

**Neural and Learning Approaches for Automotive Engine Control
and Residential Energy System Optimization**

BY

TING HUANG

B.S., Xiamen University, China, 2000

M.S., Chinese Academy of Space Technology, China, 2003

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2011

Chicago, Illinois

Defense Committee:

Derong Liu, Chair and Advisor

Rashid Ansari

Dan Schonfeld

Milos Zefran

Houshang Darabi, Mechanical and Industry Engineering

To my family

ACKNOWLEDGMENTS

I would like to express my sincere and profound thanks to my adviser, Professor Derong Liu, for his patient guidance and constant encouragement during my years of study at the University of Illinois. It is his consistent support and rigorous supervision which help me make steady progress in the Ph.D. study. Working with Professor Liu has been a highly rewarding and extremely enjoyable experience.

Also, I would like to thank Dr. Hossein Javaherian, and all of the research staff at General Motors R&D and Planning, for their support and many useful discussions during the course of my research.

I am very grateful to Dr. Rashid Ansari, Dr. Dan Schonfeld, Dr. Milos Zafran, and Dr. Houshang Darabi for serving on my Ph.D. dissertation committee and for their valuable time, support, comments and suggestions. I would like to thank my friends and colleagues at UIC with whom I have had the pleasure of working over these years.

Finally, I would like to thank my family, for their support at all times.

TH

TABLE OF CONTENTS

<u>CHAPTER</u>	<u>PAGE</u>
1 INTRODUCTION	1
1.1 Artificial Neural Networks	2
1.2 Adaptive Dynamic Programming	5
1.3 Engine and Emission Control	7
1.4 Residential Energy System	13
2 NEURAL NETWORK MODELLING FOR THE TEST ENGINE	19
2.1 Engine Description	21
2.2 Identification of Engine Torque using Feedforward Neural Network	21
2.3 Identification of Engine Air-Fuel Ratio using Nonlinear Output Error Model	26
3 ADAPTIVE CRITIC LEARNING TECHNIQUES FOR ENGINE TORQUE AND AIR-FUEL RATIO CONTROL	33
3.1 Introduction	33
3.2 Neural Network Modeling of the Test Engine	35
3.3 Adaptive Dynamic Programming	35
3.3.1 Dynamic Programming	36
3.3.2 Action-Dependent Heuristic Dynamic Programming	37
3.3.3 Tracking Control Problem	44
3.4 Simulation Studies	44
3.4.1 Critic Network	45
3.4.2 Action Network	47
3.4.3 Simulation Results	47
3.5 Conclusions	50
4 NEURAL SLIDING MODE TECHNIQUE FOR ENGINE TORQUE AND AIR-FUEL RATIO CONTROL.....	54
4.1 Introduction	54
4.2 Sliding Mode Control Design	56
4.3 Neuro-Sliding Mode Control	60
4.3.1 Neural Computation of the Equivalent Control	60
4.3.2 Neural Computation of the Corrective Control	62
4.4 Neuro-Sliding Mode Technique for Engine Control	64

TABLE OF CONTENTS (continued)

<u>CHAPTER</u>	<u>PAGE</u>
4.5 Simulation Results	66
4.6 Conclusions	76
5 BARORECEPTOR REFLEX INSPIRED ADAPTIVE CONTROL STRATEGY FOR EN- GINE TORQUE CONTROL	77
5.1 Introduction	77
5.2 Adaptive Nonlinear Control Strategy	80
5.2.1 Linear Controller Design	82
5.2.2 Nonlinear Controller Design	83
5.3 Adaptive Control System for Engine Torque Control	84
5.4 Simulation Results	87
5.5 Conclusions	94
6 A SELF-LEARNING SCHEME FOR RESIDENTIAL ENERGY SYSTEM CONTROL AND MANAGEMENT	96
6.1 Introduction	96
6.2 Description of the Residential Energy System	99
6.3 Self-Learning Scheme for the Residential Energy System	107
6.4 Simulation Results	112
6.5 Conclusions	119
7 CONCLUDING REMARKS	121
CITED LITERATURE	125
VITA	136

LIST OF FIGURES

<u>FIGURE</u>		<u>PAGE</u>
1.1	The simple functional decomposition of the system	9
1.2	Efficiency of a typical tree-way catalyst converter	11
2.3	Structure of the engine torque model	22
2.4	Illustration of inputs for the engine TRQ modelling	23
2.5	Illustration of engine TRQ validation data	24
2.6	Illustration of relative deviation of the model from TRQ*	25
2.7	The model structure of NNOE	27
2.8	The model structure of recurrent neural network	28
2.9	Illustration of inputs of NNOE for the engine AFR modelling	30
2.10	Illustration of engine AFR validation data	31
2.11	Illustration of relative deviation of the AFR model	32
3.12	A typical scheme of an action-dependent heuristic dynamic programming	38
3.13	Illustration for the backward-in-time method	40
3.14	Illustration for the forward-in-time method	41
3.15	Structure of the adaptive critic learning engine controller	46
3.16	Torque output generated by the neural network controller	48
3.17	Air-fuel ratio output generated by the neural network controller	49
3.18	Torque output generated by the refined neural network controller	51
3.19	Air-fuel ratio output generated by the refined neural network controller	52
4.20	Structure of neuro-sliding mode control	61
4.21	Structure of neural network #2 for the corrective control.	62

LIST OF FIGURES (continued)

<u>FIGURE</u>	<u>PAGE</u>
4.22	Structure of NSLMC for the TRQ control 64
4.23	Structure of NSLMC for the engine AFR control 65
4.24	Structure of neural network #2 66
4.25	Control effect of NSLMC for TRQ control. 67
4.26	Output of equivalent control, corrective control and TPS in the data. 68
4.27	Trajectory of sliding function S 69
4.28	NSLMC generalization ability for TRQ control. 70
4.29	Control effect of NSLMC for AFR control. 71
4.30	Output of equivalent control, corrective control and FPW in the data. 72
4.31	Trajectory of sliding function S 73
4.32	NSLMC generalization ability for AFR control. 74
5.33	Parasympathetic and sympathetic manipulation of heart rate 79
5.34	Biologically inspired adaptive control system 81
5.35	The adaptive nonlinear control system 85
5.36	ANCS control for TRQ 88
5.37	The output of PID control, NN control, and TPS in the data 89
5.38	The combined output of PID control and NN control and TPS in the data 90
5.39	ANCS generalization control effect 91
5.40	The output of PID control, NN control, and TPS in the data 92
5.41	The combined output of PID control and NN control and TPS in the data 93

LIST OF FIGURES (continued)

<u>FIGURE</u>		<u>PAGE</u>
6.42	Grid-connected residential energy system with battery storage	100
6.43	A typical residential load profile	102
6.44	A typical daily real-time pricing	103
6.45	Block diagram of the single critic approach	108
6.46	A typical residential load profile in one week	114
6.47	Energy changes in batteries	115
6.48	Optimal scheduling of batteries in one week	117
6.49	Scheduling of batteries of fixed daily cycle scheme	118

LIST OF ABBREVIATIONS

ADHDP	Action Dependent Heuristic Dynamic Programming
ADP	Adaptive Dynamic Programming
AFR	Air-Fuel Ratio
ANCS	Adaptive Nonlinear Control Strategy
ANN	Artificial Neural Network
ART	Adaptive Resonance Theory
DHP	Dual Heuristic Programming
EGR	Exhaust gas recirculation
FPW	Electrical Fuel Pulsewidth
GDHP	Globalized Dual Heuristic Dynamic Programming
HDP	Heuristic Dynamic Programming
MAF	Mass Air Flowrate
MAP	Intake Manifold Pressure
MVEM	Mean Value Engine Models
NN	Neural Network
NNOE	Nonlinear Output Error
NSLMC	Neuro-Sliding Mode Control
PID	Proportional Integral Derivative
RPM	Engine Speed
SLMC	Sliding Mode Control
SPA	Spark Advance

LIST OF ABBREVIATIONS (continued)

TPS	Throttle Position
TRQ	Engine Torque
TWC	Three-Way Catalyst

SUMMARY

Due to the tightened legislation on emissions, fuel economy, and diagnostic standards in spark ignition internal combustion engines from governments, the automotive industry is striving to minimize the emission and, at the same time, to achieve better fuel economy and vehicle driveability. In this dissertation, using neural networks, we develop three advanced learning algorithms for the control of automotive internal combustion engine with the objectives of reduced emissions and improved performance.

Using the actual data from a test vehicle with a V8 engine, specific neural networks are trained offline to simulate engine torque (TRQ) dynamics and exhaust air-fuel ratio (AFR) dynamics for the purpose of the identification of controllers. We first build a multi-input multi-output neural network model for the engine TRQ and AFR processes with time-lagged recurrent neural networks. Reasonably good model for the TRQ and AFR processes is generated. This model is utilized for the engine control with adaptive critic learning technique. However, considering the coupling of the TRQ and AFR signals for modeling into a multi-input multi-output system is undesirable due to the different nature of the processes involved, two separate neural network models are employed to model the TRQ and AFR processes. A multilayer feedforward neural network is utilized for the identification of TRQ process due to the fact that the torque generation process appeared to be quasi-static and governed mostly by nonlinear properties. Considering the AFR process dominated by purely dynamic effects, we choose nonlinear output error model to model AFR dynamics. Validation results demonstrate that both models simulate the engine processes with a high degree of accuracy. Thus, these two models are later utilized for the engine control with neural sliding mode control technique and the biological adaptive nonlinear control strategy.

Next, we develop adaptive critic designs for self-learning control of automotive engines. A

SUMMARY (continued)

class of adaptive critic designs that can be classified as (model-free) action-dependent heuristic dynamic programming is used. The present work uses a system, called “critic,” to approximate the cost function in dynamic programming and thus to achieve optimal control. Our research results have demonstrated that adaptive dynamic programming provide a powerful alternative approach for engine calibration and control. A distinct feature of the proposed technique is that the controllers can learn to improve their performance during the actual vehicle operations, and will adapt to uncertain changes in the environment and vehicle conditions.

We then develop a new approach for the calibration and control of spark ignition engines using a combination of neural networks and sliding mode control technique. Two parallel neural networks are utilized to realize the equivalent control and the corrective control of the sliding mode control design. The calculation of equivalent control is realized by adaptively learning without knowing the plant dynamics. The proposed adaptation scheme directly results in a chatter-free control action for the corrective control. The distinct feature of the present technique is that the learning and control are carried out simultaneously, which allows the neural network controller to be further refined and improved in real-time vehicle operation through continuous learning and adaptation.

Inspired by the functions of baroreceptor reflex, an adaptive nonlinear control strategy is developed with applications to the control of the automotive engine. The biological adaptive nonlinear control strategy is realized by a parallel controller. The controller consists of a linear controller and a nonlinear controller that interact via a reciprocal lateral inhibitory mechanism. In the linear operating region, the linear controller takes control. If the controlled process is far away from the linear regime or is disturbed by noise, the output of linear controller may be inappropriate, and therefore the nonlinear controller is activated to compensate for the inadequacy of the linear controller in a dynamic environment. These situations can be addressed by adjusting the amount

SUMMARY (continued)

of lateral inhibition and learning the characteristics of the controlled system such that desirable controller outputs are produced in any particular operating region. The novelty of the biological adaptive nonlinear control strategy is that each controller modulates the other controller via the reciprocal lateral inhibitory connections.

Finally, we apply intelligent optimization method to the intelligent price-responsive management of residential energy use, with an emphasis on the use of home battery connected to the power grid. For this purpose, a self-learning scheme that can learn from the user demand and the environment is developed for the residential energy system control and management. The idea is built upon a self-learning architecture with only a single critic neural network instead of the action-critic dual network architecture of typical adaptive dynamic programming. The single critic design eliminates the iterative training loops between the action and the critic networks and greatly simplifies the training process. The advantage of the proposed control scheme is its ability to effectively improve the performance as it learns and gains more experience in real-time operations under uncertain changes of the environment. Therefore, the scheme has the adaptability to obtain the optimal control strategy for different users based on the demand and system configuration. Simulation results demonstrate that the proposed scheme can financially benefit the residential customers with the minimum electricity cost.

1 INTRODUCTION

The tightening regulations on emissions in spark ignition internal combustion engines from governments, combined with the demanding requirements for better fuel economy and high expectations from customers, present significant challenges and opportunities for the automotive industry. In the past few years, numerous researches and applications have been conducted in the development of efficient and accurate engine control and calibration techniques. Furthermore, the introduction of microprocessor based control permits the automotive manufacturers to design cleaner, more fuel efficient, better performing and more reliable engine systems.

There has been considerable discussion of the importance of distributed energy storage, including batteries in the home, as a way to create more price-responsive demand and as a way to integrate renewable energies more effectively into power grids. Many large and small-scale energy storage resources have been installed in power systems. Distributed storage resources are now penetrating rapidly towards commercialization as competitive alternatives to the large-scale centralized generations. The key behind this development is the optimal operational scheduling of storage resources.

The present dissertation presents results obtained in the past few years and discusses future research topics. Results obtained in the present work constitute improvements to existing results in the literature and new applications. In particular, we have developed adaptive critic learning technique, neural sliding mode control algorithm and baroreceptor reflex inspired nonlinear control strategy for the automotive engine control. We also developed a self-learning scheme for the control and management of residential energy system.

Some of the research results presented in this dissertation have been reported in three journal papers [56], [58], [85] and five conference papers [57], [59], [65], [66], [67].

In the rest of this chapter, we present some background materials regarding artificial neural networks, adaptive dynamic programming, automotive engine and emission control, and residential energy system.

1.1 Artificial Neural Networks

An artificial neural network (ANN) is a computational model that tries to emulate biological neural networks found in living organisms. It is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available to users [51]. The processing capability of ANN is stored in the inter-unit connection strength obtained by a process of adaptation to a set of training patterns for specific problems. In most cases, an ANN is an adaptive system that is utilized to model complex relationships between inputs and outputs or to find patterns in data.

The two distinctive information processing capabilities, that is, massively parallel distributed computing and the ability to learn and generalize, make it possible to solve complex and large-scale problems in the real world. An ANN is composed of a large number of simple processing neurons that cooperate with each other by numerous adjustable links to perform a parallel distributed processing. Other important capabilities are:

- **Input-Output Mapping:** through learning, ANNs are capable of modelling complex functions from complicated, corrupted or imprecise data with little or no a priori knowledge for the problem at hand.

- **Nonlinearity:** an ANN, made up of an interconnection of nonlinear neurons, is itself nonlinear.
- **Adaptivity:** ANNs have a built-in capability to adapt their synaptic strength to changes in the surrounding environment.
- **Fault Tolerance:** due to the distributed nature of information stored in the network, partial destruction of a network leads to the corresponding degradation of performance rather than catastrophic failure.

The important property of a neural network is the ability of the network to learn from the environment, and to improve through learning. In general, there are three approaches for learning: supervised learning, unsupervised learning and reinforcement learning. Supervised training involves a mechanism of learning in which the task of learning is mapping from an input space to a target space with desired output. Unsupervised learning is where the network has to figure out the meaning of the inputs without outside help. Reinforcement learning concerns with how an agent take actions in an environment so as to maximize a long-term reward. The selection of learning method largely depends on the type of neural network and the characteristics of the specific problem.

There are many different types of ANNs. Each type of neural network has been designed to tackle a certain class of problems. The abilities of different networks can be related to their structure, dynamics and learning methods. Some of the most popular examples are:

- Multilayer feedforward neural networks
- Radial-basis function networks
- Hopfield networks

- Recurrent neural networks
- ART networks
- Kohonen self-organizing maps.

ANNs have seen an explosion of interest over the last few decades, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, classification or control, neural networks are being introduced. A few representative examples of problems to which neural networks have been applied successfully are:

- Detection of medical phenomena
- Stock market prediction
- Service usage forecasting
- Monitoring the condition of machinery
- Short and long-term load estimation
- Agricultural production estimations.

With the integration with conventional computing technologies, artificial intelligence, and related subjects, ANNs will be more powerful and play more important roles in people's lives in the future.

1.2 Adaptive Dynamic Programming

There have been growing interests in adaptive dynamic programming (ADP) in the past three decades [13], [16], [20], [31], [33], [85], [86], [87], [94], [108], [119], [123], [134], [137]. In the literature, ADP is also referred to by several different names including “adaptive critic designs,” “reinforcement learning,” “neuro-dynamic programming,” “approximate dynamic programming,” and “asymptotic dynamic programming.” In the present work, we will use the term “adaptive dynamic programming” as in [94] to refer to a neural network-based scheme that approximates dynamic programming solutions in the general case, i.e., approximates optimal control over time in nonlinear environments. There are many problems in practice which can be formulated as cost maximization or minimization problems. Examples include error minimization, energy minimization, profit maximization, and the like. Dynamic programming is a very useful tool in solving these problems. However, it is often computationally intractable to run dynamic programming due to the backward numerical process required for its solutions, i.e., due to the “curse of dimensionality” [18], [36]. Over the years, progress has been made to circumvent the “curse of dimensionality” by building a system, called “critic,” to approximate the cost function in dynamic programming [13], [108], [109], [119], [134]-[137]. The idea is to approximate dynamic programming solutions by using a function approximation structure such as neural networks to approximate the cost function.

Generally speaking, there are three design families of ADP: heuristic dynamic programming (HDP), dual heuristic programming (DHP), and globalized dual heuristic dynamic programming (GDHP) [108], [109], [134], [136]. A typical ADP scheme consists of three modules (critic, model, and action) that can be implemented by using neural networks. These three modules provide functions of decision, prediction, and evaluation, respectively. When in ADP the critic network (i.e., the

evaluation module) takes the action/control signal as part of its inputs, the designs are referred to as action dependent ADP (ADADP). The difference among three designs is the output of the critic module. HDP adapts critic whose output is the approximation of the cost function. DHP adapts critic the output of which is the approximation of the first-order derivatives of the cost function. GDHP which combines both HDP and DHP outputs the approximation of the cost function and the approximation of the first-order derivatives of the cost function. The distinguished feature of ADP approach is that the cost function and the control law are approximated via parametric structures, whose parameters are adapted and improved over time by solving the recurrent relationship of dynamic programming.

Researchers have shown the close relationship between ADP and reinforcement learning [16], [123], [134], [136]. For example, the well-known temporal difference method in reinforcement learning is closely related to HDP, and the famous Q-learning method is closely related to AD-HDP [123]. One of the major differences is the way in which the cost function is represented. In ADP, the cost function is approximated using networks which are built up from differentiable functions such as neural networks; while in reinforcement learning methods, the cost/value function is usually stored in look-up tables (which implies a finite or discrete set of states) [124].

An ADP consists of a parametric structure, the actor, which approximates the control law, and another parametric structure, the critic, which approximates the cost function that tries to capture the effect that the control law will have on the future cost. The ADP method determines the optimal control law for a system by successively adapting the actor and the critic. The adaptation process starts with a non-optimal, arbitrarily chosen control by the actor. The critic then provide guidance for the actor on how to improve the control law toward the optimal solution at each successive adaptation. In return, the actor is used to update the critic toward the cost function. An algorithm that successively iterates between these two operations converges to the optimal solution over

time. The essence of the ADP methodology is to iteratively build a link between present actions and future consequences via an estimate of the cost function and/or its derivatives.

1.3 Engine and Emission Control

Internal combustion engine is the most frequently used power source for automobiles. The internal combustion engine is a heat engine that converts the chemical energy from the fuel into heat energy, and then into mechanical energy for the power transmitted to the output shaft. Fuel is burned with the air inside the engine in a process of the combustion, which greatly raises the temperature and the pressure of the gases inside the engine. The result is the huge expansion within the engine that rotates the output crankshaft. The output crankshaft is connected to the transmission to propel the automotive. In general, the automotive internal combustion engine is a reciprocating engine having pistons that reciprocate back and forth in cylinders internally within the engine. Most of automotive reciprocating engines are four-stroke engines. Four-stroke engines are characterized by two alternate cycles. In the first cycle, including the intake stroke and compression stroke, the cylinder is filled with fresh air from the intake manifold and the air is compressed. In the second cycle, including power stroke and exhaust stroke, the gas is combusted and expanded, and the emissions are transferred to the exhaust pipe.

Automotive internal combustion engine control is considered as one of the most difficult and complex control problems. The difficulties of the engine control lie in the following aspects. Firstly, the processes in the engine to be controlled are usually characterized as highly nonlinear multiple input multiple output systems with complicated interaction among variables. Secondly, the production variation and ageing of the engine may cause a degradation of engine performance and an increase in emissions, which is due to the poor adaption capability of the conventional

engine control system. Thirdly, there are many different operational zones in the engine and the optimal control strategy at one such zone might perform completely differently in the other zone. Furthermore, special considerations are needed for different operating modes, including startup, idle, running, and braking, with the separate techniques. Conventional methods divide the operating regime into many zones with linear models constructed for each zone, and, accordingly, utilize separate linear controllers for each region. However, it is not practically feasible in the real implementation.

To alleviate these difficulties, the design of universal intelligent controllers with adaptability needed to be considered. Such controllers need to be general enough that it would cover the whole operational range of the controlled system and, at the same time, possess the ability to operate over highly nonlinear dynamics. Moreover, the controller should have learning ability to improve the performance during the actual vehicle operations, and will adapt to uncertain changes in the environment and vehicle conditions. Another important feature of the controller is the self-calibration of the engine without the need of service station. The neural-network-based learning control design approaches together with neural network modelling of the engine processes provide solutions for overcoming the problems associated with traditional control methods. These approaches enable us to design engine control systems with minimal a priori knowledge of the processes involved and allow realtime adaptation of engine controllers due to uncertain changes in the environmental and vehicle conditions. Once fully developed, those techniques, may offer promise for use as real-time calibration and control tools.

Emissions generated by automobiles are one of the major sources of air pollution in the United States. Ever since the introduction of emission control regulations in the 1960's, great efforts had been made to reduce the toxicity of exhaust leaving the engine. Theoretically, emissions can be controlled to a minimum possible level by controlling the engine combustion process so that the

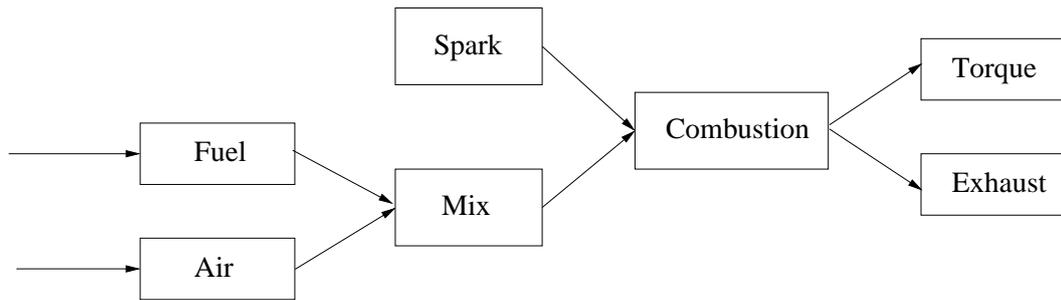


Figure 1.1: The simple functional decomposition of the system

air and fuel are mixed at certain desired ratio.

Figure 1.1 shows the simple functional decomposition of the combustion module with basic processes [74]. In conventional port fuel injection gasoline engine, fuel is injected into the intake port of each cylinder by adjusting the fuel injector plusewidth. The plusewidth is the length of the time the fuel injectors are open to supply fuel to the engine, which roughly determine the amount of the fuel being injected into the intake manifold. Air intake is controlled to combined with the fuel to yield the desirable mix. Then, the spark plugs will generate the sparks to ignite the air/fuel mixture for the start of the combustion process in cylinders. The timing of spark ignition and the ratio of air/fuel mixture determine the torque and emissions generated by the combustion process. Finally, the torque is delivered to the powertrain and the emissions to the exhaust systems. In general, air to fuel ratio (AFR) ranges from 12:1 to 18:1 in spark ignited engines, which generates different emission levels with hydrocarbons (HC) and carbon monoxide (CO) highest at rich conditions and nitrogen oxides (NO) emissions peaking near 16:1 [21].

A three-way catalyst (TWC) is widely used to simultaneously convert emissions of CO, HC and NO by both oxidizing unburned HC and CO and reducing NO contained in the precatalyst engine exhaust. With the purpose of effective operation of the TWC system, the engine must work

between excess oxygen (lean) and excess fuel (rich) cycles [97]. To prevent lean tailpipe emissions, the excess oxygen in the pre-catalyst exhaust gas is stored in the catalyst through chemisorption with the cerium oxides contained in the catalyst. To prevent rich tailpipe emissions, oxygen is released from the catalyst by oxidizing the hydrocarbons and carbon monoxide.

In theory, high simultaneous conversion efficiencies for HC, CO and NO can be obtained only in a narrow range around stoichiometric value of AFR, which is equal to approximately 14.67. Therefore, it is critical to control the AFR close to stoichiometry to minimize exhaust emissions. Figure 1.2 depicts the typical characteristics of a three-way catalytic converter [138]. As can be seen in Figure 1.2, it is desirable to maintain the air to fuel mix at 14.67 avoiding lean or rich values since both exhaust emissions and engine torque are affected by actuators simultaneously and the control of the engine torque also becomes an issue. The conversion efficiency of the catalyst is very sensitive to the variation of the ratio as Figure 1.2 shows, and even 1% deviation from the stoichiometric ratio results in up to 50% degradation in the conversion of one or more pollutants.

Generally, the goal of control strategy are given in terms of torque and emissions. Based on the data collected from several sensors measuring emissions and car dynamics under various driving conditions, embedded controllers compute the control actions to apply [14]. Given the ever increasing computational power of microcontrollers, it is possible to find and design control algorithms with guaranteed properties that can reduce substantially emissions and gas consumption while maintaining the performance of the vehicle.

Due to the complex nonlinear dynamical characteristic of modern automotive engines, the ARR and torque (TRQ) control problem, as they are known, turn out to be very difficult to solve. These control problems have been investigated for many years by many researchers (see, e.g., [2], [85], [93], [105], [116], [138], and the references cited therein). Almost every branch of the modern and

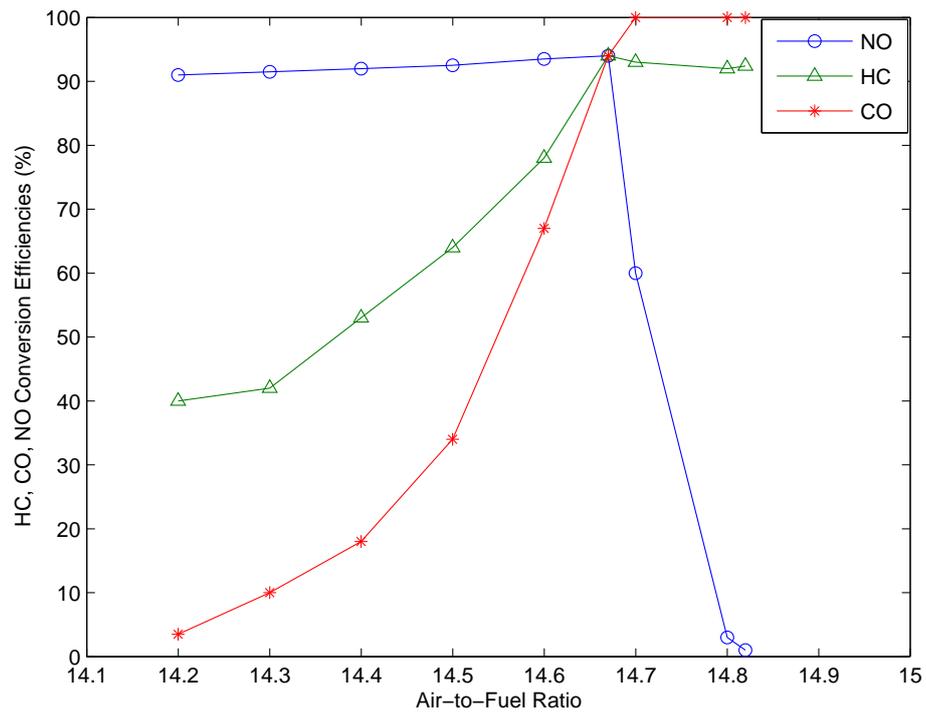


Figure 1.2: Efficiency of a typical tree-way catalyst converter

classical control theory has been researched for the control of automotive engines. In terms of control approaches, the use of nonlinear feedforward controllers [49], adaptive controllers [12], [68], [114], [115], [128], feedback linearization [48], Smith predictors [99], observer based controllers [27], [107], [29], sliding mode controllers [106], [122], [138], linear quadratic regulators [102], [103], H_∞ controllers [96], [130], and model predictive controllers [98] can be mentioned.

Classic feedback controllers utilize two controllers for an accurate control of AFR [7], [52]. The first closed-loop controller which relies on information coming from the exhaust gases oxygen sensor accounts for the steady state of AFR, while the second controller addresses the transient performance by considering an open loop controller based on transient information coming from the engine. Many of the current production fuel injection controllers utilize feedforward control based on a mass air flow sensor plus a proportional integral type feedback control to control AFR. This method cannot produce desirably accurate control performance in highly nonlinear and dynamic engines. In the early nineties, much of the research has centered around model-based AFR controller [6], [103]. A fine control of AFR requires an accurate mathematical description of an accurate model of various processes of the whole engine. However, for highly nonlinear and complex engine system, the precise mathematical models of the underlying processes are not available or not fully developed. Often, only input-output measurements are available. Furthermore, model-based controllers with no adaptive capability have significantly limited their implementation in a production line scenario. Therefore, a technique with self-adaptive and learning ability for an individual engine is clearly preferable. Neural networks with good generalization and inherent adaptivity become attractive techniques to be investigated for an optimal control of AFR. In [2] and [9], a direct inverse controller based on recurrent neural network was built. The recurrent neural network controller compensates for the wall wetting dynamics and estimates the right amount of fuel to be injected to meet the target AFR during engine transients. In [105], Park proposed feed-

back error learning neural networks, which was inspired by the way the central nervous system of the brain controls and learns voluntary movement, for AFR control. The promising attribute of the approach is the adaptation ability, which reduces the tedious calibration efforts. In [95], Manzie developed a control scheme based on radial basis function neural network for the fuel injection problems. The proposed approach requires no a priori knowledge of the engine systems, and on-line learning is achieved using gradient descent updates. Recently, adaptive critic designs, which approximate optimal control over time in noisy nonlinear environment, has been implemented for self-learning control of automotive engines [85], which is considered in the Chapter 3. The distinct feature of the technique is the controller's real-time adaptation capability based on real vehicle data which allows the neural network controller to be further refined and improved in real-time vehicle operation through continuous learning.

1.4 Residential Energy System

There has been wide attentions and increasing interest in various renewable energy resources due to the growing environmental concerns over the use of fossil fuels for electric power generation. Of the many alternatives, wind and solar photovoltaic (PV) are considered to be more promising for continually increasing energy demand than others. They are abundant, inexhaustible, and the sources for clean energy usage. However, they are not as practical as one would hope. Windmills require a windy location, and the power generation of PV is variable and intermittent suffering from uncertainty. To solve this problem, energy storage systems are installed to store energy from renewable energies.

Over the last past few years, the application of storage resources in power systems have stirred wide attentions and interest. Distributed storage resources are now penetrating rapidly towards

commercialization as competitive alternatives to the large-scale centralized generations [24]. Distributed storage resources includes energy storage resources, renewable energy resources, and other small-scale generations dispersed on the electrical distribution networks. There are many small and large-scale energy storage resources installed in power systems. The interest resulted from several reasons. First, there are more efficient storage resources with lower cost and higher power output in the market due to advances in technologies. Due to the variable and intermittent nature of renewable resources output, the integration of renewable energy resources must include corresponding energy storage resources. The efficient and low emission distributed storage resources are considered as competitive alternatives to large-scale centralized generator technologies. Second, under restructuring and deregulation, electricity supply industries in many countries have been transformed from centralized and regulated power systems to deregulated market-based systems. As competitive retail markets for electricity begin to open, the pressure to supply premium power has promoted the use of storage resources to improve power quality and reliability during periods of generation shortages or transmission congestion. Storage systems can provide energy during the high market price periods. This feature makes energy storage systems particularly suitable as a physical hedge against energy market volatility. Therefore, retailers and end users can use energy storage systems to supply demand and reduce the electricity cost instead of purchasing electrical energy with high cost from the electricity market. Meanwhile, energy storage systems are also alternatives to expensive reinforcement of generation and capacity in the transmission and distribution networks. In this case, storage systems are utilized to supply the peak power or network capacity. Finally, the tightening environmental requirements on clean energy with lower emission and environmental impacts have led to more usage of storage resources in the commercial reality. The old or less efficient conventional fossil-fuel power plants are replaced or avoided using with the energy storage systems.

Recent developments and advances in energy storage and power electronics technologies are making the application of energy storage technologies a potentially viable solution for modern power applications, allowing the system to be operated in a more flexible and controllable manner [25]. Energy storage system could be utilized for improving reliability, providing a source of energy when the production is low or during periods of high demand. Some of these energy storage systems are battery, hydrogen, and fuel cells. Among them, the battery energy storage system is one of the most promising technologies [76].

For short duration requirements, battery storage system can bring frequency control and stability, while for longer duration requirements, it can bring energy management or help to reserve energy. Battery storage system also can be used to complement primary generation as they can be used to produce energy during off peak periods. This energy produced can be stored as reserve power which will be used during the evening peak hours.

The advantages of battery used as the storage system in the electric supply network are summarized as follows.

1. It can reduce the operation cost when used as a generation resource combined with the electric utility network. It can increase the lifespan of existing transmission and distribution equipment and defer their upgrade.
2. It provides load-smoothing by storing excess energy from other energy sources, i.e., photovoltaic, when the demand is low, and to provide extra power when the demand is high.
3. It can be used to reduce the load during the peak hours. Therefore, transmission line stability and reliability of utility service for the residential households are guaranteed.
4. It can facilitate the inclusion of renewable energies into the electric power grid. When used

with renewable resources, battery can improve the efficiency and usability of PV and wind generated electricity by making its generation coinciding with peak load demand.

5. It can be used as an emergency backup power system in case of power outage. During the blackout, battery can be used as an independent electricity generation source without any warm-up time necessary.

Therefore, battery should be widely utilized as a power resource for industrial and residential customers.

Electric utilities charge time-of-use electric rates, which feature higher demand charges and energy charges during on-peak periods than during off-peak periods. For economical operation and control purposes, electric power users with hybrid systems are interested in the availability and the dispatch of different power supplies on an hourly basis to the load. This could be formulated as an optimization problem under uncertainty. Taking into account the utility rate structure, as well as the duration of the on-peak and off-peak periods, the optimization problem must determine what fraction of the on-peak period load should be met by what energy sources, how to optimize the dispatch of the hybrid generation system to minimize the cost, and when to schedule the charge or discharge of the battery.

