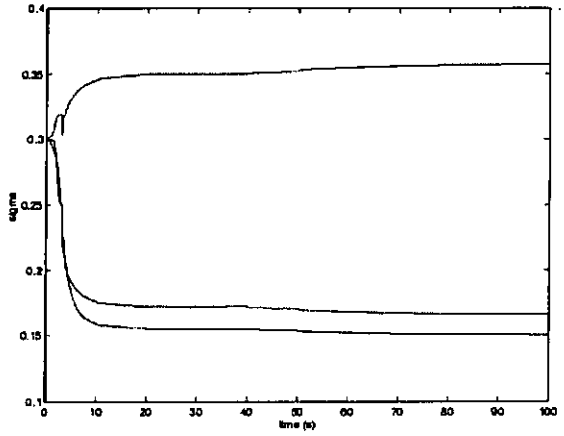
Figure 3.4: The acceleration of the controlled vehicle (left), and the spacing deviation between the two vehicles (right)

We can also see the adaptive adjustment of parameters partially from Figure 3.5.

If we consider the model disturbance (the vehicle mass is changed from 1000kg to 2000kg) and some measurement noise, the simulation results of our proposed adaptive fuzzy control are shown in the Figure 3.6.



(a) (b)



(c)

Figure 3.5: Evolution of parameters of fuzzy rules, (a) θ with initial value 0, (b) c with initial value 0, and (c) σ with initial value 0.3

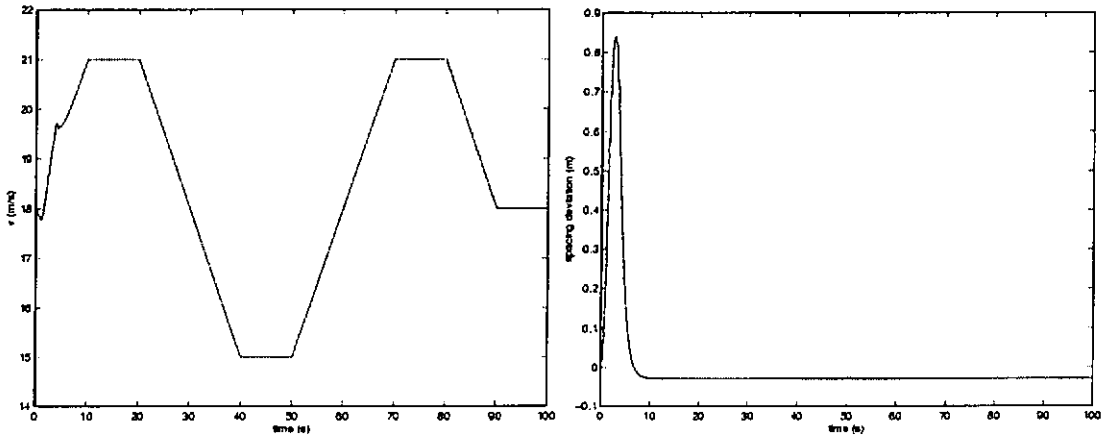


Figure 3.6: (a) The velocity of the controlled vehicle, and (b) the spacing deviation between the two vehicles with the model disturbance and measurement noise

Compared Figure 3.6 with Figure 3.4 and Figure 3.5, we can say that our proposed controller is immune to model uncertainty and measurement noise.

Our proposed adaptive fuzzy control has some advantages over some conventional adaptive control in that it can incorporate a priori knowledge into our controller. The priori knowledge can embody in the initial values of parameters, and these "good" initial values can expedite the adaptation speed.

3.5 Conclusions

We have proposed an adaptive fuzzy control in this chapter. In order that the fuzzy system exhibits a good approximation property, we tune all the parameters in the fuzzy system including parameters of Gaussian membership functions. In order to construct the adaptive laws of these parameters, linear relationship between approximation error and parameters is established. The proposed controller includes bounding control, compensating control and fuzzy control which is used to approximate the optimal ideal control. The advantage of our approach is that we do not require the complete model, and fuzzy rules can be adjusted according to the performance of the controller. Moreover, our controller is more flexible because more tuned parameters are considered. Finally, we apply our approach to vehicle longitudinal control, simulation results show that it provides satisfactory performances in car-following.

While our approach presents significant advantages, there are several aspects for us to consider. First, our control scheme is only for a class of specific continuous time

SISO nonlinear system. Extension to other nonlinear systems is an important direction. Second, the proposed method is only studied in simulation, more desirable approach to take into account the implementation aspects for real-time control is expected.

Chapter 4

A Model-Free Intelligent Adaptive Controller and its Application

In Chapter 3, we remarked on some of the drawbacks of the adaptive fuzzy controller: the control scheme is only for a class of specific continuous time SISO nonlinear system, and it needs some information of the model, such as model structure, assumptions about model parameters, etc.

In this chapter, we suggest a new approach for tuning parameters of fuzzy controllers based on reinforcement learning, in order to overcome the model constraints. The architecture of the proposed approach comprises of a Q estimator network (QEN) and a Takagi-Sugeno type fuzzy inference system (TS-FIS). Unlike other fuzzy Q-learning approaches that select an optimal action based on finite discrete actions, the proposed controller obtains the control output directly from TS-FIS. With the proposed architecture, the learning algorithms for all the parameters of the Q estimator network and the FIS are developed based on the temporal difference methods as well as the gradient descent algorithm. The performance of the proposed design technique is illustrated by simulation studies of a vehicle longitudinal control system.

The rest of this chapter is organized as follows. Section 4.1 introduces mathematical expressions of reinforcement learning and gives some remarks. The

architecture of the controller is described in section 4.2. The learning algorithms and parameter update laws are presented in section 4.3. Section 4.4 illustrates the performance of our proposed method through vehicle navigation. Finally, conclusions are drawn in section 4.5.

4.1 Foundations of Reinforcement Learning

4.1.1 The Mathematical Expressions of Reinforcement Learning

The application of reinforcement learning in control problems focuses on two main types of algorithms: actor-critic learning and Q-learning. The actor-critic learning system contains two parts: one to estimate the state-value function $V(x)$, and the another to choose the optimal action for each state. While the Q-learning system estimates action-value function $Q(x,a)$ for all state-action pairs and selects the optimal control algorithm based on $Q(x,a)$.

The state-value function $V(x)$ is the expected discounted sum of reward with the initial state x, and can be written as:

$$V(x) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid x_t = x\right\} \tag{4-1}$$

where, x is the state of the system, r_{t+k+1} is the reinforcement signal (instant reward) at time $t+k+1$, and $\gamma \in [0,1]$ is a discount factor, $E(\cdot)$ is the expected value function.

The action-value function $Q(x,a)$ is the expected discounted sum of reward with the initial state x and initial action a, and can be written as:

$$Q(x,a) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid x_t = x, a_t = a\right\} \tag{4-2}$$

where, a is the action which acts on the system.

And the optimal state-value function $V^*(x)$ and the optimal action-value function $Q^*(x,a)$ are as follows:

$$V^*(x) = \max_a \left\{r(x_{t+1}) + \gamma V^*(x_{t+1})\right\}$$

$$Q^*(x,a) = E\left\{r(x_{t+1}) + \gamma \max_{a'} Q^*(x_{t+1}, a') \mid x_t = x, a_t = a\right\}$$

The Q-learning estimates the $Q^*(x,a)$ based on TD error δ_t:

$$\delta_t = r_{t+1} + \gamma \max_{a'} Q(x_{t+1}, a') - Q(x_t, a_t) \tag{4-3}$$

And $Q(x,a)$ is updated according to the above TD error as:

$$Q_{t+1}(x_t, a_t) = Q_t(x_t, a_t) + \alpha\delta_t \tag{4-4}$$

where, α is the learning rate. Watkins [Watkins 1989] has shown that $Q_t(x,a)$ convergences to $Q^*(x,a)$ with probability one if all actions continue to be tried form all states, and the following conditions for α are satisfied:

$$0 < \alpha_k < 1, \ \sum_{k=1}^{\infty} \alpha_k = \infty, \text{ and } \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

In fact, the definition of the above TD error is TD(0), and we can extend it to TD(λ), which considers the past and the current estimated error using eligibility traces e_t:

$$e_t = \sum_{k=0}^{t} (\lambda\gamma)^{t-k} \chi_x(k) \tag{4-5}$$

where, λ is recency factor (also called eligibility rate) and $0 \le \lambda \le 1$,

$$\chi_x(t) = \begin{cases} 1 & x_t = x \\ 0 & otherwise \end{cases}$$

Then, $Q(x,a)$ is updated according to following rule:

$$Q_{t+1}(x_t, a_t) = Q_t(x_t, a_t) + \alpha\delta_t e_t \tag{4-6}$$

After obtaining the estimate of the optimal action-value function $Q^*(x,a)$, we can get the optimal action a^*, which maximizes $Q^*(x,a)$ for the current state x, i.e., $a^* = \arg\max_a Q^*(x,a)$. But during the procedure of estimating $Q^*(x,a)$, we should use a stochastic control rule that will take all possible actions to satisfy the requirements of Watkins' theorem. The Boltzmann exploration and ϵ-greedy policy are such stochastic control rules, which are often adopted [Sutton & Barto 1998, Yan et a. 2001].

The reinforcement learning was proposed to deal with discrete states and discrete actions originally based on Markovian Decision Problems (MDP) [Sutton & Barto 1998]. In the case of a large, continuous state space, the discrete representation of reinforcement learning is intractable. This problem is known as the curse of dimensionality. To solve this problem and generalize from previously experienced states to ones that have never been seen, we should combine the reinforcement learning with existing generalization methods, such as a low-order polynomial, neural network or fuzzy system instead of a look-up table. In others words, we can estimate the optimal state-value function $V^*(x)$ or the optimal action-value function $Q^*(x,a)$ with approximators.

4.1.2 Remarks

We adopt Q-learning because it is conceptually simpler, and has been found empirically to converge faster in many cases [Sutton et al. 1992]. Although many existing Q-learning approaches applied in control problems aim at tasks of continuous

domains, they select the optimal action and get the control output directly based on finite discrete actions and estimated $Q^*(x,a)$. This may bring the following problems:

1) How to discretize the action space is a problem. When a coarse discretization is used, the action is not smooth and the resulting performance is poor. When a fine discretization is used, the number of state-action pairs become large, which results in large memory storage and slow learning procedures.

2) The optimal action value is based on the current states x_t and the approximated value $Q(x_t,a)$. This means that an error in any of these approximations will be incorporated into the action value. If this is the case, the action value a' and the consequent x'_{t+1} may be slightly different from the optimal a and the consequent x_{t+1}. This error may quickly accumulate and produce a different action policy [Smart & Kaelbling 2000, Boyan & Moore 1995] because of the different states x'_{t+1} and x_{t+1}.

3) Reinforcement learning systems often perform extremely poorly in the early stage of learning due to their more-or-less random action. They can act appropriately until they acquire some experience of the world [Smart & Kaelbling 2000]. The above problem is especially serious in domains where the reward function is largely uniform and dealing with continuous systems with small sample time, because different actions have similar effects.

In our research, assumption of some candidate discrete actions is not made, this eliminates the requirement that discretizes the action space and does not have the first

problem. In order to solve the problem 2, the action is not obtained directly based on approximated value $Q(x,a)$ and candidate discrete actions. Instead, we use a fuzzy inference system (FIS) to provide the control output. The information of the estimated $Q(x,a)$ is only used to tune the parameters of FIS. A normal FIS can guarantee the performance of control output not too bad, and it can restrict the control action in a special region, and it can be thought as an initial knowledge. The initial knowledge bootstrapped into the value function approximation allows the agent to learn more effectively, and helps reduce the time spent acting randomly. So, the problem 3 is also avoided to some extent.

As a result, our proposed algorithm need less a priori knowledge because only evaluative signal is required for reinforcement learning, and is more applicable to real systems because we combine reinforcement learning with fuzzy inference system.

4.2 Architecture of the Controller Based on RL

As we mentioned before, we use reinforcement signal to tune the parameters of fuzzy controller, which provides the control output. So our proposed controller has two main responsibilities: one is to estimate the optimal action-value function $Q^*(x,a)$, and the other is to get the control output based on the estimated action-value function $Q(x,a)$. We construct the controller with two parts – neural network and fuzzy inference system - to deal with the above two different tasks respectively.

4.2.1 Neural Network for Estimating $Q^*(x,a)$

The $Q^*(x,a)$ estimator network (QEN) plays the role of approximating or predicting the optimal action-value function $Q^*(x,a)$ associated with different input states and control output.

As an approximator of value function of reinforcement learning, different types of neural networks have been proposed for different tasks and different objectives, such as the ART (adaptive resonance theory) network [Lin & Lin 1996, Lin & Chung 1999], Cerebellar Model Articulation Controller (CMAC) [Oh et al. 2002], standard multi-layer feedforward neural network [Si & Wang 2001, Lin & Lee 1994], competitive network [Yan et al. 2001], radial basis function (RBF) network [Koike & Doya 1999], and neuro-fuzzy network [Jouffe 1998], etc.

Here, we simply use a two-layer feedforward neural network to estimate and generalize the optimal action-value function. The estimation is based on the prediction TD error δ_t that is defined in section 4.1. The architecture of the QEN is shown in Figure 4.1.

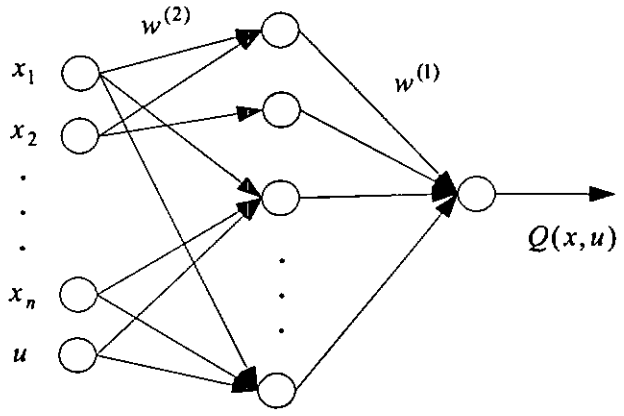


Figure 4.1: Architecture of the Q Estimator Network

Because we want to estimate $Q^*(x,a)$ using QEN, the input of the network includes system states and control output u which is same as the action a mentioned before, and the output of the network is $Q(x,u)$.

In our proposed feedforward neural network, $Q(x,u)$ will be the form of:

$$Q(x,u) = f(O_1)$$

$$O_1 = \sum_{i=1}^{N_h} w_i^{(1)} p_i$$

$$p_i = f(O_{2i})$$

$$O_{2i} = \sum_{j=1}^{n+1} w_{ij}^{(2)} x_i$$

where

O_1: the summed input of the output node,

$w_i^{(1)}$: the weight between the i th hidden node and the output node,

p_i: the output of the hidden node,

O_{2i}: the summed input of the i th hidden node,

$w_{ij}^{(2)}$: the weight between the j th input node and i th hidden node,

x_{n+1}: the $(n+1)$ th input, i.e., u,

f: the activation function of the node

Here, we adopt sigmoid function as the activation function of the node, i.e. $f(x) = \dfrac{1}{1+\exp(-x)}$.

The QEN is used to guide the fuzzy controller to tune parameters so that the fuzzy controller will achieve better performance.

4.2.2 Fuzzy Inference System for Producing the Control Output

Given the current state of the system and the action-value function $Q(x,a)$ estimated from the QEN, we may obtain the optimal action u^* based on the candidate discrete action set U and Q-learning algorithm $u^* = \arg\max_{u \in U} Q(x,u)$. Indeed, this is the basic idea of most existing algorithms which adopt Q-learning. But the candidate discrete action set U may be not known for us and the performance of the controller is affected by the selected finite candidate actions accordingly.

In our research, the information about the candidate discrete action set U is not required. The Fuzzy Inference System (FIS), which will be explained in the rest of this subsection, generate the control output u. And u will intend to maximize the action-value function $Q(x,u)$ produced by the QEN incrementally, and finally output provided by FIS will maximize $Q(x,u)$ with respect to the all possible u, instead of finite candidate discrete action set U.

We propose FIS as the controller for the following considerations:

1) Compared with obtaining control output directly based on estimated $Q(x,u)$, FIS is robust with respect to the parameters which is related to the action-value function $Q(x,u)$. This means that FIS can achieve acceptable performance even in the early stage of learning, in which the approximation error of $Q(x,u)$ is large.

