

# Adaptive Learning and Optimal Control Methods for Developing Intelligent Transportation Systems

---

A Dissertation  
Presented to  
the Faculty of the School of Engineering and Applied Science  
UNIVERSITY OF VIRGINIA

---

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in Electrical Engineering

---

by

WANSI HONG

June 2020

## APPROVAL SHEET

This dissertation is submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy in Electrical Engineering

---

Author: Wanshi Hong

This dissertation has been read and approved by the examining committee:

---

Advisor: Professor Gang Tao

---

Committee Chair: Professor Zongli Lin

---

Committee Member: Professor Harry Powell

---

Committee Member: Professor Ronald Williams

---

Committee Member: Professor Cody Fleming

Accepted for the School of Engineering and Applied Science:

---

Dean, School of Engineering and Applied Science

July, 2017

## Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Gang Tao, whose patient guidance would benefit me for the rest of my life. These four years in UVA have been one of the best times in my life. Thanks to his insight and encouragement, I was able to finish my study. His careful and patient attitude on research has set an example for me as I pursue my career.

I would like to thank our collaborator, Dr. Hong Wang, from Oak Ridge National Laboratory. He has given me a lot of valuable suggestions and helped with my research. Moreover, he has given me internship opportunities at national laboratories. I have gained valuable experiences for my research in these internship opportunities. Also, I would like to thank my colleagues in Oak Ridge National Laboratory and Pacific Northwest National Laboratory, Chieh (Ross) Wang, Meixin Zhu, and Indrasis Chakraborty for their support and collaboration.

I would like to thank my proposal committee, Professor Zongli Lin, Professor Harry Powell, Professor Ronald Williams, and Professor Cody Fleming. They have given me a lot of valuable suggestions and helped me improve my research.

I am also very grateful to our group members: Dr. Ge Song, Dr. Yu Sheng, Dr. Liyan Wen, and Dr. Chang Tan for being wonderful colleagues and friends. They gave me a lot of help in both scientific research and personal life.

Last but not least, I would like to thank my family: my father Jun Hong, my mother Yuhong Zhang, and my boyfriend Yuan Yuan. Thanks for their love and support.

Wanshi Hong

University of Virginia

June 2020

## Abstract

In recent years, intelligent transportation systems (ITSs) have a significant impact on human life as ITSs improved transportation safety and mobility. The development of ITSs lies in many different aspects. In this dissertation, an adaptive based learning algorithm and optimal control schemes are used in ITS development. The two major areas of focus are the hybrid electric vehicle (HEV) fuel-saving problem and the traffic signal control optimization problem, where both vehicle level and traffic system level are considered in the design.

HEVs have been an effective solution for improved vehicle fuel efficiency and reduced emission pollution. However, the optimization design for HEVs is complicated due to the presence of nonlinear dynamics and complicated integration of the HEV systems. Moreover, there is a trade-off between fuel optimization and emission reduction. In Chapter 2, a co-optimization scheme is proposed to optimize fuel efficiency for HEVs. The proposed optimization scheme uses vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) information as the basis to optimally tune control parameters for the existing powertrain control system. Moreover, the speed of catalyst temperature to reach its light-off level in the exhaust emission system is also considered as an additional optimization constraint to reduce emission. It has been shown that a further 9.22% fuel savings can be achieved on average for a Toyota Prius test model.

Traffic signal control is important for intersection safety and efficiency. However, most traffic signal control methods are designed for individual intersections or corridors. Although some adaptive control systems have been developed, the methods used are often proprietary and not published, making it difficult to evaluate their effectiveness. The goals of our research are to identify the unknown traffic dynamic and then develop some efficient control schemes to achieve minimal traffic delay time and smooth traffic. To met the above goals, in Chapter 3 to Chapter 5, three different approaches are developed. In Chapter 3, an adaptive linear-quadratic regulator (LQR) is designed to minimize both traffic delay and incremental changes in the control input, which is based on linear system approximation. In Chapter