For hybrid systems with batteries and renewable energy resources, and without either diesel generators or the power grid, the dispatch control strategy is simple: if there is excess energy left after meeting the residential demand, the battery gets charged, and the battery discharges if the load exceeds the renewable energy. However, the control strategies of a hybrid system can become very complex if the system includes either a diesel generator or the power grid, as it is necessary to determine how the batteries are charged/discharged and what element (batteries or diesel generator or power grid) have priority to supply energy when the load exceeds the energy generated from

renewable sources.

Operational scheduling of storage resources in the power system has been the subject of many studies. The simplest and most straightforward strategies are predefined rules [19], [28], [46], [47] [63], [92]. A set of IF-THEN rules are created according to the corresponding scenarios. When a specific scenario happens, the operating strategy employs some predetermined rules. For example, one rule would be: if there is a deficit of power and the battery performance can meet the residential demand, the load is met by the battery. Rule-based strategies are relatively simple and can be adapted to a lot of scenarios. However, limitation is obvious that every scenario has to be considered in advance, which is not practical, especially for large complex system. Furthermore, the solution obtained is, in general, not the optimal solution considering the system load and characteristics of the energy sources. Thus, predefined rules cannot fully utilize the benefits of the renewable energy sources and storage system for reducing the cost of supplying hourly load. A large number of more complex optimization techniques have been applied to solve this problem, such as dynamic programming [15], [88], [90], [132], linear programming [26], [30], Lagrange relaxation [89], and nonlinear programming [113]. These techniques aim at reducing either computation time or memory requirements. Recently, computational intelligence methodologies including fuzzy optimization, genetic algorithm, simulated annealing method and particle swarm optimization approach have been employed to deal with the operation cost of hybrid energy systems with storage systems [24], [25], [43], [76], [129]. Generally, these heuristic approaches can provide a reasonable solution. However, these approaches are not able to adapt to frequent and swift load changes and real-time pricing due to their static nature. Therefore, we develop in the Chapter 6 an operational scheme with self-learning ability and adaptability to optimize residential energy systems according to system configurations and user demand. The self-learning scheme based on adaptive dynamic programming has the capability to learn from the environment and

the residential demand so that the performance of the algorithm will be improved through further learning.

2 NEURAL NETWORK MODELLING FOR THE TEST ENGINE

The control task always requires the availability of a suitable model of the process to be controlled of the engine. The characterization of the model is an important step because the consistency of the model with respect to the actual process will affect the reliability of the controller developed for the process. Any errors in the modeling, therefore, will be transferred to the controller. Thus, appropriate identification methodologies have to be developed to adapt control or diagnosis schemes to different engines and make them robust with respect to aging effect and vehicle-to-vehicle variability.

In the framework of engine control, advanced research mainly relies on model-based techniques. Mean value engine models (MVEMs) , which allow describing the main dynamic processes as function of the mean values of the most significant engine variables, were originally proposed in [7] and [53]. MVEMs are suitable for online AFR control operation since they are based on a mean value scale and they allow observing the dynamic processes with a good level of accuracy and a limited computational demand. Despite their intrinsic accuracy, there are some disadvantages that limit the application of this method. One disadvantage is that some physical effects are not directly included, for example, EGR. Furthermore, the nonlinear parameters of the model are determined by the static maps as function of engine speed and the load. In order to solve these problems, adaptive methods have been utilized to estimate the state and tune the parameters to make use of real-time measurements, such as in [10], [107].

Neural networks, which is one of the most attractive techniques for simulating highly nonlinear and dynamic systems, are good candidates for the AFR process modelling. AFR process is usually characterized as highly nonlinear multiple input multiple output systems with high degree of interaction among the variables and lots of constraints imposed. Moreover, the so-called wall-wetting

phenomena and variable time delays compound the difficulties of the AFR estimation. Cylinder AFR is measured in an indirect way by measuring a wide-range AFR sensor mounted in the exhaust pipe. As a result, there are three time-delays included in the measured signal: the injection delay, the combustion delay and the transportation delay [44]. The measured AFR is expressed as: $AFR_s(t) = AFR_c(t - t_D)$ and $t_D = t_{inj} + t_{burn} + t_{trans}$ where t_{inj} is the injection delay, t_{burn} is the combustion delay and t_{trans} is the transport delay. The measured AFR is then divided by the stoichiometric AFR for gasoline (14.67). Obviously, a method that would be able to capture the average contribution of all cylinders and be robust, fast reacting and insensitive to noise is required for this complicated process. Neural networks with high mapping capabilities and good generalization ability provide good solutions for such modelling.

It had been stated by many [3], [4], [62], [111], that if the process can be described by a nonlinear function that represents a nonlinear mapping between inputs and outputs, then a neural network containing one hidden layer should be capable of approximating this function sufficiently well. This is almost guaranteed provided that the neural network has sufficient number of neurons in its forward part, the size and distribution of the training sequences is appropriately chosen and the training algorithm is efficient enough for the representation of the nonlinear system [55]. Further, it is proposed by many [3], [4], [17], and [62] that if a multilayer feedforward neural network can be constructed to approximate any nonlinearity, then by adding a feedback loop to construct a recurrent element, it should be able to approximate the state of the nonlinear dynamical system in the given range of the training set. This can be accomplished by different methods such as the addition of dynamical memory elements at the output, reminiscent to Hopfield networks as in [17], [62] or by constructing more complicated fully recurrent network as in [3], [4]. We consider nonlinear output error (NNOE) model in our attempts to model AFR process and feedforward neural network to model TRQ process.

2.1 Engine Description

A test vehicle with a V8 engine and 4-speed automatic transmission is instrumented with engine and transmission torque sensors, wide-range AFR ratio sensors in the exhaust pipe located before and after the catalyst on each bank, and exhaust gas pressure and temperature sensors. The vehicle is equipped with a dSPACE rapid prototyping controller for data collection and controller implementation. Data are collected at each engine event under various driving conditions, such as federal test procedure (FTP cycles), as well as more aggressive driving patterns, for a length of about 95,000 samples during each test. The engine is run under closed-loop fuel control using switching-type oxygen sensors. The dSPACE is interfaced with the powertrain control module in a by-pass model. Due to different complex nature of the TRQ and AFR processes, two neural network models are built for these processes with structures compatible with the mathematical engine model developed by Dobner [38], [39] and others.

2.2 Identification of Engine Torque using Feedforward Neural Network

In view of the fact that the torque generation process appeared to be quasi-static and governed mostly by nonlinear properties, it is assumed that the simple multilayer feedforward neural network should be sufficient for the process identification. Figure 2.3 shows the structure of the model. The TRQ model is represented by neural network containing four input neurons, one hidden nonlinear layer using *tansig* function (i.e., the hyperbolic tangent function) in Matlab [35] with ten neurons, and one linear output neuron. The inputs to the model are: TPS (throttle position), MAP (manifold absolute pressure), RPM (engine speed) and SPA (spark advance) where TPS is the control signal and the other three inputs are reference signals compatible with the control signal at any operating conditions. The output is TRQ. The target is TRQ* which represents the

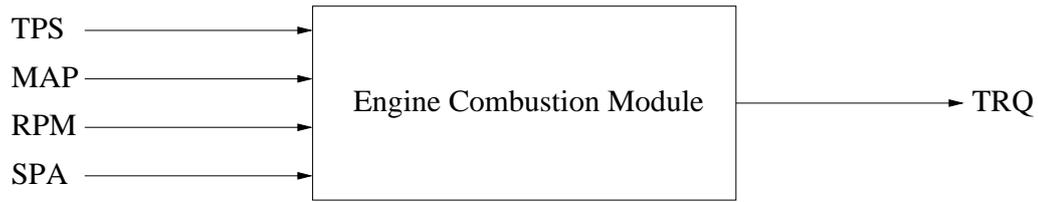


Figure 2.3: Structure of the engine torque model

measured values of engine torque generated by the engine using conventional engine controllers in open- and closed-loop operation. Other parameters have also been tried, such as manifold air temperature, coolant temperature and exhaust gas recirculation, for modelling the TRQ. However, all these extra parameters do not improve the modelling accuracy at all.

Validation results for the TRQ modelling of engine are shown in Figures 2.4–2.6. The first 30000 data points are used for training and the rest 65000 data points are used for validation. Figures 2.4–2.6 display only 10000 data points in each case. The range of these displays is 60000–70000 for validation data. In these figures and all displays to follow, all values are normalized to a range between -1 and 1 for convenience in the neural network training. Both the model networks (TRQ and AFR) were initialized to small random initial weights between -0.18 and 0.18 . The error backpropagation learning algorithm is used for the training of the network model. Figure 2.4 shows the inputs of the feedforward neural network model for TRQ process, while Figure 2.5 shows the output of neural network and the target TRQ. Fig. 2.6 shows the relative deviation of the model from TRQ^* , that is, $(TRQ - TRQ^*) / \max(TRQ^*)$. Due to the high degree of accuracy achieved from Figure 2.5 and 2.6, we feel confident that feedforward neural network is fully capable of representing nonlinear input-output mapping of the torque generation process.

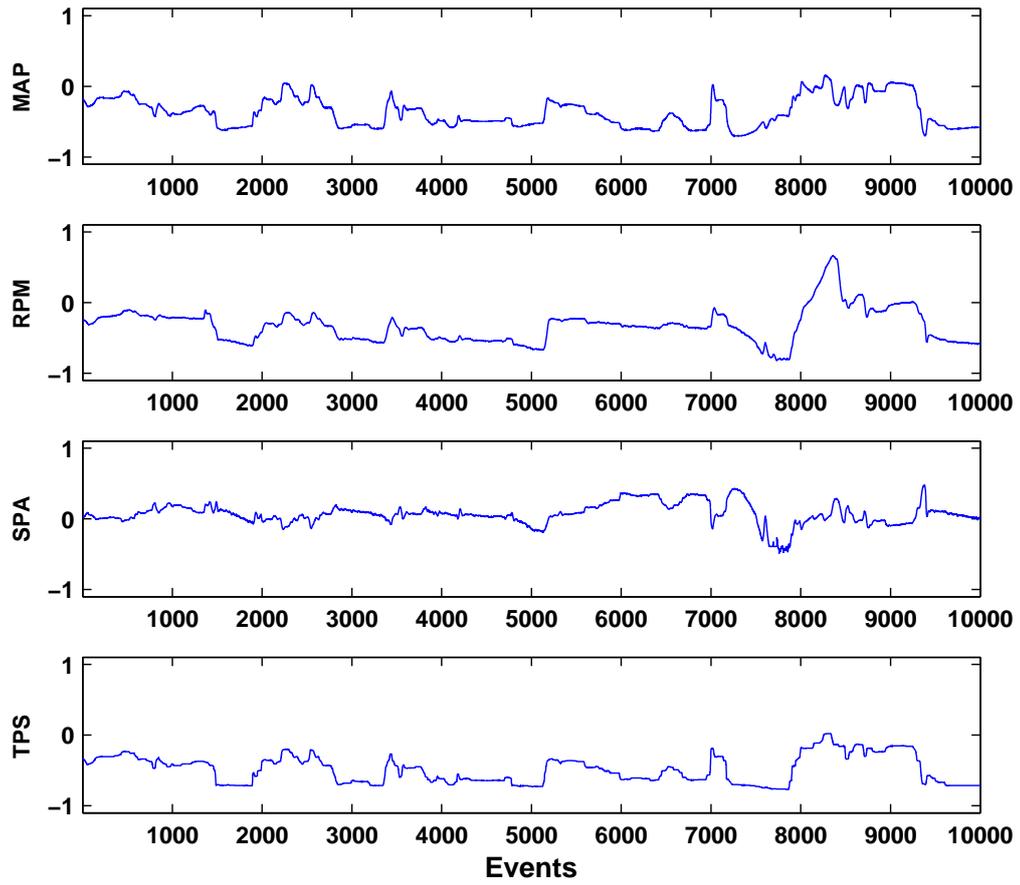


Figure 2.4: Illustration of inputs for the engine TRQ modelling

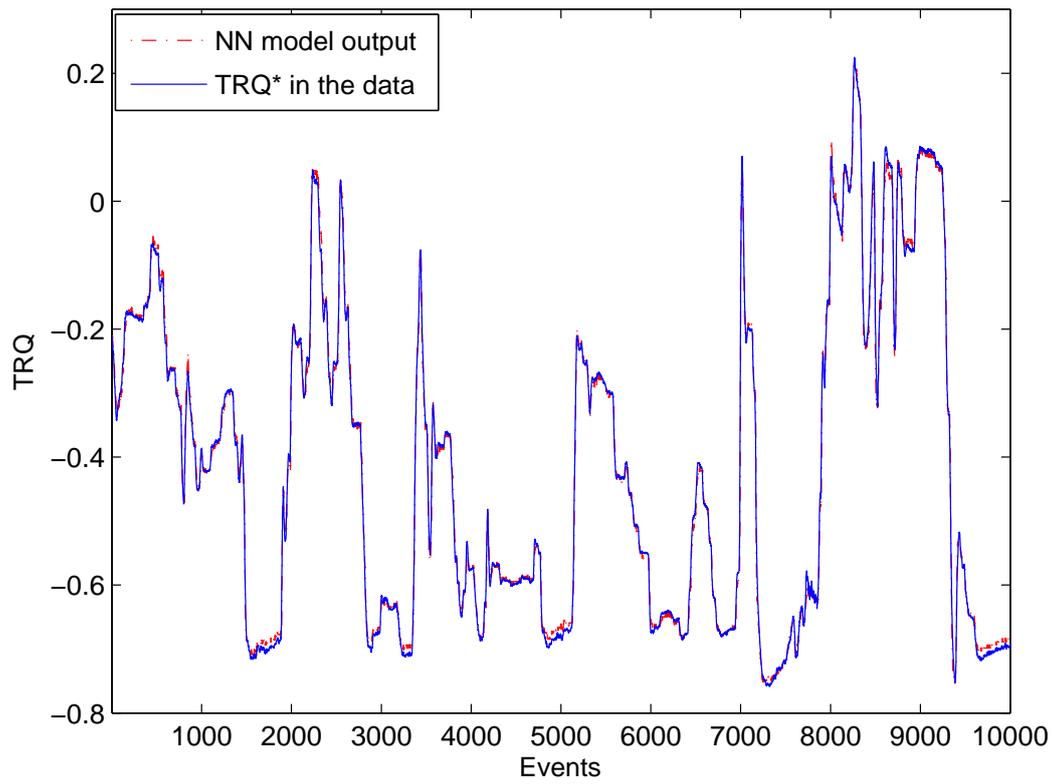


Figure 2.5: Illustration of engine TRQ validation data

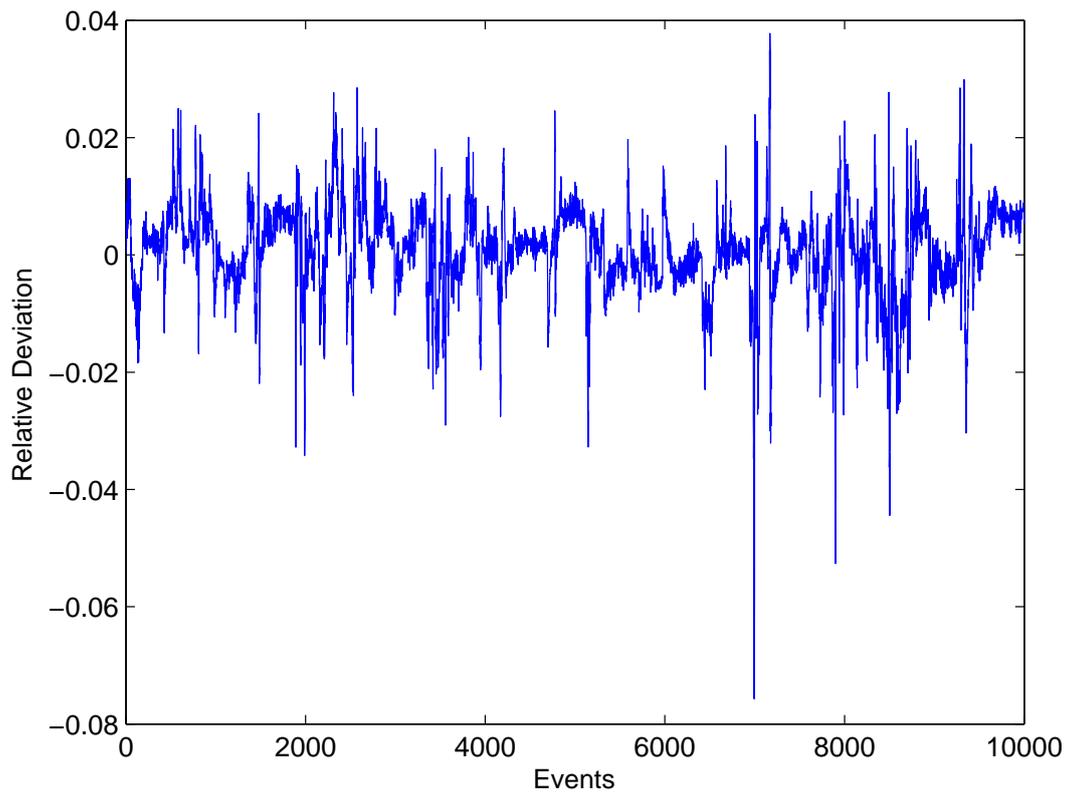


Figure 2.6: Illustration of relative deviation of the model from TRQ*

2.3 Identification of Engine Air-Fuel Ratio using Nonlinear Output Error Model

2.3.1 Manifold Fuel Film Dynamics

In this section a physical insight into the intake manifold fuel dynamics is given through the description of the mean value engine model [7], [53]. For both single and multi-point spark ignition engines, a two phase fuel flow occurs in the intake manifold, with a thin film of fuel on manifold walls and droplets transported by the main stream of air and fuel. It is assumed in [9] that 1) at any instant uniform conditions exist throughout the intake manifold; 2) a fraction D_f of the injected fuel is deposited on the wall as liquid film; 3) the evaporation rate of liquid fuel is proportional to the mass of fuel film through a first order process with time constant τ . The fuel flow entering the combustion chamber $\dot{m}_{f,e}$ is obtained by considering both the vapour flow $\dot{m}_{v,e}$ and the liquid flow $\dot{m}_{l,e}$ (Couette flow). Thus, the mass balance for liquid (m_l) and vapor (m_v) can be expressed by the following system [11]:

$$\dot{m}_l = D_f \dot{m}_{f,i} - \frac{m_l}{\tau} - \dot{m}_{l,e} \quad (2.1)$$

$$\dot{m}_v = (1 - D_f) \dot{m}_{f,i} + \frac{m_l}{\tau} - \dot{m}_{a,e} \frac{m_v}{m_a} \quad (2.2)$$

and the fuel flow entering the combustion chamber is:

$$\dot{m}_{f,e} = \dot{m}_{v,e} + \dot{m}_{l,e} = \frac{m_v}{m_a} \dot{m}_{a,e} + \dot{m}_{l,e}. \quad (2.3)$$

By following the relationship (2.3), the mixture strength of the flow entering the cylinder is given by:

$$\text{AFR} = \frac{\dot{m}_{a,e}}{\dot{m}_{f,e}}. \quad (2.4)$$

The dynamics above derived from general physical consideration can be followed to model any systems, such as continuous and pulse systems. These systems are characterized by different manifold geometry and model parameters (i.e., τ and D_f).

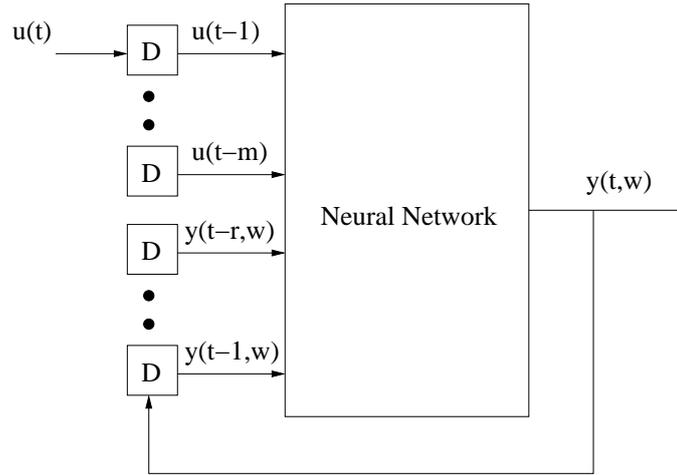


Figure 2.7: The model structure of NNOE

2.3.2 Structure of Model

Considering the AFR process dominated by purely dynamic effects, we choose nonlinear output error model (NNOE) [101], as Figure 2.7 shows, to model AFR dynamics. In practice, NNOE model has been used frequently for function approximation [8], [101]. The general form of the NNOE is:

$$\hat{y}(t|\theta) = F[\hat{y}(t-1|\theta), \hat{y}(t-2|\theta), \dots, \hat{y}(t-m|\theta), u(t-1), \dots, u(t-n)] \quad (2.5)$$

where $\hat{y}(t|\theta)$ is the output, θ is the adjustable parameter, $u(t)$ are reference inputs, the indices m, n define the lag space dimensions of external inputs and feedback variables, and F is a nonlinear mapping function realized by a recurrent neural network. Figure 2.8 shows a recurrent neural network.

Starting from the above relationships, the recurrent neural network implemented in the present

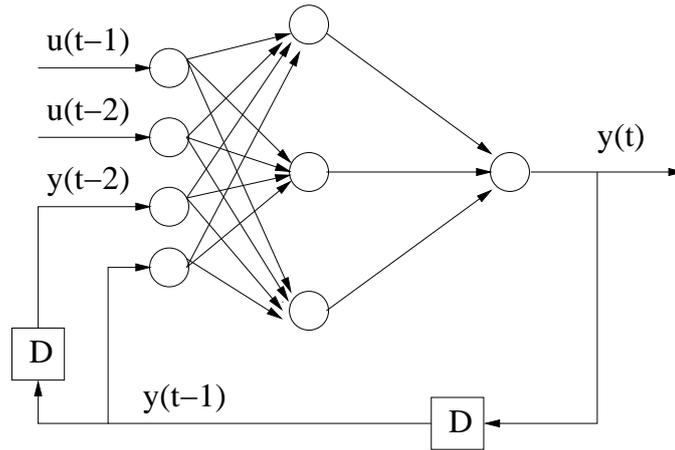


Figure 2.8: The model structure of recurrent neural network

work can be represented in the following short form:

$$\hat{Y}(t|\theta) = f\left(\sum_{i=1}^n v_i g\left(\sum_{j=1}^m w_{ij} \phi_j(t)\right)\right) \quad (2.6)$$

where n is the number of nodes in the hidden layer and m the number of input nodes, $\phi_j(t)$ denotes the inputs to the network including the recurrent input from the output $\hat{Y}(t|\theta)$, v_i represents the weight between the i th node in the hidden layer and the output node, while w_{ij} is the weight connecting the j th input node and the i th node in the hidden layer. These weights are the components of the parameters and $\phi_j(t)$ is the j th element of the input.

2.3.3 Input Variables

As it has been shown in the Section above, the actual AFR (2.4) is computed as the ratio between the air flow rate and the fuel flow rate entering the engine combustion chamber (i.e., leaving the manifold). From the past experiments and experiences, it is customary to assume that the instantaneous air mass flow rate to the engine has a strong dependence on both the manifold pressure and the engine speed (i.e., engine state variables). The actual fuel flow rate to the engine (2.3) depends on the fuel dynamics which is controlled by the amount of injected fuel and depends on

the two-phase process which takes place inside the manifold. This has been described through Equations (2.1) and (2.2) and depends on the set of parameters which in turn can be assumed to be the function of MAP and RPM. Considering the manifold as a multi-input single-output system, the important input variables are MAP, RPM and FPW. Thus, the mathematical model of AFR is:

$$\hat{\text{AFR}}(t|\theta) = F[\hat{\text{AFR}}(t-1|\theta), \hat{\text{AFR}}(t-2|\theta), \dots, \hat{\text{AFR}}(t-r|\theta), u(t-1), \dots, u(t-m)] \quad (2.7)$$

where $u = [\text{MAP}, \text{RPM}, \text{FPW}]$.

2.3.4 Simulation Results

The data range for training NNOE model is from 20000–50000. The NNOE model is trained by Levenberg-Marquardt optimization algorithm in the batch mode. Figure 2.9–2.11 show the simulation results for the validation of the NNOE model. Figure 2.9 shows the inputs of the NNOE for 5000 points, while Figure 2.10 shows the output of NNOE model and the target AFR. Figure 2.11 shows the relative deviation of the model from AFR^* , that is, $(\text{AFR} - \text{AFR}^*) / \max(\text{AFR}^*)$. All these figures indicate a very good match between the real vehicle data and the NNOE model output.

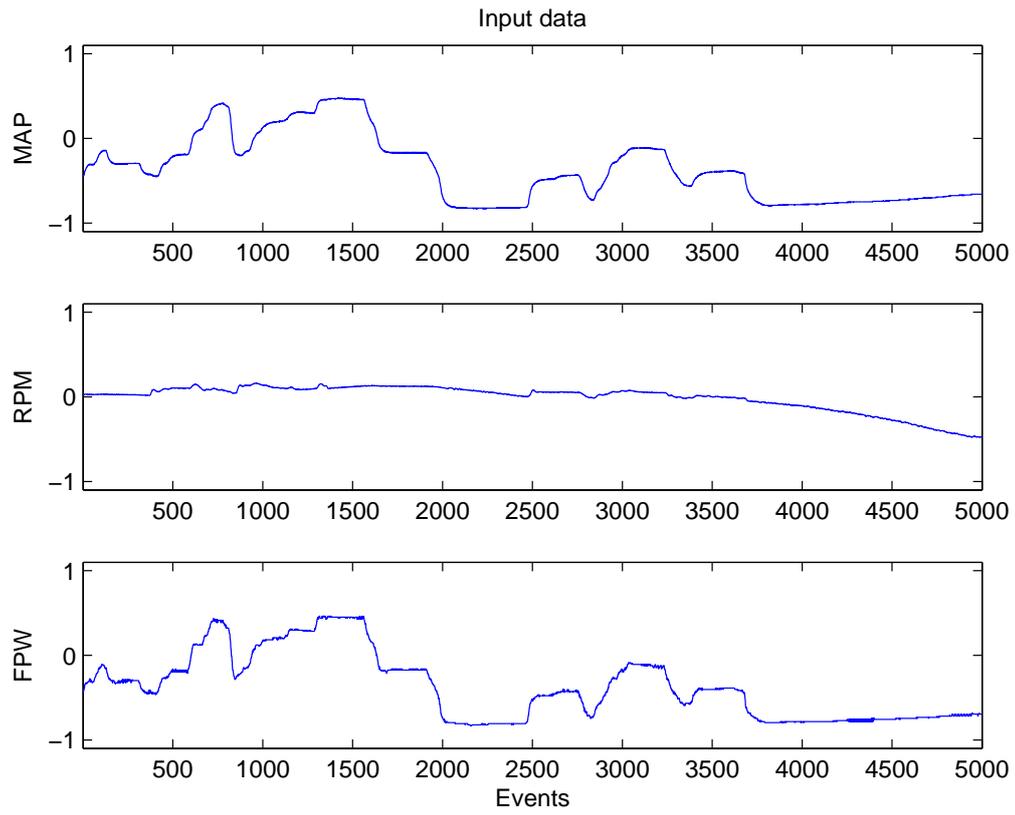


Figure 2.9: Illustration of inputs of NNOE for the engine AFR modelling

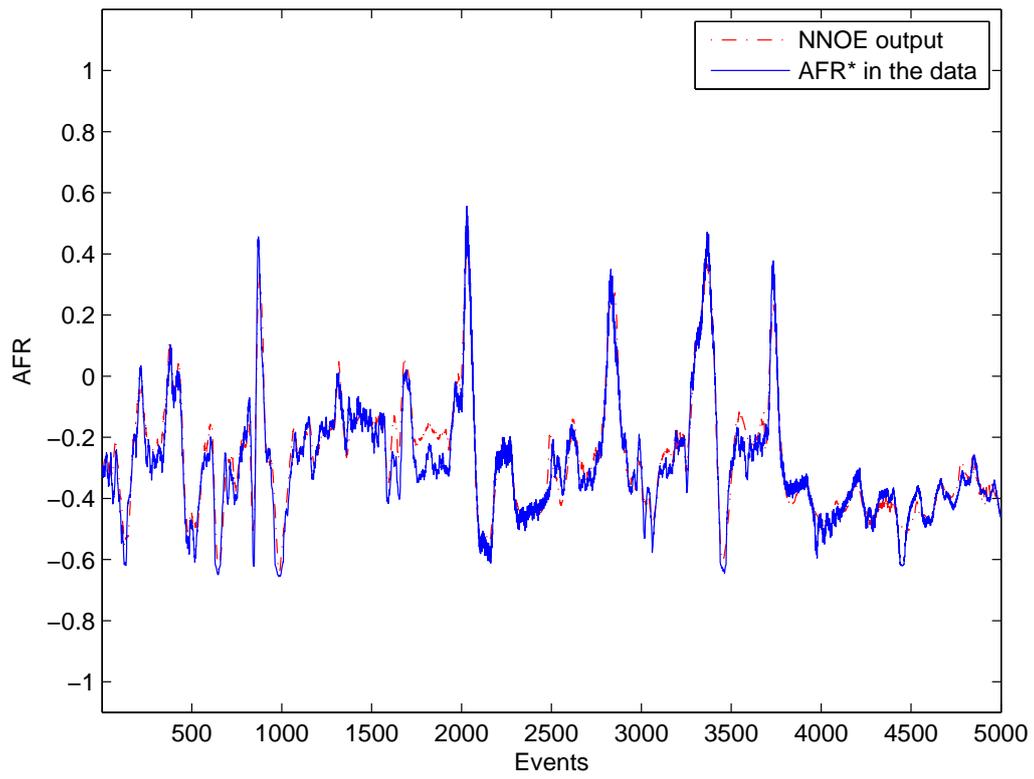


Figure 2.10: Illustration of engine AFR validation data

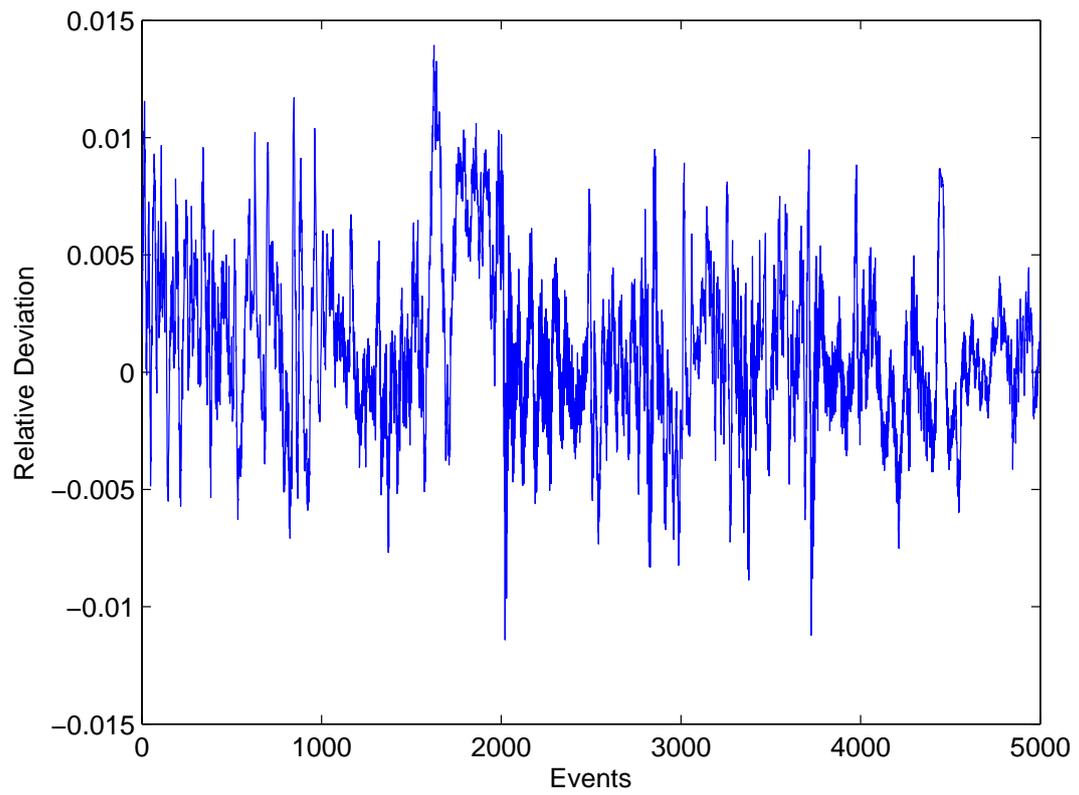


Figure 2.11: Illustration of relative deviation of the AFR model

3 ADAPTIVE CRITIC LEARNING TECHNIQUES FOR ENGINE TORQUE AND AIR-FUEL RATIO CONTROL

3.1 Introduction

In an effort to design more advanced engine control algorithms with the objectives of reduced emissions and improved performance, we develop and evaluate a learning control technique originated from dynamic programming. Dynamic programming is a theory developed back in the 1950's [18] for optimal control of nonlinear systems with the objective of minimizing a performance index that is defined as a summation of a utility function from the present time into the future. In general, using dynamic programming, such an optimal control design for nonlinear systems is only theoretically possible. Moreover, in practice, it has been known for years that due to the so-called "curse of dimensionality" [36], dynamic programming can only be applied to simple, small-scale control problems.

Automotive engines are known to be complex nonlinear dynamical systems. The control problem of automotive engines has been investigated for many years by many researchers. Almost every branch of the modern and classical control theory has been researched for the control of automotive engines. The present study adds another dimension to the existing rich literature on automotive engine control.

The uniqueness of the present study comes from the following idea. Consider a currently existing control algorithm implemented in a production vehicle. The algorithm is designed according to certain criteria and calibrated for vehicle operation over the entire operating regime. In a sense, the algorithm has been optimized for the engine in terms of its performance, fuel economy and tailpipe emissions through a significant effort in the research and development and calibration process. To

further improve the engine performance through controller design, one can go through the traditional calibration and control procedures in place today. An alternative to this traditional approach is to use the neural network-based learning control design approach initiated in the present work.

The final result of our neural network learning process is a controller that has learned to provide optimal control signals under various operating conditions. We emphasize that such a neural network controller will be obtained after a specially designed learning process that performs approximate dynamic programming. Once a controller is learned and obtained (off-line or on-line), it will be applied to perform the task of engine control. The performance of the controller can be further refined and improved through continuous learning in real-time vehicle operations. We note that continuous learning and adaptation to improve controller's performance is one of the key promising attributes of the present approach. Continuous learning and adaptation for optimal individual engine performance over the entire operating regime and vehicle conditions would be desirable for future engine controller designs. For practical reasons, during the initial stage of the controller neural network learning it is preferable to use off-line engine data for initial simulation studies. We will therefore first develop a model of the engine for the purpose of initial neural network controller learning but such a model is not necessary for the real-time engine operation.