2) Owing to the fact that reinforcement learning may spend a huge amount of time taking exploratory actions and learning nothing if it cannot search the optimal region, searching the whole space may be time-consuming. FIS,

which can incorporate some initial knowledge, may guide the agent to some "good" space, allow it to learn more effectively and expedite the learning process.

In this subsection, we present the architecture of FIS. Consider a multiple-input single-output (MISO) fuzzy controller, which performs a mapping from a state vector $\underline{x} = (x_1, x_2, ..., x_n)^T \in \Re^n$ to a control input $u \in \Re$. Using the Takagi-Sugeno model, the IF-THEN rules of the fuzzy controller may be expressed as:

$$R_l: \quad \text{IF} \quad x_1 \text{ is } F_1^l \text{ and} \cdots \text{and } x_n \text{ is } F_n^l$$

$$\text{THEN} \quad u = K_0^l + K_1^l x_1 + K_2^l x_2 + \cdots + K_n^l x_n$$

where F_i^l is the label of the fuzzy set in x_i, for $l = 1,2,...,M$. K_0^l, K_1^l, K_2^l ... and K_n^l are the constant coefficients of the consequent part of the fuzzy rule.

In this chapter, we would use product inference for the fuzzy implication and t norm, singleton fuzzifier and centre average defuzzifier, consequently, the final output value is:

$$u(\underline{x}) = \frac{\sum_{l=1}^{M}\left(\left(\prod_{i=1}^{n}\mu^{F_i^l}(x_i)\right)\cdot\left(\sum_{j=0}^{n}K_j^l x_j\right)\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\mu^{F_i^l}(x_i)\right)} \tag{4-7}$$

where, $\mu^{F_i^l}$ is the membership degree of the fuzzy set F_i^l, $x_0 = 1$.

We adopt Gaussian function as the membership function of the fuzzy system because its excellent approximation properties,

$$\mu^{F_i^l}(x_i) = \exp\left(-\left(\frac{x_i - c_i^l}{\sigma_i^l}\right)^2\right) \tag{4-8}$$

for $i = 1,2,...,n$ and $l = 1,2,...,M$.

4.2.3 Stochastic Action Modifier

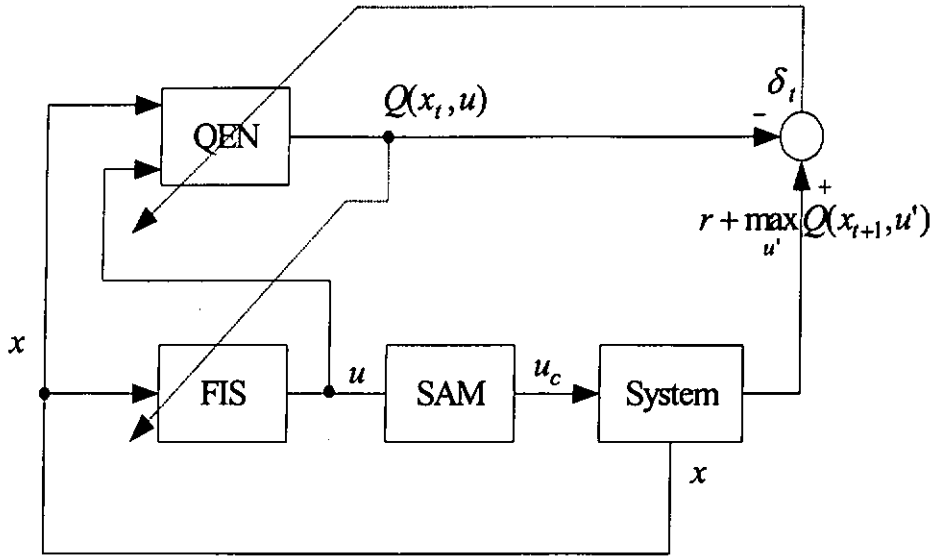


Figure 4.2: Architectures of the proposed controller

From section 4.1, we know that $Q_t(x,a)$ convergences to $Q^*(x,a)$ with probability one if all actions continue to be tried form all states. But how can we guarantee all actions to be tried?

In order to solve the above problem, we implement the exploration strategy for the control output u. However, some existing exploration strategies such as Boltzmann exploration and ϵ-greedy policy mainly deal with discrete actions.

We add a stochastic action modifier (SAM) after the FIS and before the system input. The SAM generates the control command u_c, which is a Gaussian random variable with mean u recommended by FIS and standard deviation σ_u. And σ_u satisfies the condition that it will converge to zero gradually, i.e. $u_c = u + \sigma_u(t) \cdot n(t)$, and $\lim_{t\to\infty} \sigma_u(t) = 0$.

As a result, the overall architecture of our proposed controller is shown in Figure

4.2.

4.3 Adaptation Algorithms of the Controller

In this section, we develop the learning algorithms for the QEN and the FIS. The learning mechanism of the QEN is the combination of the TD methods and the backpropagation algorithm. The learning mechanism of the FIS is based on gradient algorithm.

4.3.1 Adaptation Algorithm for Q Estimator Network

From section 4.1, we know TD methods learn their estimates in part on the basis of other estimates. We can also tune the parameters of our proposed QEN based on generalized policy iteration (GPI). We can achieve the task of approximating the optimal action-value function with the neural network by reducing the TD error δ_t continuously.

$$\delta_t = r_{t+1} + \gamma \max_{u'} Q(x_{t+1}, u') - Q(x_t, u_t) \tag{4-9}$$

In other words, the objective of the neural network is to minimize the following expression.

$$E = \frac{1}{2}\delta_t^2$$

The weight update rule for the neural network based gradient descent method is given by:

$$w(t+1) = w(t) - \eta \frac{\partial E}{\partial w} \tag{4-10}$$

$$\frac{\partial E}{\partial w} = \delta_t \frac{\partial \delta_t}{\partial w} = -\delta_t \frac{\partial Q(x_t, u_t)}{\partial w} \tag{4-11}$$

Combining the above two equations, we can obtain:

$$w(t+1) = w(t) + \eta\delta_t \frac{\partial Q(x_t, u_t)}{\partial w} \tag{4-12}$$

for $w = w_i^{(1)}$ and $w = w_{ij}^{(2)}$ respectively.

We can obtain $\frac{\partial Q(x_t, u_t)}{\partial w}$ based on chain rules for $w_i^{(1)}$ and $w_{ij}^{(2)}$ respectively.

$$\frac{\partial Q(x_t, u_t)}{\partial w_i^{(1)}} = \frac{\partial Q(x_t, u_t)}{\partial O_1} \frac{\partial O_1}{\partial w_i^{(1)}} = f'(O_1) p_i = p_i Q(x_t, u_t)[1 - Q(x_t, u_t)]$$

$$(\text{for } i = 1, \ldots, N_h) \tag{4-13}$$

$$\frac{\partial Q(x_t, u_t)}{\partial w_{ij}^{(2)}} = \frac{\partial Q(x_t, u_t)}{\partial O_1} \frac{\partial O_1}{\partial p_i} \frac{\partial p_i}{\partial O_{2i}} \frac{\partial O_{2i}}{\partial w_{ij}^{(2)}} = f'(O_1) w_i^{(1)} f'(O_{2i}) x_j$$
$$= w_i^{(1)} x_j Q(x_t, u_t)[1 - Q(x_t, u_t)] p_i [1 - p_i]$$

$$(\text{for } i = 1, \ldots, N_h, \; j = 1, \ldots, N_i) \tag{4-14}$$

where, the fact that $f'(O) = f(O)[1 - f(O)]$ is used for sigmoid function.

And we can also obtain $\frac{\partial Q(x_t, u_t)}{\partial u}$:

$$\frac{\partial Q(x_t, u_t)}{\partial u} = \frac{\partial Q(x_t, u_t)}{\partial O_1} \sum_{i=1}^{N_h} \left(\frac{\partial O_1}{\partial p_i} \frac{\partial p_i}{\partial O_{2i}} \frac{\partial O_{2i}}{u} \right) = f'(O_1) \sum_{i=1}^{N_h} \left(w_i^{(1)} f'(O_{2i}) w_{i,N_i}^{(2)} \right)$$
$$= Q(x_t, u_t)[1 - Q(x_t, u_t)] \sum_{i=1}^{N_h} \left(w_i^{(1)} w_{i,N_i}^{(2)} p_i [1 - p_i] \right) \tag{4-15}$$

where, it is noticed that control output of FIS is the N_i th input, i.e. $x_{N_i} = u$.

In the above equations (4-14) and (4-15), $w_i^{(1)} = 0$ if the i th hidden node is not connected to the output node, $w_{i,j}^{(2)} = 0$ if the j th input node is not connected to the i th hidden node.

Eligibility traces record the past and current gradients, and adopting eligibility traces could speed up a learning process, the eligibility traces for the QEN is defined as:

$$\vec{e}_t = \gamma\lambda\vec{e}_{t-1} + \frac{\partial Q(x_t, u_t)}{\partial w} \tag{4-16}$$

with $\vec{e}_0 = \vec{0}$.

Then the parameters learning algorithm for QEN is derived as follow:

$$w(t+1) = w(t) + \eta\delta_t\vec{e}_t \tag{4-17}$$

4.3.2 Adaptation Algorithm for FIS

Now, we consider how to improve the control policy using the associated value function. In other words, we consider how to tune the parameters of the fuzzy controller based on the approximated $Q(x,u)$ obtained from the subsection 4.3.1.

For the discrete finite actions, we may obtain the optimal control action based on finite comparisons of the value function we have approximated. However, this approach is obviously infeasible for continuous actions.

For continuous-time systems without discrete states and actions, we may generalize the value function ($V(x)$ or $Q(x,u)$) using neural network based on reinforcement signal δ_t, so we can deal with continuous states. But how to deal with continuous actions is not a trivial work. Gullapalli proposed the stochastic real-valued (SRV) unit algorithm, in which the parameters of the action network are updated by gradient estimation [Gullapalli 1990]. Baird proposed the advantage updating method, in which both the value function $V(x)$ and the advantage function $A(x,u)$ are updated [Baird 1994]. It should be noticed here that the above algorithms obtain the continuous actions based on state-value function $V(x)$, not the action-value function $Q(x,u)$.

In order to make the output of FIS to be optimal, we can update the parameters of FIS to maximize action-value function $Q(x,u)$ with respect to the control output u for the current state. As a result, the learning algorithms of FIS can be derived using gradient rules:

$$\xi(t+1) = \xi(t) + \beta \frac{\partial Q(x_t, u_t)}{\partial \xi} \tag{4-18}$$

$$\frac{\partial Q(x,u)}{\partial \xi} = \frac{\partial Q(x,u)}{\partial u} \frac{\partial u}{\partial \xi} \tag{4-19}$$

where, ξ is parameters to be tuned in fuzzy systems such as K_j^l, c_i^l and σ_i^l. We have obtained $\frac{\partial Q(x_t, u_t)}{\partial u}$ already in the previous subsection (see (4-15)). We will only need to deduce $\frac{\partial u}{\partial \xi}$.

If we let

$$z^l = \prod_{i=1}^{n} \exp\left(-\left(\frac{x_i - c_i^l}{\sigma_i^l} \right)^2 \right)$$

$$\bar{y}^l = K_0^l + K_1^l x_1 + \cdots + K_n^l x_n \tag{4-20}$$

$$a = \sum_{l=1}^{M} (\bar{y}^l z^l), \quad b = \sum_{l=1}^{M} z^l, \quad u = \frac{a}{b}$$

Then we can calculate $\frac{\partial u}{\partial \xi}$ by the following equations:

$$\frac{\partial u}{\partial K_j^l} = \frac{z^l}{b} x_j \tag{4-21}$$

$$\frac{\partial u}{\partial c_i^l} = \frac{\partial u}{\partial z^l} \cdot \frac{\partial z^l}{\partial c_i^l} = \frac{\bar{y}^l - u}{b} z^l \frac{2(x_i - c_i^l)}{(\sigma_i^l)^2} \tag{4-22}$$

$$\frac{\partial u}{\partial \sigma_i^l} = \frac{\partial u}{\partial z^l} \cdot \frac{\partial z^l}{\partial \sigma_i^l} = \frac{\bar{y}^l - u}{b} z^l \frac{2(x_i - c_i^l)^2}{(\sigma_i^l)^3} \tag{4-23}$$

4.3.3 Implementation Procedures

After the above discussion, we present the detailed implementation procedures as follows:

1) Initialize $Q(x_t, u_t)$, the parameters $w^{(1)}$, $w^{(2)}$ of QEN and the parameters ξ of FIS.

2) Obtain the new control output u_{t+1} based on equation (4-7) and input of the fuzzy inference system.

3) Before it is fed to the actual system, u is processed by SAM according to $u_c = u + \sigma_u(t) \cdot n(t)$.

4) SAM provides u_c, and u_c acts as the control value of the system.

5) Based on our requirements for the system, we evaluate the performance of the controller as r, which is only the "evaluative signal" instead of the "teacher" signal. And we also obtain the states of the system.

6) Obtain the approximated $Q(x_{t+1}, u_{t+1})$ from QEN based on the current control action, the current states.

7) From r, $Q(x_t, u_t)$ and $Q(x_{t+1}, u_{t+1})$, we can calculate the TD error δ_t based on equation (4-9). Here, we think $Q(x_{t+1}, u_{t+1}) \approx \max_{u'} Q(x_{t+1}, u')$ because u_{t+1} obtained from FIS which continuously maximizes $Q(x, u)$ with respect to the control output u.

8) Based δ_t obtained from step 7, we can update the parameters of QEN according to equation (4-13), (4-14), (4-16) and (4-17).

9) Tune the parameters of FIS based on equation (4-18)-(4-23).

10) Substitute $Q(x_t, u_t)$ with $Q(x_{t+1}, u_{t+1})$.

11) If the parameters are not changed any more or after predefined iterations, the learning procedure is terminated, otherwise, back to step 2.

4.4 Simulation Studies for Vehicle Following Problems

In designing the controller, the dynamic model of car is assumed to be unknown to the controller. The proposed controller adopt reinforcement learning to tune the fuzzy controller, therefore it is a model-free paradigm that is capable of learning the optimal strategy through trial-and-error interactions with a dynamic environment.

Because the objective of the longitudinal controller is to maintain a safe distance between the preceding car and following car, we consider the spacing deviation Δx and relative speed v_r as the two input variables of FIS. Then the fuzzy rules of the adaptive fuzzy controller are considered as follows:

$$R_l: \quad \text{IF} \quad \Delta x \text{ is } F_1^l \text{ and } v_r \text{ is } F_2^l$$

$$\text{THEN} \quad u = K_0^l + K_1^l \Delta x + K_2^l v_r$$

We adopt 9 fuzzy rules to construct the controller, and initially we define three fuzzy sets over the interval [-5,5] for Δx, three fuzzy sets over the interval [-0.5,0.5] for v_r.

The input vector to QEN consisted of seven components or nodes: spacing deviation Δx, relative speed v_r and the control input u. The output to QEN is $Q(x,u)$ corresponding to the action-value function of reinforcement learning. The topology of QEN is considered to be a three-layer structure having 3-10-1 nodes.

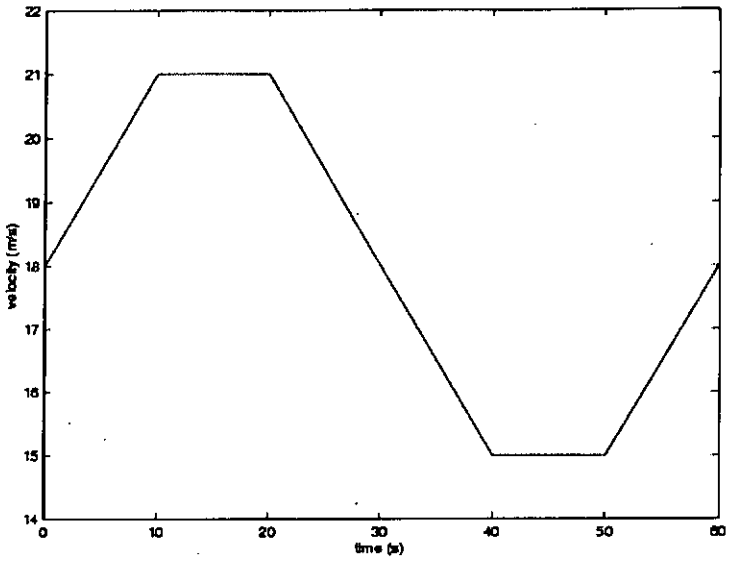


Figure 4.3: The velocity profile of the leading car

In our simulation, the velocity profile of the leading car is shown in Figure 4.3.