4, a traffic signal optimal control scheme with an adaptive on-line learning scheme using multiple-model neural networks is designed to achieve traffic delay minimization. In this work, a non-linear neural network model is used to represent the unknown traffic dynamics. In Chapter 5, the above problem is solved from another aspect, where the goal is to have the distributions of every intersection follows a target distribution. To this end, a stochastic traffic signal control algorithm is designed. The above three methods are not only practically meaningful for traffic signal control design but also have some theoretic contributions to the field of adaptive learning techniques.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research Backgrounds and Motivations . . . . .	1
1.1.1	Transportation System . . . . .	1
1.1.2	Adaptive Learning Techniques . . . . .	2
1.1.3	Optimization and Control . . . . .	3
1.2	Literature Review . . . . .	4
1.2.1	HEV Power Management . . . . .	4
1.2.2	Traffic Signal Control . . . . .	6
1.2.3	Summary . . . . .	8
1.3	Dissertation Outline . . . . .	8
<b>2</b>	<b>System Identification and Fuel Optimization for HEV Systems</b>	<b>10</b>
2.1	Hybrid Electric Vehicle System Modeling . . . . .	11
2.1.1	Dynamic Model of HEV . . . . .	11
2.1.2	First-Principles Model Validation . . . . .	12
2.2	Objective and Problem Statement . . . . .	15
2.2.1	Control Objective . . . . .	15
2.2.2	Parameter Selection . . . . .	17
2.2.3	Use of V2V/V2I Information . . . . .	20
2.2.4	Co-Optimization Problem Formulation . . . . .	21
2.3	Co-Optimization Scheme Design . . . . .	23

2.3.1	Equivalent NN Models . . . . .	23
2.3.2	Equivalent Optimization Problem Description . . . . .	28
2.3.3	Co-Optimization Based Parameter Tuning Strategy . . . . .	30
2.4	Simulation Study . . . . .	32
2.5	Conclusions and Future Work . . . . .	35
<b>3</b>	<b>System Identification and Traffic Signal Control for Large Traffic Networks in Urban Areas</b>	<b>36</b>
3.1	Urban Traffic Network and Simulation Model . . . . .	38
3.1.1	Traffic Network . . . . .	38
3.1.2	Microscopic Traffic Simulation Model . . . . .	39
3.2	Traffic Signal Controller Design . . . . .	40
3.2.1	Traffic System Modeling . . . . .	40
3.2.2	Online Parameter Estimation Scheme . . . . .	43
3.2.3	LQR Controller Design . . . . .	45
3.2.4	Adaptive LQR Algorithm Summary . . . . .	45
3.2.5	Robustness of the Proposed Algorithm . . . . .	47
3.2.6	Coordination and Scalability Issues . . . . .	47
3.3	Baseline Methods and Ablation Study . . . . .	48
3.3.1	Baseline Methods . . . . .	48
3.3.2	Ablation Study . . . . .	50
3.4	Results . . . . .	51
3.4.1	Comparison with Baseline Methods . . . . .	52
3.4.2	Ablation Study . . . . .	54
3.5	Conclusion and future work . . . . .	57
<b>4</b>	<b>Network Traffic Signal Control with Adaptive On-line Learning Scheme Using Multiple-Model Neural Networks</b>	<b>60</b>
4.1	System Model and Control Objective . . . . .	61

4.1.1	Traffic Network Model . . . . .	61
4.1.2	Control Objective . . . . .	63
4.2	Nominal Neural-Network Based Design and Parameterization . . . . .	63
4.2.1	Nominal Neural Network Approximation . . . . .	64
4.2.2	Nominal Control Design . . . . .	66
4.3	Multiple-Model Neural Network Based Design and Parameterization . . . . .	67
4.3.1	Multiple-Model based Neural Network Approximation . . . . .	68
4.4	Adaptive Control Design . . . . .	69
4.4.1	Single Neural Network Based Adaptive Control Design . . . . .	69
4.4.2	Multiple-Model Neural Network Based Adaptive Control Design . . . . .	71
4.5	Simulation Study . . . . .	73
4.5.1	Simulation System . . . . .	73
4.5.2	Simulation Results . . . . .	76
4.5.3	Discussion . . . . .	82
4.6	Conclusions . . . . .	83
<b>5</b>	<b>Stochastic Traffic Signal Control Design for Traffic Networks</b>	<b>85</b>
5.1	Problem Statement . . . . .	87
5.1.1	Traffic Flow Modeling . . . . .	87
5.1.2	Traffic Network Modeling . . . . .	91
5.1.3	Static Model . . . . .	93
5.1.4	Control Objective . . . . .	94
5.2	Static Traffic System Modeling . . . . .	95
5.2.1	B-splines Artificial Neural Networks . . . . .	95
5.2.2	Traffic Network Static Modeling . . . . .	96
5.3	Stochastic Control Design for Four Intersection Traffic Network . . . . .	98
5.3.1	Stochastic Control Design . . . . .	99
5.3.2	Simulation Results . . . . .	100
5.4	Conclusion . . . . .	100