This chapter is organized as follows. In Section 3.2, the neural network model of the test engine is described. The model is different from the models in the Chapter 2. In the Chapter 2, we build two separated models for the engine TRQ and AFR processes. Here, a multi-input multi-output model based on time-lagged recurrent neural network is constructed. In Section 3.3, dynamic programming will be introduced. Engine torque and exhaust air-fuel ratio tracking control using adaptive dynamic programming will be developed. In Section 3.4, simulation results for engine torque and air-fuel ratio tracking control using adaptive dynamic programming will be presented. In Section 3.5, the chapter will be concluded with a few remarks.

3.2 Neural Network Modeling of the Test Engine

We build a neural network model for the test engine compatible with the mathematical engine model developed by Dobner [38] [39] and others. Due to the complexity of modern automotive engines, in the present work, we use the time-lagged recurrent neural networks (TLRNs) for engine modeling. In practice, TLRNs have been used often for function approximation and it is believed that they are more powerful than the networks with only feedforward structures (cf. [110, 135]).

For the neural network engine model, we choose AFR and TRQ as the two outputs. We choose throttle position (TPS), fuel pulse width(FPW), and spark advance (SPA) as the three control inputs. These are input signals to be generated using our new adaptive critic learning control algorithm. We choose intake manifold pressure (MAP), mass air flow rate (MAF), and engine speed (RPM) as reference inputs. The time-lagged recurrent neural network used for the engine combustion module has six input neurons, a single hidden layer with eight neurons, and two output neurons. Validation results for the outputs TRQ and AFR of the neural network engine model indicate very good match between the real vehicle data and the neural network model outputs during the validation phase [75].

3.3 Adaptive Dynamic Programming

Dynamic Programming is a recursive method for solving sequential decision problems. The method searches the optimal path by tracking backward from the final step, retaining all sub-optimal paths from any given point to the finish in memory, until the starting point is reached. The result of this process is that the procedure may incur the exponential growth in the number of calculations for the complex problems, which is too computationally expensive for the most real-world problems. Efforts have been made to overcome this problem. In recent years, adaptive

dynamic programming which approximates dynamic programming arouses wide attentions and interest. The technique uses a critic module to learns the cost function in dynamic programming.

3.3.1 Dynamic Programming

Suppose that one is given a discrete-time nonlinear dynamical system

$$x(t + 1) = F[x(t), u(t), t] \quad (3.8)$$

where $x \in R^n$ represents the state vector of the system and $u \in R^m$ denotes the control action.

Suppose that one associates with this system the performance index (or cost)

$$J[x(i), i] = \sum_{k=i}^{\infty} \gamma^{k-i} U[x(k), u(k), k] \quad (3.9)$$

where U is called the utility function or local cost function and γ is the discount factor with $0 < \gamma \leq 1$. Note that J is dependent on the initial time i and the initial state $x(i)$, and it is referred to as the cost-to-go of state $x(i)$. The objective is to choose the control sequence $u(k)$, $k = i, i + 1, \dots$, so that the function J (i.e., the cost) in (3.9) is minimized. Dynamic programming is based on Bellman's principle of optimality [18], [36]: An optimal (control) policy has the property that no matter what previous decisions (i.e., controls) have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.

Suppose that one has computed the optimal cost $J^*[x(t + 1), t + 1]$ from time $t + 1$ on for all possible states $x(t + 1)$, and that one has also found the optimal control sequences from time $t + 1$ on. The optimal cost results when the optimal control sequence $u^*(t + 1), u^*(t + 2), \dots$, is applied to the system with initial state $x(t + 1)$. Note that the optimal control sequence depends on $x(t + 1)$. If one applies an arbitrary control $u(t)$ at time t and then uses the known optimal control

sequence from $t + 1$ on, the resulting cost will be

$$U[x(t), u(t), t] + \gamma J^*[x(t + 1), t + 1],$$

where $x(t)$ is the state at time t and $x(t + 1)$ is determined by (3.8). According to Bellman's principle of optimality, the optimal cost from time t on is equal to

$$J^*[x(t), t] = \min_{u(t)} \left(U[x(t), u(t), t] + \gamma J^*[x(t + 1), t + 1] \right). \quad (3.10)$$

The optimal control $u^*(t)$ at time t is the $u(t)$ that achieves this minimum, i.e.,

$$u^*(t) = \arg \min_{u(t)} \left(U[x(t), u(t), t] + \gamma J^*[x(t + 1), t + 1] \right). \quad (3.11)$$

Equation (3.11) is the principle of optimality for discrete-time systems. Its importance lies in the fact that it allows one to optimize over only one control vector at a time by working backward in time. In other words, any strategy of action that minimizes the function J in the short term will also minimize the sum of U over all future times. Equation (3.10) is called the functional equation of dynamic programming and is the basis for computer implementation of dynamic programming.

In the computations in (3.11), whenever one knows the function J in (3.9) and the model F in (3.8), it is a simple problem in function minimization to pick the action $u^*(t)$ that achieves the minimum in (3.11). However, this procedure requires a backward numerical process and it is too computationally expensive to determine the solutions due to the so-called "curse of dimensionality" [18, 36].

3.3.2 Action-Dependent Heuristic Dynamic Programming

A typical design of ADP consists of three modules: 1) critic (for evaluation); 2) model (for prediction); and 3) action (for decision). When in ADP, the critic network (i.e., the evaluation module)

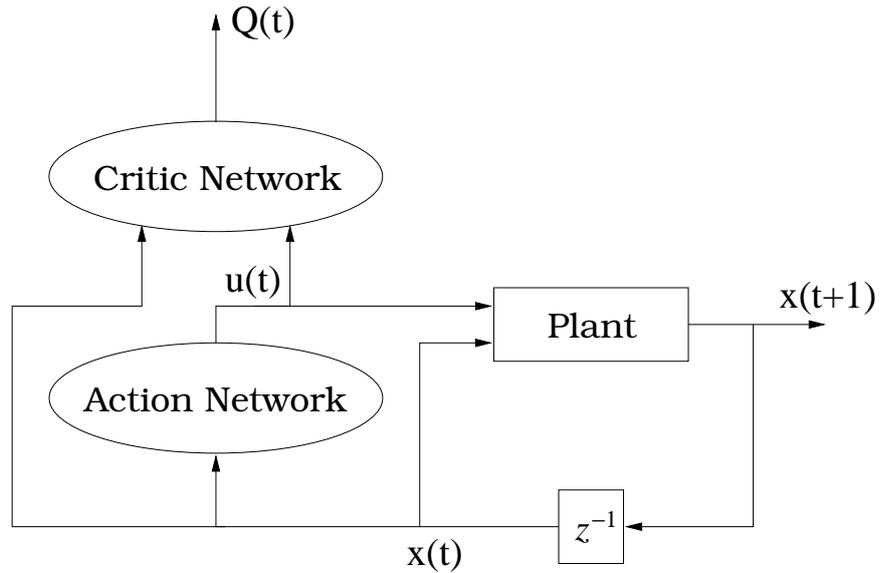


Figure 3.12: A typical scheme of an action-dependent heuristic dynamic programming

takes the action/control signal as part of its inputs, the designs are referred to as action dependent ADP.

In the present work we use an action-dependent version of ACDs that does not require the explicit use of the model network in the design. Consider the action-dependent heuristic dynamic programming (ADHDP) shown in Figure 3.12 (cf. [87]). The critic network in this case will be trained by minimizing the following error measure over time,

$$\begin{aligned} \|E_q\| &= \sum_t E_q(t) \\ &= \sum_t [Q(t-1) - U(t) - \gamma Q(t)]^2 \end{aligned} \quad (3.12)$$

where $Q(t) = Q[x(t), u(t), t, W_C]$ and W_C represents the weight vector of the critic network

obtained through training. When $E_q(t) = 0$ for all t , (3.12) implies that

$$\begin{aligned}
 Q(t-1) &= U(t) + \gamma Q(t) \\
 &= U(t) + \gamma[U(t+1) + \gamma Q(t+1)] \\
 &= \dots \\
 &= \sum_{k=t}^{\infty} \gamma^{k-t} U(k).
 \end{aligned} \tag{3.13}$$

Clearly, comparing (3.9) and (3.13), we have $Q(t-1) = J[x(t), t]$. Therefore, after the minimization of error function in (3.12), the output of neural network trained becomes an estimate of the performance cost defined in dynamic programming for $i = t + 1$, i.e., the value of the performance cost in the immediate future.

The input-output relationship of the critic network in Figure 3.12 is given by

$$Q(t) = Q[x(t), u(t), t, W_C]$$

where W_C represents the weight vector of the critic network. In off-line applications, the critic network is trained to obtain an initial fixed weight vector W_C . However, in real time applications, this vector can be adaptively modified to account for slow changes in the process. There are two methods to train the critic network according to the error function (3.12) in the present case.

(1) *Backward-in-time*: We can train the critic network at time t , with the desired output target given by $[Q(t-1) - U(t)]/\gamma$. The training of the critic network is to realize the mapping given by

$$C_f: \left\{ \begin{array}{c} x(t) \\ u(t) \end{array} \right\} \rightarrow \left\{ \frac{1}{\gamma} [Q(t-1) - U(t)] \right\}. \tag{3.14}$$

In this case, we consider $Q(t)$ in (3.12) as the output from the network to be trained and the target output value for the critic network is calculated using its output at time $t-1$. Thus, we refer to this method as the backward-in-time method. Note that $Q(t-1) = Q[x(t-1), u(t-1), t-1, W_C]$ in (3.14) is obtained at time $t-1$ and stored in the memory. This method is illustrated in Figure 3.13.

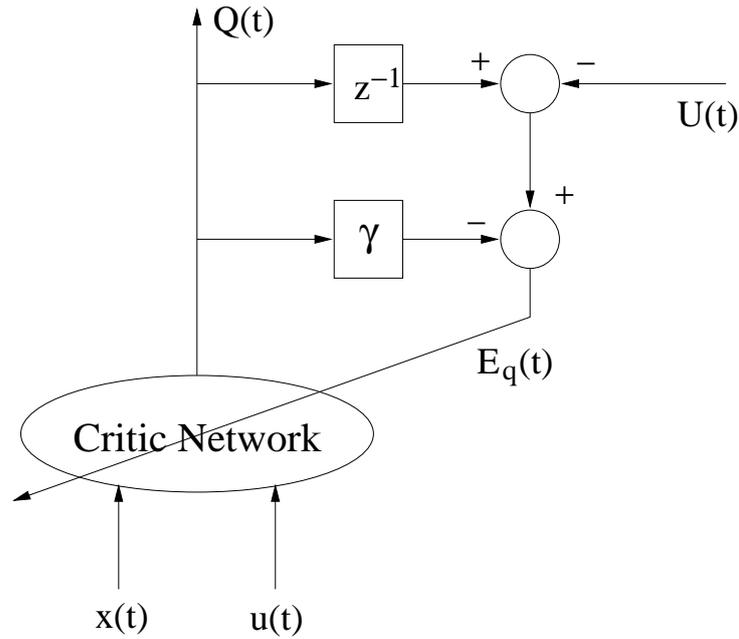


Figure 3.13: Illustration for the backward-in-time method

(2) *Forward-in-time*: We can train the critic network at time $t - 1$, with the desired output target given by $U(t) + \gamma Q(t)$. The training of the critic network is to realize the mapping given by

$$C_f: \begin{Bmatrix} x(t-1) \\ u(t-1) \end{Bmatrix} \rightarrow \{U(t) + \gamma Q(t)\}. \quad (3.15)$$

In this case, we consider $Q(t - 1)$ in (3.12) as the output from the network to be trained and the target output value for the critic network is calculated using its output at time t . Thus, we refer to this method as the forward-in-time method. Note that the training in this case still happens at time t and $x(t - 1)$ and $u(t - 1)$ in (3.15) are obtained at time $t - 1$ and stored in the memory. This method is illustrated in Figure 3.14.

Our experiments have shown that if $U(t)$ is chosen as an error function (which will be non-zero until the control objective is reached), the training of the critic network using backward-in-time method will sometimes be numerically unstable due to the increase of the target value in (3.14). For example, when $0 < \gamma < 1$, the target value in (3.14) will tend to increase in magnitude unless

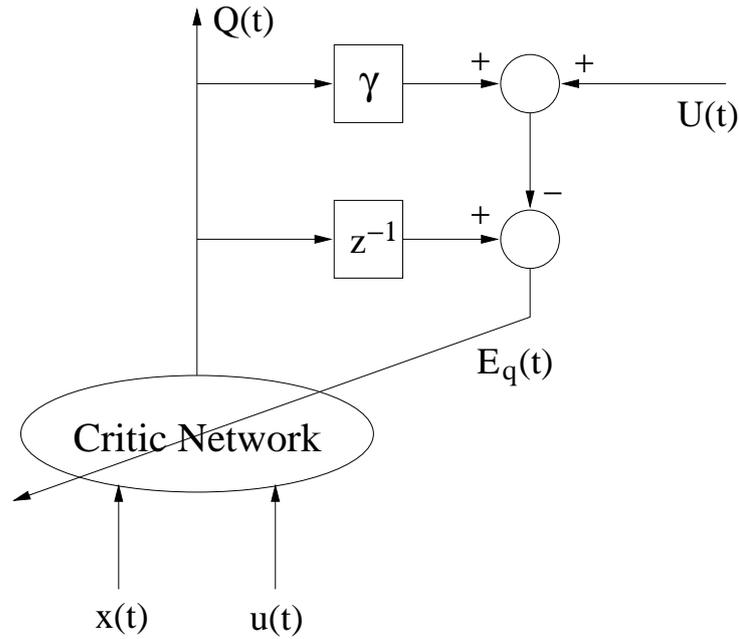


Figure 3.14: Illustration for the forward-in-time method

if we choose $U(t) = 0$ for most t . In the following we will provide discussions concerning the forward-in-time method. Similar discussions are applicable to the backward-in-time method as well. Initially, the target for the critic network training given in (3.15) for time step t is calculated using $Q(t) = Q[x(t), u(t), t, W_C^{(0)}]$, where $W_C^{(0)}$ is the initial value of the critic network weights. After one step of weight update using either sequential mode training or batch mode training, the critic network weight vector will become $W_C^{(1)}$. This weight updating process will be repeated until no further improvements in the critic network training can be achieved. This is determined by either no more weight update during training or no more improvement in the error function during training. A good critic network obtained after training will satisfy the following properties:

$$1) |Q(t-1) - U(t) - \gamma Q(t)| < \varepsilon, \text{ i.e.,}$$

$$\left| Q[x(t-1), u(t-1), t-1, W_C^{(p)}] - U[x(t), u(t), t] - \gamma Q[x(t), u(t), t, W_C^{(p)}] \right| < \varepsilon$$

for each t and for some p , where ε is the maximum error tolerance, and

2) $\|W_C^{(p+1)} - W_C^{(p)}\| < \eta$ where η is very small and $\|\cdot\|$ denotes an appropriate norm.

The first condition implies that with the weight vector $W_C^{(p)}$ obtained after training, we are able to balance the equation

$$Q(t-1) \approx U(t) + \gamma Q(t)$$

with an error strictly below the error tolerance ε . We note that the key to adaptive dynamic programming is to find a critic network so that its outputs will balance this equation for all t as is derived in (3.13). The second condition implies that the weight update of the critic network indicated by the training algorithm from the p th cycle to the $(p+1)$ st cycle is so small that it can be practically ignored. This is because the training has reached the global minimum or a local minimum in the weight space of the error function. If the training reaches a local minimum and the performance of the critic network is not satisfactory, we will reinitialize the critic network using random weights and start the critic network training again.

The training samples for the critic network are obtained over a trajectory starting from $x(0)$ at $t = 0$. Starting from $x(0)$, we can generate $u(0)$ from the action network and we can apply $\{x(0), u(0)\}$ to the plant to be controlled to obtain $x(1)$ (cf. Figure 3.12). We can then generate $u(1)$ from the action network using $x(1)$ and apply $\{x(1), u(1)\}$ to the plant to be controlled to obtain $x(2)$, and so on. Initially, the action signals $u(t)$, $t = 0, 1, \dots$, will be generated given $x(t)$, $t = 0, 1, \dots$, from an action network that is initialized with random weights. We can collect training samples either over a fixed number of time steps (e.g., 300 consecutive points [108]) or from $t = 0$ until the final state is reached (e.g., the fuel cut-off mode is activated in our engine control problem). For each time instant t , the local cost function will be computed as $U(t) = U[x(t), u(t), t]$. Some or all of the data collected, $\{x(t), u(t), U(t)\}$ for $t = 0, 1, 2, \dots$, are stored in the computer memory for the purpose of the critic network and action network training to be

described next.

After the critic network's training is finished, the action network's training starts with the objective of minimizing $Q(t)$. To collect training samples for the action network, we start with $x(0)$ and we apply $x(0)$ to the action network to obtain $u(0)$. We then apply $x(0)$ and $u(0)$ to the plant to be controlled to obtain $x(1)$. Using the action network, we obtain $u(1)$ by applying $x(1)$ and we obtain $x(2)$ from the plant at the next time step. Clearly, such a data collection process can be combined with the data collection process for the critic network training. This process continues until all the necessary training patterns are collected. The goal of the action network training is to minimize the critic network output $Q(t)$. In this case, we can choose the target of the action network training as zero, i.e., we will train the action network so that the output of the critic network becomes as small as possible. In general, a good critic network should not output negative values if $U(t)$ is non-negative. This is particularly true when $U(t)$ is chosen as the square error function in tracking control problems. The desired mapping which will be used for the training of the action network in the present ADHDP is given by

$$A: \{x(t)\} \rightarrow \{0(t)\} \quad (3.16)$$

where $0(t)$ indicates the target values of zero for the critic network output. We note that during the training of action network, it will be connected to the critic network as shown in Figure 3.12. The target in (3.16) is for the output of the whole network, i.e., the output of the critic network after it is connected to the action network as shown in Figure 3.12.

After the action network's training cycle is completed, one may check the system's performance, then stop or continue the training procedure by going back to the critic network's training cycle again, if the performance is not acceptable yet.

3.3.3 Tracking Control Problem

Assume that the control objective is to have $x(t)$ in (3.8) track another signal given by $x^*(t)$. We can define in this case the local cost function $U(t)$ as

$$U(t) = \frac{1}{2}e^T(t)e(t) = \frac{1}{2}[x(t) - x^*(t)]^T[x(t) - x^*(t)].$$

Using the ADHDP introduced earlier in this section, we can design a controller to minimize

$$J(t) = \sum_{i=t}^{\infty} \gamma^{i-t} U(i)$$

where $0 < \gamma < 1$. We note that in this case our control objective is to minimize an infinite summation of $U(t)$ from the current time to the infinity future, while in conventional tracking control designs, the objective is often to minimize $U(t)$ itself.

The present neural network engine controller will be obtained through the following three steps:

- (1) The optimal controller is designed based on the idea of adaptive dynamic programming;
- (2) The initial controller is trained using a neural network engine model developed from off-line engine data;
- (3) In real-time operation of the vehicle, the controller is further refined through continuous learning and adaptation.

3.4 Simulation Studies

The objective of the present engine controller design is to provide control signals, so that the torque generated by the engine will track the torque measurement as in the data and the AFR will track the required values also as in the data. The measured torque values in the data are generated by

the engine using the existing controller. Our learning controller will assume no knowledge about the control signals provided by the existing controller. It will generate a set of control signals that are independent of the control signals in the measured data. Based on the data collected, we use our learning controller to generate control signals TPS, FPW, and SPA, with the goal of producing exactly the same torque and AFR as in the data set. That is to say, we keep our system in the same requirements as the data collected and build a controller that provides control signals that achieve the torque control and AFR control performance of the engine.

As described in the previous section, the development of an adaptive critic learning controller involves two stages: 1) the training of a critic network and 2) the development of a controller/action network. We describe in the rest of this section the learning control design for tracking the TRQ and AFR measurements in the data set. This is effectively a torque-based controller, i.e., a controller that can generate control signals given the torque demand. The block diagram of the present adaptive critic engine control (including AFR control) is shown in Figure 3.15. The diagram shows how adaptive critic designs can be applied to engine control through adaptive dynamic programming.

3.4.1 Critic Network

The critic network is chosen as a 8–15–1 structure with eight input neurons and fifteen hidden layer neurons.

- The eight inputs to the critic network are TRQ, TRQ*, MAP, MAF, RPM, TPS, FPW, and SPA, where TRQ* is read from the data set, indicating the desired torque values for the present learning control algorithm to track.
- The hidden layer of the critic network uses sigmoidal function, i.e., the `tansig` function in

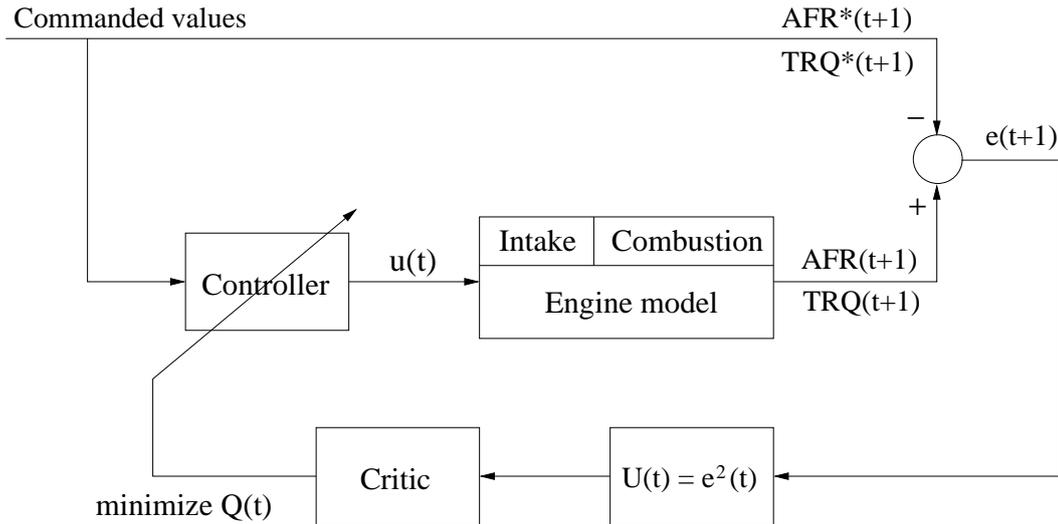


Figure 3.15: Structure of the adaptive critic learning engine controller

MATLAB [35], and the output layer uses the linear function `purelin`.

- The critic network outputs the function Q , which is an approximation to the function $J(t)$ defined as in (3.9).
- The local cost function U defined in (3.9) in this case is chosen as

$$U(t) = \frac{1}{2} [\text{TRQ}(t) - \text{TRQ}^*(t)]^2 + \frac{1}{2} [\text{AFR}(t) - \text{AFR}^*(t)]^2$$

where $\text{TRQ}(t)$ and $\text{AFR}(t)$ are the TRQ and AFR generated using the proposed controller, respectively, and TRQ^* and AFR^* are the demanded TRQ value and the desired AFR value, respectively. Both TRQ^* and AFR^* are taken from the actual measured data in the present case. The utility function chosen in this way will lead to a control objective of TRQ following TRQ^* and AFR following AFR^* .

- Utilizing the MATLAB Neural Network Toolbox, we have applied `trainngdx` (gradient descent algorithm) for the training of the critic network. We note that other algorithms implemented in MATLAB, such as `trainngd`, `trainngda`, `trainngdm`, `trainlm` are also

applicable. We employ batch training for the critic network, i.e., the training is performed after each trial of certain number of steps (e.g., 10000 steps). We choose $\gamma = 0.9$ in the present experiments.

3.4.2 Action Network

The structure of the action network is chosen as 6–12–3 with six input neurons, 12 hidden layer neurons, and three output neurons.

- The 6 inputs to the action network are TRQ, TRQ*, MAP, MAF, THR, and RPM, where THR indicates the driver's throttle command.
- Both the hidden layer and the output layer use the sigmoidal function `tansig`.
- The outputs of the action network are TPS, FPW, and SPA, which are the three control input signals used in the engine model.
- The training algorithm we choose to use is `traingdx`. We employ batch training for the action network as well.

3.4.3 Simulation Results

In the present simulation studies, we first train a critic network for many cycles with 500 training epochs in each cycle. At the end of each training cycle, we check the performance of the critic network. Once the performance is found satisfactory, we stop critic network training. This process usually takes about 6–7 hours. We use the forward-in-time training approach for the critic training.

After the critic network training is finished, we start the action network training. We train the

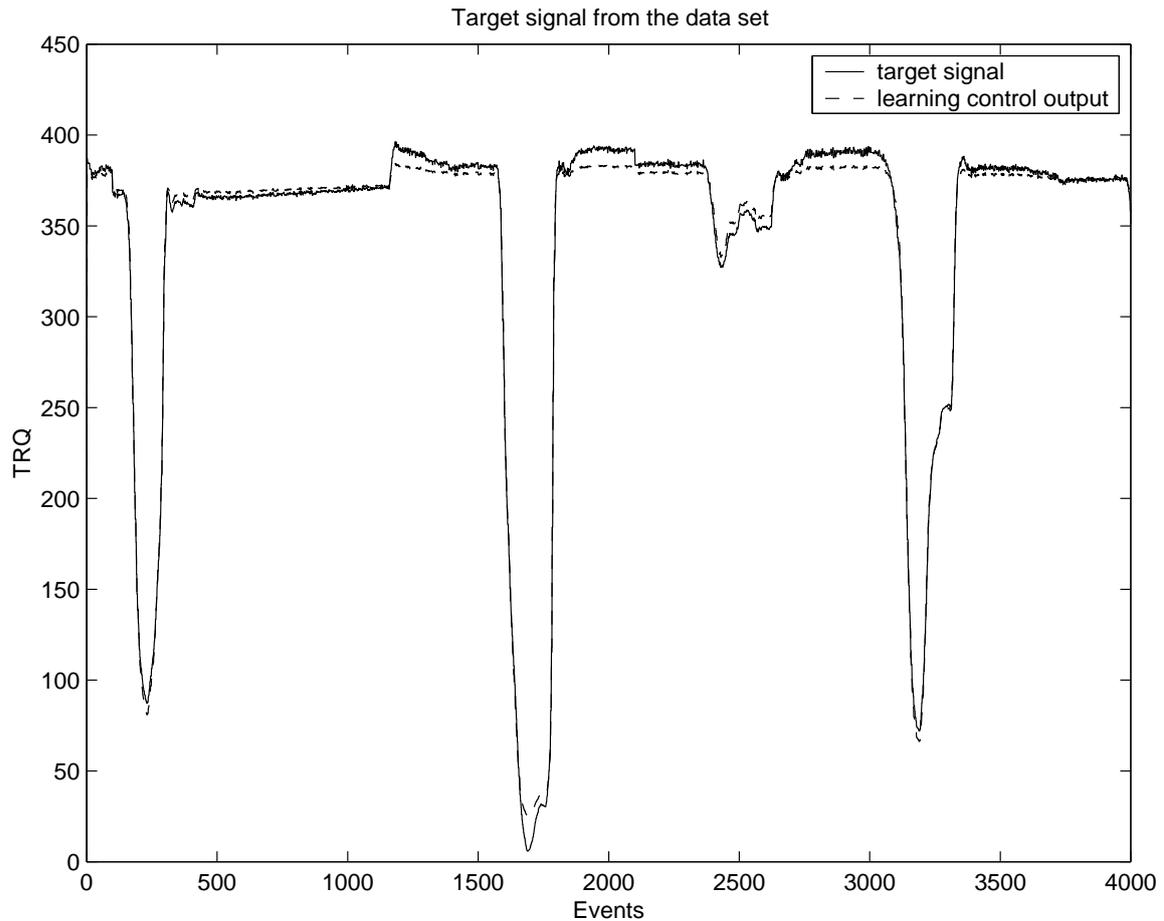


Figure 3.16: Torque output generated by the neural network controller

controller network for 200 epochs after each trial. We check to see the performance of the neural network controller at the end of each trial.

We choose to use 4000 data points from the data (16000–20000 in the data set) for the present critic and action network training.

We first show the TRQ and AFR outputs due to the initial training of our neural network controller when TRQ^* and AFR^* are chosen as random signals during training. Figures 3.16 and 3.17 show the controller performance when it is applied with TRQ^* and AFR^* chosen as the measured values in the data set. The neural network controller in this case is trained for 15 cycles

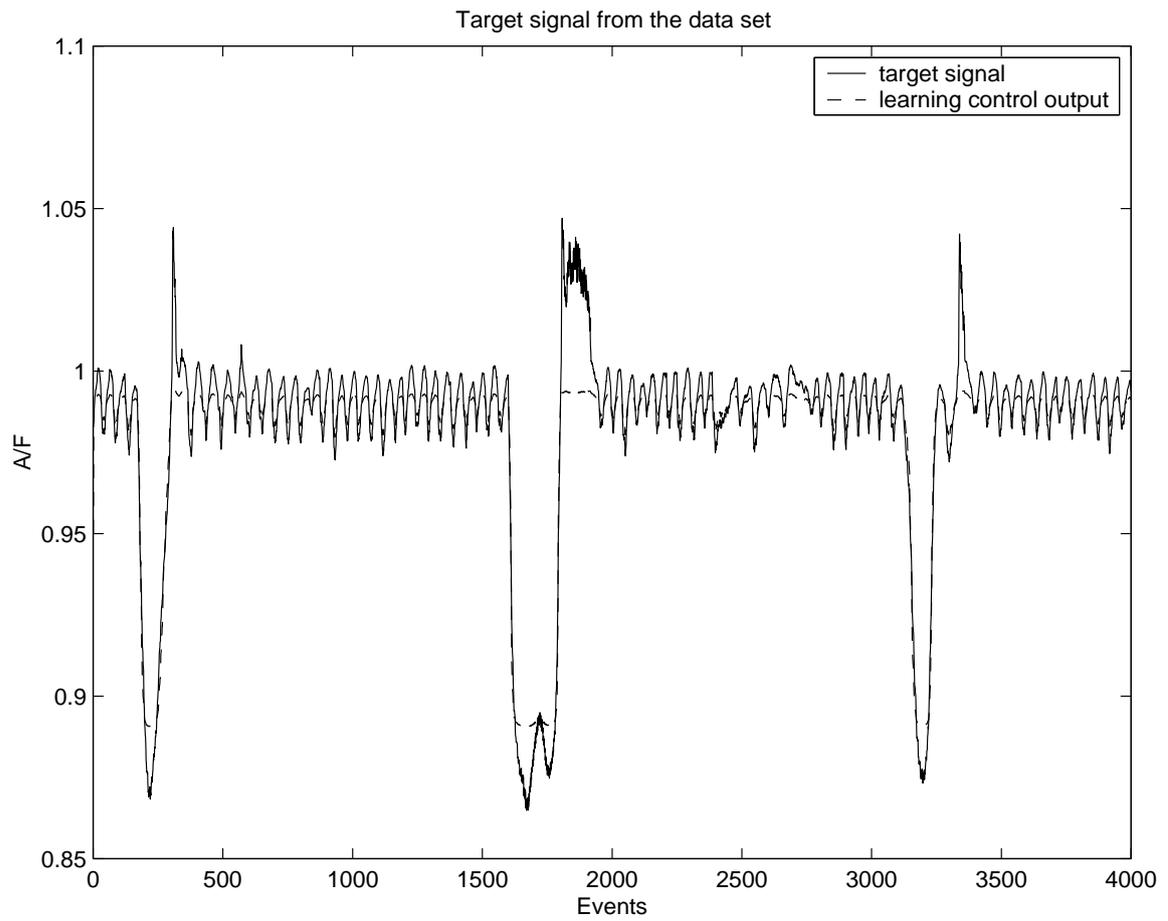


Figure 3.17: Air-fuel ratio output generated by the neural network controller

using randomly generated target signal TRQ^* and AFR^* . Figures 3.16 and 3.17 show that very good tracking control of the commanded torque signal (TRQ) and the exhaust AFR are achieved. We note that at the present stage of the research we have not attempted to regulate the AFR at the stoichiometric value but to track a given command. In these experiments we simply try to track the measured engine-out AFR values so that the control signal obtained can directly be validated against the measured control signals in the vehicle. In Figure 3.17, it appears that better tracking of AFR was achieved on the rich side of stoichiometric value possibly due to more frequent rich excursions encountered during model training. This could also have been caused by intentional fuel enrichments (i.e., wall-wetting compensation) during vehicle accelerations.

Figures 3.18 and 3.19 show the TRQ and AFR outputs after refined training when TRQ^* and AFR^* are chosen as the measured values in the data. The neural network controller in this case is trained for 15 cycles using target signal TRQ^* and AFR^* as in the data. Figures 3.18 and 3.19 show that excellent tracking control results for the commanded TRQ and AFR are achieved.

The figures shown in this section indicate that the present learning controller design based on adaptive dynamic programming is effective in training a neural network controller to track the desired TRQ and AFR sequences through proper control actions.