The next step is to define the reinforcement signal r. We evaluate the performance of the controller by the maximum absolute value of spacing deviation $|\Delta x|$, because the objective of the longitudinal controller is to maintain a safe distance and make the spacing deviation as small as possible. We think the larger spacing $|\Delta x|$, the poorer performance is for the controller. In our simulations, we think the maximum allowed spacing deviation is 6m. Therefore, failure is defined as the spacing deviation between the preceding car and the following car is larger than 6m. The reinforcement signal r may be defined as follows:

$$r(t) = \begin{cases} -\frac{1}{6}|\Delta x| & |\Delta x| \le 6 \\ -1 & |\Delta x| > 6, \textit{fail} \end{cases}$$

In our simulation, the fuzzy controller based on reinforcement learning was tested for 200 trials (periods) of the velocity profile of the leading car. The controller learns

TABLE 4.1

SUMMARY OF PARAMETERS USED IN THE SIMULATION

Parameters	γ	λ	$\eta(0)$	$\eta(f)$
Value	0.95	0.9	0.25	0.005

the experience, and update the fuzzy parameters continuously based on the reinforcement signal. In our learning procedures, the system is reset to the initial states and resume learning once the "fail" signal happens. The simulation was conducted to evaluate the effectiveness of our control design. The parameters used in the simulation are summarized in Table 4.1 with the proper notations defined in the following:

γ: discount factor of the TD error δ_t;

λ: eligibility rate

$\eta(0)$: initial learning rate of the QEN

$\eta(t)$: learning rate of the QEN at time t which is decreased with t until it reaches 0.005 and it stays at $\eta(f) = 0.005$;

We can see from Figure 4.4 that failures occurred frequently at the beginning of the learning. This means the performance of the initially proposed controller is poor, in other words, the parameters of the controller is not optimal. And the failures often occurred at time near 10s because the acceleration of the leading vehicle is changed suddenly at this time. However, the controller was able to successfully drive the vehicle with the spacing deviation less than 6m all the while after 68 trials.

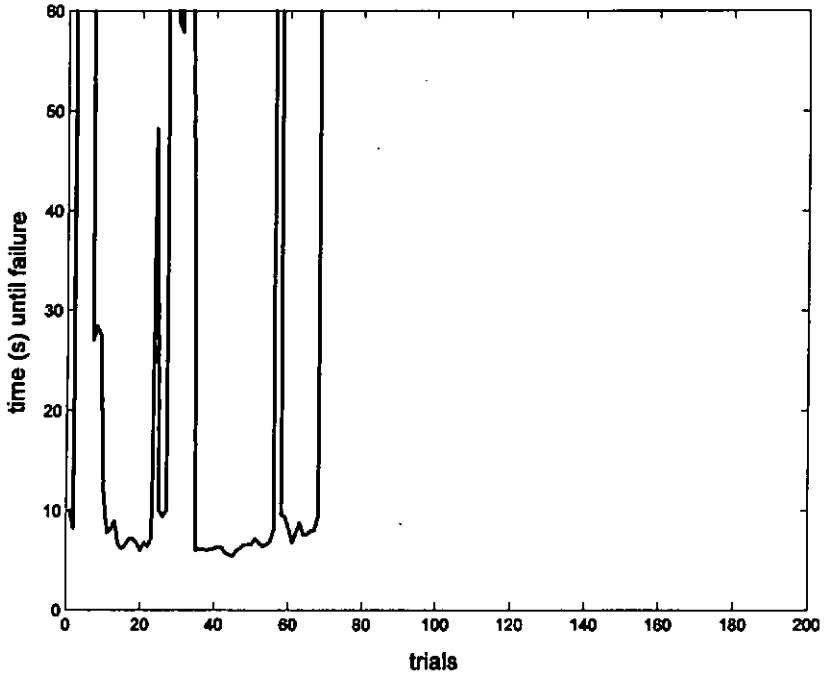


Figure 4.4: Performance of the proposed controller for vehicle following problems

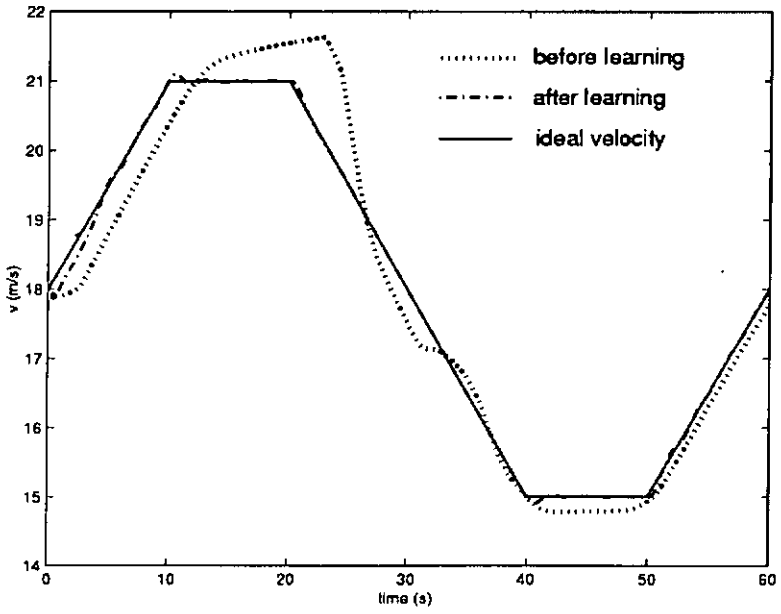


Figure 4.5: Velocity response of the controller before and after learning takes place

We can illustrate the effectiveness of our learning approach by comparing the performance of the controller before learning takes place and after learning takes place. We can see the simulation results clearly from Figure 4.5 and Figure 4.6.

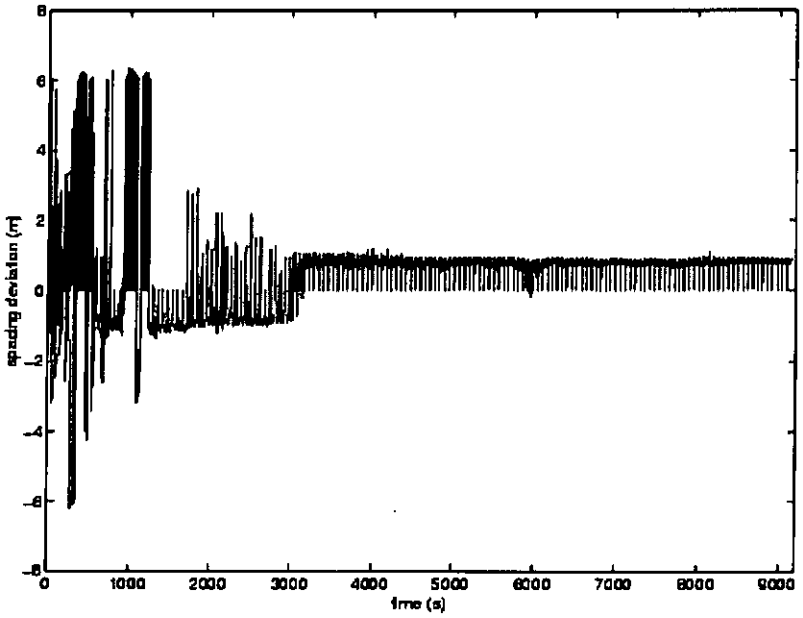


Figure 4.6: Evolution of spacing deviation in simulation

In Figure 4.5, the dot line is the velocity response of the fuzzy controller before learning takes place, the dash-dot line is the velocity response of the fuzzy controller after learning takes place, and the solid line is the leading car velocity. From simulation results, we can conclude:

1) Not only the spacing deviation is decreased gradually, but also the velocity response is better accordingly; this is basic characteristic of reinforcement learning which can continuously improve its performance without a teacher purely based on trial and error.

2) The velocity response of the controlled car after learning is not identical to the leading velocity, this is due to the sudden velocity change of the leading car (we can see the deviation is especially obvious for these unexpected situations) and the limit of number of fuzzy rules,

3) During the learning procedure, the maximum spacing deviation may be increased,

this is due to the exploration strategy (stochastic action modifier) we have implemented. However, the maximum spacing is decreased as a whole.

4) The final spacing deviation is not zero because we only take few fuzzy rules.

It is worth noting here, our proposed method needs no model information and training data, but it can achieve better performance with few fuzzy rules (only 9 rules in our simulations).

4.5 Conclusions

This chapter presents the controller architecture comprised of the QEN and the FIS for solving continuous space reinforcement learning problems. The QEN is used to estimate the optimal action-value function, and the FIS is used to get the control output based on the estimated action-value function provided by the QEN. With the proposed architecture, the parameters adaptation algorithms for the QEN and the FIS are developed based on techniques of temporal difference and gradient descent algorithm. Finally, the simulation studies of vehicle longitudinal control demonstrated the validity and performance of the proposed learning algorithms.

The main advantage of the proposed RL based controller is the model of the process is not required for the adaptation of the controller. The only information available for learning is the system feedback, which describes in terms of reward and punishment the task the fuzzy controller has to realize. And the controller can tune itself to achieve better performance based on these evaluative signals during its

interaction with environment.

A disadvantage of this method is a longer training time since adaptation is only performed after many trials. From the simulation, we know that the performance and the learning speed of our proposed controller are affected by the parameters of the controller, such as the standard deviation of the stochastic action modifier and the learning rate. Currently, we choose these parameters based on a heuristic approach.

Chapter 5

Improvement on Vehicle Longitudinal Controller by DHP

In chapter 4, we have made some achievements for a vehicle longitudinal control system based on the principle of reinforcement learning. The main advantage is that the model of the process was not required for the adaptation of the controller. There are still some problems for us to consider. From the simulation studies, we observed that the performance and the learning speed of our proposed controller were affected by the parameters of the controller, such as the standard deviation of the stochastic action modifier and the learning rate. We selected these parameters based on a heuristic approach. If we choose these parameters not well, the leaning algorithm may not be robust and even fail completely.

Therefore, the efficiency of applying reinforcement learning directly may not be good. In this chapter, we hope to improve the performance of the proposed vehicle control based on dual heuristic programming (DHP), one approach of Adaptive Critic Designs (ACDs).

To deal with control problems, Adaptive Critic Designs have been emerged recently as a synthesis of reinforcement learning, approximate dynamic programming, and backpropagation [Werbos 1990, Prokhorov et al. 1995, Prokhorov & Wunsch 1997]. Adaptive critic methods design controllers in a manner that we need only

include the problem-domain control task and constraints in the Utility Function, the controller will optimize itself gradually based on the two distinct loops: a control training loop and a critic training loop.

It should be noted that the controller design based on Q-learning can be subsumed by adaptive critic designs. But the controller design based on DHP is more effective.

The rest of this chapter is organized as follows. In section 5.1, the basic idea of adaptive critic designs is described. The critic network and action network are given in section 5.2, including their architectures and their adaptation. In section 5.3, the training procedures, simulation results and performance analysis are presented. In section 5.4, the controller proposed in chapter 4 and that in this chapter is compared. Finally, some conclusion remarks are given in section 5.5.

5.1 Foundations of Adaptive Critic Designs

5.1.1 Basic Idea of ACD

The one-step cost (or it can be called "primary" utility function) $U(t)$ should be known for ACD. $U(t)$ represents the instant cost related with the control objective. And $U(t)$ may also contain the imposed constraints such as stability, energy consumption etc. Therefore, the choice of $U(t)$ is an important aspect for controller design. $U(t)$ is similar to r_t in reinforcement learning (see chapter 4). The guideline for controller design is not to optimize the instant cost $U(t)$ only, but optimize the overall cost $J(t)$, which can be written as:

$$J(t) = \sum_{k=0}^{\infty} \gamma^k U(t+k) \tag{5-1}$$

where γ is a discount factor for finite horizon problems ($0 < \gamma < 1$), and $J(t)$ is called secondary utility function, and is similar to Q_t in reinforcement learning (see chapter 4). Here, J and U may be related with state of the plant and control action.

We can approximate $J(t)$ by approximate dynamic programming based on the following equation:

$$J(t) = U(t) + \gamma J(t+1) \tag{5-2}$$

The above equation can be called Bellman's Recursion. We can estimate the overall cost $J(t)$ based on the above equation and function approximators such as neural networks, fuzzy systems or any other building blocks which have the universal approximation properties. The function approximator is named critic network, which has the similar function with Q estimator network (QEN) in chapter 4. We can get the optimal control action based on the principle that the control action minimizes the estimation of $J(t)$. The structure which produces the optimal control action based on the critic network is named action network, which has the similar function with the Takagi-Sugeno type fuzzy inference system (TS-FIS) in chapter 4.

From the above, we notice that the adaptive critic method determines optimal control laws for a system by successively adapting two networks, namely, an action network (which dispenses the control signals) and a critic network (which "learns" the desired performance index for some function associated with the performance index). These two networks approximate the Hamilton-Jacobi-Bellman equation associated

with optimal control theory. The adaptation process starts with a nonoptimal, arbitrarily chosen, control by the action network; the critic network then guides the action network toward the optimal solution at each successive adaptation. During the adaptations, neither of the networks need any "information" of an optimal trajectory, only the desired cost needs to be known. This is the basic idea of adaptive critic designs. The critic network and the action network comprise the architecture of adaptive critic designs.

5.1.2 Remarks

Although multiplayer neural networks are well known to be universal approximators of not only a function itself but also its derivatives with respect to the network's input, Prokhoriov and Wunsch [Prokhorov & Wunsch 1997, Prokhorov 1997] noted that the quality of a direct approximation of derivatives of a function is always better than that of any indirect approximation for given sizes of the network and the training data. In other words, it is expected that DHP should be more effective and contribute a more superior performance than HDP, because the small estimated error of $J(t)$ may result in large error for the derivative of $J(t)$ [Prokhorov & Wunsch 1997, Eaton et al. 2000].

In this chapter, we focus on the ADDHP approach. The reason behind this is that the performance of HDP may be poor although it is the simplest one, and training the critic of GDHP is a complex task [Prokhorov & Wunsch 1997, Prokhorov 1997]. The advantages for adopting ADDHP instead of DHP are that, as we can see clearly later,

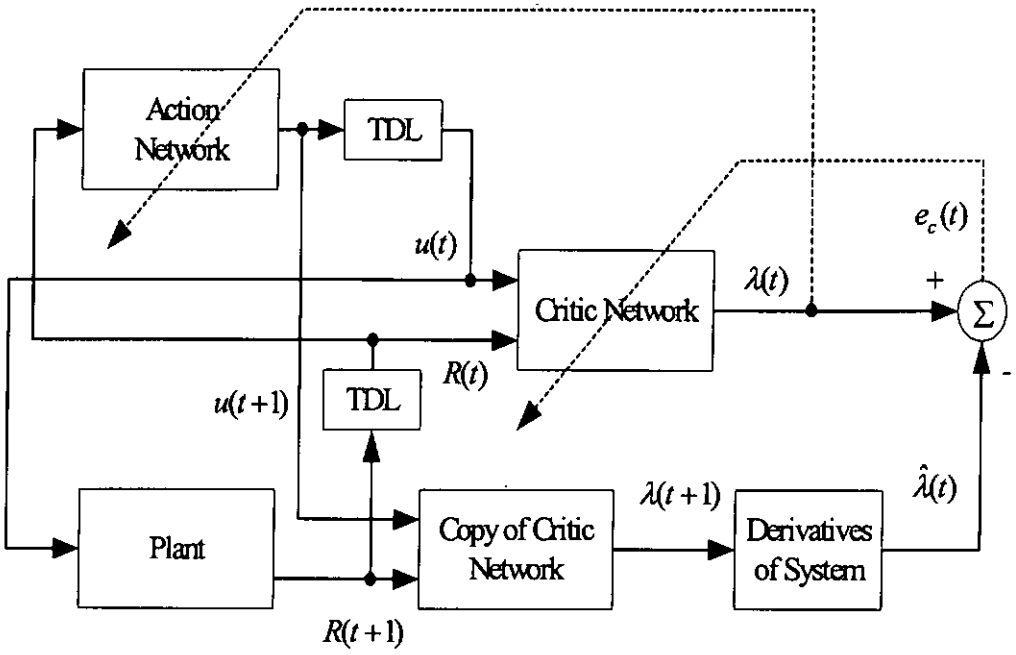


Figure 5.1: The architecture and adaptation in the proposed controller

not only just one output node is needed, but also the training of the action network does not require the system model.