3.5 Conclusions

Our research results have demonstrated that adaptive dynamic programming provide a powerful alternative approach for engine calibration and control. The design is based on neural network learning using approximate dynamic programming. After the network is fully trained, the present controller may have the potential to outperform existing controllers with regard to the following three aspects. (1) The proposed technique will automatically learn the inherent dynamics and non-

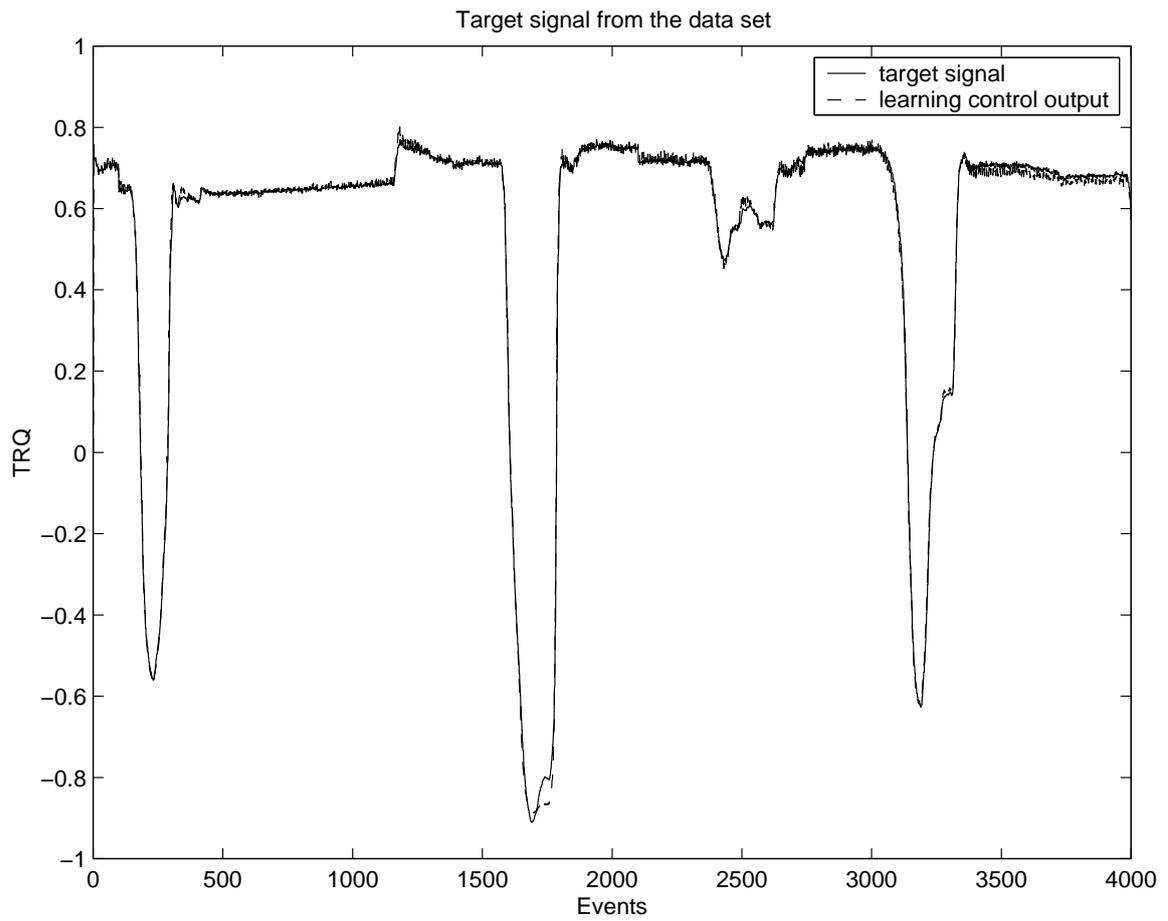


Figure 3.18: Torque output generated by the refined neural network controller

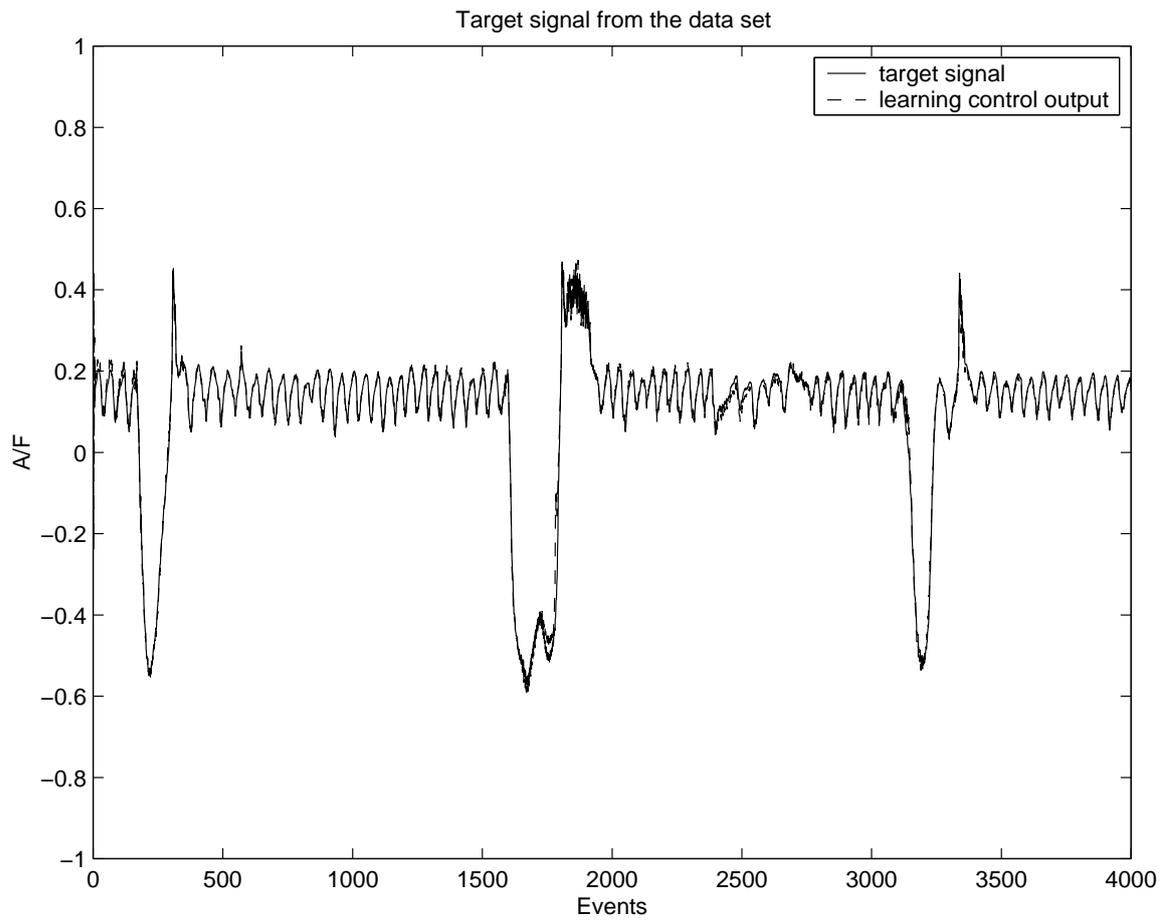


Figure 3.19: Air-fuel ratio output generated by the refined neural network controller

linearities of the engine from real vehicle data and, therefore, do not require a mathematical model of the system to be developed. (2) The methods developed will further advance the development of a virtual powertrain for performance evaluation of various control strategies through the development of neural network models of engine and transmission in a prototype vehicle. (3) The proposed controllers can learn to improve their performance during the actual vehicle operations, and will adapt to uncertain changes in the environment and vehicle conditions. This is an inherent feature of the proposed neural network learning controller. As such, these techniques may offer promise for use as real-time engine calibration tools.

Simulation results show that the proposed self-learning control approach is effective in achieving tracking control of engine torque and air-fuel ratio control through neural network learning.

4 NEURAL SLIDING MODE TECHNIQUE FOR ENGINE TORQUE AND AIR-FUEL RATIO CONTROL

4.1 Introduction

In this chapter, we consider a learning control algorithm based on sliding mode control methods (SLMCs) for the engine TRQ and AFR control with the goals of reduced emissions and improved performance. The theory of sliding mode control has been developed and widely used for more than three decades due to its robustness to system parameter uncertainties and external disturbances [42], [61], [73], [77], [78], [83], [121], [131]. Essentially, SLMCs utilize a high-speed switching control law to drive state trajectory of the nonlinear system onto a specified surface in the state space, called the sliding surface, and to maintain the system state trajectory on this surface for all subsequent times. The system dynamics restricted to this surface result in very robust control systems. By proper design of the sliding surface, SLMCs achieve the conventional goals of control such as stabilization, tracking and regulation. In practical control applications, however, the problem of chattering and difficulty in the calculation of equivalent control limit the implementation of SLMCs. The equivalent control cannot be calculated accurately because limited knowledge of the system is available. The chattering brought by high frequency oscillation of controller output results in low control accuracy and potential instability. The most common approach to reduce the chattering is using saturation function [140]. In order to compensate the uncertainties of the plant and avoid the computational burden, we use an estimation technique based on a neural network for the calculation of the equivalent control. Meanwhile, the continuous output of the corrective controller replaces the discontinuous sign term in conventional SLMCs to eliminate chattering.

The present study considers the neuro-sliding mode control (NSLMC) for both engine TRQ and AFR control of automotive engines. A good deal of work has been reported for the engine

control using sliding mode control methods (SLMCs), a few examples of which are cited in [72], [104], [106], [122], [133], [138]. The application of model-based SLMCs is the emphasis in these works. SLMCs also have been applied to motion control and robotics.

Integration of a neural network into SLMCs can alleviate the problems associated with SLMCs, which can be classified into three main categories. The first method is the use of different kind of neural networks to approximate the plant nonlinearities or uncertainties and improve the control performance consequently [37], [60], [138]. The second method utilizes a neural network for the adaptation of the SLMCs parameters where the SLMCs parameters are progressively updated [70] [84]. The third approach is the use of neural network together with SLMCs either in parallel to act as compensator of the conventional SLMCs controller [1], [77] or to compute the equivalent and corrective control [32], [40], [127]. Both in [40] and [127], two parallel neural networks were used to realize the equivalent control and the corrective control of sliding mode control (SLMCs) design. The difference between these two works is the error for updating the neural network for equivalent control. In [40], the corrective control is handled as the error. While in [127], it is the sliding function S . However, the speed of convergence of either algorithm is slower than the one proposed in this study where no special parameter tuning is needed.

The present work uses two parallel neural networks to realize the equivalent control and the corrective control of the SLMCs design. The calculation of the equivalent control is realized by adaptively learning without a priori knowledge of the plant dynamics. The proposed adaptation scheme directly results in a chatter-free control action for the corrective control. The distinct feature of the present technique is that the learning and control are carried out simultaneously, which allows the neural network controller to be further refined and improved in real-time vehicle operation through continuous learning and adaptation.

This chapter is organized as follows: In Section 4.2, The sliding mode control design is briefly introduced. In Section 4.3, neuro-sliding mode controller will be developed. In Section 4.4, neuro-sliding mode technique for the engine control is presented. In Section 4.5, simulation results for engine TRQ and exhaust AFR tracking control using NSLMC will be presented. In Section 4.6, the chapter will be concluded with some remarks.

4.2 Sliding Mode Control Design

Sliding mode control has been widely used due to its robustness to system parameter uncertainties and external disturbances. The most salient feature of an SLMCs is that the feedback control is discontinuous, switching on one or more manifolds in the state space. When the state crosses each discontinuity surface, the structure of the feedback system is altered. Under certain circumstances, all motions in the neighborhood of the manifold are directed toward the manifold and, thus, a sliding motion on a predefined subspace of the state space is established in which the system state repeatedly crosses the switching surface. This mode has useful invariance properties in the face of uncertainties in the plant model and, therefore, is a good candidate for tracking control of uncertain nonlinear systems [71]. In general, the phase trajectory of the system with SLMCs consists of two parts. The first part is the reaching mode in which the trajectory starting from anywhere on the phase plane moves toward a switching surface and reaches the surface in finite time. The second part is the sliding mode in which the trajectory slides along the surface to the origin of the phase plane.

Consider the following nonlinear, multi-input multi-output system:

$$\dot{X} = G(X, U) \quad (4.17)$$

where $X \in R^p$ and $U \in R^q$ are the states and the control, respectively. The control problem is to

find a control law so that the states X can track the desired trajectory X_d . Let the tracking error of the system be

$$e = X_d(t) - X(t). \quad (4.18)$$

The aim of SLMCs is to drive the system states to the sliding surface $S = 0$ and remain on it. Once the states are on the sliding surface, the system is insensitive to parameter variations or external disturbances. The sliding mode control design approach consists of two steps. The first step is to select a sliding surface that models the desired closed-loop performance in the state variable space according to design specifications. This is a surface that is invariant of the controlled dynamics, where the controlled dynamics are exponentially stable, and where the system tracks the desired set-point. The second step is concerned with the selection of a control law which will drive the system state trajectory toward the sliding surface and remain on it.

The sliding surface S is defined in the state space by the scalar function $S(e) = 0$, where

$$S(e) = c^T e \quad (4.19)$$

$c = [c_1 \ c_2 \ \dots \ c_n]^T$, $e = [e_1 \ e_2 \ \dots \ e_n]^T$, and n is the number of parameters. The vector c are chosen so that $S(e) = 0$, which means the subsequent system is stable [131]. Hence, the control input can drive the system (4.17) to converge on the sliding surface.

The design of SLMCs is based on the selection of Lyapunov function. The control should be chosen such that the candidate Lyapunov function satisfy Lyapunov stability criteria.

Lyapunov theorem for global stability: assume that there exists a scalar function V of S , with continuous first-order derivatives such that (1) $V(S)$ is positive definite, (2) $dV(S)/dt$ is negative definite, (3) $V(S) \rightarrow \infty$ as $\|S\| \rightarrow \infty$, then the equilibrium at the $S = 0$ is globally asymptotically stable [120].

Let the Lyapunov function be selected as below:

$$V(s) = \frac{S^T S}{2}. \quad (4.20)$$

It can be noted that this function is positive definite ($V(S = 0) = 0$ and $V(S) > 0 \forall S \neq 0$). It is aimed that the derivative of the Lyapunov function is negative definite. This can be assured if one can assure that

$$\frac{dV(S)}{dt} = -S^T L \text{sign}(S). \quad (4.21)$$

where L is positive number and $\text{sign}(S)$ is defined as:

$$\text{sign}(S) = \begin{cases} 1 & S > 0 \\ -1 & \text{otherwise.} \end{cases}$$

Taking the derivative of (4.20) and equating this to (4.21), the following equation is obtained:

$$S^T \frac{dS}{dt} = -S^T L \text{sign}(S). \quad (4.22)$$

The time derivative of S can be obtained using (4.19) and the plant equation as given below:

$$\frac{dS}{dt} = C^T \dot{e} = C^T (\dot{X}_d - \dot{X}) = C^T \dot{X}_d - C^T (G(X) + BU). \quad (4.23)$$

By putting (4.23) into (4.22), the control input signal can be written as

$$U(t) = U_{eq}(t) + U_c(t) \quad (4.24)$$

where U_{eq} is the equivalent control, which is the control action necessary to maintain an ideal sliding motion on sliding surface, given by

$$U_{eq}(t) = (C^T B)^+ (C^T \dot{X}_d - C^T G(X)) \quad (4.25)$$

and U_c is the corrective control, which drives the phase trajectory towards the sliding surface, given by

$$U_c(t) = (C^T B)^+ L \text{sign}(S) = K \text{sign}(S) \quad (4.26)$$

where $(C^T B)^+$ is Moore-Penrose pseudo-inverse of the matrix $C^T B$.

SLMCs has many attractive properties, such as insensitivity to parameter variations and good nonlinear control performance. However, in applications of practical control, the SLMCs suffer from three main disadvantages: chattering, difficulty in the calculation of equivalent control and lack of learning ability. The difficulty in the calculation of equivalent control lies in the complete knowledge of the system dynamics required. In implementation, complex systems with multiple variables and nonlinear coupling are hard to understand thoroughly. The calculation of equivalent control requires a good mathematical model of the system, G in (4.17), which is hard to obtain. Moreover, it is difficult to learn the intricacy and the complexity of the real world industrial systems using a conventional linearly parametrized adaptive framework. Neural networks with learning capability represent good candidate solutions to the problems discussed. The corrective control, given by $K \text{sign}(S)$, exhibits high-frequency oscillations in its output, brought about by the high-speed switching necessary for the establishment of a sliding mode. The introduction of boundary layer corresponding to substituting the sign function could give a chattering-free system [140]. However, a finite steady error would always exist. It is suggested that a saturation function or a sigmoid function is used to eliminate chattering instead of the sign function [41]. In our design, the shifted sigmoid function $T(S)$ is used to compute the corrective control:

$$U_c(t) = KT(S) \quad (4.27)$$

where $T(S)$ is chosen as follows:

$$T(S) = \frac{1 - e^{-S}}{1 + e^{-S}}. \quad (4.28)$$

Since the output of the corrective controller is continuous and variable, the chattering is eliminated.

4.3 Neuro-Sliding Mode Control

In this study, considering the difficulty of the calculation of the equivalent control, we use neural networks to generate the equivalent control and the corrective control in SLMCs. The combination of neural network and SLMCs provides the desirable properties from both neural network and SLMCs. Neural network based equivalent control and corrective control possess the features of self-adaptation ability to system uncertainties and robustness to parameter variations and external disturbances. The aim of NSLMC design is to minimize the value of sliding function S such that the system states reach the sliding surface as soon as possible. Two neural networks in parallel are used to realize the equivalent control and corrective control of SLMCs design as in Fig. 4.20 which shows where neural network 1 (NN1) is used to estimate the equivalent control (U_{eq}), and neural network 2 (NN2) is employed to generate the corrective control (U_c). The sliding surface is chosen as:

$$S(e) = c^T e + d^T \dot{e} \quad (4.29)$$

which its value reflects the difference between the actual and the targeted trajectory.

4.3.1 Neural Computation of the Equivalent Control

The structure will be chosen as a two-layer feedforward neural network with one hidden layer and one output layer. The inputs to the neural network are the desired target and actual values of the states. The output of the neural network is the equivalent control U_{eq} . The weight adaptation of the neural network is based on a minimization of the cost function as follows:

$$E = \frac{1}{2}(U_{eq} - \hat{U}_{eq})^2 = \frac{1}{2}\zeta^2 \quad (4.30)$$

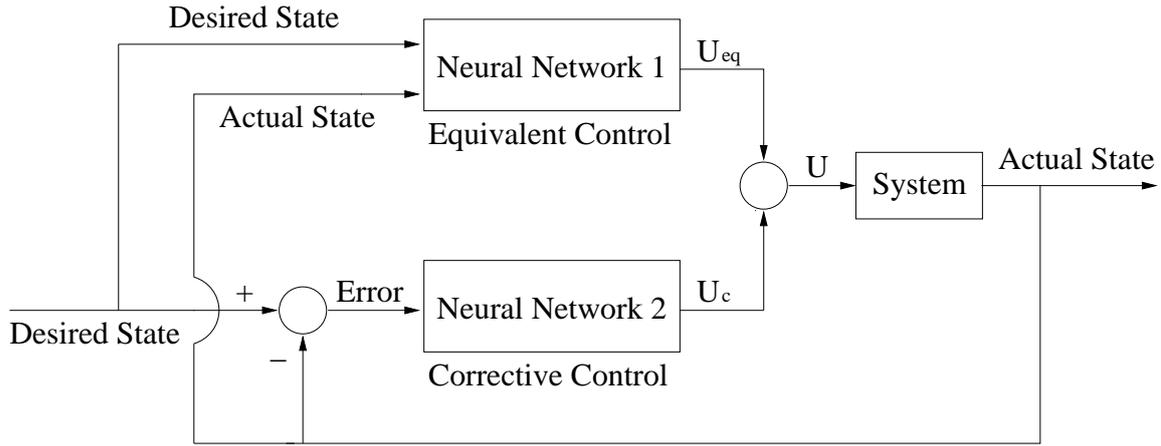


Figure 4.20: Structure of neuro-sliding mode control

where \hat{U}_{eq} is the estimated value of the equivalent control and $\zeta = U_{eq} - \hat{U}_{eq}$.

The Levenberg-Marquardt (L-M) algorithm is used to update the weights of NN1 instead of the error backpropagation (BP) algorithm. The selection of L-M algorithm is based on the fact that the L-M algorithm is widely accepted as the most efficient one in the sense of realization accuracy for nonlinear least squares.[50]

The formula for updating weights is given as follows:

$$\Delta W = [J^T(W)J(W) + \mu I]^{-1} J^T(W)\zeta \quad (4.31)$$

where the parameter μ is adjustable, W is adjustable weight vector and $J(W)$ is the Jacobian matrix. $J(W)$ can be expressed as follows:

$$J(W) = \left[\frac{\partial \zeta}{\partial W_1} \quad \frac{\partial \zeta}{\partial W_2} \quad \cdots \quad \frac{\partial \zeta}{\partial W_n} \right]^T. \quad (4.32)$$

From Eq. (4.30),

$$\frac{\partial \zeta}{\partial W_i} = \frac{\partial (U_{eq} - \hat{U}_{eq})}{\partial W_i} = -\frac{\partial \hat{U}_{eq}}{\partial W_i}. \quad (4.33)$$

Eq. (4.33) can be calculated using the standard BP algorithm. Thus, the Jacobian matrix can be computed by Eq. (4.32) and Eq. (4.33).

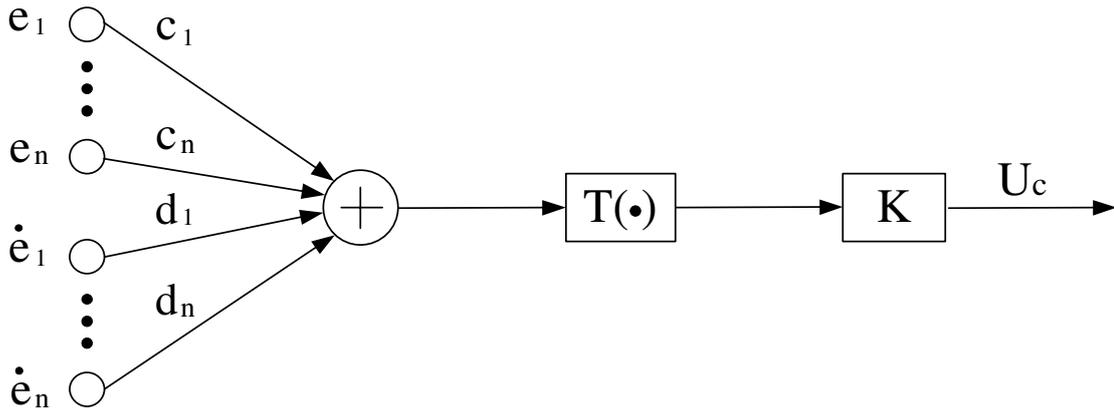


Figure 4.21: Structure of neural network #2 for the corrective control.

From Eq. (4.30), we find that the desired equivalent control is unknown. To overcome this problem, it is suggested that $U_{eq} - \hat{U}_{eq}$ is replaced by the value of sliding function S since the characteristics of $U_{eq} - \hat{U}_{eq}$ and S are similar, that is, when S is close to 0, $U_{eq} - \hat{U}_{eq} \rightarrow 0$. [127]

4.3.2 Neural Computation of the Corrective Control

The structure of the NN2 is decided by the design of SLMC. From the sliding surface (4.29) and equation (4.27), the gains of SLMC are represented as the weights of neural network 2 as in Fig. 4.21 shows. In this way, the gains of SLMC are adapted gradually to the best values.

The cost function is chosen to drive the states converging to sliding surface as follows:

$$J_c = \frac{1}{2} S S^T. \quad (4.34)$$

Minimization of J_c results in minimization of S . To minimize J_c , the weights are changed in the direction of the negative gradient,

$$\Delta K = -\mu \frac{\partial J_c}{\partial K} \quad (4.35)$$

$$\Delta c_i = -\mu \frac{\partial J_c}{\partial c_i} \quad (4.36)$$

$$\Delta d_i = -\mu \frac{\partial J_c}{\partial d_i} \quad (4.37)$$

where K is defined in Eq. (4.27), c_i and d_i are both defined in Eq. (4.29), and μ is the learning rate.

The gradient descent for c_i can be derived using Eq. (4.29) as:

$$\Delta c_i = -\mu \frac{\partial J_c}{\partial c_i} = -\mu \frac{\partial J_c}{\partial S} \frac{\partial S}{\partial c_i} = -\mu S \frac{\partial S}{\partial c_i} = -\mu \cdot S \cdot e_i. \quad (4.38)$$

The gradient descent for d_i can be derived using Eq. (4.29) as:

$$\Delta d_i = -\mu \frac{\partial J_c}{\partial d_i} = -\mu \frac{\partial J_c}{\partial S} \frac{\partial S}{\partial d_i} = -\mu S \frac{\partial S}{\partial d_i} = -\mu \cdot S \cdot \dot{e}_i. \quad (4.39)$$

The gradient descent for K can be derived as:

$$\Delta K = -\mu \frac{\partial J_c}{\partial K} = -\mu \frac{\partial J_c}{\partial S} \frac{\partial S}{\partial K} = -\mu S \frac{\partial S}{\partial K}. \quad (4.40)$$

From Eq. (4.29),

$$S(e) = c^T e + d^T \dot{e} = c^T (X_d - X) + d^T \dot{e}. \quad (4.41)$$

That is, from Eq. (4.17), Eq. (4.24), Eq. (4.27) and Eq. (4.41)

$$\frac{\partial S}{\partial K} = -c^T \frac{\partial X}{\partial K} = -c^T \frac{\partial X}{\partial U} \frac{\partial U}{\partial U_c} \frac{\partial U_c}{\partial K} = -c^T \frac{\partial X}{\partial U} T(S) \quad (4.42)$$

where X is the state of the system and $T(S)$ is defined in Eq. (4.28). Since TRQ and AFR dynamics are represented by neural networks, according to the BP algorithm, $\partial X / \partial U$ can be derived easily.

Finally,

$$\Delta K = -\mu S c^T \frac{\partial X}{\partial U} T(S). \quad (4.43)$$

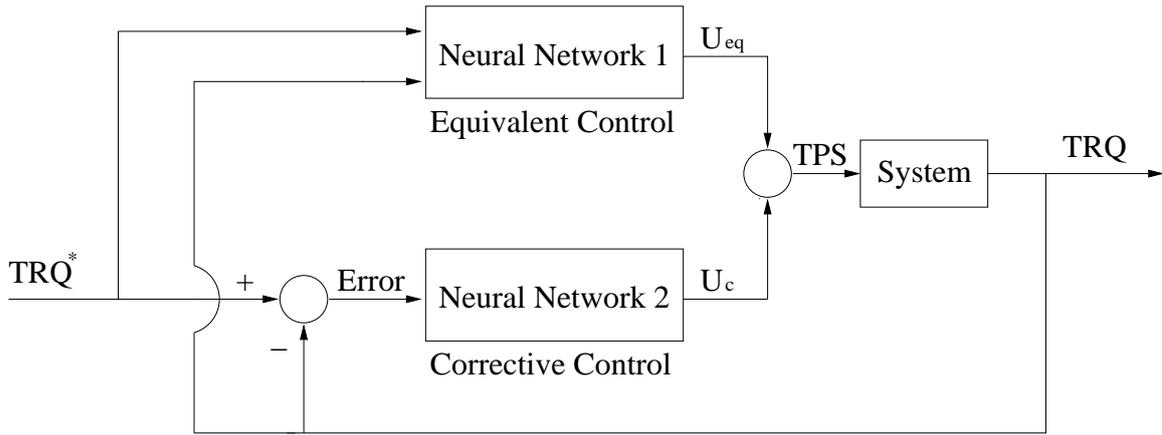


Figure 4.22: Structure of NSLMC for the TRQ control

4.4 Neuro-Sliding Mode Technique for Engine Control

In the present study, we design an intelligent controller that would generate desired control signals, so that the TRQ signals generated by the engine will track the TRQ measurement as in the data and the AFR signals will track the required values also as in the data. The TRQ and AFR measurements in the data are generated by using the existing controller in the car. It is assumed that no a prior knowledge of the control signals generated by the existing controller. Our learning controller will provide control signals TPS and FPW with the control objectives of TRQ and AFR which follow the corresponding measurements as in the data.

The neural network for equivalent control for both TRQ and AFR is chosen as a 2-9-1 structure with 2 inputs and 9 hidden layer neurons. Both the hidden layer and the output layer use the *tansig* function. The block diagram of the present NSLMC controller for TRQ control is shown in Figure 4.22. The inputs to the NN1 are TRQ* (TRQ* is the demanded TRQ value) and actual TRQ. The sum of two outputs of the NN1 and NN2 is TPS which is the control signal for the TRQ. The block diagram for the AFR control is shown in Figure 4.23. The inputs to the NN1 are MAP(t-1) and RPM(t-1) from the data. The sum of two outputs of the NN1 and NN2 is FPW which is the control

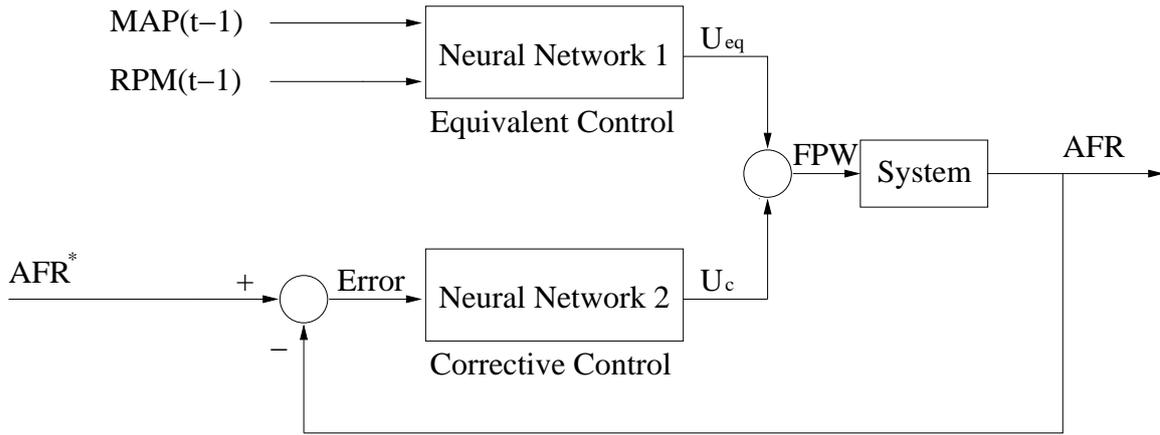


Figure 4.23: Structure of NSLMC for the engine AFR control

signal for the AFR. AFR^* is the desired AFR value. Both TRQ^* and AFR^* are read from the data set, indicating the desired values for the present control algorithm to track. The TRQ and AFR dynamic systems are neural network models we built in Section II.

The sliding surface S is defined as:

$$S = c_1 e + d_1 \dot{e} \quad (4.44)$$

where e represents the difference between target (TRQ^* or AFR^*) and actual values (TRQ or AFR). The sliding surface chosen in this way will lead to control objectives of TRQ which follows TRQ^* and AFR which follows AFR^* . From the defined S , the structure of NN2 can be decided and it is shown in Figure 4.24. The inputs to the NN2 are the error e (between the target TRQ and the actual TRQ for TRQ control or between the target AFR and the actual AFR for AFR control) and the derivative of the error \dot{e} . The overall procedures of NSLMC technique for engine control are given as follows:

Step 1. Initialize: Set all weights of NN1 to small random values in the range of $[-0.18 \ 0.18]$, Set c_1 and K weights of NN2 as 1, 2, respectively. Set learning rate $0 < \mu < 1$. All values of signals were normalized to a range between -1 and 1 for convenience in the neural network

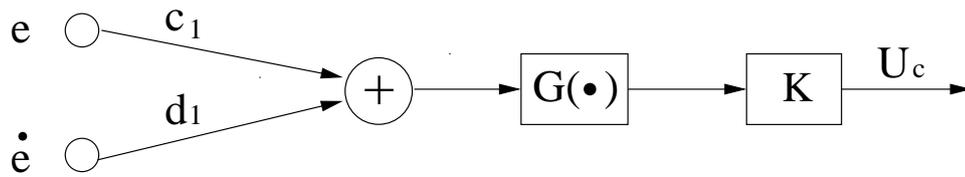


Figure 4.24: Structure of neural network #2

training.

Step 2. Compute the equivalent control U_{eq} from NN1.

Step 3. Compute the corrective control U_c from NN2.

Step 4. Apply the sum of equivalent control and corrective control to the engine system.

Step 5. Measure the state of the engine system (TRQ or AFR).

Step 6. Adjust the weights of NN1 and NN2 according to the weight adaptation rules described in Section IV.

Step 7. Repeat by going to step 2 until the criterion is reached.

4.5 Simulation Results

We randomly choose to use 4000 points from the data (1000-5000 in the data set) for TRQ control and 3000 points (25000-28000 in the data set) for AFR control. Fig. 4.25 and Fig. 4.29 show the controller performance for TRQ and AFR, respectively. Both figures show that excellent tracking control performance is achieved. It is important to note that, at the present stage of the research, we have not attempted to regulate the AFR at the stoichiometric value but to track a given command. In these simulation studies, we simply try to track the measured engine-out AFR values. In this way, the control signal obtained can be directly validated against the measured control signals in

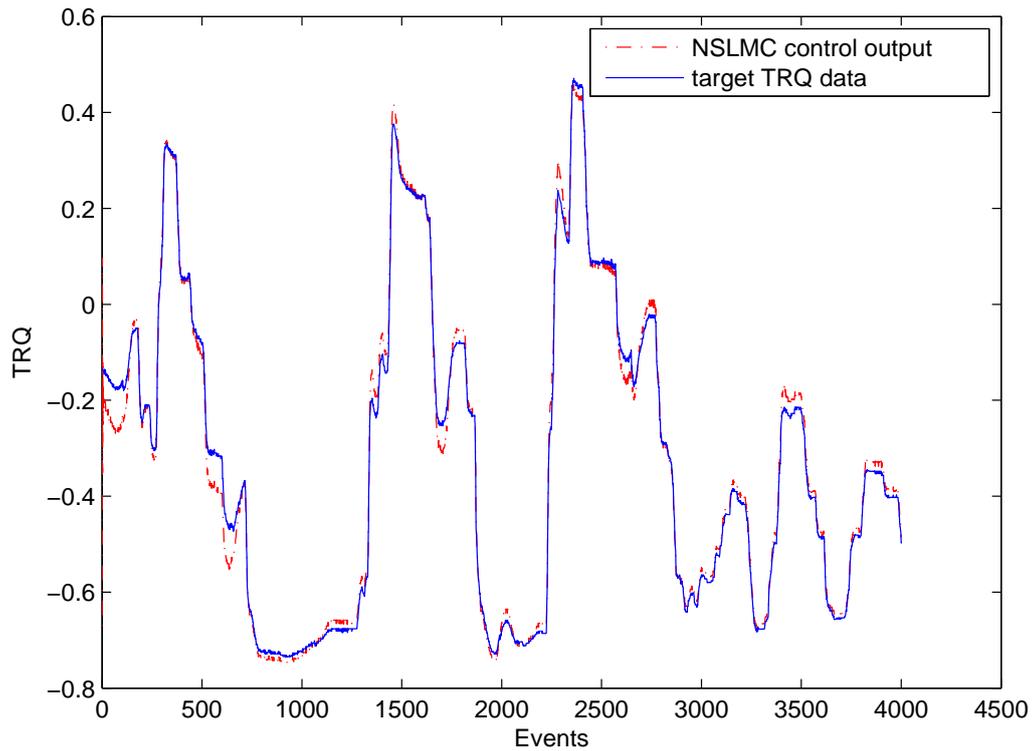


Figure 4.25: Control effect of NSLMC for TRQ control.

the vehicle. Fig. 4.26 and Fig. 4.30 show the output of the equivalent control and corrective control compared to the TPS or FPW values in the dataset. From the figures, we can see that most of the time, the corrective control is around zero. Only when the system states deviate from the sliding mode, the controller takes action to pull the system states back to the sliding surface. The trajectory of the sliding function S is as in Fig. 4.27 and Fig. 4.31. From the figures, we can see that only after 5 events, the sliding function S reaches the acceptable value (under 0.005 is generally regarded as acceptable). Fig. 4.28 and Fig. 4.32 demonstrate the good generalization ability of the NLSMC controller.

The figures shown in this section indicate that the present learning controller design based on the combination of neural network and sliding mode control is effective in training a neural

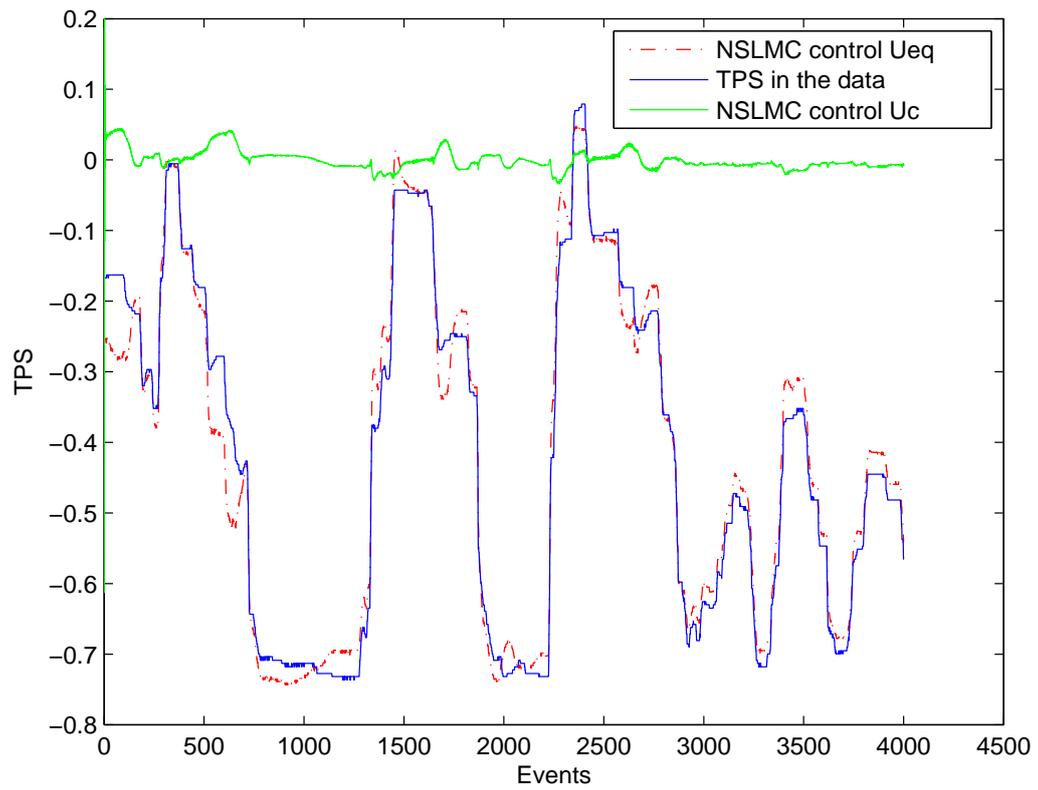


Figure 4.26: Output of equivalent control, corrective control and TPS in the data.

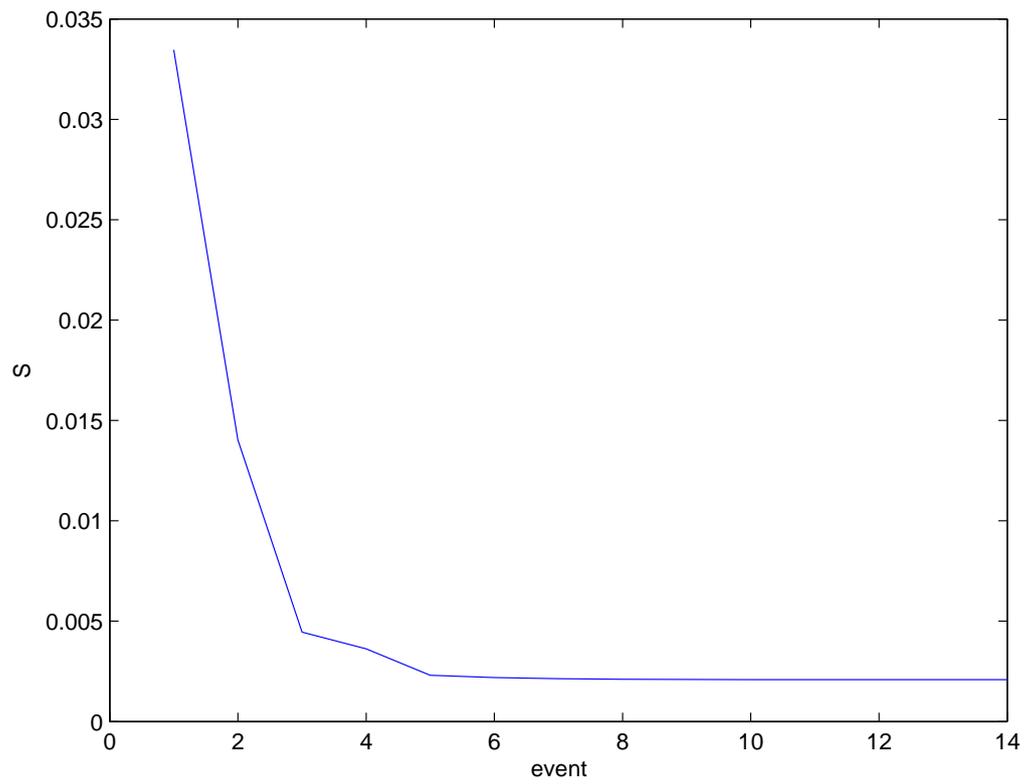


Figure 4.27: Trajectory of sliding function S .

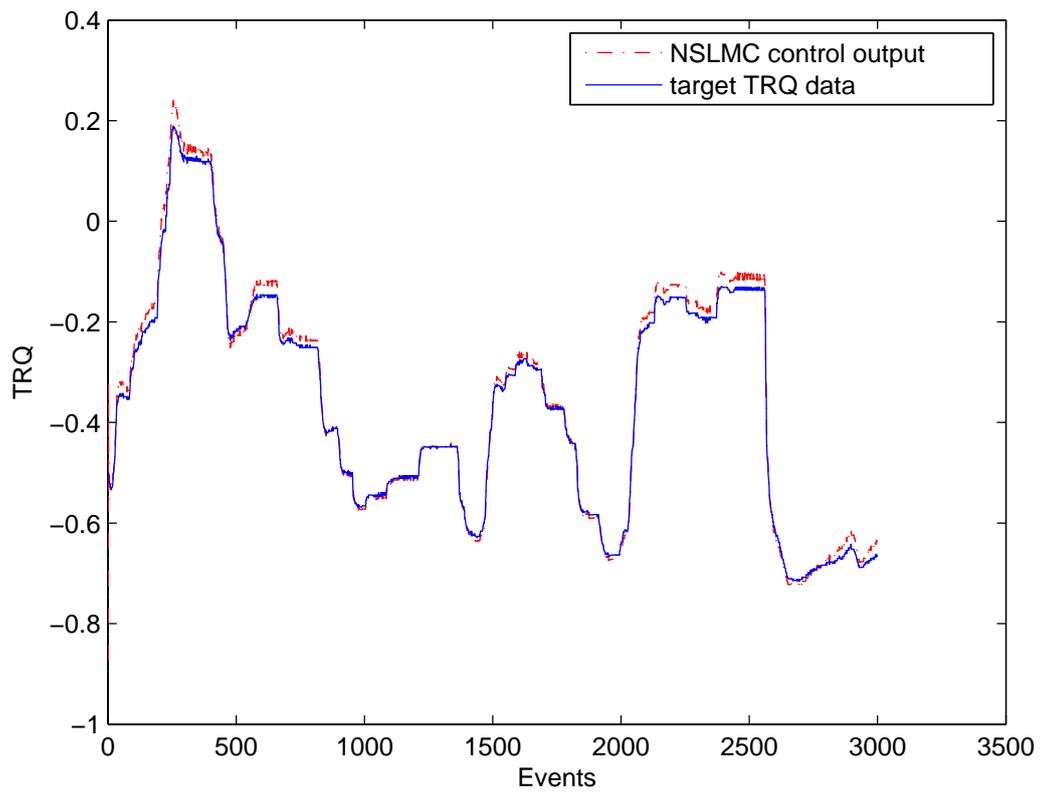


Figure 4.28: NSLMC generalization ability for TRQ control.

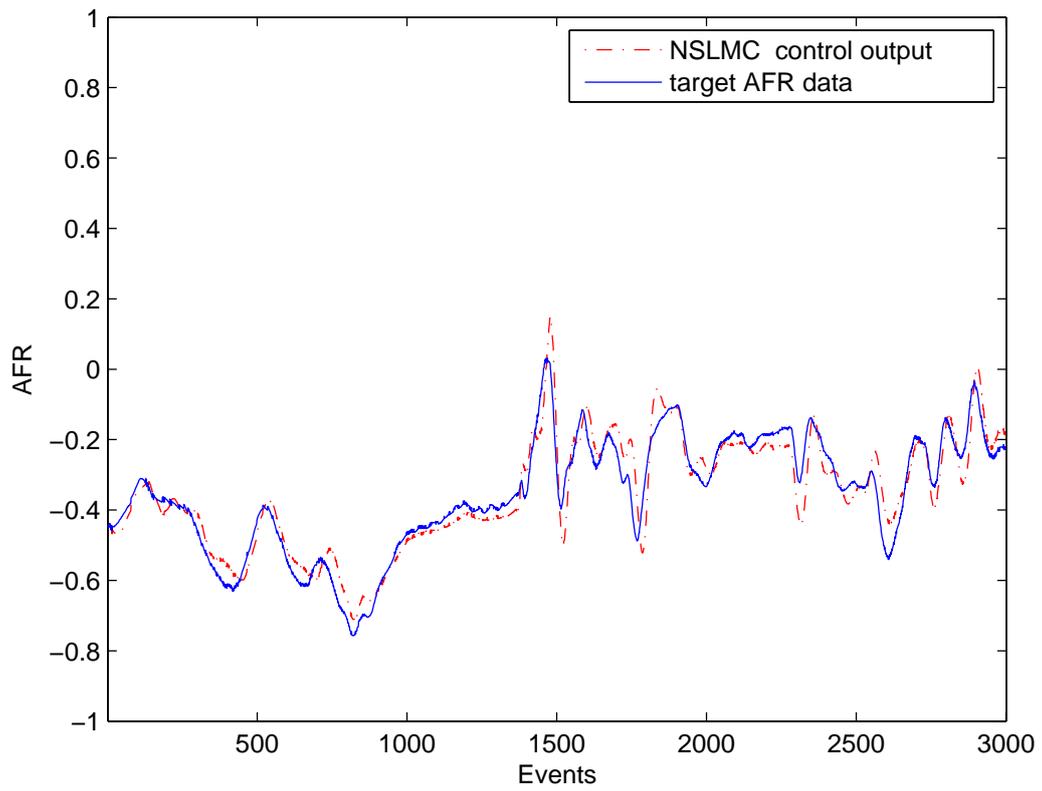


Figure 4.29: Control effect of NSLMC for AFR control.

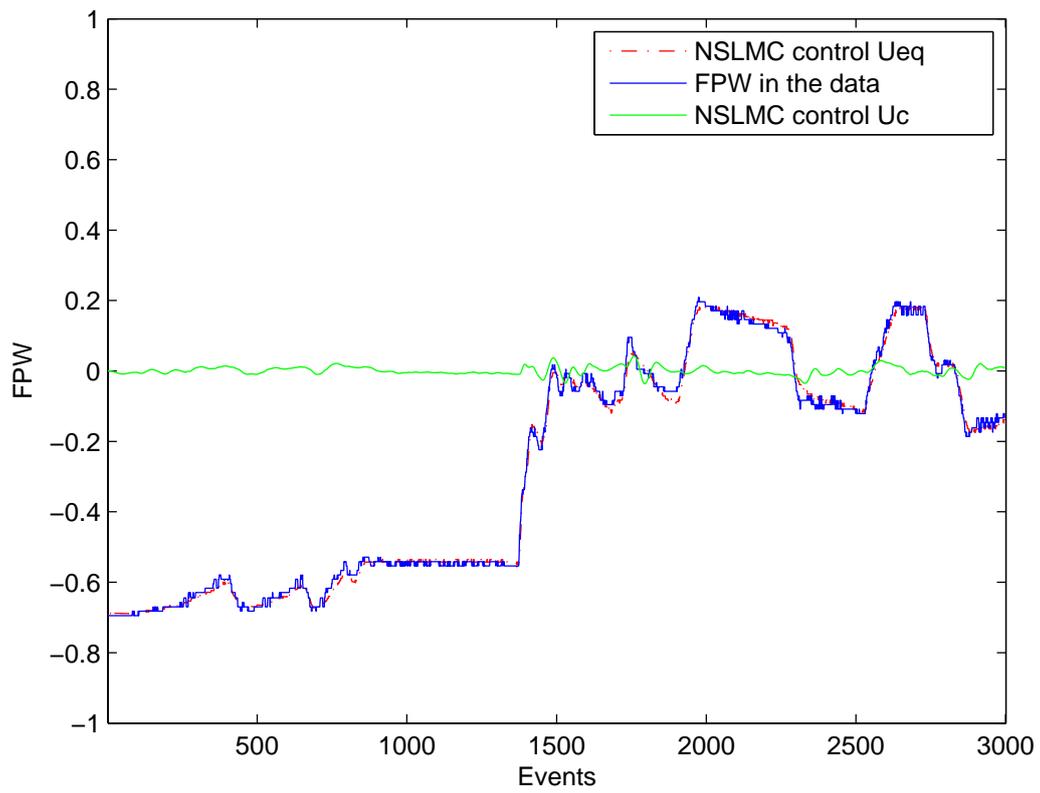


Figure 4.30: Output of equivalent control, corrective control and FPW in the data.

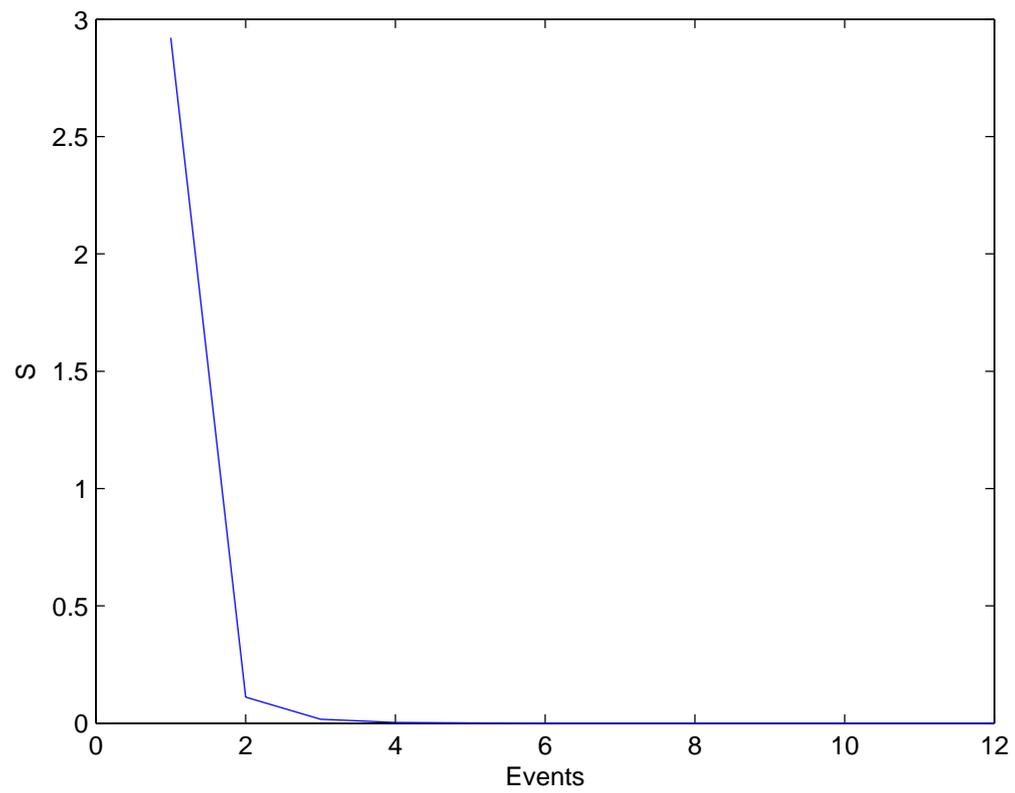


Figure 4.31: Trajectory of sliding function S .

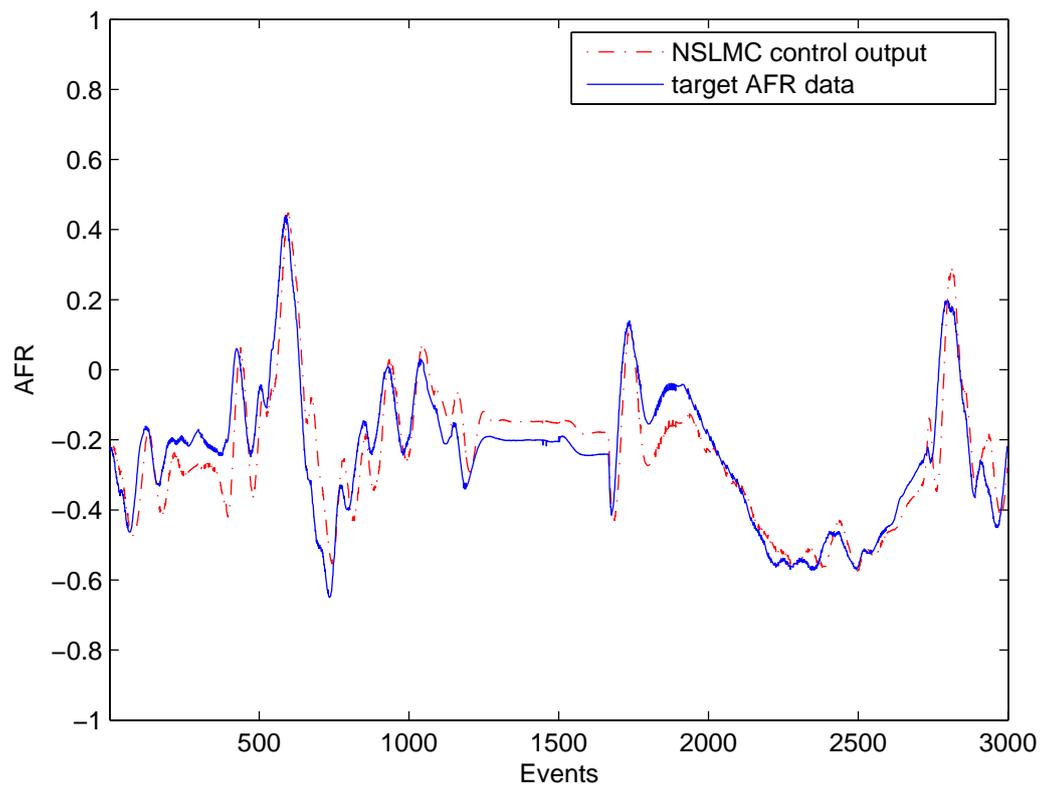


Figure 4.32: NSLMC generalization ability for AFR control.

network controller to track the desired TRQ and AFR sequences through proper control actions. The performance of TRQ controller is a little better than the performance of AFR controller mainly due to the fact that the process of AFR is far more complex than the TRQ process. Combustion TRQ is mostly governed by process nonlinearities whereas the AFR process is mostly dominated by dynamic effects.

Fig. 4.27 and Fig. 4.31 show the trajectory of the sliding function S . For a sliding mode-based control system, the dynamic behavior is determined by the sliding surface when the system is in the sliding mode. The sliding surface is chosen as $S(e) = c^T e + d^T \dot{e}$ which its value reflects the difference between the actual and the targeted trajectory. The sliding surface chosen in this way will lead to control objectives of TRQ which follows TRQ^* and AFR which follows AFR^* . When the sliding curve is near zero, it means that TRQ follows TRQ^* and AFR tracks AFR^* .

The adoption of L-M optimization method instead of BP for updating the weights of neural network speed up the convergence greatly. It took only 15 steps (about 1 minute) to achieve very good results for TRQ control and 20-30 steps (around 5 minutes) for AFR control (The configuration of computer is Pentium 4 3.2G with 1G RAM). Compared to the training technique used in [127] which took one hour for TRQ control, the training speed is a lot quicker and there's no oscillation during the training [59]. The parameters of neural network for the corrective control are set by trial and error in [40] and [127] which is a time-consuming task, and they are selected as large values to achieve fast convergence. In our scheme, the parameters are selected as 1 or 2 which is enough to guarantee convergence. Since the controlled systems are modelled using neural network, the way of computation of K here is different from the method in [40] and [127] where the computation of the gradient of K is acquired from the integral of $G(s)$. The integral of $G(s)$ would bring unpredictably big values in real experiments, resulting in potential instability. Boundary is set to assure stability in [40] and [127].

4.6 Conclusions

Our research results have demonstrated that NSLMC provides a powerful alternative approach for engine calibration and control. The design is the combination of neural networks and sliding mode control. Two parallel neural networks are utilized to realize the equivalent control and the corrective control of SLMC. The successful application of NSLMC to TRQ and AFR control enables us to formulate the following two conclusions: 1) The proposed technique will automatically learn the inherent dynamics and nonlinearities of the engine from real vehicle data and therefore no prior model and the characteristics of the system are required. 2) Learning and control are carried out simultaneously with very fast convergence speed in the proposed controller, which allows it to be further refined and improved in real-time vehicle operation through continuous learning and adaptation.

The simulation results indicate the proposed NSLMC is effective in achieving tracking control of TRQ and AFR control through neural network learning.

5 BARORECEPTOR REFLEX INSPIRED ADAPTIVE CONTROL STRATEGY FOR ENGINE TORQUE CONTROL

5.1 Introduction

Biological processes have evolved sophisticated mechanisms for solving difficult control problems with precise control abilities, robust stability and strong fault tolerance. By analyzing and understanding these natural systems, it is possible that principles can be derived which are applicable to general control systems. The methodologies based on the inspiration of some biological mechanism include: artificial neural network, gene algorithm, artificial immune system, etc. The artificial neural network, gene algorithm, and artificial immune system have been researched for quite a long time by many researchers and are being widely employed ([22], [23], [34], [51], [54], [91]). However, due to the high complexity and the scarce cognition of biological prototype, there are still many open problems under consideration, such as the processing mechanism of the each part of the brain. Even so, it is believed that control techniques derived from biological systems offer great potential for solving complex and nonlinear problems in the modern field of control.

The peripheral autonomic nervous system functions in a dynamic balance aiming at homeostasis. It is divided into two nervous systems: the parasympathetic nervous system and the sympathetic nervous system ([126]). The sympathetic system originates from the thoracic and upper lumbar spinal segments, while the parasympathetic system originates from the brain stem and sacral spinal cord. Both systems are efferent and consist of a chain of preganglionic and postganglionic neurons, which are synaptically connected in peripheral autonomic ganglia. Sympathetic ganglia are situated remote from the target organs and organized bilaterally in the sympathetic chains and in the prevertebral ganglia. Parasympathetic ganglia are situated close to the target organs. Stimulation of sympathetic neurons produces many distinct effector responses elicited from a variety

of cell and tissue types including blood vessels, heart, exocrine epithelia, etc., while stimulation of parasympathetic neurons leads to activation of most exocrine glands, pacemaker and atria of the heart, and some other target cells ([64]).

The baroreflex is responsible for short-term regulation of arterial blood pressure ([117]) though recent research in [125] points out that baroreceptors are also important for the regulation of long-term mean arterial blood pressure. Baroreceptor reflex provides a negative feedback loop in which an elevated blood pressure reflexively causes blood pressure to decrease; similarly, decreased blood pressure depresses the baroreflex, causing blood pressure to rise.

Regulation of arterial blood pressure by baroreceptor reflex would be regarded as tracking a desired curve or value. When blood pressure rises, the carotid and aortic sinuses are distended, resulting in stretch and therefore activation of the baroreceptors. Active baroreceptors fire action potentials that are relayed to the nucleus of the tractus solitarius (NTS), which uses frequency as a measure of blood pressure. The increased activation of the NTS inhibits the vasomotor center and stimulates the vagal nuclei. The end result of baroreceptor activation is inhibition of the sympathetic nervous system and activation of the parasympathetic nervous system. Sympathetic inhibition leads to a reduction of total peripheral resistance and cardiac output via increased contractility of the heart, heart rate, and arterial vasoconstriction, which tends to decrease blood pressure. At the same time, parasympathetic activation leads to a decreased cardiac output via decrease in contractility and heart rate, resulting in a tendency to decrease blood pressure.

By coupling sympathetic inhibition and parasympathetic activation, the baroreflex maximizes blood pressure reduction. Sympathetic inhibition leads to a drop in peripheral resistance, while parasympathetic activation leads to a depressed heart rate and contractility. The combined effects will dramatically decrease blood pressure to the normal level. Similarly, sympathetic activation

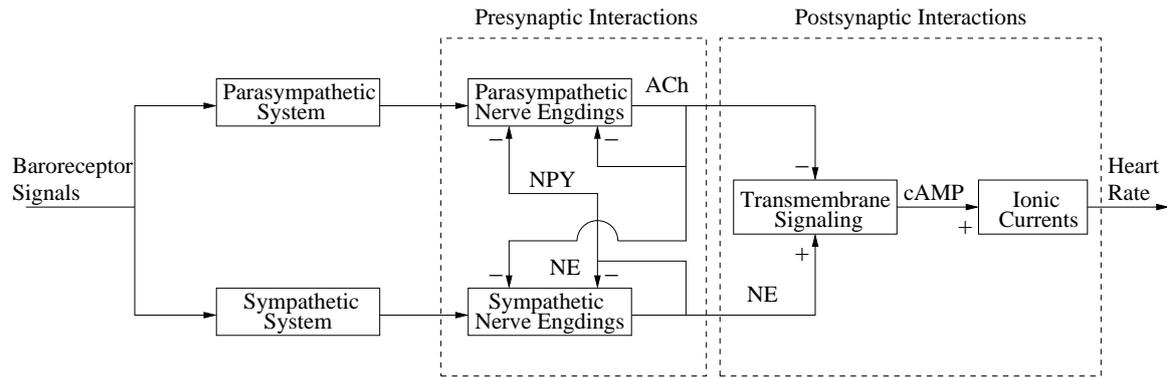


Figure 5.33: Parasympathetic and sympathetic manipulation of heart rate

with parasympathetic inhibition allows the baroreflex to elevate blood pressure to the desired level.

A particularly distinctive feature of the baroreflex is the interactions that occur between the parasympathetic and sympathetic systems at the level of the cardiac pacemaker cells [80], [81]. A simplified representation is shown in Figure 5.33. The two controllers adjust heart rate by releasing neurotransmitters from nerve fibers. Acetylcholine (ACh) is released from parasympathetic nerve endings, while norepinephrine (NE) and neuropeptide Y (NPY) are released from sympathetic nerve endings. ACh and NE bind to receptors on the surface of cardiac pacemaker cells, thereby altering the production of second messenger cyclic AMP (cAMP). cAMP affects heart rate by modulating ionic currents in pacemaker cells. Note that the overall effects of the sympathetic and parasympathetic systems on heart rate are facilitatory and inhibitory, respectively.

Complex interactions occur before and after the synapses between the nerve endings and the cardiac cells. Presynaptically, ACh inhibits the release of NE from sympathetic nerves, while NPY inhibits the release of ACh from parasympathetic nerves. Postsynaptically, ACh inhibits the production of cAMP and therefore diminishes the heart rate response to a given level of NE. In addition, parasympathetic nerves have ACh autoreceptors which subserve a negative feedback function; analogous NE autoreceptors are present in sympathetic nerves. This inhibition mechanism

depicted in Figure 5.33 is present in other biological control systems, including the vestibulo-ocular reflex (VOR) [118] and the auditory system [100].

The regulation by reciprocal lateral inhibition of sympathetic and parasympathetic nervous systems exhibit the following interesting control characteristics:

- (1) The sympathetic and parasympathetic nervous systems have opposite effects.
- (2) The effects of the parasympathetic nervous system are considerably faster than those of the sympathetic nervous system.
- (3) The parasympathetic nervous system is employed only for dynamic control, while the sympathetic nervous system is used primarily for steady-state control.
- (4) The effects of the two nervous systems do not sum linearly due to complex interactions which occur in both the autonomic nervous system and the heart itself.

This body's homeostatic mechanisms and characteristics for maintaining blood pressure at the desired value showed superior performances in the highly complex cardiovascular system, which proves baroreceptor reflex is an adaptive, nonlinear, multivariable control system. Automotive engine control are known to be highly nonlinear complex systems. Thus, it is potentially suitable to develop an effective control strategy for the control of such nonlinear dynamic processes as TRQ generation processes.

5.2 Adaptive Nonlinear Control Strategy

From the optimal control views, the role of parasympathetic nervous system is to refine the performances of the sympathetic nervous system by discovering optimal corrections to be added to the output of whole neural control system.

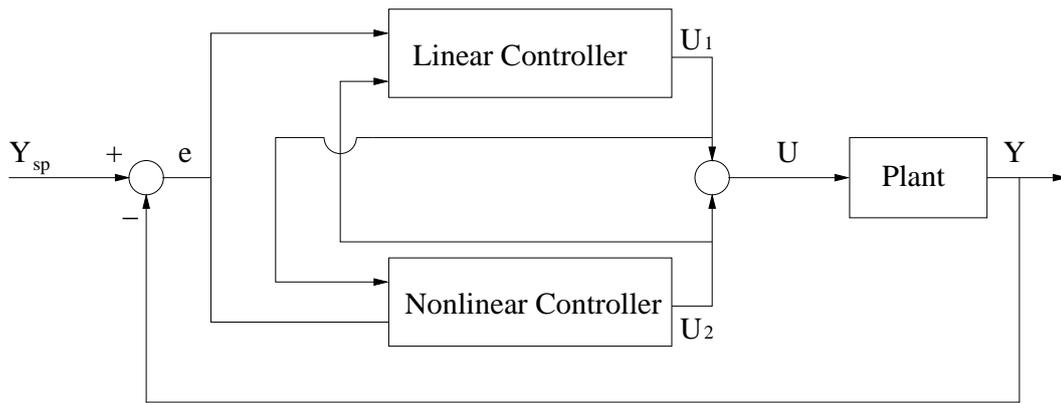


Figure 5.34: Biologically inspired adaptive control system

The proposed adaptive nonlinear control strategy derived from a biological control system consists of a linear controller and a nonlinear controller which interact via a reciprocal lateral inhibitory mechanism as shown in Figure 5.34, where Y is the control output, U is the control command, Y_{sp} is the target. The control system has three components: a linear controller, an adaptive nonlinear controller, and a summation that produces U from the outputs of the two parallel controllers (U_1 , U_2). By analogy to the baroreflex, the linear controller plays role of the sympathetic system, the nonlinear controller plays the role of the parasympathetic system, and the summation represents the postsynaptic interactions. Note that each controller modulates the other controller via reciprocal lateral inhibitory connections.

The control output of the linear controller to the nonlinear controller is regarded as a complex expression of the error of the controlled plant in the way that includes the integral of the errors, the derivative of the errors and the proportional error. These three terms represent the whole history of errors, the nonlinear trend of the change of the error and the current error, respectively. Using error information provided, the nonlinear controller would, in a sense, see the possible situation of the system and know the compensation will be input to the controlled plant by the other controller. In this way, the nonlinear controller learns the characteristics of the plant and adjusts itself according

to the other linear controller. What's more, in the nonlinear operating regions or the situation with disturbance, the linear controller may be inappropriate, and therefore, the nonlinear controller is added to compensate the insufficiency of the linear controller and the dynamic environment such as the parameter variations of the process and disturbances. These situations can be addressed by adjusting the amount of lateral inhibition and learning the characteristics of the controlled system such that desirable controller outputs are produced in particular operating regions.

The linear controller learns the action of the other controller by watching an input from the nonlinear controller. The adjustable parameter attached to this input increases the degree of freedom of the linear fixed controller.

By coupling linear controller and nonlinear controller, the whole controller adjusts the output of the controlled plant. The linear controller leads to a basic control output, while the nonlinear controller refines the performances of the linear controller. The combined effects will dramatically increase the systems learning ability and adjustability to the noisy environment.

5.2.1 Linear Controller Design

The possible linear controller would be a proportional-integral-derivative controller (PID controller). The PID controller is a generic control loop feedback mechanism widely used in industrial control systems. The PID controller attempts to correct the error between a measured process variable and a desired setpoint by calculating and then outputting a corrective action that can adjust the process accordingly.

The PID controller algorithm involves three separate parameters; the proportional, the integral and derivative gains. The proportional gain determines the reaction to the current error, the integral determines the reaction based on the sum of recent errors and the derivative determines the reaction

to the rate at which the error has been changing.

The general form of the PID controller in the time domain is:

$$U_{pid} = K_p e(t) + K_i \int e(t) dt + K_d \dot{e}(t) \quad (5.45)$$

where $e(t)$ is tracking error between the target and the actual output of the controlled plant, K_p is proportional gain, K_i is integral gain and K_d is derivative gain. By tuning the three gains in the PID controller algorithm, the PID controller can provide control action designed for specific process requirements.

The modulatory effect of the nonlinear controller is incorporated by adding an interaction term to the linear PID controller. The form of the PID controller with reciprocal lateral impact is:

$$U_1 = U_{pid} + K_1 U_2 \quad (5.46)$$

where U_2 is the control output of the nonlinear controller, K_1 is a lateral impact weight which determine the amount of the lateral impact from nonlinear controller.

5.2.2 Nonlinear Controller Design

The nonlinear controller is designed without the aid of an explicit dynamic model. The motivation for this approach is that nonlinear models are much more difficult to obtain than linear models.

The controller could be chosen to have the following form:

$$U_2 = f(e, \dot{e}) + K_2 U_1 \quad (5.47)$$

where $f(e, \dot{e})$ represents a neural network which would be chosen according to the specific control goal and K_2 is another lateral impact weight which determine the amount of the lateral impact from linear PID controller.

The control law from equation (5.47) would be represented as:

$$U_2 = y^T(e, \dot{e})\omega + K_2U_1 \quad (5.48)$$

where $y^T(e, \dot{e})$ is the regressor vector and ω is the vector of nonlinear controller parameters. The objective is to adapt the nonlinear parameters (ω) and lateral impact weights (K_1, K_2) such that the system output matches the target.