5.2 Structure and Adaptation of the Proposed Controller

As we mentioned in section 5.1, our proposed controller is designed based the principle of ADDHP. The Architecture of the controller is composed of two parts: one is the critic network and the other is the action network. And they have two different roles respectively: the critic network is used to estimate the derivative of the overall cost; the action network is used to provide the control action which would be fed into the system. We adopt three-layer feedforward neural network as critic network, and we construct fuzzy controller as action network.

The architecture of the critic network and the action network is shown in Figure 5.1.

The adaptation of the critic network and action network is also shown in this figure.

5.2.1 Critic Network

The role of the critic network is to estimate the derivative of J (overall cost). The input of the critic network is the states, or observable variables, and because we adopt ADDHP approach, the input of the neural network also includes the control action. The output of the neural network is the derivative of $J(t)$ with respect to $u(t)$, i.e. $\lambda(t) = \frac{\partial J(t)}{\partial u(t)}$. From Bellman's Recursion (equation 5-2), we have

$$\frac{\partial}{\partial u(t)} J(t) = \frac{\partial}{\partial u(t)} \left(U(t) + \gamma J(t+1) \right) \tag{5-3}$$

Because the instant cost (primary utility function) $U(t)$ may be written as the function of state variables and control action, using the chain rule we have:

$$\frac{\partial}{\partial u(t)} U(t) = \sum_i \frac{\partial U(t)}{\partial R_i(t)} \frac{\partial R_i(t)}{\partial u(t)} + \frac{\partial U(t)}{\partial u(t)} \tag{5-4}$$

where, $R_i(t)$ is the input variable of the critic network.

Because the input of the critic network contains state variables and control action, using the chain rule we have:

$$\frac{\partial}{\partial u(t)} J(t+1) = \sum_i \frac{\partial J(t+1)}{\partial R_i(t+1)} \frac{\partial R_i(t+1)}{\partial u(t)} + \frac{\partial J(t+1)}{\partial u(t+1)} \frac{\partial u(t+1)}{\partial u(t)} \tag{5-5}$$

After combination with (5-3), (5-4), and (5-5), we have:

$$\lambda(t) = \sum_i \frac{\partial U(t)}{\partial R_i(t)} \frac{\partial R_i(t)}{\partial u(t)} + \frac{\partial U(t)}{\partial u(t)} + \gamma \left(\sum_i \frac{\partial J(t+1)}{\partial R_i(t+1)} \frac{\partial R_i(t+1)}{\partial u(t)} + \frac{\partial J(t+1)}{\partial u(t+1)} \frac{\partial u(t+1)}{\partial u(t)} \right) \tag{5-6}$$

Because the Bellman's Recursion is only satisfied for the optimal $J(t)$, the value

obtained by (5-6) is the desired value $\hat{\lambda}(t)$ for $\lambda(t)$ actually. So, we can rewrite (5-6) as

$$\hat{\lambda}(t)=\sum_i \frac{\partial U(t)}{\partial R_i(t)}\frac{\partial R_i(t)}{\partial u(t)}+\frac{\partial U(t)}{\partial u(t)}+\gamma\left(\sum_i \lambda(t+1)\frac{\partial u(t+1)}{\partial R_i(t+1)}\frac{\partial R_i(t+1)}{\partial u(t)}+\lambda(t+1)\frac{\partial u(t+1)}{\partial u(t)}\right) \tag{5-7}$$

The fact $\lambda(t+1)=\dfrac{\partial J(t+1)}{\partial u(t+1)}$ is used in the above equation.

To evaluate the right side of (5-7), the system dynamics which contains the terms $\dfrac{\partial R_i(t+1)}{\partial u(t)}$, $\dfrac{\partial u(t+1)}{\partial u(t)}$, $\dfrac{\partial u(t+1)}{\partial R_i(t+1)}$ and $\dfrac{\partial R_i(t)}{\partial u(t)}$ are required, and this may need the system model. For this reason, DHP may be also called model-based method.

In most of the literature of ACD, the complete system model must be needed to get the plant derivatives. This model may be assumed to know [Lendaris et al. 1999], or may be obtained by system identification [Prokhorov & Wunsch 1997, Eaton et al. 2000, Venayagamoorthy et al. 2002], or by perturbation through plant model [Schultz et al. 2001]. However, it may be unavailable or difficult to obtain the system derivatives by obtaining the system model first. Much more, if the model is known, some other design approach may be better.

Nevertheless, our proposed controller does not require the complete vehicle model, and we do not construct a vehicle model using neural networks or fuzzy systems. As we can see later, the model information for the proposed controller is very little about vehicle longitudinal model.

Now, we discuss the critic network design for the special case of vehicle longitudinal control. From chapter 2, we know the task of vehicle longitudinal control

can be described as $v_r + k \cdot \Delta x = 0$. Therefore, we define $e(t)$ and construct the primary utility function as:

$$e(t) = v_r(t) + k \cdot \Delta x(t) \tag{5-8}$$

$$U(t) = e^2 = (v_r + k \cdot \Delta x)^2 \tag{5-9}$$

For the critic network design, it worth noting here that we should keep the number of the input variables as small as possible. This is because we use the critic network as a function approximator. If the neural network has large number of input nodes, the training procedure would be very slow because more training samples are required (increase with the input nodes exponentially) for the same estimation accuracy, we may call this phenomenon as "curse of dimensionality". For the vehicle longitudinal control problem, only one input node is needed besides the control action $u(t)$. We take spacing deviation $\Delta x(t)$ as input variable. This information may be enough for estimating the derivative of $J(t)$, because spacing deviation can reflect the performance of the controller clearly for longitudinal control.

So, we choose the critic network structure for the vehicle longitudinal problem as a three-layer feedforward neural network with 2 inputs, a hidden layer of 5 nodes, and 1 output.

After we have the configuration of the critic network, we discuss the training of the critic network.

Combining with (5-7), (5-8), (5-9) and (2-1), we have the expected $\lambda(t)$ as follows:

$$\hat{\lambda}(t) = 2e(t)\frac{\partial e(t)}{\partial u(t)} + \gamma\lambda(t+1)\left(\frac{\partial u(t+1)}{\partial u(t)} - \frac{\partial u(t+1)}{\partial \Delta x(t+1)}\frac{\partial x(t+1)}{\partial u(t)}\right) \tag{5-10}$$

The critic network tires to minimize the following error measure at time t :

$$E_c(t) = \left(\lambda(t) - \hat{\lambda}(t)\right)^2 \tag{5-11}$$

The parameters update equations of the critic network are as follows:

$$\theta(t+1) = \theta(t) - \eta_c \frac{\partial E_c(t)}{\partial \theta(t)} = \theta(t) - \eta_c \left(\lambda(t) - \hat{\lambda}(t)\right) \frac{\partial \lambda(t)}{\partial \theta(t)} \tag{5-12}$$

where, θ is the weights of the neural network, η_c is learning rate of the neural network. Backpropagation can determine all the update of the parameters of the critic network.

In order to obtain the desired $\partial J(t)/\partial u(t)$ from (5-10), we need to know the value of $\lambda(t+1)$, which corresponds to the derivative of J with $\Delta x(t+1)$ and $u(t+1)$. This seems difficult because we do not know $\Delta x(t+1)$ and $u(t+1)$ at time t if the system model is not provided. But we can use the "forward-in-time" method to solve this problem [Liu 2002]. The basic idea of "forward-in-time" method is to obtain the actual value of $\Delta x(t+1)$ and $u(t+1)$ after $u(t)$ is fed into the system. Then, we can obtain $\lambda(t+1)$ from the copy of the critic network at time t based on $\Delta x(t+1)$ and $u(t+1)$. Finally, we could update the critic network at time t instead of time $t + 1$.

From (5-10), we can see that the much troublesome problem for vehicle longitudinal control is how to obtain $\partial e(t)/\partial u(t)$, $u(t+1)/u(t)$, $u(t+1)/\Delta x(t+1)$ and $x(t+1)/u(t)$ to get the desired $\partial J(t)/\partial u(t)$. The details will be discussed in section 5.2.2.

5.2.2 Action Network

The role of action network is to produce the control action which would be fed into the system. Here, "network" means the controller of any forms whose output is differential with respect to the tuned parameters, although neural network is adopted frequently [Eaton et al. 2000, Venayangamoorthy et al. 2002]. The strength of neural network is the promise of fast computation, versatile representational ability of nonlinear maps, fault tolerance, and the capability to generate quick, robust, suboptimal solutions [Venayangamoorthy et al. 2002]. But the interpretation of neural networks may be rather opaque. It is also difficult to incorporate a priori knowledge into neural networks. The above drawbacks of neural networks may make the training procedure slow. More badly, it may even make the system unstable. Some strategies are taken to deal with this problem, in [Venayangamoorthy et al. 2002], the action neural network is pretrained with conventional controllers controlling the plant in a linear region. Another promising strategy is to adopt fuzzy controller as the action network, because the fuzzy controller offers a way around the difficulty of opaque interpretation in many application contexts.

However, the application of fuzzy control to complex system may be not a trivial task because the size of the rule base in a typical fuzzy control architecture will be increase exponentially with the size of variables [Jamshidi 1997]. To handle this exponential explosion of the size of the rule base, sensory fusion or hierarchical fuzzy control may be adopted.

In this chapter, we also employ Takagi-Sugeno (TS) fuzzy controller as the action

network for the special case of vehicle longitudinal control. TS model offers an analytical expression for adaptive critic designs. The control action $u(t)$ is related with the relative velocity $v_r(t)$ and the spacing deviation $\Delta x(t)$. We also know the current velocity of the vehicle $v(t)$ also affects $u(t)$. Meanwhile, in order to reduce the fuzzy rules, we fuse $\Delta x(t)$, $v_r(t)$ as $e(t) = v_r(t) + k\Delta x(t)$ (see equation (5-8)). From the above consideration, the input variables of TS type fuzzy controller fro longitudinal control is $e(t)$ and $v(t)$.

The consequence of TS fuzzy controller is a mathematical expression. We choose the form of the consequence based on the works of Yanakiev and Kanellakopoulos [Yanakiev & Kanellakopoulos 1996, 2001]. They did not employ fuzzy controller, but the performance of their proposed adaptive PIQ controller is good based on a nonlinear reference model with autonomous operation. Because the performance of the conventional PI controller is not acceptable, a signed quadratic (Q) term of the form $(v_r + k\Delta x)|v_r + k\Delta x|$ is added to the PI controller. Then the controller is more aggressive at large errors, but does not have the undesirable side effect of overshoot with the signed Q term. The form of their proposed adaptive PIQ is as follows:

$$u = \hat{k}_p(v_r + k \cdot \Delta x) + \hat{k}_i + \hat{k}_q(v_r + k \cdot \Delta x)|v_r + k \cdot \Delta x| \qquad (5\text{-}13)$$

As we know, the effective aerodynamic drag of vehicle is proportional to the square of the current velocity of the vehicle $v(t)$, and the aerodynamic drag is negative to the control action. Therefore, we compensate the aerodynamic drag using the term $k_v v^2(t)$ in the consequence of the fuzzy rules. This can be looked as a priori knowledge which is incorporated into the fuzzy controller.

Then, the fuzzy rules of the action network are as follows:

$$R_l: \quad \text{IF} \quad e(t) \text{ is } F_1^l \text{ and } v(t) \text{ is } F_2^l,$$

$$\text{THEN} \quad u(t) = k_p^l e(t) + k_i^l + k_q^l e(t)|e(t)| + k_v^l v^2(t) \qquad (5\text{-}14)$$

where, F_1^l and F_2^l are the labels of the fuzzy sets for $e(t)$ and $v(t)$ respectively. k_p^l, k_i^l k_q^l and k_v^l are the constant coefficients of the consequent part of the fuzzy rules.

We can obtain the total output of fuzzy controller as the following expression:

$$u(t) = \frac{\sum_l m_l \left(k_p^l e(t) + k_i^l + k_q^l e(t)|e(t)| + k_v^l v^2(t)\right)}{\sum_l m_l} \qquad (5\text{-}15)$$

where, m_l is the membership degree for the l th rule.

The goal of the action network is to minimize $J(t)$, thereby optimizing the overall cost expressed as a sum of all $U(t)$ over the horizon of the problem. This is achieved by training the action with $\lambda(t) = \partial J(t)/\partial u(t)$, which we can obtain directly from the output of the critic network. Therefore, we can obtain the update of the parameters of the action network as follows:

$$\alpha(t+1) = \alpha(t) - \eta_A \frac{\partial J(t)}{\partial \alpha(t)} = \alpha(t) - \eta_A \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial \alpha(t)} = \alpha(t) - \eta_A \lambda(t) \frac{\partial u(t)}{\partial \alpha(t)} \qquad (5\text{-}16)$$

where, α is the parameters of action network, η_A is the learning rate. In order to calculate (5-16), we also need to get $\partial u(t)/\partial \alpha(t)$.

For the special case of longitudinal control, it is easy to get $\partial u(t)/\partial \alpha(t)$ from the (5-16).

$$\frac{\partial u(t)}{\partial k_p^l} = \frac{m_l}{\sum_k m_k} e(t), \quad \frac{\partial u(t)}{\partial k_i^l} = \frac{m_l}{\sum_k m_k}, \quad \frac{\partial u(t)}{\partial k_q^l} = \frac{m_l}{\sum_k m_k} e(t)|e(t)|, \quad \frac{\partial u(t)}{\partial k_v^l} = \frac{m_l}{\sum_k m_k} v^2(t)$$

(5-17)

Combining (5-16) and (5-17), we can obtain the update rules of the parameters of the fuzzy rules. Here, we assume the membership functions are fixed, and we only tune the sequence of fuzzy rules.

We can also obtain the derivative of control action $u(t)$ with respect to the input variables $v(t)$ and $e(t)$:

$$\frac{\partial u(t)}{\partial v(t)} = \frac{\sum_l 2m_l k_v^l v(t)}{\sum_l m_l}, \quad \frac{\partial u(t)}{\partial e(t)} = \frac{\sum_l m_l \left(k_p^l + 2k_q^l |e(t)|\right)}{\sum_l m_l} \tag{5-18}$$

Now, we return to the problem of how to obtain $\partial e(t)/\partial u(t)$, $u(t+1)/u(t)$, $u(t+1)/\Delta x(t+1)$ and $x(t+1)/u(t)$ to get the desired $\partial J(t)/\partial u(t)$, which was mentioned in section 5.2.1.

1) We can get $\partial u(t)/\partial e(t)$ from (5-18), but how to get $\partial e(t)/\partial u(t)$? We think the absolute value of $\partial e(t)/\partial u(t)$ can be determined by the reciprocal of the absolute value of $\partial u(t)/\partial e(t)$, and the sign of $\partial e(t)/\partial u(t)$ is the inverse of the sign of $\partial u(t)/\partial e(t)$, we can explain this as follows: $u(t)$ will increase when $e(t)$ increase, but $e(t)$ will decrease when $u(t)$ increase. So, we have

$$\frac{\partial e(t)}{\partial u(t)} = -\left(\frac{\partial u(t)}{\partial e(t)}\right)^{-1} \tag{5-19}$$

2) Based on $x(t+1) = x(t) + v \cdot \Delta t + \frac{1}{2} a(\Delta t)^2$ and $a \propto \frac{u}{M}$, we can estimate $\partial x(t+1)/\partial u(t)$ as:

$$\frac{\partial x(t+1)}{\partial u(t)} = \frac{(\Delta t)^2}{2M} \tag{5-20}$$

where, M is the mass of the vehicle, Δt is the step size.

Based on $v(t+1) = v(t) + a \cdot \Delta t$, we can estimate $\partial v(t+1)/\partial u(t)$ as:

$$\frac{\partial v(t+1)}{\partial u(t)} = \frac{\Delta t}{M} \tag{5-21}$$

3) Based on $e(t) = v_r(t) + k \cdot \Delta x(t)$, we have the following expression from (5-20) and (5-21):

$$\frac{\partial e(t+1)}{\partial u(t)} = -\frac{\partial v(t+1)}{\partial u(t)} - k\frac{\partial x(t+1)}{\partial u(t)} = -\frac{2\Delta t + k(\Delta t)^2}{2M} \tag{5-22}$$

Using the chain rule, we have:

$$\frac{\partial u(t+1)}{\partial u(t)} = \frac{\partial u(t+1)}{\partial e(t+1)}\frac{\partial e(t+1)}{\partial u(t)} + \frac{\partial u(t+1)}{\partial v(t+1)}\frac{\partial v(t+1)}{\partial u(t)} \tag{5-23}$$

In (5-23), we can get $\partial u(t+1)/\partial e(t+1)$ and $\partial u(t+1)/\partial v(t+1)$ from (5-17), so we can get $\partial u(t+1)/\partial u(t)$ by combing (5-18), (5-21) and (5-22).

4) We can obtain $\partial u(t+1)/\partial \Delta x(t+1)$ based on $e(t) = v_r(t) + k \cdot \Delta x(t)$:

$$\frac{\partial u(t+1)}{\partial \Delta x(t+1)} = \frac{\partial u(t+1)}{\partial e(t+1)}\frac{\partial e(t+1)}{\partial \Delta x(t+1)} = k\frac{\partial u(t+1)}{\partial e(t+1)} \tag{5-24}$$

where, $\partial u(t+1)/\partial e(t+1)$ can be determined from (5-18) with $e(t+1)$ instead of $e(t)$.

For the above estimations of $\partial e(t)/\partial u(t)$, $u(t+1)/u(t)$, $u(t+1)/\Delta x(t+1)$ and $x(t+1)/u(t)$, we have the following remarks.

Remarks:

1) In most of the previous papers, the system which may be obtained before hand or by system identification is required. The function of the system is to obtain the derivative values of the system for the training of critic network and

network, and get the next state for obtaining $\lambda(t+1)$. However, in this chapter the system model is not required throughout the development of the critic network and the action network. The only information needed is some common sense such as the acceleration of the vehicle is proportion to the control effort, and is inverse proportion to the mass of the vehicle ($a \propto \frac{u}{M}$).

2) It is pointed out by T. Shannon and G. Lendaris that as long as the approximate derivative values obtained had the correct sign (positive or negative) most of the time, the model was adequate for use in DHP, in other words, only qualitative model is required [Shannon 1999, Lendaris & Shannon 1999]. Although we do not obtain (5-21), (5-22) and (5-23) based on the accurate vehicle model, this is not very important, and the estimated derivative values are enough for successful controller training.

3) Because TS type fuzzy controller is employed in this chapter, we can incorporate the priori knowledge into the consequences of fuzzy rules. We can adopt the expression of the controller developed by other researchers as the consequence of fuzzy rules. We can also add the term in the fuzzy controller to eliminate the aerodynamic drag. Obviously, all of these make the controller more interpretable, training much faster and even more stable compared with the neural network controller.

4) To avoid "curse of dimensionality", the input variables of the critic network and action network should be kept as small as possible, so some measures such as sensor fusion are taken.

5.3 Simulation Studies of Vehicle Longitudinal Control

5.3.1 Training Procedures for the Critic and Action Network

Before training the critic and action network, we may prestructure the consequence parameters of TS type fuzzy controller using a priori knowledge such as the experience of human operators or some developed conventional controllers.

There are two approaches to train the critic and action network: one consists of two separate training cycles; and another trains the critic network and action network simultaneously. The former approach only tunes the critic's parameters initially with the prestructured action network, to ensure the whole system not to introduce instabilities. After the critic training, the action network is trained further while keeping the critic's parameters fixed. This process of training the critic and action one after the other, is repeated until an acceptable performance is reached.

In this chapter, we adopt the latter approach to train the critic and action network simultaneously, since the simultaneous stepping approach is about twice as fast as the alternating approach [Shannon & Lendaris 2000], and no training process becomes instable in our simulation.

The training procedures for the critic and action network are as follows:

1) Repeat the following steps until the acceptable performance is reached.
2) Initialize $t = 0$ and the state variables of the plant.
3) Obtain the measurement of $v(t)$, $e(t) = v_r(t) + k \cdot \Delta x(t)$ based on (2-1), (5-8), then get the output $u(t)$ of TS fuzzy controller at time t based on (5-15).

4) Calculate the critic output $\lambda(t)$ based on $\Delta x(t)$, $u(t)$ and the critic network at time t.

5) Obtain the next state $v(t+1)$, $v_r(t+1)$ and $\Delta x(t+1)$ when $u(t)$ from step 4 is fed into the vehicle system.

6) Get the output $u(t+1)$ of TS fuzzy controller based on (5-15), the parameters of the fuzzy control is still unchanged, i.e. the parameters are same as those at time t.

7) Calculate the critic output $\lambda(t+1)$ based on $\Delta x(t+1)$, $u(t+1)$ and the critic network at time t.

8) Update the parameters of the fuzzy controller based on (5-16) and (5-17).

9) Calculate the desired critic output $\hat{\lambda}(t)$ based on (5-10), (5-19)-(5-24).

10) Update the parameters of the critic neural network based on $\lambda(t)$ obtained from step 4, $\hat{\lambda}(t)$ obtained from step 9 and equation (5-12).

11) If t less than the termination time (the duration of one trial) return to step 3), else return to step 2).

5.3.2 Simulation Results

We demonstrate the proposed controller on a vehicle longitudinal control system. The profile of the velocity and acceleration of the preceding car is shown Figure 5.2.

$$v_l(t) = \frac{75}{2\pi}\left(1-\cos(0.04\pi t)\right), \quad a_l(t) = 1.5\sin(0.04\pi t) \tag{5-25}$$

The whole simulation lasts 100 s. The system is sampled at 20Hz and the numerical simulation is performed with a fixed step size 4th order Runge-Kutta algorithm.

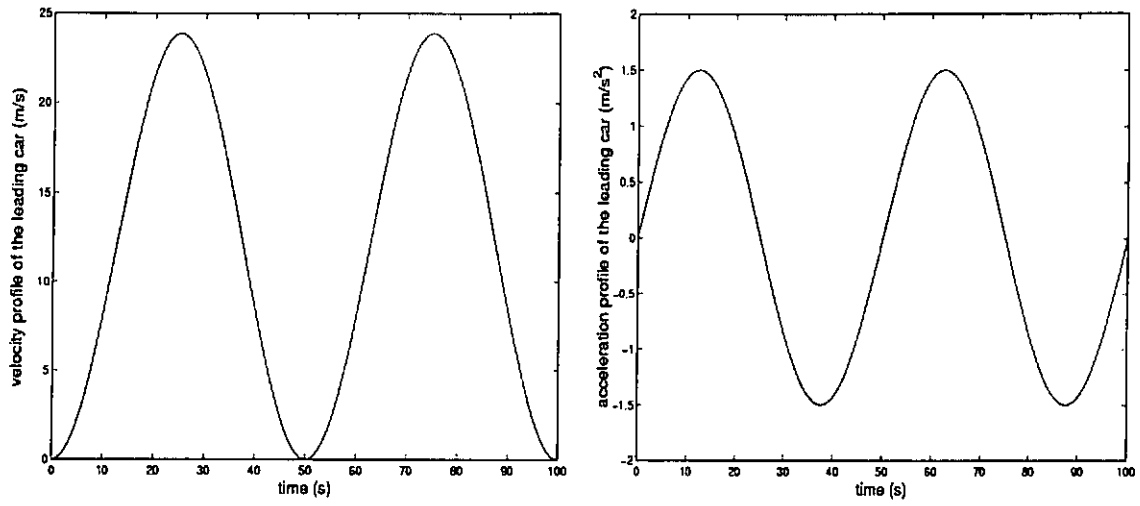


Figure 5.2: The velocity (left) and acceleration (right) profile of the leading car

As we mentioned in section 5.2.1, the critic network structure for the vehicle longitudinal problem is a three-layer feedforward neural network with 2 inputs $u(t)$ and $\Delta x(t)$, a hidden layer of 5 nodes, and 1 output $\lambda(t) = \partial J(t)/\partial u(t)$. The activation function of the hidden nodes is sigmoidal function, and that of the output node is linear function.

We define 9 fuzzy sets over the interval [-10, 10] for $e(t)$, 4 fuzzy sets over the interval [0, 30] for $v(t)$. We simply take the membership of $e(t)$ and $v(t)$ as triangle shape (see Figure 5.3). So, the TS type fuzzy controller consists of 36 rules totally.

In our simulation, the critic and action network were trained simultaneously for 100 trials. Based on the analysis of the previous sections and [Yanakiev & Kanellakopoulos 2001, Shannon & Lendaris 2000], the parameters used in the simulation are summarized in Table 5.1 with the proper notations defined in the following:

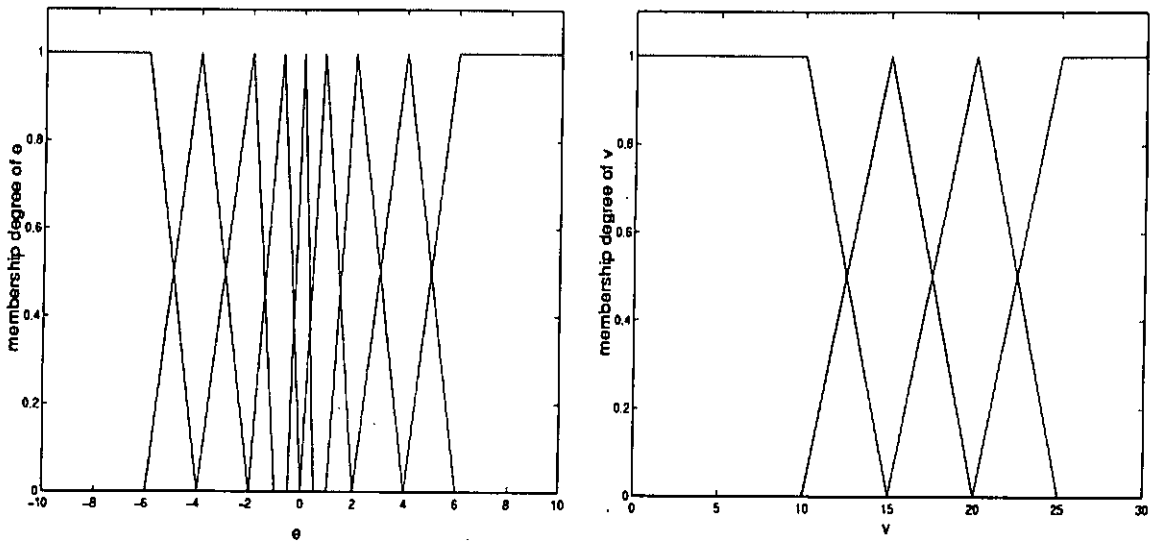


Figure 5.3: The membership degree of $e(t)$ (left) and $v(t)$ (right)

γ : a discount factor for finite horizon problems

k : the parameter of fusing relative speed $v_r(t)$ and spacing deviation $\Delta x(t)$

η_C : the learning rate of critic neural network

k_p^l : one of the coefficients of consequence of l th fuzzy rule in (3-14)

k_i^l : one of the coefficients of consequence of l th fuzzy rule in (3-14)

k_q^l : one of the coefficients of consequence of l th fuzzy rule in (3-14)

k_v^l : one of the coefficients of consequence of l th fuzzy rule in (3-14)

η_{Ap} : the learning rate for k_p^l

η_{Ai} : the learning rate for k_i^l

η_{Aq} : the learning rate for k_q^l

η_{Av} : the learning rate for k_v^l

TABLE 5.1

SUMMARY OF PARAMETERS USED IN SIMULATION

Parameters	γ	k	η_C	k_p^l	k_i^l	k_q^l
Value	0.9	1	0.001	100	0	0
Parameters	k_v^l	η_{Ap}	η_{Ai}	η_{Aq}	η_{Av}	
Value	0	1	1	0.05	0.001	

Here, we only select k_p^l as 100, other parameters in fuzzy consequence all equal to 0, this means the priori knowledge of us is very limited, we only take the controller similar with P controller initially.

We can illustrate the effectiveness of our learning approach by comparing the vehicle velocity after 1st learning procedure and 100th learning procedure. We can see the simulation result from Figure 5.4.

We observe that the velocity of the controlled vehicle is nearly identical to the velocity of the leading car after training, although there is a large deviation when the controller is initially trained. This means that the fuzzy controller learns the information from the critic neural network, then tune itself towards good performance. The critic information may be not accurate at the beginning of the training procedure, however, the critic neural network is also trained continuously to estimate the derivative of the overall cost with respect to the current control. The simultaneous training of the critic and action network makes the controller better and better.

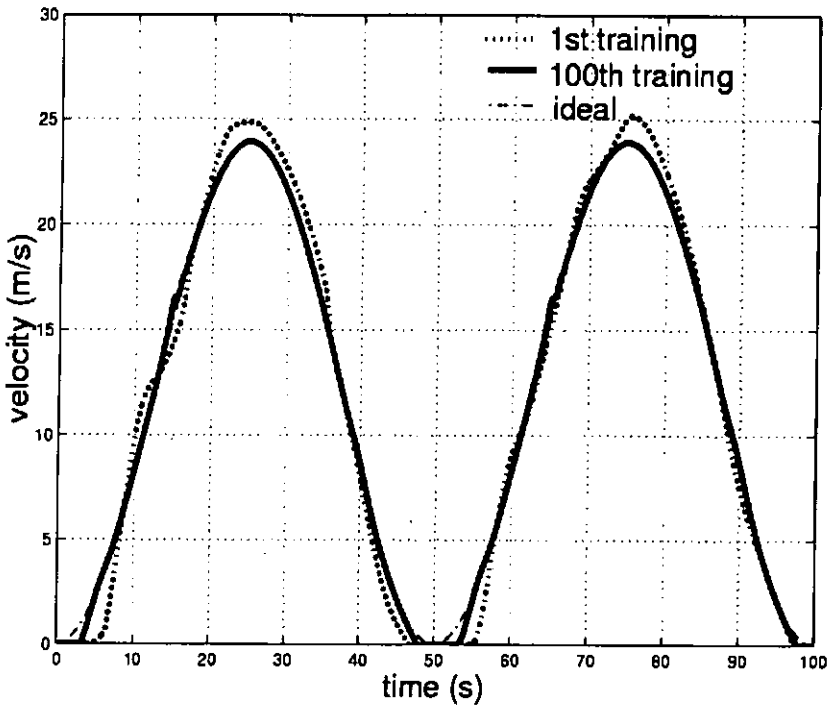


Figure 5.4: The comparison of the vehicle velocity after 1st training procedure and 100th training procedure

We can see the spacing deviation and relative speed between the preceding car and following car after 100 trials clearly from Figure 5.5 and Figure 5.6 respectively. The maximum spacing deviation is less than 2.0 m and the maximum relative speed is less than 1.0 m/s. We also observe that there are some obvious overshoots for spacing deviation and relative speed. This is expected due to the maximum acceleration of the leading car at these points.

The evolutions of the spacing deviation and relative speed with trials are shown in Figure 5.7 and Figure 5.8 respectively. We are glad to see that both of the spacing deviation and relative speed decrease monotonously when trials increase. This explains the efficiency of the proposed controller.