5.3 Adaptive Control System for Engine Torque Control

The structure of the torque control system is shown in Figure 5.35. The control system consists of three components: a linear controller (PID controller), an adaptive nonlinear controller (neural network controller), and a summation that produces TPS, command signal, from the outputs of the two parallel controllers (U_1, U_2). By analogy to the baroreflex, the linear controller plays role of the sympathetic system, the nonlinear controller plays the role of the parasympathetic system, and the summation represents the postsynaptic interactions. In Figure 5.35, TRQ* is the desired TRQ to be tracked by the parallel biologically inspired controller system, TPS is the command signal which is formed by the sum of U_1 and U_2 . The two controllers interact via reciprocal lateral inhibitory connections, that is, U_1 , the output of PID controller, is one of inputs of neural network controller while U_2 , the output of neural network controller, is incorporated in the PID controller. The idea of coupling controller is the tracking error of the system is asymptotically convergent to zero by linear and nonlinear controller cooperating with each other using the feedback control output of the each controller. When the error of the system is zero, since the input of the PID controller from the NN controller is not zero, the output of the PID controller will not be zero either. The same rule applies to the NN controller. In this way, the system is controlled to track the desired TRQ in the normal situation. When there is disturbance in the system, the PID controller

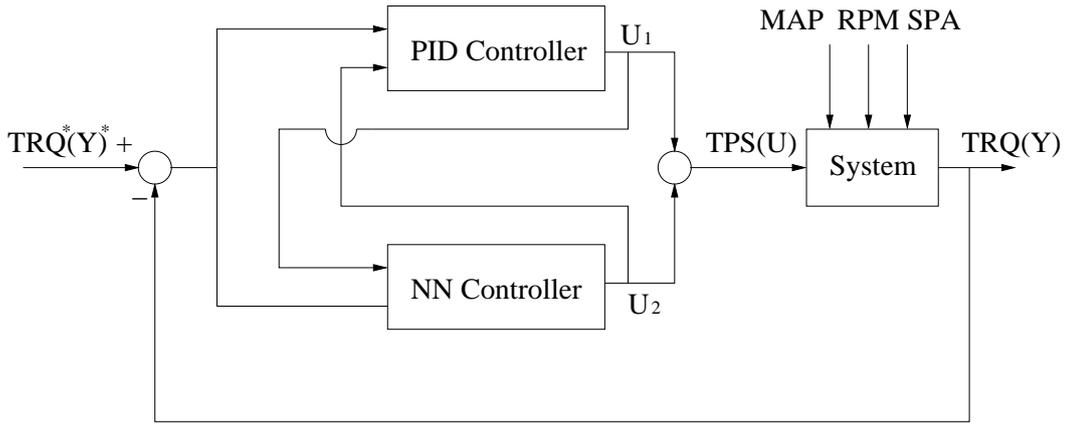


Figure 5.35: The adaptive nonlinear control system

and NN controller will cooperate with each other to pull the disturbed system back to the normal situation.

The linear PID controller could take the following form:

$$U_1 = K_p e(t) + K_i \int e(t) dt + K_d \dot{e}(t) + K_1 U_2 \quad (5.49)$$

where $e(t) = Y^*(t) - Y(t)$ ($TRQ^*(t) - TRQ(t)$) and K_p , K_i , K_d , K_1 are proportional gain, integral gain, derivative gain and lateral impact weight respectively. The lateral impact weight K_1 is tuned by adaptive learning algorithm.

The nonlinear controller would have the following form:

$$U_2 = f(e, \dot{e}, U_1) \quad (5.50)$$

where $f(e, \dot{e}, U_1)$ represents a feedforward neural network with inputs: the error, the derivative of the error and the control output of the PID controller. Here, the control output of the PID controller is incorporated into the neural network with more parameters to be tuned which have the same effect as equation (5.47).

The weight adaption of the adaptive nonlinear control strategy is based on a minimization of

the cost function as follows:

$$E = \frac{1}{2}(Y^* - Y)^2 = \frac{1}{2}\zeta^2 \quad (5.51)$$

where $\zeta = Y^* - Y$. The Levenberg-Marquardt algorithm is used to update the weights of feed-forward neural network instead of the backpropagation algorithm which is a steepest descent algorithm.

The formula for updating neural network weights is given as follows:

$$\Delta W = [J^T(W)J(W) + \mu I]^{-1} J^T(W)\zeta \quad (5.52)$$

where the parameter μ is adjustable and $J(W)$ is the Jacobian matrix. μ is multiplied by some factor β whenever a step would result in an increased E . When a step reduces E , μ is divided by β . By adjusting μ in this way, the search direction interpolates between the gradient and the Gaussian-Newton direction. That is the reason that the rate of convergence is satisfactory. Jacobian matrix $J(W)$ can be expressed as follows:

$$J(W) = \left[\frac{\partial \zeta}{\partial W_1} \quad \frac{\partial \zeta}{\partial W_2} \quad \cdots \quad \frac{\partial \zeta}{\partial W_n} \right]^T. \quad (5.53)$$

From equation (5.51),

$$\frac{\partial \zeta}{\partial W_i} = \frac{\partial(Y^* - Y)}{\partial W_i} = -\frac{\partial Y}{\partial U} \frac{\partial U_2}{\partial W_i}. \quad (5.54)$$

where $U = U_1 + U_2$. $\frac{\partial U_2}{\partial W_i}$ could be easily calculated using the standard backpropagation algorithm.

Thus, the Jacobian matrix can be computed by (5.54).

The formula for updating lateral weight K_1 to minimize E in the direction of the negative gradient

is given as follows :

$$\begin{aligned}
 \Delta K_1 &= -\mu \frac{\partial E}{\partial K_1} \\
 &= \mu e \frac{\partial Y}{\partial K_1} \\
 &= \mu e \frac{\partial Y}{\partial U} \frac{\partial U}{\partial K_1} \\
 &= \mu e \frac{\partial Y}{\partial U} \frac{\partial U}{\partial U_1} \frac{\partial U_1}{\partial K_1} \\
 &= \mu e \frac{\partial Y}{\partial U} \frac{\partial U_1}{\partial K_1}
 \end{aligned} \tag{5.55}$$

where μ is the learning rate.

From (5.46),

$$\frac{\partial U_1}{\partial K_1} = U_2 \tag{5.56}$$

That is, the change of lateral impact weight is:

$$\Delta K_1 = \mu e \frac{\partial Y}{\partial U} U_2 \tag{5.57}$$

Since the module of the engine TRQ is represented by a neural network, according to the backpropagation algorithm, $\frac{\partial Y}{\partial U}$ could be derived easily for updating weights of the algorithm.

5.4 Simulation Results

The objective of the present engine controller is to provide control signals so that torque generated by the engine will track the torque measurement as in the data. The measured values in the data are obtained using the commercial engine controller under warmup conditions. Based on the data collected we use the biological adaptive controller to generate control signal TPS with the goal of producing exactly the same torque as in the data set. The performance is defined as variations of

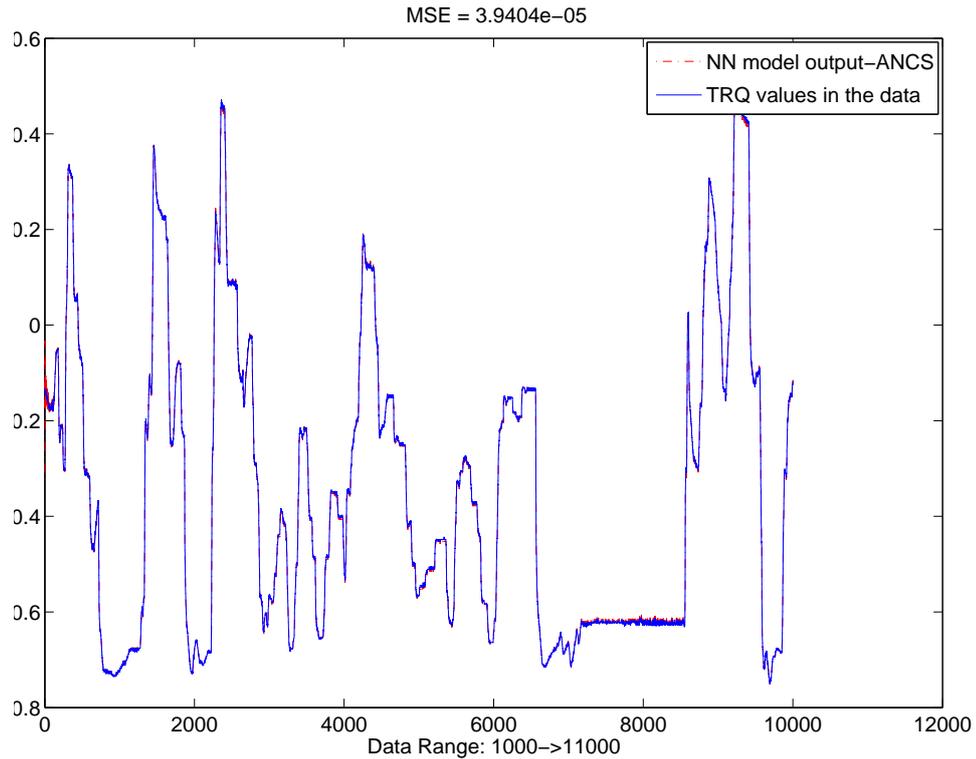


Figure 5.36: ANCS control for TRQ

the torque generated from the measured values in the data set. The engine torque model is as the one in the Chapter 2.

The data range for the training part of the control algorithm on the TRQ control is from 1000 to 11000 from the data set without any special purpose for the selection. Figure 5.36 is the output (TRQ) of the engine using adaptive nonlinear control strategy compared to the TRQ data in the data set. From the figure 5.36, we can see that the TRQ controlled by the adaptive nonlinear controller tracks the TRQ measurement in the data set very well. Figure 5.37 is the output of the PID controller and neural network controller compared to the TPS in the data. Figure 5.38 is the combined control to the engine compared to TPS in the data. From the figure 5.38, we can see that the combined control output of controllers is very close to the TPS measurement in the data set.

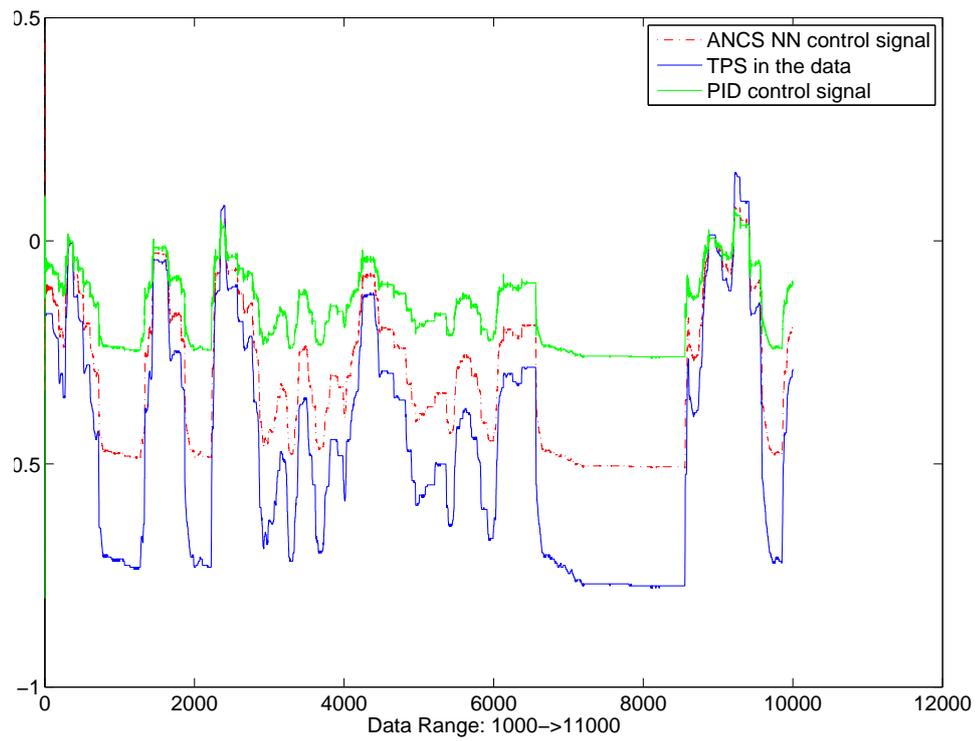


Figure 5.37: The output of PID control, NN control, and TPS in the data

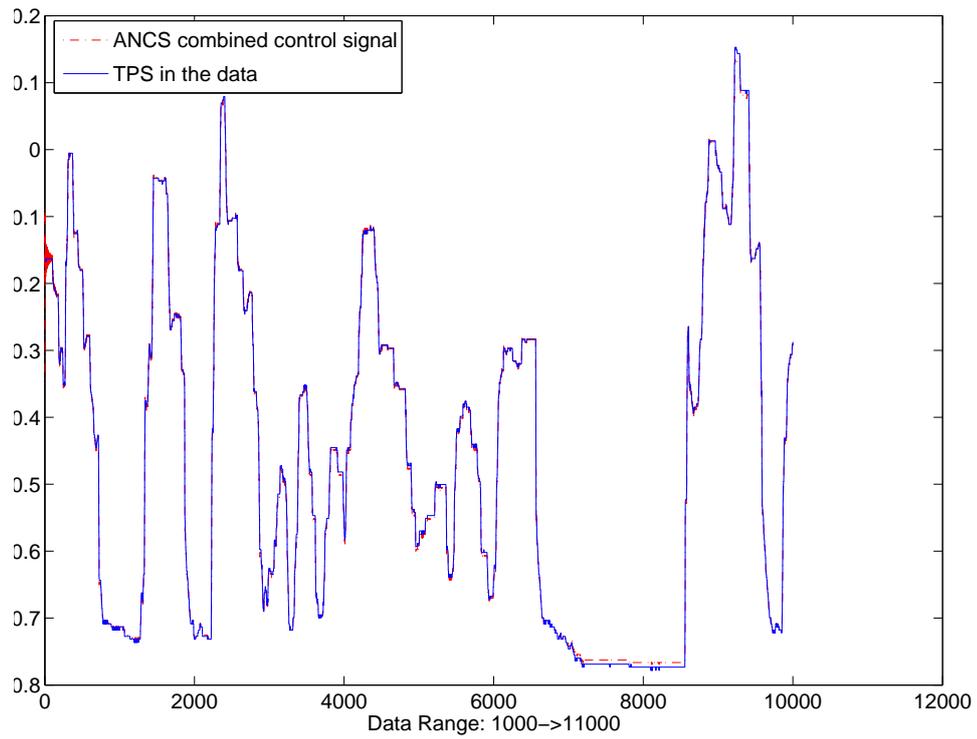


Figure 5.38: The combined output of PID control and NN control and TPS in the data

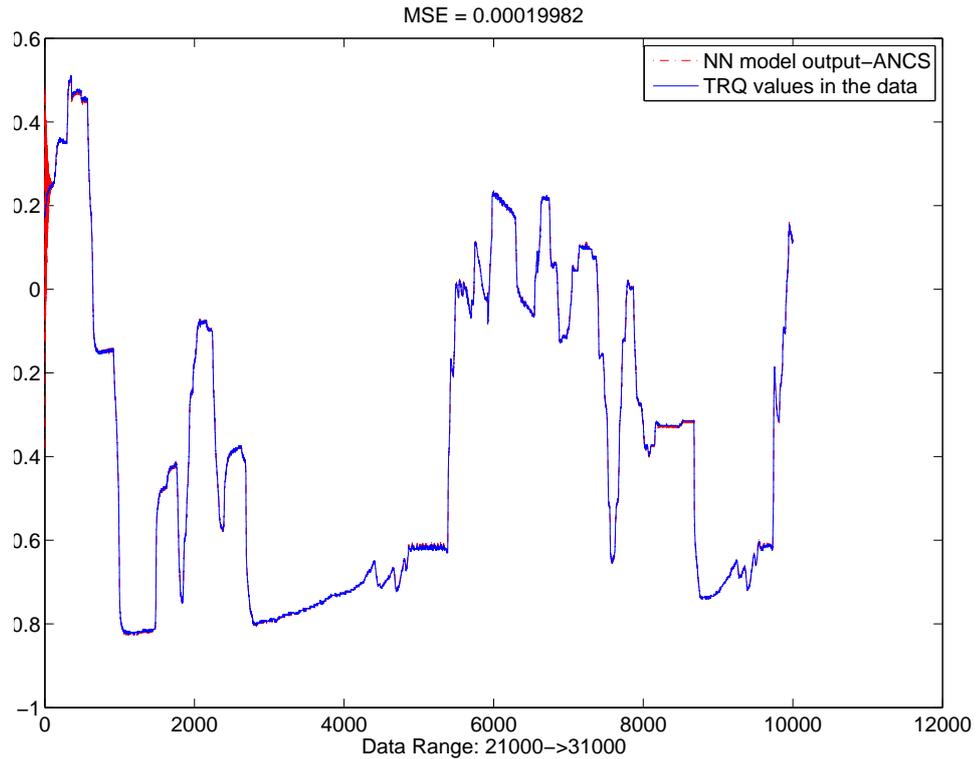


Figure 5.39: ANCS generalization control effect

From figure 5.37 and 5.38, we can see that the neural network and PID controller interact with each other all the time. When there is disturbance in the system, the PID controller and neural network controller cooperate with each other to pull the disturbed system back to the normal situation.

The data range for the generalization part of the control algorithm on the TRQ control is from 21000 to 31000 from the data set without any special purpose for the selection. Figure 5.39, Figure 5.40 and Figure 5.41 are the results from which we can say that the control effect is pretty good. We can also see that the neural network controller and PID controller cooperate with each other all the time.

From the Figures, we can tell that the role of neural network controller is to refine the performances of the linear PID controller by discovering optimal corrections to be added to the linear

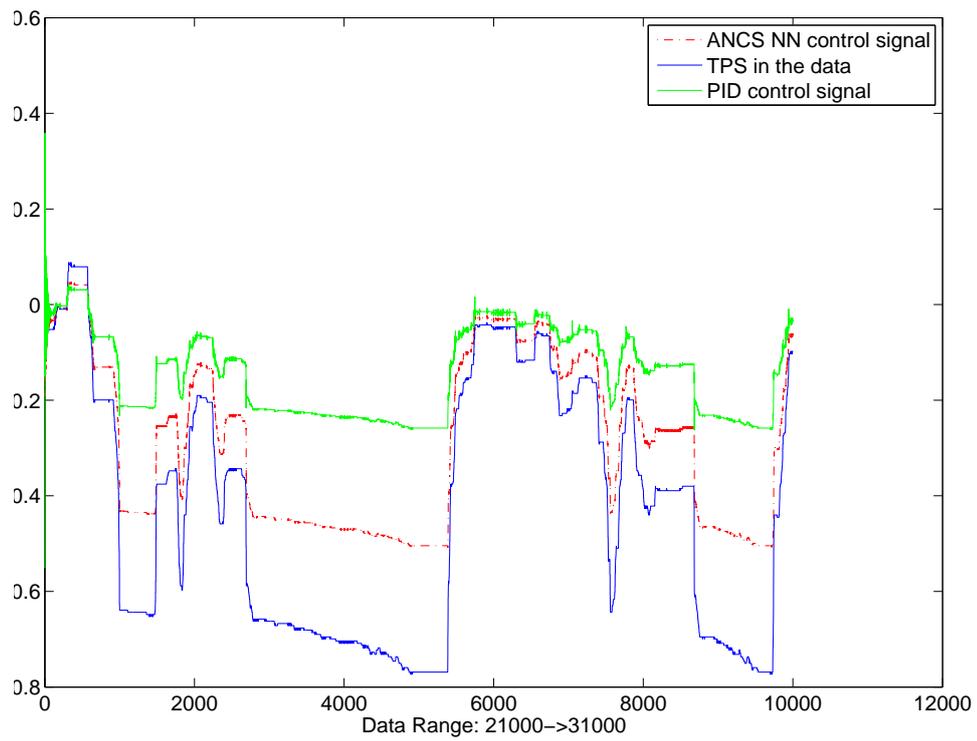


Figure 5.40: The output of PID control, NN control, and TPS in the data

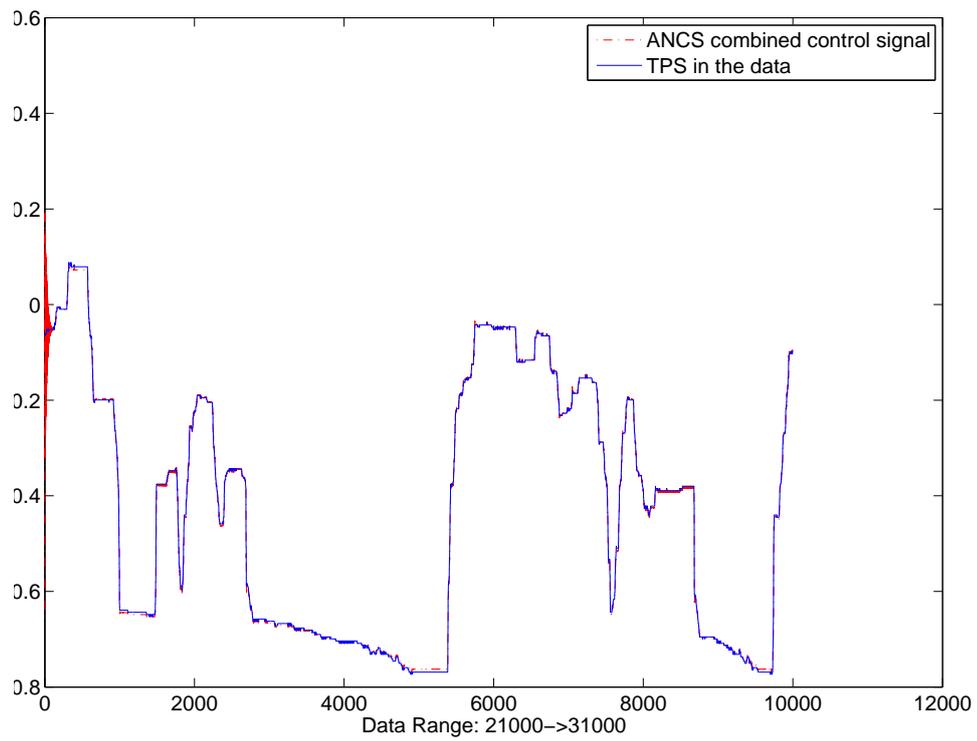


Figure 5.41: The combined output of PID control and NN control and TPS in the data

controllers output, which bears an analogy with the mechanism of the baroreceptor reflex to regulate arterial blood pressure. When there is a deviation of the states of the system, the neural network controller takes actions to pull the systems back to the normal conditions.

Compared to the method of neuro-sliding mode control, the training speed is almost the same, but the tracking effect of biological adaptive controller is better. It took only 15 steps (about 1 minute) to get very good results using Levenberg-Marquardt training algorithm. While using backpropagation (delta training rule), it took around 2 hours steps to make MSE below 0.0004 during the training. Considering the speed, the adaptive nonlinear control strategy can be a good candidate used for the online learning control.

The tuning of the parameters (the gains of the proportional, integral and derivative terms) of PID controller is a very time-consuming task. If the PID controller parameters are chosen incorrectly, the controlled process input can be unstable, i.e. its output diverges, with or without oscillation, and is limited only by saturation or mechanical breakage. In our control strategy, the parameters are selected as $K_p = 1.6$, $K_i = 0$ and $K_d = -0.48$ without any special purpose for the selection. The other random selections of PID parameters are $K_p = 0.6$, $K_i = 0.12$ and $K_d = 0.18$. We have also got very good results. We do not have to choose the parameters of PID controller by trials and errors which is a very time-consuming task. In fact, the corrections produced by the neural controller will compensate the insufficiency of the PID controller, plant-model mismatches and disturbances from the environments.

5.5 Conclusions

Our research results show that the method of adaptive nonlinear control strategy is a good approach for engine control. The parallel controllers inspired by the functions of baroreceptor reflex consists

of a linear controller and a nonlinear controller that interact via a reciprocal lateral inhibitory mechanism. The linear controller design is based on a PID controller, while the nonlinear controller is constructed from neural network that are updated online. The nonlinear controller is designed without the aid of an explicit dynamic model. In the linear operating region, the PID controller takes control. If the controlled process is far away from linear regime or is disturbed by the noises, the output of linear controller may be inappropriate, and therefore, the nonlinear controller is added to compensate the insufficiency of the linear controller and the dynamic environment such as the parameter variations of the process and disturbance. These situations can be addressed by adjusting the amount of lateral inhibition and learning the characteristics of the controlled system such that desirable controller outputs are produced in particular operating regions. From these two controllers, the objective of tracking the desired TRQ could be realized.

The following advantages are observed in the experiments:

- (1) Computational efficiency, suitable for real-time implementation.
- (2) Real-time adaptability, insensitive to the changing characteristics of the controlled plant or the noisy environment.
- (3) Superior learning ability, even in the situation of a very poor linear controller
- (4) No prior knowledge of the controlled plant needed
- (5) Control and learning are done simultaneously.
- (6) No need to tune the parameters of PID controller by trails and errors which is a very time-consuming task.

6 A SELF-LEARNING SCHEME FOR RESIDENTIAL ENERGY SYSTEM CONTROL AND MANAGEMENT

6.1 Introduction

Over the last decade, the human beings have become more and more dependent on the electricity for their daily life. The rising cost, the environmental concerns, and the reliability issues all underlie the needs and the opportunities for developing new intelligent control and management system of residential hybrid energy usage. There has been considerable discussion of the importance of distributed energy storage, including batteries in the home, as a way to create more price-responsive demand and as a way to integrate more renewable energy resources more effectively into power grids. It is envisioned that distributed energy storage technologies could reduce the combustion of fossil fuel, supply reliable energy in concert with other energy sources and financially benefit residential customers.

The development of an intelligent power grid, i.e., the smart grid, has attracted significant amount of attention recently. Considerable research and development activities have been carried out in both industry and academia [5], [45], [82], [112]. Along with the development of smart grid, more and more intelligence has been required in the design of the residential energy management system. Smart residential energy management system provides end users the optimal management of energy usage by means of robust communication capability, smart metering and advanced optimization technology.

There has been a growing interest in the development and application of renewable energy sources for residential households in recent years. Renewable energy sources like solar cells and wind have many advantages over nonrenewable energy sources like oil and coal. They are nonde-

pleting, modular, and nonpolluting. Although renewable energy source provide attainability, they depend on nature and, therefore, are intermittent. For example, PV panels provide electricity only when the sun is available and wind systems generate electricity only when there is enough wind. To solve this problem, energy storage systems are installed to store energy from renewable energies. Energy storage systems bring an additional degree of flexibility to power systems, providing a means of optimizing the way energy is used by decoupling energy production from utilization. In many of these systems, energy storage has been considered as a means of optimizing the overall use of energy, with the hope of providing additional value to the system, and limiting the dependence on the power grid. As [69] point out that the system can be handled cost-efficiently if storage is incorporated into the electrical network.

Time-of-use pricing (or residential real-time pricing as the utility company ComEd calls it) is one of the load management policies used to shift electricity use from peak load hours to light load hours in order to improve power system efficiency and allow new power system construction projects [76]. For the residential families, in general, there is a distinction between night and day consumption for which different electricity prices apply. Due to the variable and intermittent nature of renewable energy outputs, hybrid energy system will only function as grid-connected battery energy system in some peak hours with the absence of renewable energy output. The battery that stores large amounts of energy during low rate hours could discharge during periods of high demand with full electricity rate. Therefore, it will financially benefit the residential customers. The short run value of this scheme would be its ability to off-load peaking units that require more pricey electricity. The long run value would be its ability to reduce the electricity cost for residential households.

The main focus of this chapter is on proposing a computationally feasible and self-learning optimization-based optimal operating control scheme for the residential energy system with bat-

teries. We aim to minimize the total operating cost over the scheduling period in a residential household by optimally scheduling the operation of batteries, while satisfying a set of constraints imposed by the requirements on the system and the capacities of individual components of the system. We consider the application of adaptive dynamic programming to the self-learning operation control and management of the grid-connected energy system in residential households. ADP is the method which approximate dynamic programming for solving problems of optimization over time, in the general case of nonlinearity with random disturbance.

The advantages of the optimal energy controller based on ADP are reflected in many aspects. First, the controller provides optimal control signals under various operating conditions. Such a neural network controller will be obtained after a specially designed learning process that performs approximate dynamic programming. Once a controller is learned and obtained (offline or online), it will be applied to perform the task of energy optimization control. Second, the performance of the controller can be further refined and improved through continuous learning in real-time operations. One of the key promising attributes of the present approach is the ability of the continuous learning and adaptation to improve controllers performance during real-time operations under uncertain changes in the environment of the residential household. Last, but not least, the optimal controller can be adapted to different scenarios of different residential customers. Traditional fixed or rule-based control strategy is generally based on the expert knowledge according to the state-of-the-art. As a result, the same control strategy is applied for all system configurations. However, this procedure cannot ensure an optimum system design because it ignores the differences among the different system configurations. With continuous learning and adaptation for residential household energy system, the controller based on ADP will obtain the optimal control strategy according to the system configuration and energy utilization of the residential customer.

The high-level objective of this work is to understand the economics and the security of hybrid

energy system for the residential households. The specific goal addressed in this chapter is to help reduce the operational cost and ensure the reliability of the supply for the residential household. This paper is organized as follows. In Section 2, the residential energy system used in the paper is briefly described. The control and management problem of residential energy system is formulated. In Section 3, the ADP scheme that is suitable for the application to the residential energy system control and management problem is introduced. In Section 4, our self-learning control algorithm for grid-connected energy system in residential households is developed. The present work will assume the use of artificial neural networks as a means for function approximation in the implementation of ADP. In particular, multilayer feedforward neural networks are considered, even though other types of neural networks are also applicable in this case. In Section 5, the performance of our algorithm is studied through simulations. The simulation results indicate that the proposed self-learning algorithm is effective in achieving the optimal cost. Finally, in Section 6, the paper will be concluded with a few remarks.

6.2 Description of the Residential Energy System

The objective of this paper is to apply ADP intelligent optimization method to the challenge of intelligent price-responsive management of residential energy use. Specifically, it is to minimize the sum of system operational cost over the scheduling period, subject to technological and operational constraints of grids and storage resource generators and subject to the system constraints such as power balance and reliability. For this purpose, we focus our research on finding the optimal battery charge/discharge strategy of the residential energy system with batteries and power grids configuration.

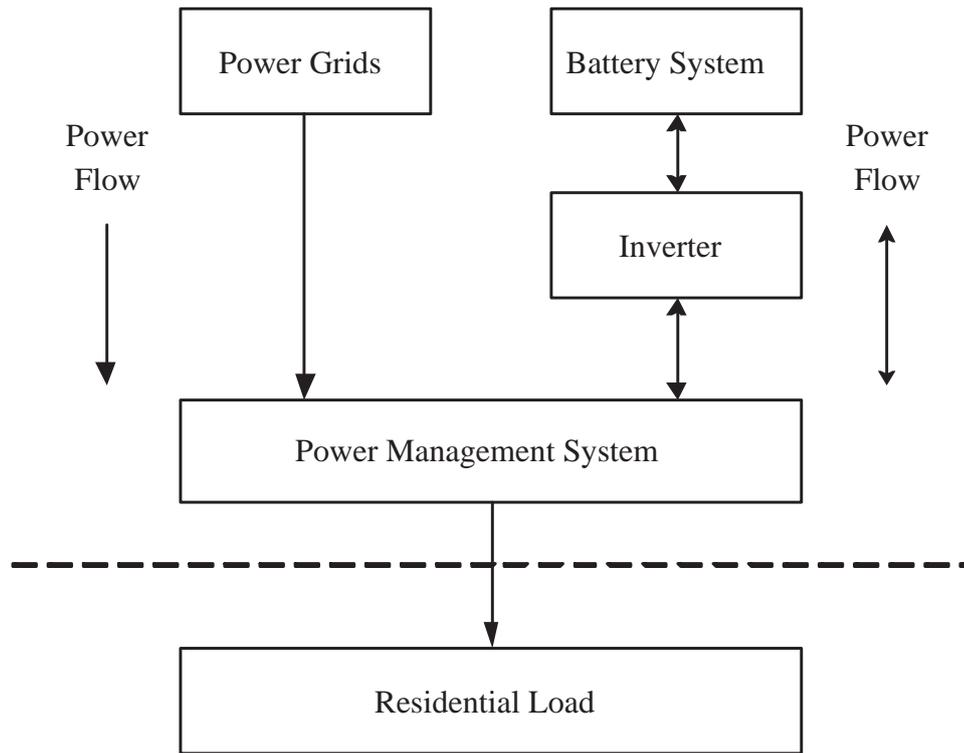


Figure 6.42: Grid-connected residential energy system with battery storage

6.2.1 Residential Energy System

The residential energy system uses AC utility grid as the primary source of electricity and is intended to operate in parallel with the battery storage system. Fig. 6.42 depicts the schematic diagram of a residential energy system. The system consists of power grids, a sinewave inverter, a battery system and a power management unit. The battery storage system is connected to power management system through an inverter. The inverter functions as both charger and discharger for the battery. The construction of the inverter is based upon power MOSFET technology and pulse-width-modulation technique [43]. The quality of the inverter output is comparable to that delivered from the power grids. The battery storage system consists of lead acid batteries, which are the most commonly used rechargeable battery type. The optimum battery size for a particular residential household can be obtained by performing various test scenarios, which is beyond the scope of the

present paper. Generally, the battery is sized to enable it to supply power to the residential load for a period of twelve hours.

There are three operational modes for the residential energy system under consideration.

- (1) Charging mode: when system load is low and the electricity price is inexpensive, the power grids will supply the residential load directly and, at the same time, charge the batteries.
- (2) Idle mode: the power grids will directly supply the residential load at certain hours when, from the economical point of view, it is more cost effective to use the fully-charged batteries in the evening peak hours.
- (3) Discharging mode: by taking the subsequent load demands and time-varying electricity rate into account, batteries alone supplies the residential load at hours when the cost of grid power is high.

This system can easily be expanded; i.e., other power sources along with the power grid and batteries like PV panels or wind generators can be integrated into the system when they are available.

6.2.2 Load Profile

For this study, the optimal scheduling problem is treated as a discrete-time problem with the time step as one-hour and it is assumed that the residential load over each hourly time step is varying with noise. Thus, the daily load profile is divided into twenty-four hour period to represent each hour of the day. Each day can be divided into a greater number of periods to have higher resolution. However, for simplicity and agreement with existing literature [15], [25], [30], [89], we use a twenty-four hour period each day in this work. A typical weekday load profile is shown in Fig.

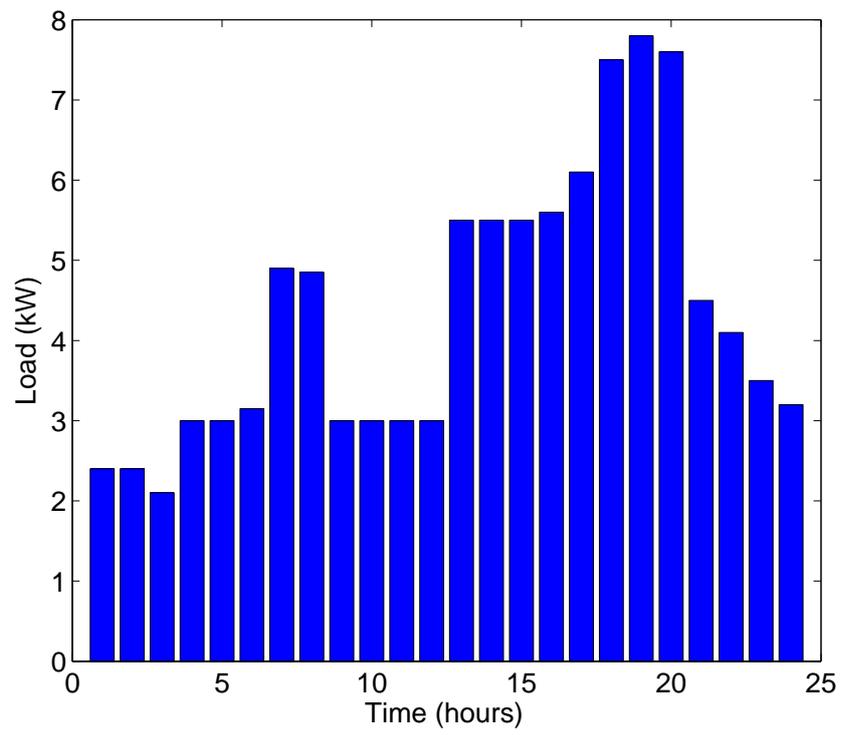


Figure 6.43: A typical residential load profile

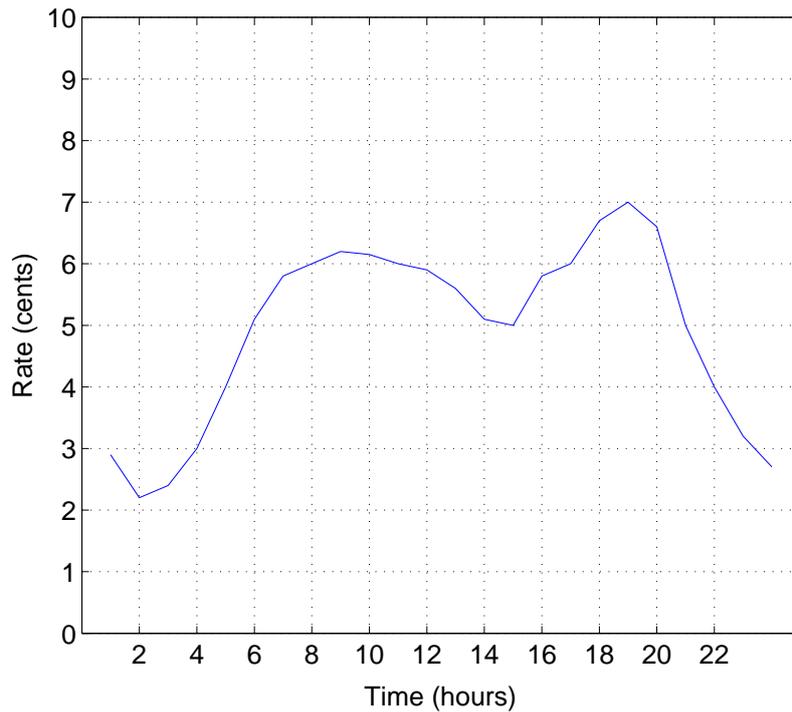


Figure 6.44: A typical daily real-time pricing

6.43. The load factor P_L is expressed as $P_L(t)$ during hour t ($t = 1, 2, \dots, 24$). For instance, at time $t = 19$, the load is 7.8kW which would require 7.8kWh of energy. Since the load profile is divided into one hour steps, the units of the power of energy sources can be represented equally by kW or kWh.

6.2.3 Real-Time Pricing

Residential real-time pricing is one of the load management policies used to shift electricity useage from peak load hours to light load hours in order to improve power system efficiency and allow new power system construction projects [76]. With real-time pricing, the electricity rate varies from hour to hour based on wholesale market prices. Hourly, market-based electricity prices typically change as the demand for electricity changes; higher demand usually means higher hourly prices.

In general, there tends to be a small price spike in the morning and another slightly larger spike in the evening when the corresponding demand is high. Fig. 6.44 demonstrates a typical daily real-time pricing from [141]. The varying electricity rate is expressed as $C(t)$, the energy cost during the hour t in cents. For the residential customer with real-time pricing, energy charges are functions of the time of electricity use. Therefore, for the situation where batteries are charged during the low rate hours and discharged during high rate hours, one may expect, from an economical point of view, the profits will be made by storing energy during low rate hours and releasing it during the high rate hours. In this way, the battery storage system can be used to reduce the total electricity cost for residential household.

6.2.4 Battery Model

The energy stored in a battery can be expressed as [76], [139]:

$$E_b(t) = E_{b0} - \sum_{i=0}^t P_b(i) \quad (6.58)$$

$$P_b(i) = V_0 I \alpha(i) \quad (6.59)$$

$$\alpha(i) = \begin{cases} 1 & (i \leq i_0) \\ K_1(I) dV_0/di & (i > i_0) \end{cases} \quad (6.60)$$

$$V_0 = V_s - (K_c(Q/(Q - Ji) + N)J + A \exp(-BQ^{-1}Ji)) \quad (6.61)$$

where $E_b(t)$ is the battery energy at time t , E_{b0} is the peak energy level when the battery is fully charged (capacity of the battery), $P_b(i)$ is the battery power output at time i , V_0 is the terminal

voltage of the battery, I is the battery discharge current, $\alpha(i)$ is the current weight factor as a function of discharge time, i_0 is the battery manufacturer specified length of time for constant power output under constant discharge current rate, $K_1(I)$ is the weight factor as a function of the magnitude of the current, V_s is the battery internal voltage, K_c is the polarization coefficient (ohm cm^2), Q is the available amount of active material (coulombs per cm^2), J is the apparent current density (amperes per cm^2), N is the internal resistance per cm^2 , and A and B are constants.

Apart from the battery itself, the loss of other equipments such as inverters, transformers, and transmission lines should also be considered in the battery model. The efficiency of these devices was derived in [139] as:

$$\eta(P_b(t)) = 0.898 - 0.173 |P_b(t)|/P_{\text{rate}}, \quad P_{\text{rate}} > 0 \quad (6.62)$$

where P_{rate} is the rated power output of the battery, $\eta(P_b(t))$ is the total efficiency of all the auxiliary equipments in the battery system.

Assume that all the loss caused by these equipments occur during the charging period. The battery model used in this work is expressed as follows: when the battery is charged

$$E_b(t+1) = E_b(t) - P_b(t+1) \times \eta(P_b(t+1)), \quad P_b(t+1) < 0 \quad (6.63)$$

and when the battery discharges

$$E_b(t+1) = E_b(t) - P_b(t+1), \quad P_b(t+1) > 0. \quad (6.64)$$

In general, to improve battery efficiency and extend the battery's lifetime as far as possible, two constraints need to be considered:

- (1) Battery has storage limit. A battery lifetime may be reduced if it operated at lower amount of charge. In order to avoid damage, the energy stored in the battery must always meet

constraint as follows:

$$E_b^{\min} \leq E_b(t) \leq E_b^{\max}. \quad (6.65)$$

- (2) For safety, battery cannot be charged or discharged at rate exceeding the maximum and minimum values to prevent damage. This constraint represents the upper and lower limit for the hourly charging and discharging power. A negative $P_b(t)$ means that the battery is being charged, while a positive $P_b(t)$ means the battery is discharging,

$$P_b^{\min} \leq P_b(t) \leq P_b^{\max}. \quad (6.66)$$

6.2.5 Load Balance

At any time, the sum of the power from the power grids and the batteries must be equal to the demand of residential user

$$P_L(t) = P_b(t) + P_g(t), \quad (6.67)$$

where $P_g(t)$ is the power from the power grids, $P_b(t)$ can be positive (in the case of batteries discharging) or negative (batteries charging) or zero (idle). It explains the fact that the power generation (power grids and batteries) must balance the load demand for each hour in the scheduling period. We assume here that the supply from power grids is enough for the residential demand.

6.2.6 Optimization Objectives

The objective of the optimization policy is, given the residential load profile and real-time pricing, to find the optimal battery charge/discharge/idle schedule at each time step which minimize the total cost

$$C_T = \sum_{t=1}^T C(t) \times P_g(t) \quad (6.68)$$

while satisfying the load balance equation (6.67) and the operational constraints (6.62)–(6.66). C_T represents the operational cost to the residential customer in a period of T hours. To make the best possible use of batteries for the benefit of residential customers, with time of day pricing signals, it is a complex multistage stochastic optimization problem. Adaptive dynamic programming which provides approximate optimal solutions to dynamic programming is applicable to this problem. Using ADP, we will develop a self-learning optimization strategy for residential energy system control and management. During real-time operations under uncertain changes in the environment, the performance of the optimal strategy can be further refined and improved through continuous learning and adaptation.

6.3 Self-Learning Scheme for the Residential Energy System

The learning control architecture for residential energy system control and management is based on ADP. However, only a single module will be used instead of two or three modules in the original scheme. The single critic module technique retains all the powerful features of the original ADP, while eliminating the action module completely. There is no need for the iterative training loops between the action and the critic networks and, thus, greatly simplify the training process. There exists a class of problems in realistic applications that have a finite dimensional control action space. Typical examples include inverted pendulum or the cart-pole problem, where the control action only takes a few finite values. When there is only a finite control action space in the application, the decisions that can be made are constrained to a limited number of choices, e.g., a ternary choice in the case of residential energy control and management problem. When there is a power demand from the residential household, the decisions can be made are constrained to three choices, i.e., to discharge batteries, to charge batteries, or to do nothing to batteries. Let us denote the three options by using $u(t) = 1$ for “discharge”, $u(t) = -1$ for “charge”, and $u(t) = 0$ for “idle”. In the

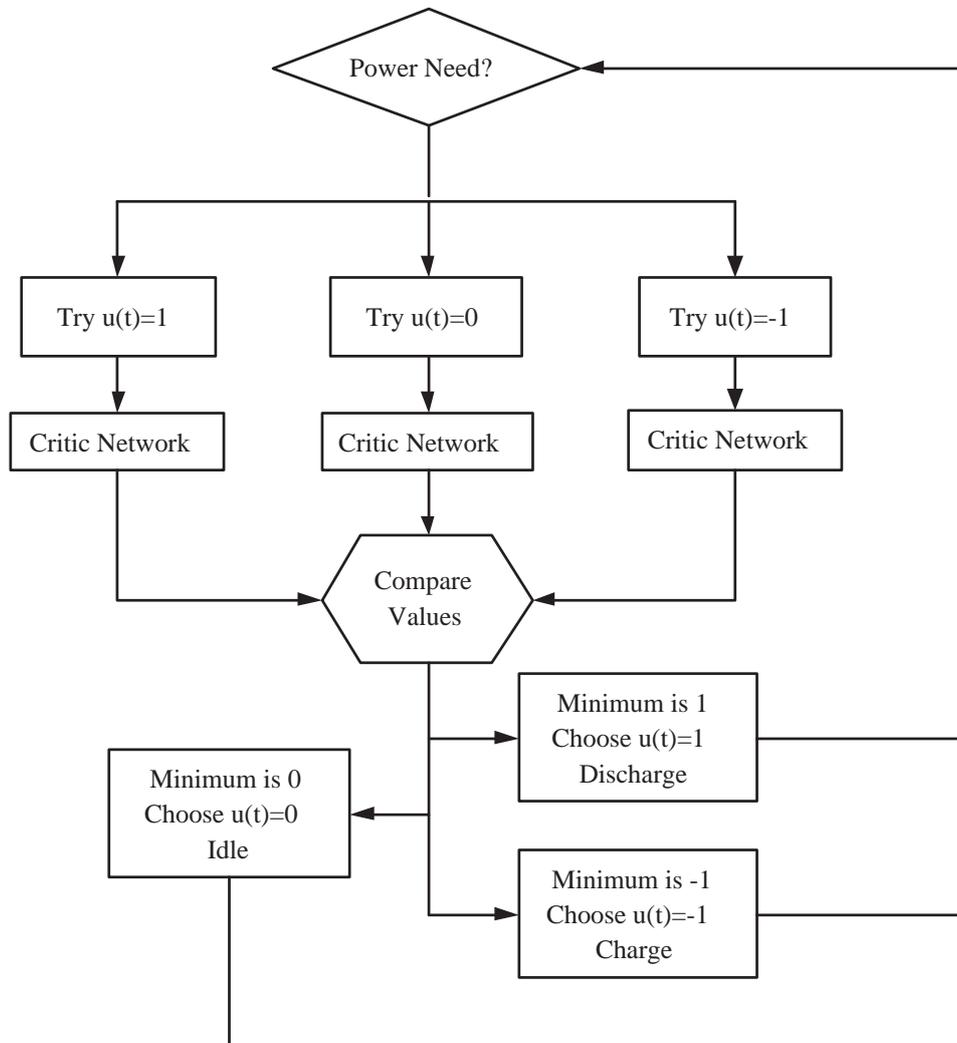


Figure 6.45: Block diagram of the single critic approach

present case, we note that the control actions are limited to a ternary choice, or to only three possible options. Therefore, we can further simplify the ADP introduced in Fig. 3.12 so that only the critic network is needed in the ADP design. Fig. 6.45 illustrates our self-learning control scheme for residential energy system control and management using ADP. The control scheme works in this way: when there is a power demand from the residential household, we will first ask the critic network to see which action (discharge, charge and idle) generates the smallest output value of the critic network; then, the control action from $u(t) = 1, -1, 0$ that generates the smallest critic network output will be chosen. As in the case of Fig. 3.12, the critic network in our ADP design will also need the system states as input variables. It is important to realize that Fig. 6.45 is only a diagrammatic layout that illustrates how the computation takes place while making battery control and management decisions. In Fig. 6.45, the three blocks for the critic network stand for the same critic network or computer program. From the block diagram in Fig. 6.45, it is clear that the critic network will be utilized three times in calculations with different values of $u(t)$ to make a decision about whether to discharge or charge batteries or keep it idle. The previous description is based on the assumption that the critic network has been successfully trained. Once the critic network is learned and obtained (offline or online), it will be applied to perform the task of residential energy system control and management as in Fig. 6.45. The performance of the overall system can be further refined and improved through continuous learning as it learns more experience in real-time operations when needed. In this way, the overall residential energy system will achieve optimal individual performance now and in the future environments under uncertain changes.

In stationary environment, where residential energy system configuration remains unchanged, a set of simple static if-then rules will be able to achieve the optimal scheduling as described previously. However, system configuration including user power demand, capacity of the battery, power rate, etc., may be significantly different from time to time. To cope with uncertain changes of en-

vironments, static energy control and management algorithm would not be proper. The present control and management scheme based on ADP will be capable of coping with uncertain changes of the environment through continuous learning. Another advantage of the present self-learning scheme is that, through further learning as it gains more and more experience in real-time operations, the algorithm has the capability to adapt itself and improve performance. We note that continuous learning and adaptation over the entire operating regime and system conditions to improve the performance of the overall system is one of the key promising attributes of the present method.

The development of the present self-learning scheme for residential energy system control and management involves the following four steps.

Step 1) Collecting data: During this stage, whenever there is a power demand from residential household, we can take any of the following actions: discharge batteries, charge batteries or keep batteries idle and calculate the utility function for the system. The utility function in the present work is chosen as:

$$U(t) = \frac{\text{the electricity charge at time } t}{\text{the possible maximum cost}} \quad (6.69)$$

During the data collection step, we simply choose actions 1, -1 , 0 randomly with the same probability of $1/3$. In the meanwhile, the states corresponding to each action are collected. The environmental states we collect for each action are the electricity rate, the residential load, and the energy level of the battery.

Step 2) Training the critic network: We use the data collected to train the critic network as presented in the previous section. The input variables chosen for the critic network are states including the electricity rate, the residential load, the energy level of the battery and the action.

Step 3) Applying the critic network: We apply the trained critic network as illustrated in Fig.

6.45. Three values of action $u(t)$ will be provided to the critic network at each time step. The action with the smallest output of the critic network is the one the system is going to take.

Step 4) Further updating critic network: We will update the critic network as needed while it is applied in the residential energy system to cope with environmental changes, for example, user demand changes or new requirements for the system. We note that the data has to be collected again and the training of critic network has to be performed as well. In such a case, the previous three steps will be repeated.

Once the training data is collected, we use the forward-in-time method to train the critic network. The critic network is trained at time $t - 1$, with the output target given by $U(t) + \gamma Q(t)$. The training of the critic network is to realize the mapping given by

$$C_f: \left\{ \begin{array}{l} x(t-1) \\ u(t-1) \end{array} \right\} \rightarrow \{U(t) + \gamma Q(t)\}. \quad (6.70)$$

In this case, we consider $Q(t - 1)$ as the output from the network to be trained and $x(t - 1)$ and $u(t - 1)$ as the input to the network to be trained. We calculate the target output value for the training of the critic network by using its output at time t as indicated in (6.70). The goal of learning the function given by (6.70) is to have the critic network output satisfy

$$Q(t - 1) \approx U(t) + \gamma Q(t) \text{ for all } t$$

which is required by (3.13) for approximating dynamic programming solutions.

Using the strategy of [79], the training procedure for the critic network is presented in the following steps:

Step 1) initialize two critic networks: $cnet1 = cnet2$;

Step 2) collect data as in (6.70) including states and action for training;

Step 3) use $cnet2$ to get $Q(t)$, and then train $cnet1$ for five epochs using the Levenberg-Marquardt algorithm [50];

Step 4) copy $cnet1$ to $cnet2$, i.e., let $cnet2 = cnet1$;

Step 5) repeat Steps 3) and 4), e.g., five times;

Step 6) repeat Steps 2)–5), e.g., fifty times;

Step 7) pick the best $cnet1$ as the trained critic network.

Note that the training for the critic network we describe here can be applied to both the initial training of the critic network and further training of the critic network when needed in the future.

6.4 Simulation Results

The performance of the proposed algorithm is demonstrated by simulation studies for a typical residential family. The objective is to minimize the electricity cost from power grids over one week horizon by finding the optimal battery operational strategy of the energy system while satisfying load conditions and system constraints. The focus of the present paper is on residential energy system with home batteries connected to the power grids. For the residential energy system, the cost to be minimized is a function of real-time pricing and residential power demands. The optimal battery operation strategy refers to the strategy of when to charge batteries, when to discharge batteries and when to keep batteries idle to achieve minimum electricity cost for the residential user.

The residential energy system consists of power grids, an inverter, batteries and a power management unit as shown in Fig. 6.42. We assume that the supply from power grid is guaranteed for the residential user demand at any time. The capacity of batteries used in the simulations is

100kWh and a minimum of 20% of the charge is to be retained. The rated power output of batteries and the maximum charge/discharge rate is 16kWh. The initial charge of batteries is at 80% of batteries' full-charge. We assume that the batteries and the power grids will not simultaneously provide power to the residential user. At any time, residential power demand is supplied by either batteries or power grids. The power grids would provide the supply to the residential user and, at the same time, charge batteries. It is expected that batteries are charged during the low rate hours, idle in some mid-rate hours, discharged during high rate hours. In this way, energy and cost savings are both achieved.

The critic network in the present application is a multilayer feedforward neural network with 4–9–1 structure, i.e., four neurons at the input layer, nine neurons at the hidden layer, and one linear neuron at the output layer. The hidden layer uses the hyperbolic tangent function as the activation function. The critic network outputs function Q , which is an approximation to the function $J(t)$. The four inputs to the critic network are: energy level of batteries, residential power demand, real-time pricing, and the action of operation (1 for discharging batteries, -1 for charging batteries, 0 for keeping batteries idle). The local utility function defined is

$$U(t) = \frac{C(t) \times P_g(t)}{U_{\max}}$$

where $C(t)$ is real-time pricing rate, $P_g(t)$ is the supply from power grids for residential power demand, and U_{\max} is the possible maximum cost for all time. The utility function chosen in this way will lead to a control objective of minimizing the overall cost for the residential user.

The typical residential load profile in one week is shown in Fig. 6.46 [141]. We add up to $\pm 10\%$ random noise in the load curve. From the load curve, we can see that, during weekdays, there are two load peaks occurring in the period of 7:00–8:00 and 18:00–20:00, while during weekend, the residential demand gradually increases until the peak appears at 19:00. Thus, the

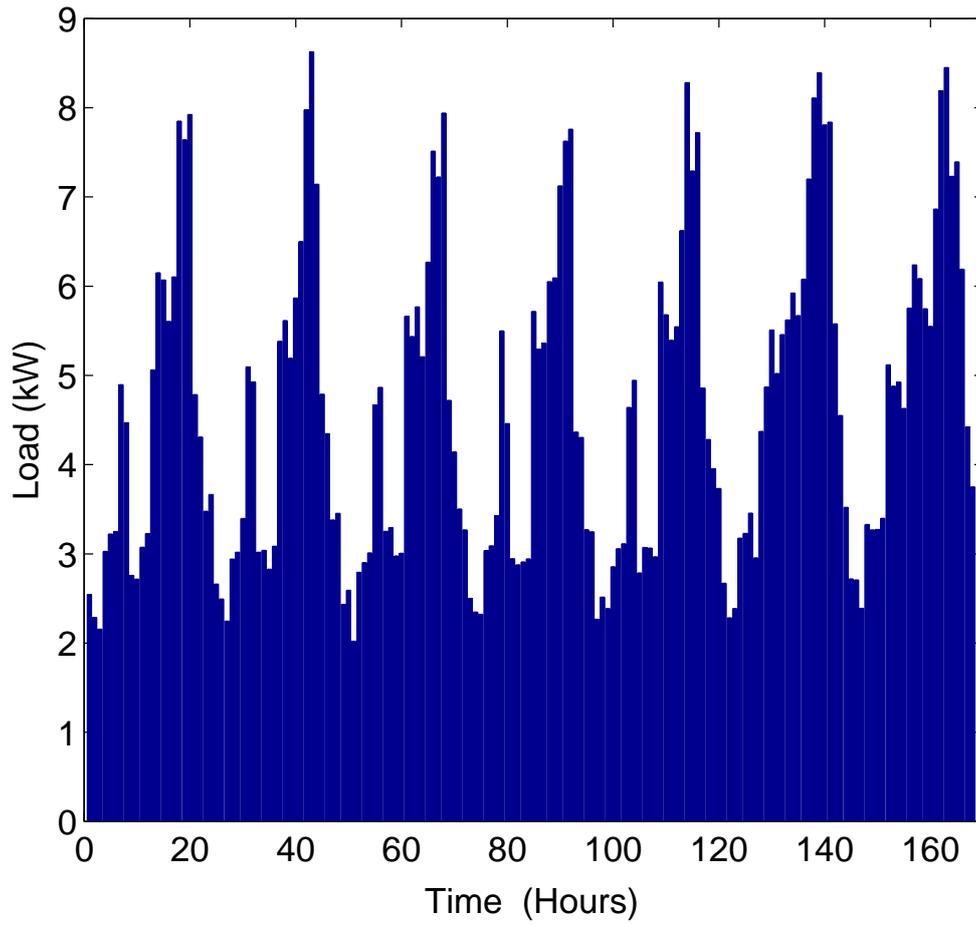
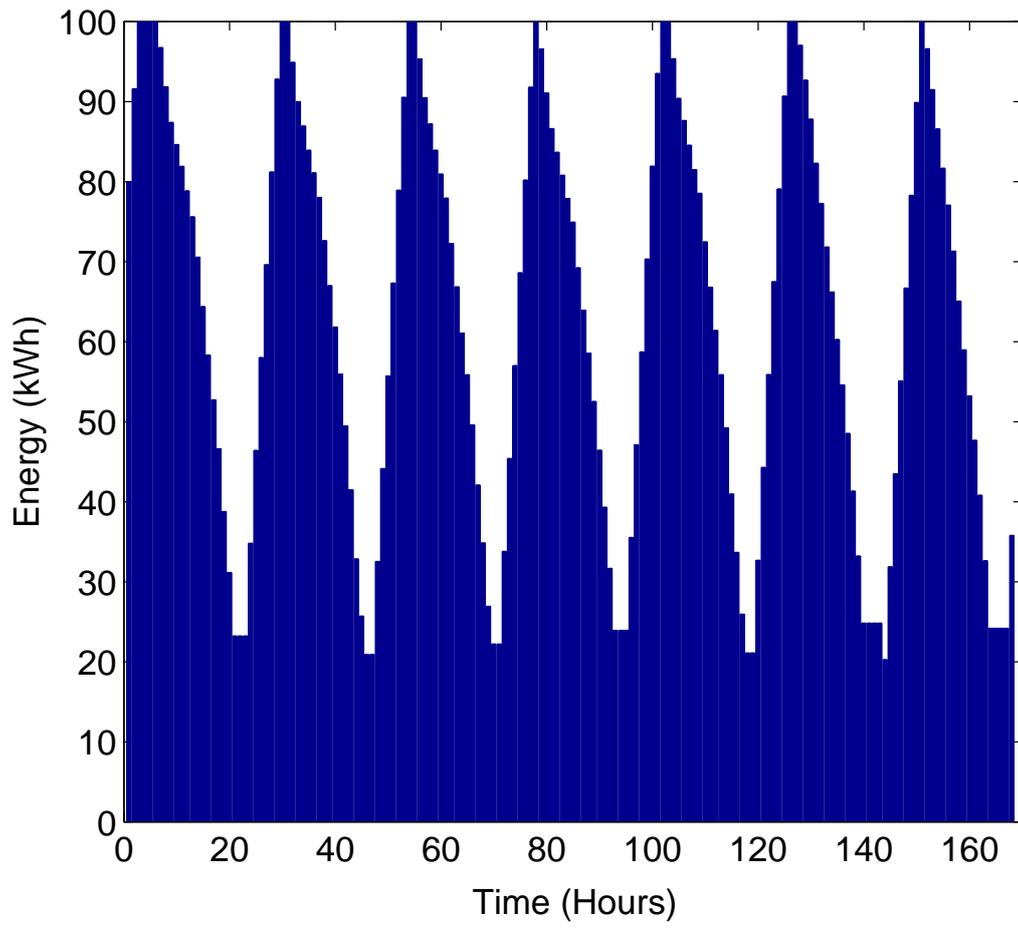


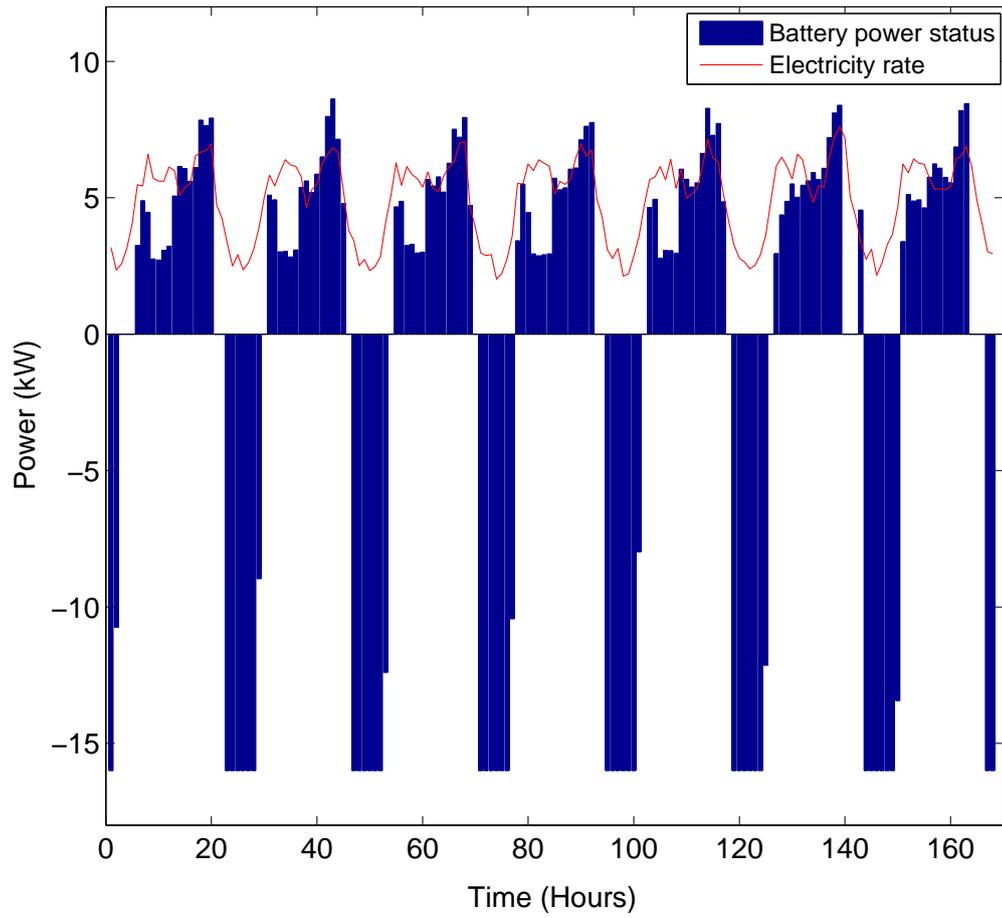
Figure 6.46: A typical residential load profile in one week



inin

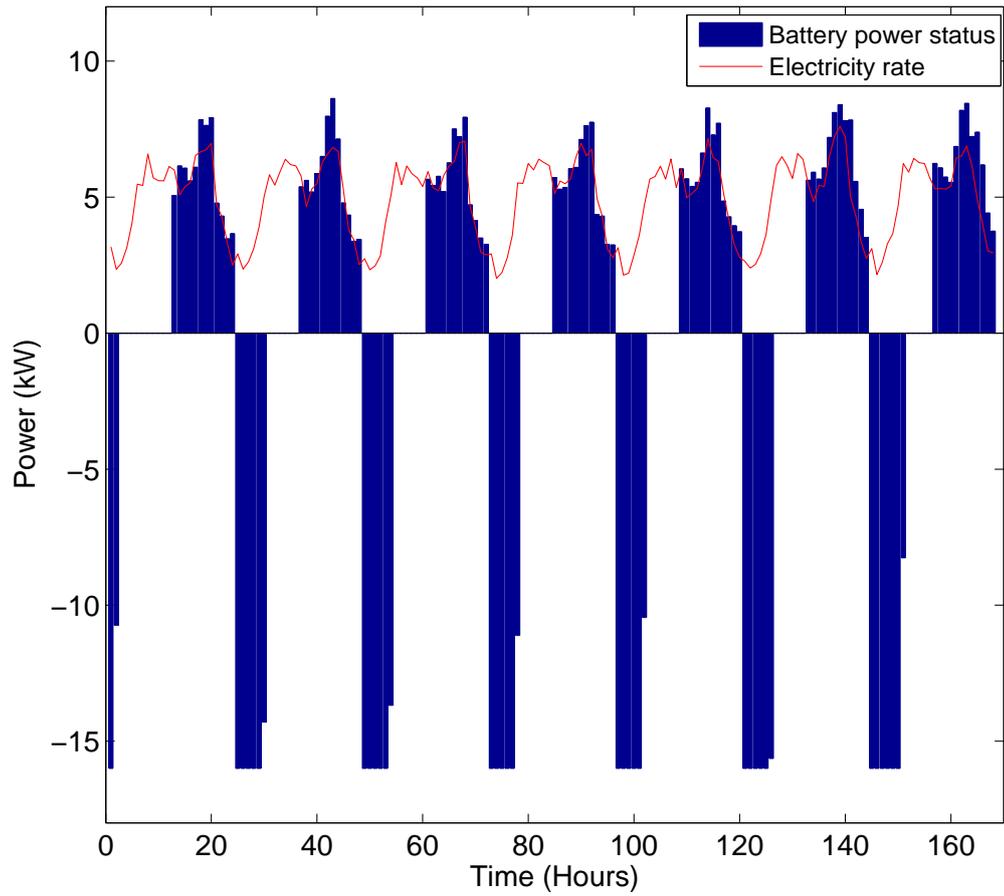
Figure 6.47: Energy changes in batteries

residential demand pattern during weekdays and during weekend is different. Fig. 6.47 shows the change of the electrical energy level in batteries during a typical one week residential load. From Fig. 6.47, it can be seen that batteries are fully charged during the midnight when the price of electricity is cheap. After that, batteries discharge during peak load hours or medium load hours, and are charged again during the midnight light load hours. This cycle repeats, which means that the scheme is optimized with evenly charging and discharging. Therefore, the peak of the load curve is shaved by the output of batteries, which results in less consumption of power from the power grids. Fig. 6.48 illustrates the optimal scheduling of home batteries. The bars in Fig. 6.48 represent the power output of batteries, while the dotted line denotes the electricity rate in real-time. From Fig. 6.48, we can see that batteries are charged during hours from 23:00 to 5:00 next day when the electricity rate is in the lowest range and discharge when the price of electricity is expensive. It is observed that batteries discharge from 6:00 to 20:00 during weekdays and from 7:00 to 19:00 during weekend to supply the residential power demand. The difference lies in the fact that the power demand during the weekend is generally bigger than the weekdays' demand, which demonstrates that the present scheme can adapt to varying load conditions. From Fig. 6.48, we can also see that there are some hours that the batteries are idle, such as from 3:00 to 5:00 and from 21:00 to 22:00. Obviously, the self-learning algorithm believe that, considering the subsequent load demand and electricity rate, keeping batteries idle during these hours will achieve the most economic return which result in the lowest overall cost to the customer. The cost of serving this typical residential load in one week is 2866.64 cents. Comparing to the cost using the power grids alone to supply the residential load which is 4124.13 cents, it gives a savings of 1257.49 cents in a week period. This illustrates that a considerable saving on the electricity cost is achieved. In this case, the self-learning scheme has the the ability to learn the system characteristics and provide the minimum cost to the residential user.



inin

Figure 6.48: Optimal scheduling of batteries in one week



inin

Figure 6.49: Scheduling of batteries of fixed daily cycle scheme

In order to better evaluate the performance of the self-learning scheme, we conduct comparison studies with a fixed daily cycle scheme. The daily cycle scheme charges batteries during the day time and releases the energy into the residential user load when required during the expensive peak hours at night. Fig. 6.49 shows the scheduling of batteries by the fixed daily cycle scheme. The overall cost is 3284.37 cents. This demonstrates that the present ADP scheme has lower cost. Comparing Fig. 6.48 with Fig. 6.49, we can see the self-learning scheme is able to discharge batteries one hour late from 7:00 to 19:00 during the weekend instead of from 6:00 to 20:00 during weekdays to achieve optimal performance, while the fixed daily cycle scheme ignore the differences of the demand between weekdays and weekend due to the static nature of the algorithm. Therefore, we conclude that the present self-learning algorithm performs better than the fixed algorithm due to the fact that the self-learning scheme can adapt to the varying load consideration and environmental changes.