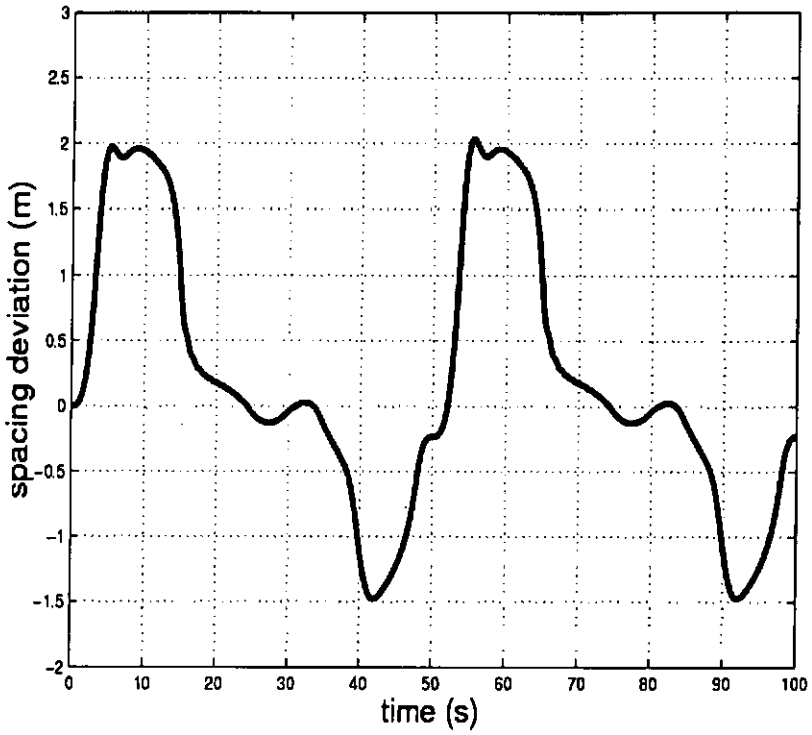


Figure 5.5: The spacing deviation between the preceding car and following car after 100 training trials

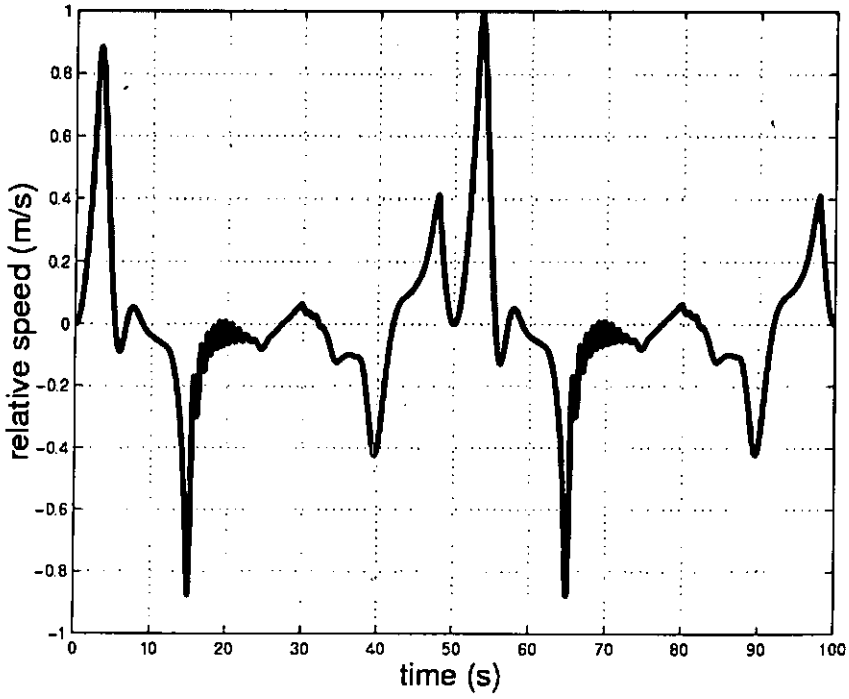


Figure 5.6: The relative speed between the preceding car and following car after 100 training trials

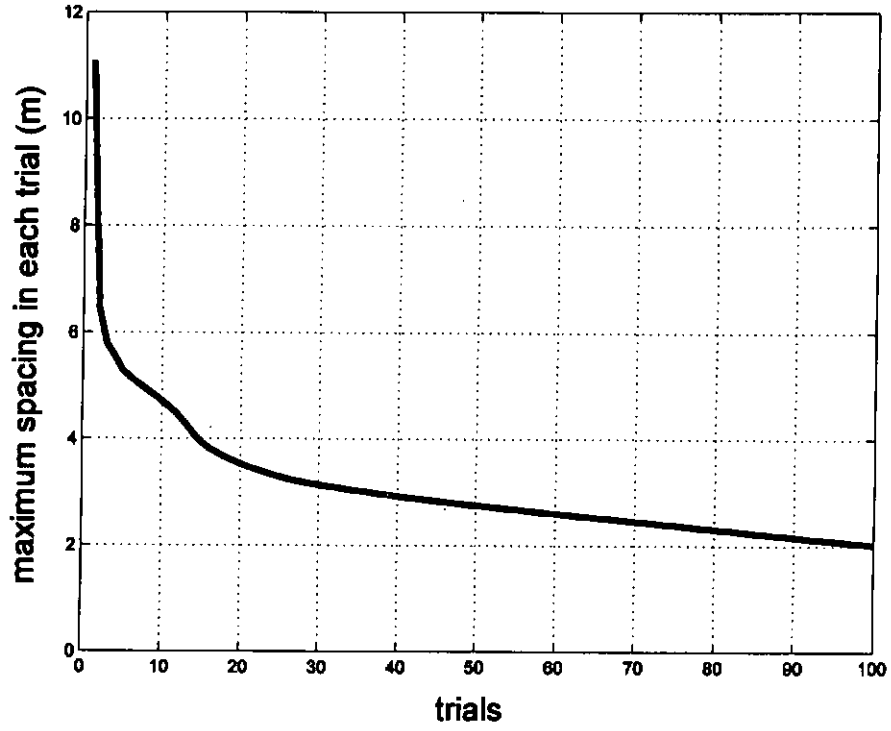


Figure 5.7: The evolution of the maximum spacing deviation in each trial with the training trials

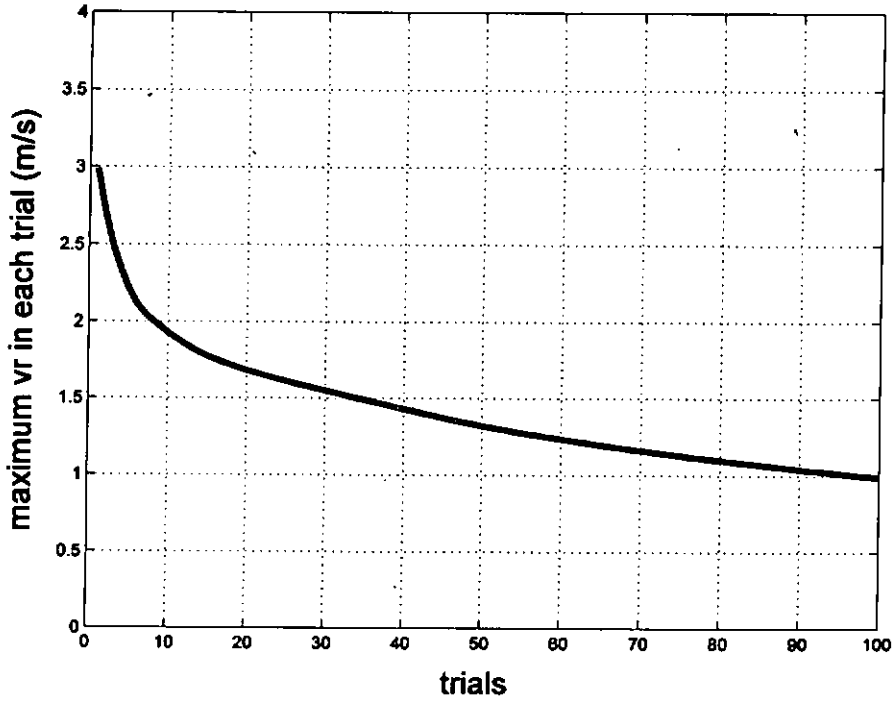


Figure 5.8: The evolution of the maximum relative speed in each trial with the training trials

We should emphasize here, the performance of our controller is limited by (or depends on) the following factors:

1) The measurement values we can get: it is obvious that the performance of the controller will be better if more measurement values are obtained. For the special case of vehicle longitudinal control, if we could know the future velocity of the leading car, the performance of the longitudinal controller must be better.

2) The structure of the controller: for the special case of vehicle longitudinal control, if we select more fuzzy rules or obtain a more optimal form of consequence, the performance may be improved.

3) The model information we used: if we can obtain more accurate derivative information based on accurate model, the performance may be better. So, it may be not realistic to expect that the spacing deviation and relative speed tend to be zero for our simulations.

In our simulation, we assume that we can only obtain the following input variables: the current position and velocity of the controlled car and the leading car. In addition, we think we do not know the vehicle models. Nevertheless, the proposed controller achieved satisfied performance. The "performance cost ratio" of the proposed controller is high.

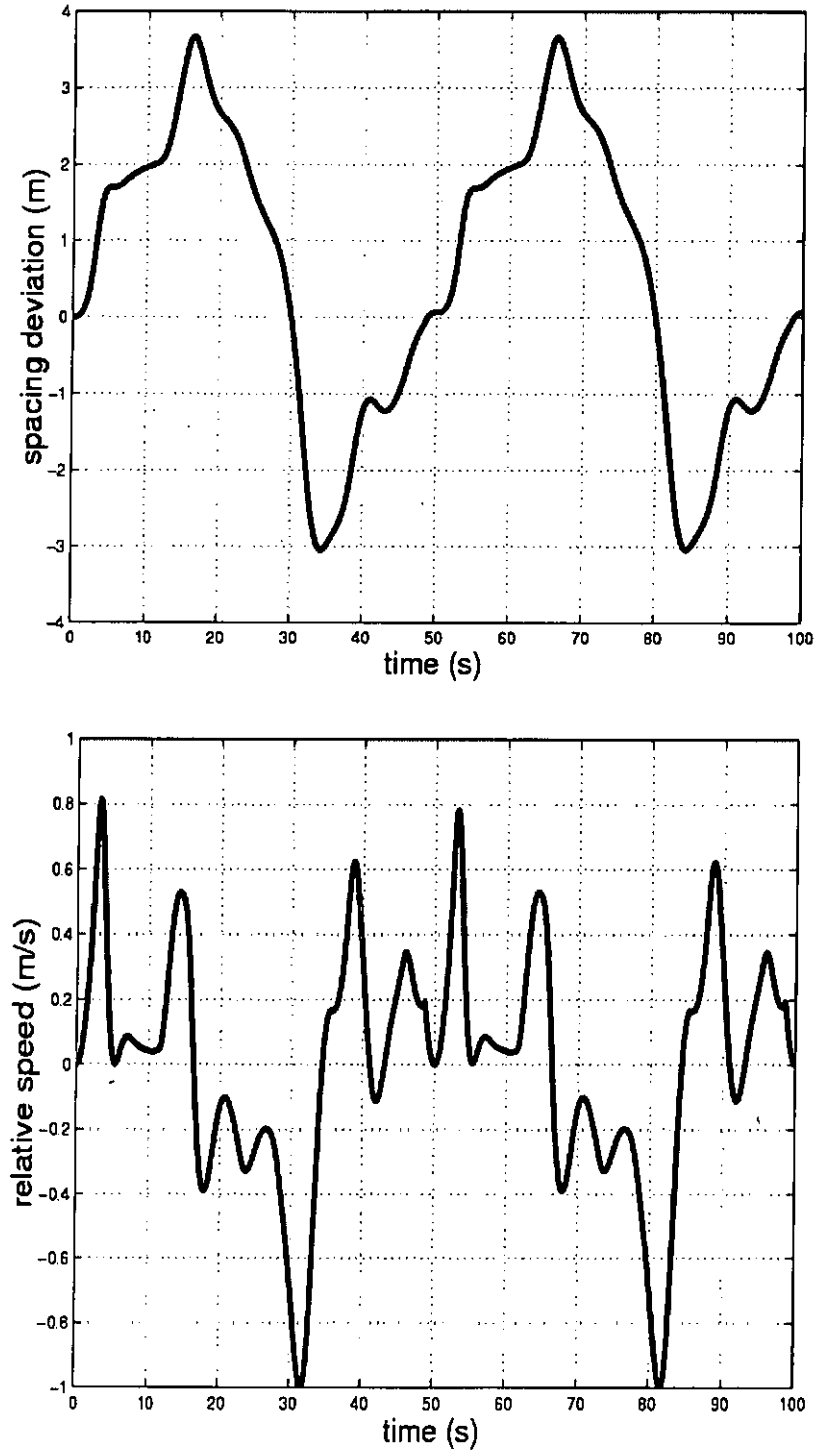


Figure 5.9: The spacing deviation (top) and relative speed (bottom) between the preceding car and following car after 100 training trials without k_q^l and k_v^l

As comparison, if we take the fuzzy rules without "Q" term and aerodynamic drag compensating term (i.e. no k_q^l and k_v^l), the performance of the controller is

degraded, which is shown in Figure 5.9. The maximum spacing deviation with k_q^l and k_v^l in Figure 5.5 is about 2.0 m, while the maximum spacing deviation without k_q^l and k_v^l in Figure 5.9 is about 3.7 m. And the relative speed is more oscillated in Figure 5.9 than that in Figure 5.6. So, we should keep it in mind that a priori knowledge should be incorporated into the controller design to enhance the performance. This also explains the advantage of taking fuzzy controller as the action network.

5.4 Comparisons

At the end of this chapter, we would compare the controller proposed in chapter 4 and that in chapter 5. We abbreviate the former controller as RLC (reinforcement learning controller) since it is based on reinforcement learning; and abbreviate the latter controller as DHPC (dual heuristic programming controller) since it is based on dual heuristic programming.

First, we list the similarities between RLC and DHPC:

1) The main structures of RLC and DHPC both comprise two networks, namely, an action network (which dispenses the control signals) and a critic network (which "learns" the desired performance associated with the performance index).

2) The parameters update laws are both developed based on the output of critic network.

3) For RLC and DHPC design, we need only include the problem-domain

control task and constraints in the cost (r for RLC and $U(t)$ for DHPC), and system model may not required.

4) RLC and DHPC offer a unified approach to dealing with the controller's nonlinearity.

5) RLC and DHPC can tune themselves to achieve better performance during its interaction with environment.

Although the above similarities, there are some obvious differences between these two controllers:

1) Derivative information is required for DHPC, while this is not required for RLC.

2) Stochastic action modifier (SAM) is included in the controller structure of RLC, the performance of RLC may be unacceptable if we do not employ SAM to implement the exploration strategy. While this is not necessary for DHPC.

3) For RLC, the parameters of the architecture adapt only by means of the scalar cost, so it has been shown to converge very slowly [Werbos 1990]. While for DHPC, the critic network approximates the derivatives of $J(t)$ with respect to the state, thereby correlating the adjustable parameters in the architecture to a larger number of dependent variables. And also because we use the derivatives of $J(t)$ to tune the parameters of action network, there are some advantages of DHPC over RLC from a theoretical point of view.

4) It is expected that the performance of DHPC is better than that of RLC, we

can verify this from the following simulations.

We will compare their performance by simulation studies. It should be noted here that the velocity profile of preceding car and parameters of vehicle longitudinal control simulation studies are same for RLC and DHPC.

The performance of RLC after training of 100 trials is shown in Figure 5.10, which includes spacing deviation and relative speed between the preceding car and the controlled car. Comparing with Figure 5.5 and Figure 5.6, which are the performances of DHPC with the same conditions, we find that although the spacing deviation of RLC is nearly same as that of DHPC, the relative speed of RLC is more fluctuant than that of DHPC. The fluctuation of relative speed means the control smoothness is not good for RLC. Control smoothness affects the comfort of passengers and drivers. And improving control smoothness can enhance safety and reducing fuel consumption.

Besides the controller performance after learning, we should also pay attention to the learning procedure since it is important for practical problems. The evolution of the maximum spacing deviation and the maximum relative speed during the learning procedure of RLC is shown in Figure 5.11. Although the maximum spacing deviation and the maximum relative speed decrease from the whole training, we can see some large oscillations during the learning of RLC, especially at the beginning of learning procedure. This is due to the stochastic action modifier of RLC, which implements exploration strategies thereby increasing the uncertainties and oscillations for the controller. These oscillations may destroy the practical system before it can achieve good performance. This is not expected, although we can not see the negative effects

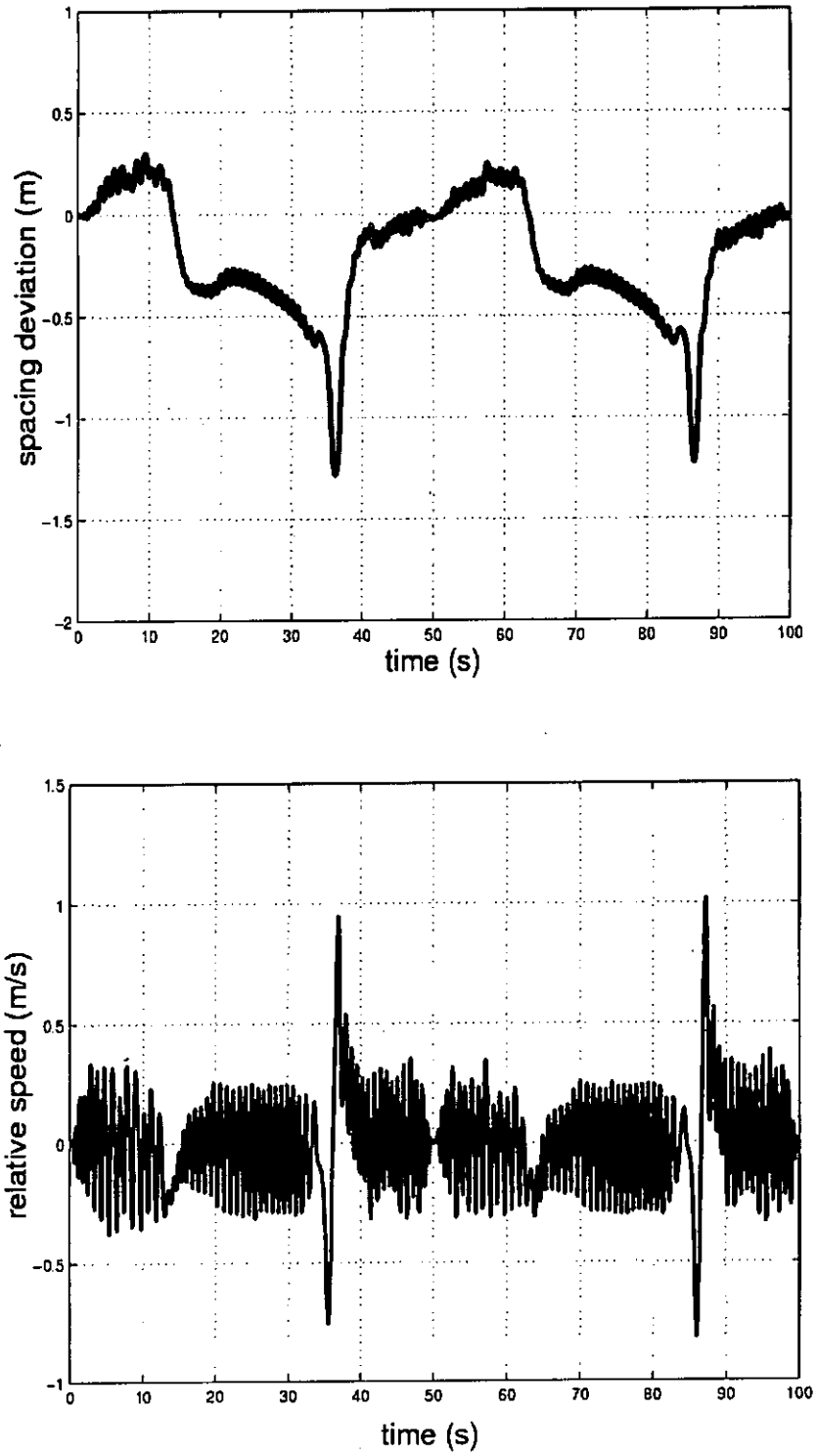


Figure 5.10: The spacing deviation (top) and relative speed (bottom) of RLC

for simulation studies. As a comparison, we can refer the learning procedure of DHPC from Figure 5.7 and Figure 5.8, the maximum spacing deviation and the maximum

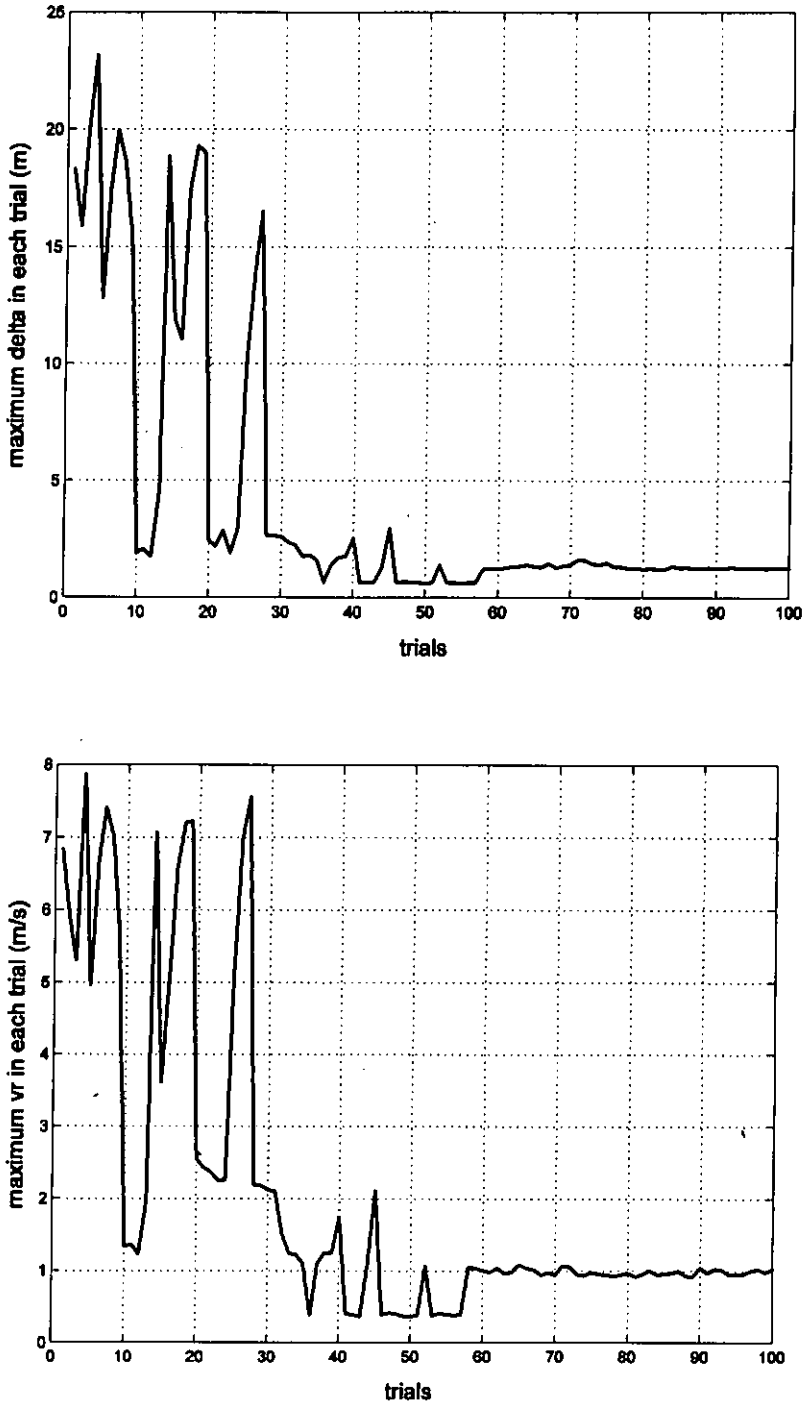


Figure 5.11: The evolution of the maximum spacing deviation (top) and relative speed (bottom) in each trial with the training trials

relative speed both decrease monotonously. This is the case we prefer. So, we may conclude that the learning efficiency of RLC is worse than that of DHPC.

The introduction of stochastic action modifier increases the uncertainty, but this is

necessary since the performance of RLC is very poor without the stochastic action modifier. The choice of the controller parameters such as learning rate and the magnitude of the stochastic action modifier affects the performance sensitively. We can see this from Figure 5.12 clearly. In this figure, we change the magnitude of the stochastic action not too much, but the evolution of the maximum spacing deviation and relative speed increase at 60^{th} trial suddenly. The learning procedure becomes unstable and the performance becomes unacceptable. Currently, it seems there is no systematic approach to select the parameters of SAM. This problem does not exist for DHPC since there is no need to add SAM.

After the above simulation comparisons and analysis between RLC and DHPC, we may conclude that:

1) The structures of RLC and DHPC are similar. They both comprise critic network and action network.

2) The requirements of controller design for RLC and DHPC are different. The only information for RLC design may be evaluative information r (scalar value) of the controller during its interaction with environment, while some information about system model may be required for DHPC design to get the derivative information.

3) The performance of RLC and DHPC are different. DHP should be more effective and contribute a more superior performance than HDP, the reason may exist in that the output of the critic network of DHPC is the derivative of the overall cost J directly instead of J, which makes the derivative of J

for RLC intrinsically less accurate. And also RLC introduces a stochastic action modifier, which makes the learning efficiency not good but is a must for RLC.

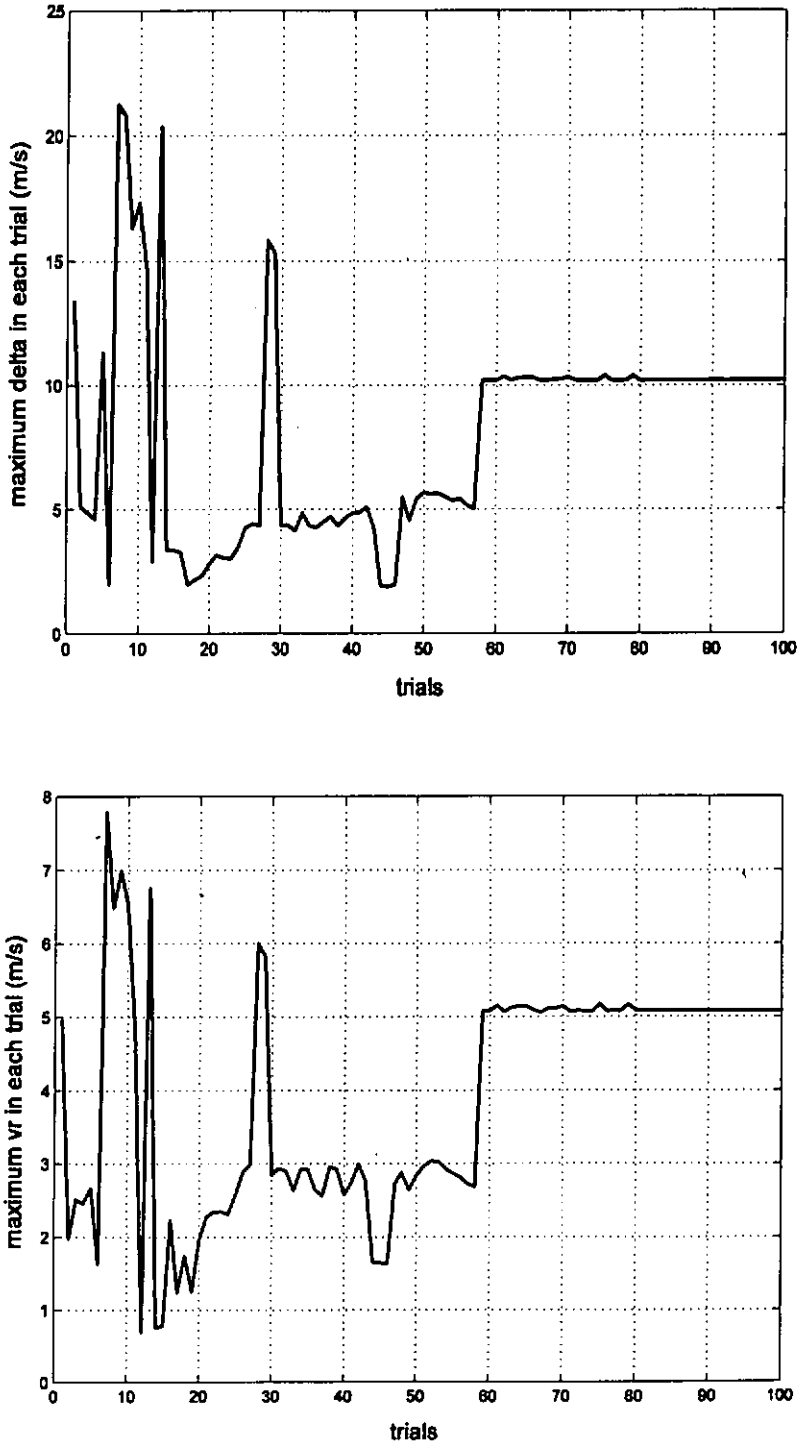


Figure: 5.12 The evolution of the maximum spacing deviation (top) and relative speed (bottom) with the change of some parameters

5.5 Conclusions

This chapter has presented a novel vehicle longitudinal controller based on dual heuristic programming. A brief overview of adaptive critic designs is introduced first, and we delve into the details of the specific technique by which we develop our proposed controller. We describe the architectures of critic network and action network, and propose their adaptation. The training procedures, simulation results are presented and the performance of the controller is discussed. There are two distinct differences between our proposed controller and other controllers: 1) the system model, which may be obtained in advance or by system identification, is not required for the proposed controller which is applied to a vehicle longitudinal control system, 2) we can incorporate a priori knowledge into the controller, since TS type fuzzy controller is employed as the action network instead of neural network. As a result, it is convenient for us to adopt the idea or structure of conventional controllers for action network design. This makes the controller more interpretable, training much faster and even more stable compared with the neural network controller. Finally, we compare the controller proposed in chapter 4 (RLC) and the controller in this chapter (DHPC) from the aspects of control structure, design requirements and performance, etc.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

In the last five chapters, the author presented his modest contributions to the area of autonomous intelligent control. The focus has been developing algorithms with more flexibility and less a priori knowledge for the problem of vehicle following control. To deal with the complexity of vehicle dynamics, we employed intelligent control technologies including fuzzy control, neural networks, reinforcement learning and adaptive critic design. To accommodate for a higher degree of uncertainty, we have integrated the idea of "adaptive control" into the vehicle longitudinal controller. This thesis describes three possible approaches to implementing advanced decision-making processes on an autonomous vehicle.

The first stage of the research involved the design of an adaptive fuzzy control algorithm. The vehicle longitudinal control problem falls into a class of specific continuous-time SISO nonlinear system with some unknown parameters. We have extended the studies by other researchers to tune all the parameters of fuzzy controller. As a result, a flexible and stable fuzzy controller is achieved.

However, the weakness of the proposed adaptive fuzzy controller is that some model information is required and it only aims at a specific nonlinear system. To this end, we have investigated Q-learning, a model free reinforcement learning (RL)

method, and its applicability as a controller design approach in a knowledge-poor environment. The focus has been on two issues: (i) the structure of the Q estimator network and the fuzzy controller, and (ii) the development of learning algorithms for both of them.

Nevertheless, the learning efficiency of applying RL directly may not be sufficient. Furthermore, we propose a controller based on dual heuristic programming (DHP) to enhance its performance. The structure and adaptation algorithms of the controller for vehicle following problems are also presented. The proposed controller has two advantages compared with other controllers based on adaptive critic designs: (i) the system model is not required directly or indirectly, and (ii) it takes advantage of the TS type fuzzy controller to incorporate a priori knowledge. We solved the problem of requirement for the vehicle model dexterously by some estimation. In the design procedure, we can expedite the learning process by the TS type fuzzy controller.

Finally, we presented comparisons between the controller based on RL directly and the controller based on DHP. It has been shown, by simulation studies, that the performance of the controller based on DHP has some advantages over the controller based on RL directly, although the latter is much simpler than the former.

6.2 Future Directions

Our research indicates the possibility of combining the intelligent control with adaptive control to deal with incomplete information and dynamic environment. However, there are other aspects that may be explored in further depth in the future.

One important task is the stability analysis of the controllers design based on RL or DHP. This objective is not trivial since the RL algorithms used for the control of continuous state space tasks are based either on heuristic or on discrete RL methods and DHP are based on RL algorithms. Moreover, one of the characteristics of RL (DHP) is that it optimizes the controller based on interactions with the system. Both the action and the critic networks have to be trained, where the approximation of the action network is trained based on the critic network. This makes the training procedure rather tedious and its outcome hard to analyze. Moreover, some parameters of the controller are selected based on a heuristic approach for the time being. We could not give the exact meaning and effect of these parameters to the controller performance.

The aforesaid means the performance of the controller is not very well understood when RL (DHP) is applied to a control system. Stability during learning may not be guaranteed. Therefore, solid theoretical studies to prove convergence to the optimal controller should be carried out in the future.