6.5 Conclusions

In this chapter, we developed a self-learning scheme based on ADP for the new application of residential energy system control and management. Such a neural network scheme will be obtained after a specially designed learning process that performs approximate dynamic programming. Once the scheme is learned and obtained (offline or online), it will be applied to perform the task of energy cost optimization. The simulation results indicate that the proposed self-learning scheme is effective in achieving minimization of the cost through neural network learning. The key promising feature of the present approach is the ability of the continuous learning and adaptation to improve the performance during real-time operations under uncertain changes in the environment or new system configuration of the residential household. We note that changes in residential demand are inevitable in real-time operations. Therefore, fixed scheme which cannot take demand

changes and system characteristics into account is less preferable in practical applications. Another important benefit of the present algorithm is that it can be adapted to different scenarios of different residential customers. Traditional fixed control strategies apply the same control strategy for all system configurations, ignoring the different demands and system configurations. Therefore, this procedure cannot ensure an optimum system design for all customers. With continuous learning and adaptation for residential household energy system, the control scheme based on ADP can obtain the optimal control strategy according to the system configuration and energy utilization of the residential customer. This scheme is customer-centered, unlike the utility-centered, yet effective and simple enough for a real-life use of residential consumers.

7 CONCLUDING REMARKS

In this dissertation, we addressed two research problems and presented our corresponding methods to solve them. The first research problem is related to automotive engine control. We proposed three advanced learning control algorithms for the automotive engine TRQ and AFR control. The first step required for a successful identification of the neural controller is the characterization of a suitable model for the processes under control. We tried a multi-input multi-output neural network model which coupled the TRQ and AFR together to model the engine processes. Based on this model, we developed the adaptive critic design to address the engine control problems. However, due to the different nature of the TRQ and AFR process, we built two neural network models for the engine TRQ process and AFR dynamics. Combustion torque is governed mostly by nonlinearities whereas the air-fuel ratio process dominated by purely dynamic effects. Simulation results demonstrate that good matches between the real vehicle data and the outputs of the models. Based on these two neural network models, we applied neural sliding mode control technique and baroreceptor reflex inspired adaptive control strategy for the engine TRQ and AFR control.

The goals of the present learning control design for automotive engines include improved performance, reduced emissions and maintained optimum performance under various operating conditions. More specifically, The goal of engine TRQ control is to track the commanded torque. The objective of the AFR control is to regulate the engine air-fuel ratio at specified setpoints. We applied three different learning control algorithms to solve these problems: adaptive critic learning technique, neural sliding mode control technique and biological nonlinear adaptive control technique. The model-free action-dependent heuristic dynamic programming which approximates the cost function with respect to the states of the process to be controlled was used at first in the Chapter 3. Action-dependent heuristic dynamic programming is very useful in solving the cost

minimization problems. By carefully defining the cost function, it realizes the optimal control of the engine TRQ and AFR regulation in our research. In simulation studies, with the developed model of the TRQ and AFR process, we have been able to achieve excellent tracking performance. Meanwhile, excellent neural network controller transient performance also been achieved.

Considering the different characteristics of TRQ and AFR process, it is potential to improve the results by the separation of the model and building individual controller for the engine TRQ and AFR control. Therefore, two separated controller each dealing with TRQ and AFR independently will be beneficial in achieving a better control performance. We investigated an approach of cooperative control based on the concept of combining neural networks and the methodology of sliding mode control in the Chapter 4. Two neural networks in parallel are utilized to implement neural sliding mode control. The equivalent control and the corrective control terms of SLMCs are the outputs of the neural networks. The weight adaptations of neural networks are based on the SLMCs equations and Levenberg-Marquardt algorithm. Simulation results indicate that the suggested approach has considerable advantages compared to the classical sliding mode control and is capable of achieving a good chattering free trajectory following performance without a prior detailed knowledge of the controlled plant.

The control techniques derived from biological systems can offer great potential for solving complex and nonlinear problems in the real world. By analogy to the mechanism of baroreflex for maintaining the blood pressure at the desired level, we studied an adaptive nonlinear control strategy in the Chapter 5. The controller consists of a linear controller and a nonlinear controller that interact via a reciprocal lateral inhibitory mechanism. By cooperating with each other in the operating region, the controller achieved desirable performance. The most important feature of the strategy is that each controller adjusts the control output of the other controller via the lateral inhibitory connections. The computational efficiency, real-time adaptability and superior learn-

ing ability are illustrated through extensive numerical simulations for engine torque management driven by the biological adaptive nonlinear control strategy.

Comparing those different control techniques for the engine control, the adaptive critic learning control technique needs lengthy training time to achieve a good result, while neural sliding mode control technique and biological nonlinear adaptive control technique takes considerable shorter training time. The training for adaptive critic design usually takes about 6-7 hours, while, with neural sliding mode control technique, it takes around 5-10 minutes to achieve similar result. Compare the method of neuro-sliding mode control with biological nonlinear adaptive control technique for the TRQ control, the training speed is almost the same, but the tracking effect of control with biological nonlinear adaptive control technique is better.

Development of reliable techniques for calibration and control of systems for which mathematical models of the underlying physical processes are either not available or not fully developed have always posed significant challenges in engine control design and synthesis. The adaptive critic learning control technique, neural sliding mode control technique and biological nonlinear adaptive control technique, together with neural network modeling of the engine processes will enable us to design engine control systems with minimal a priori knowledge of the processes involved and allow real-time adaptation of engine controllers due to uncertain changes in the environmental and vehicle conditions. These techniques, once fully developed, may find potential applications as engine calibration tools for modern propulsion systems such as CIDI, HCCI and even Fuel Cells.

Intelligent energy management systems can help to minimize energy costs for the residential customers and reduce emissions by efficiently using renewable energy resources and distributed energy storage systems. If distributed energy storage resources, including batteries in the home, would be adopted smartly and efficiently, they would be an effective method to create more price-

responsive demand and integrate more renewable energy resources more effectively into power grids. In the Chapter 6, we applied a self-learning scheme for the control and management of residential energy system. The self-learning scheme is based on a single critic neural network. The distinctive advantage of the scheme is that the intelligent method has the capability to achieve the optimal control for different customers based on the demand and system configuration. Simulation results confirmed that our proposed learning scheme can greatly benefit the residential customers with the minimum electricity charge. This technique has the potential to revolutionize the residential energy management as it reduces and shifts demand automatically, provides valuable insights to customers which ultimately save the environment by reducing the carbon footprint of power companies.

CITED LITERATURE

- [1] N. Al-Holou, T. Lahdhiri, D. S. Joo, J. Weaver, and F. Al-Abbas, "Sliding mode neural network inference fuzzy logic control for active suspension systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 259–268, Apr. 2000.
- [2] C. Alippi, C. D. Russis, and V. Piuri, "A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 33, no. 2, pp. 259–268, May 2003.
- [3] C. Alippi and V. Piuri, "Neural methodology for prediction and identification of non-linear dynamical systems," *Proceedings of the International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing*, Venice, Italy, Aug. 1996, pp. 305–313.
- [4] C. Alippi, C. De Russis, and V. Piuri, "A fine control of the air-to-fuel ratio with recurrent neural networks," *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, St. Paul, MN, May 1988, pp. 924–929.
- [5] S.M. Amin, B.F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE Power Energy Mag.*, vol. 3, no. 5, pp. 34–41, 2005.
- [6] A. Amstutz, N. P. Fekete, and J. D. Powell, "Model based airCfuel ratio control in SI engines with a switch type EGO sensor," *SAE paper 940972*, 1994.
- [7] C. F. Aquino, "Transient A/F control characteristics of the 5 liter central fuel injection engine," *SAE Paper 810494*, 1981.
- [8] I. Arsie, C. Pianese, and M. Sorrentino, "A procedure to enhance identification of recurrent neural networks for simulating air-fuel ratio dynamics in SI engines," *Eng. Appl. Artif. Intell.*, vol. 19, no. 1, pp. 65–77, Feb. 2006.
- [9] I. Arsie, C. Pianese, and M. Sorrentino, "Nonlinear recurrent neural networks for air fuel ratio control in SI engines," *SAE Paper 2004-01-1364*, 2004.
- [10] I. Arsie, C. Pianese, G. Rizzo, and V. Cioffi, "An adaptive estimator of fuel film dynamics in the intake port of a spark ignition engine," *Control Engineering Practice*, vol. 11, no. 3, pp. 303–309, Mar. 2003.
- [11] I. Arsie, R. Flora, C. Pianese, G. Rizzo, and G. Serra, "A computer code for S.I. engine control and powertrain simulation," *SAE Paper 2000-01-0938*, 2000.
- [12] B. Ault, V. K. Jones, J. D. Powell, and G. F. Franklin, "Adaptive air-fuel ratio control of a spark-ignition engine," *SAE Paper 940373*, 1994.

- [13] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 4, pp. 893–898, July–Aug. 1996.
- [14] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, C. Pinello, and A. L. Sangiovanni-Vincentelli, "Automotive engine control and hybrid systems: challenges and opportunities," *Proceedings of the IEEE*, vol. 88, no. 7 pp. 888–912, 2000.
- [15] A. G. Bakirtzis and P. S. Dokopoulos, "Short term generation scheduling in a small autonomous system with unconventional energy system," *IEEE Trans. on Power System*, vol. 3, no. 3, Aug, 1988.
- [16] A. G. Barto, "Reinforcement learning and adaptive critic methods," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches* (Chapter 12), Edited by D. A. White and D. A. Sofge, New York, NY: Van Nostrand Reinhold, 1992.
- [17] V. M. Becerra, J. M. F. Calado, P. M. Silva, and F. Garces, "System identification using dynamic neural networks: training and initialization aspects," *Proc. of 15th World Congress IFAC 2002*, Barcelona, Spain, June 2002, pp. 235–240.
- [18] R. E. Bellman, *Dynamic Programming*, Princeton, NJ: Princeton University Press, 1957.
- [19] B. Belvedere, M. Bianchi, A. Borghetti, and M. Paolone, "A microcontroller-based automatic scheduling system for residential microgrids," in *Proc. IEEE Power Tech.*, Bucharest, Romania, June 2009, pp. 1–6.
- [20] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Belmont, MA: Athena Scientific, 1996.
- [21] J.H. Buckland and J.A. Cook, "Automotive emissions control," *Proc. Amer. Contr. Conf.*, Portland, OR, June 2005, pp. 3290–3295.
- [22] S. Cagnoni, *Applications of Evolutionary Computing*, Springer, 2002.
- [23] L. N. de Castro and J. Timmis, *Artificial Immune Systems*, Springer, 2002.
- [24] T. D. H. Cau and R. J. Kaye, "Multiple distributed energy storage scheduling using constructive evolutionary programming," *IEEE Power Eng. Soc. Int. Conf. Power Ind. Comput. Appl.*, May 2001, pp. 402–407.
- [25] F. A. Chacra, P. Bastard, G. Fleury, and R. Clavreul, "Impact of energy storage costs on economical performance in a distribution substation," *IEEE Trans. Power System*, vol. 20, no. 2, pp. 684–691, 2005.
- [26] S. Chakraborty, M. D. Weiss, and M. G. Simoes, "Distributed Intelligent Energy Management System for a Single-Phase High-Frequency AC Microgrid," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 1, pp. 97–109, Jan 2007.

- [27] C.F. Chang, N. P. Fekete, A. Amstutz, and J. D. Powell, "Airfuel ratio control in spark-ignition engines using estimation theory," *IEEE Trans. Contr. Syst. Tech.*, vol. 3, no. 1, Mar. 1995.
- [28] S. J. Chiang, K. T. Chang, and C. Y. Yen, "Residential photovoltaic energy storage system," *IEEE Trans. Ind. Electron.*, vol. 45, no. 3, pp. 385–394, June 1998.
- [29] S. B. Choi and J. K. Hedrick, "An observer-based controller design method for improving air/fuel characteristics of spark ignition engines," *IEEE Trans. Contr. Syst. Tech.*, vol. 6, no. 3, pp. 325–334, May 1998.
- [30] P.M. Corrigan and G.T. Heydt, "Optimized Dispatch of a Residential Solar Energy System," *North American Power Symposium*, Sept. 2007, pp. 4183–188.
- [31] C. Cox, S. Stepniewski, C. Jorgensen, R. Saeks, and C. Lewis, "On the design of a neural network autolander," *International Journal of Robust and Nonlinear Control*, vol. 9, pp. 1071–1096, Dec. 1999.
- [32] F. P. Da, "Decentralized sliding mode adaptive controller design based on fuzzy neural networks for interconnected uncertain nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1471–1480, Nov. 2000.
- [33] J. Dalton and S. N. Balakrishnan, "A neighboring optimal adaptive critic for missile guidance," *Math. Comput. Modell.*, vol. 23, no. 1-2, pp. 175–188, Jan. 1996.
- [34] D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer, 1998.
- [35] H. Demuth and M. Beale, *Neural Network Toolbox User's Guide*. Natick, MA: MathWorks Inc., 1998. (Version 3).
- [36] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*, New York, NY: Academic Press, 1977.
- [37] H. Du and S. S. Nair, "A neuro-sliding control approach for a class of nonlinear systems," in *Proc. of 1st Int. Conf. on Knowledge-Based Intell. Elect. Syst.*, Adelaide, Australia, May 1997, pp. 331–337.
- [38] D. J. Dobner, "A mathematical engine model for development of dynamic engine control," *SAE Paper 800054*, 1980.
- [39] D. J. Dobner, "Dynamic engine models for control development Part I: Non-linear and linear model formation," *Int. J. Veh. Des.*, Special Publication SP4, pp. 54–74, 1983.
- [40] M. Ertugrul and O. Kaynak, "Neuro sliding mode control of robotic manipulators," *Mechatron*, vol. 10, no. 12, pp. 243–267, Feb. 2000.

- [41] M. Ertugrul, O. Kaynak, A. Sabanovic, and K. Ohnishi, "A generalized approach for Lyapunov design of sliding mode controllers for motion control applications," *Proc. of Adv. Motion Control*, Mar. 1996, pp. 243–267.
- [42] A. Ferrara and P. Pisu, "Minimum sensor second-order sliding mode longitudinal control of passenger vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no.1, pp. 20-32, Mar. 2004.
- [43] C. C. Fung, S. C. Y. Ho, and C. V. Nayar, "Optimisation of a hybrid energy system using simulated annealing technique," *Proc. IEEE Int. Conf. Comput. Commun. Control Power Eng.*, Oct. 1993, pp. 235–238.
- [44] A. Di Gaeta, S. Santini, L. Glielmo, F. De Cristofaro, C. Di Guisepppe, and A. Caraceni, "An algorithm for the calibration of wall-wetting model parameters," *SAE Paper 2003-01-1054*, 2003.
- [45] Garrity TF (2008) Getting smart. *IEEE Power Energy Mag* 6(2):38–45
- [46] M. E. Glavin, P. K.W. Chan, S. Armstrong, and W. G. Hurley, "A stand-alone photovoltaic supercapacitor battery hybrid energy storage system," *Power Electronics and Motion Control Conference*, Sept. 2008, pp. 1688–1695.
- [47] A. Gupta, R. P. Saini, and M. P. Sharma, "Computerized modelling of hybrid energy system† Part II: Combined dispatch strategies and solution algorithm," *International Conference on Electrical and Computer Engineering*, vol. 3, no. 3, Dec. 2008, pp. 13–18.
- [48] L. Guzzella, M. Simons, and H. P. Geering, "Feedback linearizing air/fuel-ratio controller," *Control Eng. Practice* vol. 5 no. 8, pp. 1101–1105, Aug. 1997.
- [49] L. Guzzella, "Models and model-based control of IC-engines a nonlinear approach," *SAE Paper 950844*, 1995.
- [50] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.
- [51] S. Haykin, *Neural networks: a comprehensive foundation*, Prentice Hall, 1999.
- [52] E. Hendricks and T. Vesterholm, "Non-Linear Transient Fuel Film Compensation," *SAE Paper TP 930767*, 1993.
- [53] E. Hendricks and S. C. Sorenson, "SI engine controls and mean value engine modeling," *SAE Paper 910258*, 1991.
- [54] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.
- [55] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward network are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

- [56] T. Huang and D. Liu, "A self-learning scheme for residential energy system control and management," *Neural Computing and Applications*, accepted for publication.
- [57] T. Huang and D. Liu, "Residential energy system control and management using adaptive dynamic programming," *Proceeding of INNS-IEEE International Joint Conference on Neural Networks*, San Jose, CA, July 2011.
- [58] T. Huang, H. Javaherian, and D. Liu, "Nonlinear torque and air-to-fuel ratio control of spark ignition engines using neuro-sliding mode technique," *Inter. J. Neural Syst.*, vol.21, no.3, pp. 213–224, June 2011.
- [59] T. Huang, D. Liu, H. Javaherian, and N. Jin, "Neural sliding-mode control of the engine torque," *Proc. of the 2008 IFAC Triennial World Congr.*, Jul. 2008, pp. 9453–9458.
- [60] C. L. Hwang, "A novel Takagi-Sugeno-based robust adaptive fuzzy sliding-mode controller," *IEEE Trans. Fuzzy Syst.* vol. 12, no. 5, pp. 676–687, Oct. 2004.
- [61] S. Janardhanan and B. Bandyopadhyay, "Discrete sliding mode control of systems with unmatched uncertainty using multirate output feedback," *IEEE Trans. Automat. Contr.*, vol. 51, no. 6, pp. 1030–1035, Jun. 2006.
- [62] D. Jiang and J. Wang, "On-line learning of dynamical systems in the presence of model mismatch and disturbances," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1272–1283, Nov. 2000.
- [63] M. James, B. Jacob, and S. G. Scott, "Dynamic analyses of regenerative fuel cell power for potential use in renewable residential applications," *Int. J. Hydrogen Energy*, vol. 31, no. 8, pp. 994–1009, Jul. 2006.
- [64] W. Janig, *The integrative action of the autonomic nervous system: neurobiology of homeostasis*, Cambridge University Press, 2006.
- [65] H. Javaherian, T. Huang, and D. Liu, "A biologically inspired adaptive nonlinear control strategy for applications to powertrain control," *Proc. 2009 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, Oct. 2009, pp.2007–2013.
- [66] N. Jin, D. Liu, Z. Pang, and T. Huang, "Wavelet basis function neural networks," *Proc. 2007 International Joint Conference on Neural Networks*, Orlando, FL, Aug. 2007, pp.500–505.
- [67] N. Jin, D. Liu, T. Huang, and Z. Pang, "Discrete-time adaptive dynamic programming using wavelet basis function neural networks," *Proc. 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, Honolulu, HI, Apr. 2007, pp.135–142.
- [68] V. K. Jones, B. A. Ault, G. F. Franklin, and J. D. Powell, "Identification and air-fuel ratio control of a spark ignition engine," *IEEE Transactions on Control Systems Technology* vol. 3, no. 1, pp. 14–21, Mar. 1995.

- [69] A. Joseph and M. Shahidehpour, "Battery storage systems in electric power systems," *IEEE Power Engineering Society General Meeting*, 2006.
- [70] A. Karakasoglu and M. K. Sundareshan, "A recurrent neural network based adaptive variable structure model-following control of robotic manipulators," *Automatica*, vol. 31, no. 10, pp. 1495–1507, Oct. 1995.
- [71] O. Kaynak, K. Erbatur, and M. Ertugrul, "The fusion of computationally intelligent methodologies and sliding mode control—a survey," *IEEE Trans. Ind. Electron.*, vol. 48, no. 1, pp. 4–17, Feb. 2001.
- [72] M. K. Khan and S. K. Spurgeon, "MIMO control of an IC engine using dynamic sliding modes," *Proc. of Intell. Syst. and Control*, Salzburg, Austria, June 2003.
- [73] A. J. Koshkouei and A. S. I. Zinober, "Sliding mode control of discrete-time systems," *J. Dyn. Syst. Meas. Contr.*, vol. 122, no. 4, pp. 793–802, Dec. 2000.
- [74] O. A. Kovalenko, *Adaptive Critic Learning Techniques for Automotive Engine Control*, Master Thesis, University of Illinois at Chicago, Chicago, IL, 2005.
- [75] O. Kovalenko, D. Liu, and H. Javaherian, "Neural network modeling and adaptive critic control of automotive fuel-injection systems," *Proceedings of the IEEE Int. Symp. Intelligent Contr.*, Taipei, Taiwan, Sept. 2004, pp. 368–373. (Invited paper)
- [76] T. Y. Lee, "Operating Schedule of Battery Energy Storage System in a Time-of-Use Rate Industrial User With Wind Turbine Generators: A Multipass Iteration Particle Swarm Optimization Approach," *IEEE Transaction on Energy Conversion*, vol. 22, no. 3, pp. 774–782, 2007.
- [77] M. Lee and H. S. Choi, "A robust neural controller for underwater robot manipulators," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1465–1470, Nov. 2000.
- [78] P. Lee, S. Hong, Y. Lim, C. Lee, B. Jeon, and J. Park, "Discrete-time quasi-sliding mode control of an autonomous underwater vehicle," *IEEE J. Oceanic Eng.*, vol. 24, no. 3, pp. 388–395, July 1999.
- [79] G. G. Lendaris and C. Paintz, "Training strategies for critic and action neural networks in dual heuristic programming method," in *Proc. Int. Conf. Neural Netw.*, Houston, TX, June 1997, pp. 712–717.
- [80] M. N. Levy, "Autonomic interactions in cardiac control," *Annals of the New York Academy of Sciences*, vol. 601, pp. 209–221, 1990.
- [81] M. N. Levy, "Cardiac sympathetic and parasympathetic interactions," *Fed. Proc.*, vol. 43, pp. 2598–2602, 1984.

- [82] F. Li, W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang, "Smart transmission grid: vision and framework," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 168–177, 2010.
- [83] X. Li and S. Yurkovich, "Neural network based, discrete adaptive sliding mode control for idle speed regulation in IC engines," *J. Dyn. Syst. Meas. Contr.*, vol. 122, no. 2, pp. 269–275, 2000.
- [84] F. J. Lin and R. J. Wai, "Sliding-mode-controlled slider-crank mechanism with fuzzy neural network," *IEEE Trans. Ind. Electron.*, vol. 48, no. 1, pp. 60–70, Feb. 2001.
- [85] D. Liu, H. Javaherian, O. Kovalenko, and T. Huang, "Adaptive critic learning techniques for engine torque and air-fuel ratio control," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, no. 4, pp. 988–993, Aug. 2008.
- [86] D. Liu, Y. Zhang, and H. Zhang "A self-learning call admission control scheme for CDMA cellular networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1219–1228, 2005.
- [87] D. Liu, X. Xiong, and Y. Zhang, "Action-dependent adaptive critic designs," *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, Washington, DC, July 2001, pp. 990–995.
- [88] C. H. Lo and M. D. Anderson, "Economic dispatch and optimal sizing of battery energy storage systems in utility load-leveling operations," *IEEE Trans. Energy Convers.*, vol. 14, no. 3, pp.824–829, Sep. 1999.
- [89] B. Lu and M. Shahidehpour, "Short-term scheduling of battery in a grid-connected PV/battery system," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1053–1061, 2005.
- [90] D. K. Maly and K. S. Kwan, "Optimal battery energy storage system (BESS) charge scheduling with dynamic programming," *IEE Proc. Sci. Meas. Technol.*, vol. 142, no. 6, pp. 454–458, Nov. 1995.
- [91] T. Miller, R. Sutton, and P. Werbos, *Neural network for Control*, The MIT Press, 1990.
- [92] J. A. Momoh, Y. Wang, and F. Eddy-Posey, "Optimal power dispatch of photovoltaic system with random load," in *Proc. IEEE Power Eng. Soc.*, Piscataway, NJ, June 2004, pp. 1939–1945.
- [93] J. J. Moskwa and J. K. Hedrick, "Nonlinear algorithms for automotive engine control," *IEEE Control Syst. Mag.*, vol. 10, pp. 88–93, Apr. 1990.
- [94] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 32, pp. 140–153, May 2002.

- [95] C. Manzie, M. Palaniswami, and H. Watson, "Gaussian networks for fuel injection control," *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, Vol. 215, no. 10, pp. 1053–1068, 2001.
- [96] L. Mianzo, H. Peng, and I. Haskara, "Transient air-fuel ratio H_∞ preview control of a drive-by-wire internal combustion engine," *Proc. Amer. Control Conf.*, 2001, pp. 2867–2871.
- [97] K. R. Muske, J. C. P. Jones, and E. M. Franceschi, "Adaptive Analytical Model-Based Control for SI Engine Air-Fuel Ratio," *IEEE Trans. Contr. Syst. Technol.*, vol. 16, no. 4, pp. 763–768, 2008.
- [98] K. R. Muske and J. C. P. Jones, "A model-based si engine air fuel ratio controller," *Proc. Amer. Control Conf*, 2006, pp. 3284–3289.
- [99] S. Nakagawa, K. Katogi, M. Oosuga, "A new air-fuel ratio feedback control for ulev/sulev standard," SAE Paper 2002-01-0194, 2002.
- [100] R. Hecht-Nielsen. *Neurocomputing*, Addison Wesley, Reading, MA, 1990.
- [101] M. Norgaard, O. Ravn, N. L. Poulsen, and L. K. Hansen, *Neural Networks for Modeling and Control of Dynamic Systems*. Berlin: Springer, 2000.
- [102] A. Ohata, M. Ohashi, M. Nasu, and T. Inoue, "Crank-angle domain modeling and control for idle speed" *SAE Paper 950075*, 1995.
- [103] C. Onder and H. Geering, "Model-based multivariable speed and air-to-fuel ratio control of an SI engine," *SAE paper 930859*, 1993.
- [104] S. Ouenou-Gamo, A. Rachid, and M. Ouladsine, "A nonlinear controller of a turbocharged diesel engine using sliding mode," *Proc. of IEEE Int. Conf. on Control Appl.*, Oct. 1997, pp. 803–805.
- [105] S. Park, M. Yoon, and M. Sunwoo, "Feedback error learning neural networks for air-to-fuel ratio control in SI engines," *SAE Technical Paper 2003-01-0356*, 2003.
- [106] J. K. Pieper and R. Mehrotra, "Air/fuel ratio control using sliding mode methods," *Proc. Amer. Control Conf.*, San Diego, CA, June 1999, pp. 1027–1031.
- [107] J. D. Powell, N. P. Fekete, and C. F. Chang, "Observer-based air-fuel ratio control," *IEEE Transactions on Control Systems*, vol. 18, pp. 72–83, Oct. 1998.
- [108] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Transactions on Neural Networks*, vol. 8, pp. 997–1007, Sept. 1997.
- [109] D. V. Prokhorov, *Adaptive Critic Designs and Their Applications*, Ph.D. Dissertation, Texas Tech University, Lubbock, TX, Oct. 1997.

- [110] G. V. Puskorius, L. A. Feldkamp, and L. I. Davis, "Dynamic neural network methods applied to on-vehicle idle speed control," *Proceedings of the IEEE*, vol. 84, pp. 1407–1420, Oct. 1996.
- [111] I. Revals and L. Personnaz, "Nonlinear internal model control using neural networks: application to processes with delay and design issues," *IEEE Transactions on Neural Networks*, vol. 11, no.1, pp. 80–90, Jan. 2000.
- [112] Roncero JR (2008) Integration is key to smart grid management. In: Proc IET-CIRED Seminar SmartGrids for Distrib, pp 1–4
- [113] P. Rupanagunta, M. L. Baughman, and J. W. Jones, "Scheduling of cool storage using nonlinear programming techniques," *IEEE Trans. Power Syst.*, vol. 10, no. 3, pp. 1279–1285, Aug. 1995.
- [114] D. Rupp, *Model-based adaptive air/fuel ratio control for an automotive gasoline engine*, Ph.D. thesis, ETH Zurich, Feb. 2009.
- [115] D. Rupp, C. Onder, and L. Guzzella, "Iterative adaptive air/fuel ratio control," *Proc. Advances in Automotive Control*, Seascape Resort, USA, 2008.
- [116] B. Saerens, J. Vandersteen, T. Persoons, J. Swevers, M. Diehl and E. Van den Bulck, "Minimization of the fuel consumption of a gasoline engine using dynamic optimization," *Applied Energy*, vol. 86, no. 9, Sep. 2009, pp. 1582–1588.
- [117] K. Sagawa, "Baroreflex Control of Systemic Arterial Pressure and Vascular Bed," *Handbook of Physiology, Sec. 2 The Cardiovascular System, Vol. III Peripheral Circulation and Organ Blood Flow; Shepherd, J., Abboud, F., Eds.*, American Physiological Society: Bethesda, MA, pp 453-496, 1983.
- [118] S. Shamma, "Spatial and temporal processing in central auditory networks," C. Koch and I. Segev, Eds, *Methods in Neuronal Modeling: From Synapses to networks*, pp. 247-289, MIT Press, Cambridge, MA, 1989.
- [119] J. Si and Y.-T. Wang, "On-line learning control by association and reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, pp. 264–276, Mar. 2001.
- [120] J. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice-Hall, 1991.
- [121] J. J. E. Slotine, "Sliding controller design for non-linear systems," *Int. J. of Contr.*, vol. 40, no. 2, pp. 421–434, Aug. 1984.
- [122] J. S. Souder and J. K. Hedrick, "Adaptive sliding mode control of air-fuel ratio in internal combustion engines," *Int. J. Robust Nonlinear Control*, vol. 14, no. 6, pp. 525–541, Feb. 2004.

- [123] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA: The MIT Press, 1998.
- [124] R.S. Sutton, "Learning to predict by the methods of temporal differences", *Machine Learning*, vol. 3, pp. 9–44, 1988.
- [125] T.N. Thrasher, "Baroreceptors and the long-term control of blood pressure", *Exp. Physiol.*, Vol. 89, 2004.
- [126] C. Van Toller. *The nervous body : an introduction to the autonomic nervous system and behaviour*, Chichester ; New York : Wiley, 1979.
- [127] C. Tsai, H. Chung, and F. Yu, "Neuro-sliding mode control with its applications to seesaw systems," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 124–134, Jan. 2004.
- [128] R. Turin and H. Geering, "Model-reference adaptive A/F ratio control in an SI engine based on Kalman-Filtering techniques," *Proc. Amer. Control Conf.*, 1995, pp. 4082–4090.
- [129] M. Urbina and Zuyi Li, "A Fuzzy Optimization Approach to PV/Battery Scheduling with Uncertainty in PV Generation," *North American Power Symposium*, Sept. 2006, pp. 561–566.
- [130] C. W. Vigild, K. P. H. Andersen, E. Hendricks, and M. Struwe, "Towards robust H-infinity control of an SI engine's air/fuel ratio," *SAE Paper (1999- 01-0854)*, 1999.
- [131] V. I. Utkin, *Sliding Modes in Control and Optimbation*. New York: Springer, 1992.
- [132] S. Wakao and K. Nakao, "Reduction of fuel consumption in PV/diesel hybrid power generation system by dynamic programming combined with genetic algorithm," *IEEE Fourth World Conference on Photovoltaic Energy Conversion*, 2006, pp. 2335–38.
- [133] S. Wang and D. L. Yu, "An Application of Second-Order Sliding Mode Control for IC Engine Fuel Injection," *Canadian Conf. on Electr. and Comput. Eng.*, May 2006, pp. 1035–1038.
- [134] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches* (Chapter 13), Edited by D. A. White and D. A. Sofge, New York, NY: Van Nostrand Reinhold, 1992.
- [135] P. J. Werbos, T. McAvoy, and T. Su, "Neural networks, system identification, and control in the chemical process industries," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches* (Chapter 10), Edited by D. A. White and D. A. Sofge, New York, NY: Van Nostrand Reinhold, 1992.

- [136] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control* (Chapter 3), Edited by W. T. Miller, R. S. Sutton, and P. J. Werbos, Cambridge, MA: The MIT Press, 1990.
- [137] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *General Systems Yearbook*, vol. 22, pp. 25–38, 1977.
- [138] M. Won, S. B. Choi, and J. K. Hedrick, "Air-to-fuel ratio control of spark ignition engines using Gaussian network sliding control," *IEEE Trans. Contr. Syst. Technol.*, vol. 6, no. 5, pp. 678–687, Sept. 1998.
- [139] T. Yau, L. N. Walker, H. L. Graham, and R. Raithel, "Effects of battery storage devices on power system dispatch," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 1, pp. 375–383, 1981.
- [140] D.Q. Zhang and S. K. Panda, "Chattering-free and fast-response sliding mode controller," *IEE Proc. of Control Theory Appl.*, Vol. 146, No. 2, Mar. 1999, pp. 171–177.
- [141] ComEd, USA, <http://www.thewattspot.com/>. Accessed 16 May 2010

VITA

NAME: Ting Huang

EDUCATION: Ph.D., Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, Illinois, 2011

M.S., Aerospace Engineering Engineering, Chinese Academy of Space Technology, Beijing, China, 2003

B.S., Electrical Engineering, Xiamen University, Xiamen, China, 2000

PUBLICATIONS: Ting Huang and Derong Liu, “A self-learning scheme for residential energy system control and management,” *Neural Computing and Applications*, accepted for publication.

Ting Huang and Derong Liu, “Residential energy system control and management using adaptive dynamic programming,” *Proceeding of INNS-IEEE International Joint Conference on Neural Networks*, San Jose, CA, July 2011.

Ting Huang, Hossein Javaherian, and Derong Liu, “Nonlinear torque and air-to-fuel ratio control of spark ignition engines using neuro-sliding mode technique,” *Inter. J. Neural Syst.*, vol.21, no.3, pp. 213–224, June 2011.

Hossein Javaherian, Ting Huang, and Derong Liu, "A biologically inspired adaptive nonlinear control strategy for applications to powertrain control," *Proc. 2009 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, TX, Oct. 2009, pp.2007–2013.

Ting Huang, Derong Liu, Hossein Javaherian, and Ning Jin, "Neural sliding-mode control of the engine torque," *Proc. of the 2008 IFAC Triennial World Congr.*, Jul. 2008, pp. 9453–9458.

Ning Jin, Derong Liu, Zhongyu Pang, and Ting Huang, "Wavelet basis function neural networks," *Proc. 2007 International Joint Conference on Neural Networks*, Orlando, FL, Aug. 2007, pp.500–505.

Ning Jin, Derong Liu, Ting Huang, and Zhongyu Pang, "Discrete-time adaptive dynamic programming using wavelet basis function neural networks," *Proc. 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, Honolulu, HI, Apr. 2007, pp.135–142.

Derong Liu, Hossein Javaherian, Olesia Kovalenko, and Ting Huang, "Adaptive critic learning techniques for engine torque and air-fuel ratio control," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, no. 4, pp. 988–993, Aug. 2008.