The applications of several intelligent adaptive controllers for autonomous vehicle control systems have been described in this thesis. However, current studies have been studied only via simulations. Therefore, another future direction of this research is to improve the algorithms for practical implementation on a real vehicle. This may include:

1) A reliable controller is needed for real control tasks. If we cannot guarantee the stability of the controller, it may be useless in the practical situations. This

also explains the hesitation in applying the algorithms based on RL directly on real systems in the existing literatures.

2) We should speed up the training procedure. One disadvantage of algorithms based on RL (DHP) is their inherent time-consuming processes due to the nature of RL (DHP), which requires large amount of interactions with the environment. One way to speed this up may be to use algorithms that require less training.

As we mentioned before, automated vehicle control includes longitudinal control and lateral control. In this thesis, we have mainly focused on vehicle longitudinal control, although we have studied vehicle lateral control in lesser detail. In the next stage, we may investigate the control algorithms for vehicle lateral control. Compared with longitudinal control, lateral control may contain more uncertainty, such as the parameters of vehicle dynamics, road conditions (road curvature, road adhesion), and even nonlinear tire characteristics or wind gusts. Moreover, existing literature also suggest that the control is especially difficult when the car speed is high.

This brings to end the author's work at this stage. The autonomous vehicle control is a fast emerging area of research and development. The work reported in this volume is a contribution towards the design of a completely autonomous vehicle to address the problems of highway congestion and alleviating traffic accidents.

Bibliography

Anderson, C., Hittle, D., Katz, A., and Kretchmar, R. "Synthesis of Reinforcement Learning, Neural Networks, and PI Control Applied to a Simulated Heating Coil". *Journal of Artificial Intelligence in Engineering,* Vol.11, pp. 423-431 (1996)

Astrom, K.J. and Wittenmark, B. *Adaptive control (Second Edition).* Addison-Wesley (1995)

Baird, L.C. "Reinforcement learning in continuous time: advantage updating". *1994 IEEE International Conference on Neural Networks*, Vol.4, pp.2448-2453 (1994)

Bellman, R. E. *Dynamic Programming.* Princeton University Press (1957)

Bellman, R. *Methods of Nonlinear Analysis: Volume II,* Academic Press (1973)

Bender, J.G. "An overview of system studies of automated highway systems". *IEEE Trans. Veh. Technol.*, Vol. 40, pp.82-99 (1991)

Bertsekas, D.P. and Tsitsiklis, J.N. *Neuro-dynamic Programming*, Athena Scientific, Belmont, MA (1996)

Boyan, J.A. and Moore, A.W. "Generalization in Reinforcement Learning: Safely Approximating the Value Function". In Tesauro, G, Touretzky, D.S., and Leen, T.K., eds., *Advances in Neural Information Processing Systems 7*, MIT Press, Cambidge MA (1995)

Cai, Z.X. *Intelligent Control: Principles, Techniques and Applications.* World Scientific, Singapore (1997)

Chen, B.S., Lee, C.H., and Chang, Y.C. "H-inf tracking design of uncertainty nonlinear SISO systems: Adaptive fuzzy approach". *IEEE Trans. Fuzzy Systems*, Vol.4, pp.32-43 (1996)

Chiang, C.K., Chung, H.Y., and Lin, J.J. "A self-learning fuzzy logic controller using genetic algorithms with reinforcements". *IEEE Tran. On Fuzzy Systems*, Vol.5, pp.460-467 (1997)

Crites, R. H. and Barto, A. G. "Improving elevator performance using reinforcement learning". In Touretzky, D.S., Mozer, M.C., and Hasselmo M.E., eds., *Advances in Neural Information Processing Systems 8*, MIT Press (1996)

Driankov, D. and Palm, R. *Advances in fuzzy control.* Physica-Verlag, New York (1998)

Eaton, P., Prokhorov, D., and Wunsch, D.C. "NeuroController Alternatives for 'Fuzzy' Ball-and Beam Systems with Nonuniform Nonlinear Friction". *IEEE Trans. Neural Networks*, Vol.11, pp.423-435 (2000)

Feng, G. "An Approach to Adaptive Control of Fuzzy Dynamic Systems". *IEEE Transactions on Fuzzy Systems*, Vol.10, pp.268-275 (2002)

Guldner, J., Tan, H., and Patwardhan, S. "On fundamental issues of vehicle steering control for highway automation". *California PATH Working Paper, UCB-ITS-PWP-97-11*, University of California, Berkeley (1997)

Gullapalli, V. "A stochastic reinforcement learning algorithm for learning real-valued functions". *Neural Networks*, Vol.3, pp.671-692 (1990)

Han, H., Su, C., and Stepanenko, Y. "Adaptive control of a class of nonlinear systems with nonlinearly parameterized fuzzy approximators". *IEEE Transactions on Fuzzy Systems*, Vol.9, pp.315-323 (2001)

Hangos, K.M., Lakner, R., and Gerzson, M. *Intelligent Control Systems: An Introduction with Examples*. Kluwer Academic Publishers, Netherlands (2001)

Hedrick, J.K., McMahon, D.H., and Swaroop, D. "Vehicle Modeling and Control for Automated Highway Systems". *Technical Report of University of California, Berkeley, UCB-ITS-PRR-93-24*, University of California, Berkeley (1993)

Hedrick, J.K., Tomizuka, M., and Varaiya, P. "Control issues in automated highway systems". *IEEE Control Systems Magazine*, Vol.14, pp.21-32 (1994)

Hedrick, J.K. "Nonlinear controller design for automated vehicle applications". *UKACC International Conference on Control '98*, Vol.1, pp.23-32 (1998)

Homaifar, A. and McCormick, E. "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms". *IEEE Trans. on Fuzzy Systems*, Vol.3, pp.129-139 (1995)

Horowitz, R. and Varaiya, P. "Control design of an automated highway system". *Proceedings of the IEEE*, Vol.88, pp.913 -925 (2000)

Howard, R. *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA (1960)

Huang, S. and Ren, W. "Use of Neural Fuzzy Networks with Mixed Genetic/Gradient Algorithm in Automated Vehicle Control". *IEEE Transactions on Industrial Electronics*, Vol.46, pp.1090-1102 (1999)

Ioannou, P. and Chien, C.C. "Autonomous intelligent cruise control". *IEEE Tran. Veh. Technol.*, Vol.42, pp.657-672 (1993)

Ioannou, P., Xu, Z., Eckert, S., Clemons, D., and Sieja, T. "Intelligent cruise control: theory and experiment". *Proceedings of the 32nd Conference on Decision and Control*, pp.1885-1890 (1993)

Ioannou, P. and Bose, A. "Automated Vehicle Control". In Hall, R.H. ed., *Handbook of transportation science*, Kluwer Academic, pp.186-232 (1999)

Jamshidi, M. "Fuzzy Control of Complex Systems". *Soft Computing*, Vol.1, pp.24-56 (1997)

Jouffe, L. "Fuzzy inference system learning by reinforcement methods". *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, Vol.28 , pp.338-355 (1998)

Kehtarnavaz, N., Nakamura, E., Griswold, N., and Yen, J. "Autonomous Vehicle Following by a Fuzzy Logic Controller". *North American Fuzzy Information Processing Society Biannual Conference*, pp.333 -337 (1994)

Kim, H.M., Dickerson, J., and Kosko, B. "Fuzzy throttle and brake control for platoons of smart cars". *Fuzzy Sets Syst.*, Vol.84, pp.209-234 (1996)

Kim, E., Park, M., Ji, S., and Park, M. "A New Approach to Fuzzy Modeling". *IEEE Transactions on Fuzzy Systems*, Vol.5, pp.328-337 (1997)

Kirk, D. E. *Optimal Control Theory: an Introduction*, Prentice-Hall, Englewood Cliffs, NJ (1970)

Koike, Y. and Doya, K. "Multiple state estimation reinforcement learning for driving model: driver model of automobile". *1999 IEEE International Conference on Systems, Man, and Cybernetics*, Vol.5, pp.504-509 (1999)

Lendaris, G. and Shannon, T. "Qualitative Models for Adaptive Critic Neurocontrol". *Proceedings of IEEE SMC'99 Conference*, Tokyo (1999)

Lendaris, G., Shannon, T., and Rustan, A. "A Comparison of Training Algorithms for DHP Adaptive Critic Neuro-Control". *Proceedings IJCNN'99*, Washington D.C. (1999)

Liu, B. and Si, J. "The best approximation to C^2 function and its error bounds using regular-center Gaussian networks". *IEEE Trans. Neural Networks*, Vol.5, pp.848-847 (1994)

Lin, C.T. and Lee, C.S.G. "Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems". *IEEE Transactions on Fuzzy Systems*, Vol.2 , pp.46-63 (1994)

Lin, C.J. and Lin, C.T. "Reinforcement learning for an ART-based fuzzy adaptive

learning control network". *IEEE Transactions on Neural Networks*, Vol.7, pp.709 -731 (1996)

Lin, C.T. and Chung, I.F. "A reinforcement neuro-fuzzy combiner for multiobjective control". *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol.29, pp.726-744 (1999)

Liu, D. "Adaptive Critic Designs for Problems with Known Analytical Form of Cost Function". *Proc. INNS-IEEE International Joint Conference on Neural Networks 2002*, Honolulu, HI, pp.1808-1813 (2002)

Mar, J. and Lin, F. "An ANFIS Controller for the Car-Following Collision Prevention System". *IEEE Transactions on Vehicular Technology*, Vol.50, pp.1106-1113 (2001)

Oh, S., Lee, J., and Choi, D. "A new reinforcement learning vehicle control architecture for vision-based road following". *IEEE Transactions on Vehicular Technology*, Vol.49, pp.997-1005, (2000)

Passino, K.M. and Yurkovich, S. *Fuzzy Control*. Addison-Wesley, Menlo Park (1998)

Passino, K.M. "Intelligent Control: An Overview of Techniques". In Samad, T. ed., *Perspectives in Control: New Concepts and Applications*, IEEE Press, NJ (2001)

Prokhorov, D., Santiago, R., and Wunsch, D.C. "Adaptive critic designs: A case study fro neurocontrol". *Neural Networks*, Vol.8, pp.1367-1372 (1995)

Prokhorov, D. and Wunsch, D.C. "Adaptive critic designs". *IEEE Trans. Neural Networks*, Vol.8, pp.977-1007 (1997)

Prokhorov, D. "Adaptive Critic Designs and their application". *Ph.D dissertation*, Department of Electrical Engineering, Texas Tech University (1997)

Raza, H. and Ioannou, P. "Vehicle Following Control Design for Automated Highway Systems". *IEEE Control Systems Magazine*, Vol.16, pp.43-60 (1996)

Samuel, A. L. "Some studies in machine learning using the game of checkers". *IBM Journal on Research and Development*, No.3 (1959)

Schultz, L.J., Shannon, T.T., and Lendaris, G.G. "Using DHP Adaptive Critic Methods to Tune a Fuzzy Automobile Steering Controller". *Proceedings of IFSA/NAFIPS Conference*, Vancouver, B.C. (2001)

Shannon, T. "Partial, Noisy and Qualitative Models for Adaptive Critic Neurocontrol". *Proceedings of IJCNN'99*, Washinton, D.C. (1999)

Shannon, T. and Lendaris, G. "Adaptive Critic Based Approximate Dynamic Programming for Tuning Fuzzy Controllers". *The Ninth IEEE International*

Conference on Fuzzy Systems, 2000, Vol.1, pp.25-29 (2000)

Sheikholeslam, S. and Desoer, C. A. "Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study". *IEEE Tran. Veh. Technol.*, Vol.42, pp.546-554 (1993)

Shladover, S.E. "Review of the State of Development of Advanced Vehicle Control System(AVCS)". *Vehicle System Dynamics*, Vol.24, pp.551-595 (1995)

Si, J. and Wang, Y. "Online learning control by association and reinforcement". *IEEE Transactions on Neural Networks*, Vol.12, pp.264-276 (2001)

Singh, S. and Bertsekas, D. "Reinforcement learning for dynamic channel allocation in cellular telephone systems". Touretzky, D.S., Mozer, M.C., and Hasselmo M.E., eds., *Advances in Neural Information Processing Systems 8*, MIT Press (1996)

Smart, W.D. and Kaelbling, L.P. "Practical Reinforcement Learning in Continuous Spaces". *Proceedings of the Seventeenth International Conference on Machine Learning* (2000)

Spooner, J.T. and Passino, K.M. "Stable adaptive control using ufzzy systems and neural networks". *IEEE Transactions on Fuzzy Systems*, Vol.4, pp.339-359 (1996)

Su, C. and Stepanenko, Y. "Adaptive control of a class of nonlinear systems with fuzzy logic". *IEEE Transactions on Fuzzy Systems*, Vol.2, pp.285-294 (1994)

Sutton, R.S., Barto, A.G, and Williams, R.J. "Reinforcement learning is direct adaptive optimal control". *IEEE Control Systems Magazine*, Vol.12, pp.19-22 (1992)

Sutton, R.S. and Barto, A.G *Reinforcement Learning: an Introduction*. MIT Press (1998)

Swaroop, D., Hedrick, J.K., and Choi, S.B. "Direct adaptive longitudinal control of vehicle platoons". *IEEE Tran. Veh. Technol.*, Vol.50, pp.150-161 (2001)

Takagi, M. and Sugeno, M. "Fuzzy identification of systems and its application to modeling and control". *IEEE Trans. Systems, Man, Cybernetics*, Vol.SMC-15, pp.116-132 (1985)

Tesauro, G "TD-Gammon, a self teaching backgammon program, achieves master-level play". *Neural Computation*, Vol.6, pp.215-219 (1994)

Thorndike, E. L. *Animal Intelligence*. Hafner (1911)

Tsay, D., Chung, H., and Lee, C. "The adaptive control of nonlinear systems using the Sugeno-type of fuzzy logic". *IEEE Transactions on Fuzzy Systems*, Vol.7, pp.225-229 (1999)

Tzafestas, S. G. *Advances in Intelligent Autonomous Systems*. Kluwer Academic Publishers (1999).

Varaiya, P. "Smart cars on smart roads: problem of control". *IEEE Tran. Veh. Technol.*, Vol.38, pp.195-207 (1993)

Venayangamoorthy, G.K., Harley, R.G, and Wunsch, D.C. "Comparison of Heuristic Dynamic Programming and Dual Heuristic Programming Adaptive Critics for Neurocontrol of a Turbogenerator". *IEEE Trans. Neural Networks*, Vol.13, pp.764-773 (2002)

Vlacic, L., Parent, M., and Harashima, F. *Intelligent vehicle technologies: theory and applications*. Butterworth-Heinemann, Boston (2001)

Wang, L. and Mendel, J.M. "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning". *IEEE Trans. Neural Networks*, Vol.3, pp.807 -814 (1992)

Wang, L. "Stable adaptive fuzzy control of nonlinear systems". *IEEE Trans. Fuzzy Systems*, Vol.1, pp.146-155 (1993)

Wang, L. *A course in Fuzzy System and Control*. Prentice-Hall, Inc., NJ (1997)

Watkins, C.J.C.H. "Learning with Delayed Rewards," *Ph.D. thesis*, Cambridge Univ., Psychology Dept. (1989)

Werbos, P.J. "A menu of designs for reinforcement learning over time". In Miller, W.T., Sutton, R., and Werbos, P. , eds., *Neural Network for Control*, MIT Press, Cambridge, pp.67-95 (1990)

Werbos, P. J. "Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence". *General Systems Yearbook* (1997)

Yan, X.W., Deng, Z.D., and Sun, Z.Q. "Competitive Takagi-Sugeno fuzzy reinforcement learning". *Proceedings of the 2001 IEEE International Conference on Control Applications*, pp.878 -883 (2001)

Yanakiev, D. and Kanellakopoulos, I. "Speed Tracking and Vehicle Follower Control Design for Heavy-Duty Vehicles". *Vehicles System Dynamics*, Vol.25, pp.251-276 (1996)

Yanakiev, D. and Kanellakopoulos, I. "Longitudinal control of automated CHVs with significant actuator delays". *IEEE Tran. Veh. Technol.*, Vol.50, pp.1289-1297 (2001)

Zalama, E., Gomez, J., Paul, M., and Peran, J.R. "Adaptive behavior navigation of a mobile robot". *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol.32, pp.160-169 (2002)