**6 Conclusions and Future Work****102**

# List of Figures

2.1	Integrated powertrain and exhaust Simulink model [1]. . . . .	13
2.2	Integration of the hybrid controller. . . . .	14
2.3	Driving path for powertrain model validation [1]. . . . .	15
2.4	Powertrain validation results [1]. . . . .	16
2.5	NOx sensor . . . . .	17
2.6	Exhaust system validation results . . . . .	18
2.7	Parameter co-optimization scheme structure. . . . .	19
2.8	Neural network example. . . . .	24
2.9	Neural network fuel prediction results. . . . .	27
2.10	Neural network training performances for fuel consumption predictions. . . . .	28
2.11	Neural network state constraint prediction results. . . . .	29
2.12	Neural network training performances for state constraint predictions. . . . .	30
2.13	NN catalyst temperature prediction results. . . . .	31
2.14	Optimal system performance. . . . .	33
2.15	Optimal parameter trajectory. . . . .	34
2.16	Fuel saving results. . . . .	34
3.1	Illustration of (a) traffic count by movement data (left) and (b) traffic volume data in the road network [2]. . . . .	39
3.2	Illustration of the VISSIM simulation model for the investigated urban traffic network [2]. . . . .	40

3.3	Illustration of the notations for signal control inputs and traffic delay measurements. . . . .	42
3.4	The adaptive LQR closed loop control structure and the information flow chart. . . . .	46
3.5	Changing of average vehicle delay during the simulation with normal off-peak traffic volume. Solid colored lines represent the mean, and shaded areas represent the mean $\pm$ one standard deviation intervals. Results of the IDQN method were not visualized because of its large delays. . . . .	54
3.6	Distribution of average vehicle delays over the simulation test period under normal off-peak traffic volume for comparison among different initial green times. Results of IDQN were not visualized due to its much larger delays. . . . .	55
3.7	LQR control and changes of N-S green time of the 35 intersections during the simulation test period with different initial N-S green time under normal off-peak traffic volume. . . . .	56
3.8	Results of ablation study: changing of average vehicle delay during the simulation test period with 150% off-peak traffic volume. Solid colored lines represent the mean and shaded areas represent the mean $\pm$ standard deviation interval. . . . .	58
4.1	Traffic network system . . . . .	62
4.2	Two intersection traffic network example. . . . .	74
4.3	Estimation result for $\Delta x$ . . . . .	77
4.4	Estimation error for $\Delta x$ . . . . .	78
4.5	Optimization result with single Neural network. . . . .	80
4.6	Estimation result for $\Delta x$ . . . . .	81
4.7	Estimation error for $\Delta x$ . . . . .	82
4.8	Optimization result with multiple Neural network. . . . .	84
5.1	(a) unbalanced intersection; (b) highly congested intersection; (c) balanced and low wait-time intersection. . . . .	86

5.2	Single intersection example. . . . .	86
5.3	Traffic control signal. . . . .	89
5.4	Traffic queue simulation results . . . . .	90
5.5	Traffic network example. . . . .	91
5.6	Traffic network with four intersections. . . . .	93
5.7	Actual PDF vs target PDF. . . . .	95
5.8	travel queue PDF using B-spline approximation . . . . .	97
5.9	Actual PDF vs target PDF. . . . .	101

# Chapter 1

## Introduction

This chapter gives a brief introduction to the basic concepts used in this dissertation. First, the research background on HEV fuel optimization, traffic signal control, adaptive control, system identification, and stochastic control is introduced and the research motivation is stated. Then, a literature review on these topics is presented. Finally, the dissertation outline is given.

### 1.1 Research Backgrounds and Motivations

This section introduces the research background on the transportation system and control, then addressed the research motivation.

#### 1.1.1 Transportation System

Intelligent Transportation Systems (ITSs) have a great impact on human life as it improves transportation safety effectiveness [3–5]. Developing ITS is a big subject which is formulated with a set of smaller aspects. In this work, vehicle level and traffic network-level research are primarily considered. The two main focuses are *hybrid electric vehicle (HEV) system fuel optimization* and *traffic signal control design for travel delay minimization*.

**Hybrid Electric Vehicle System.** Hybrid electric vehicles (HEVs) have recently exhibited a rising trend in the market because they can achieve higher fuel efficiency than

that of conventional vehicles [6, 7]. The fast development of connected vehicle (CV) techniques has inspired researchers to use traffic information (i.e., vehicle-to-vehicle/vehicle-to-infrastructure [V2V/V2I] information) to develop advanced fuel efficiency schemes for HEVs. Moreover, many state of the art fuel optimization schemes for the HEV did not take the effect on exhaust system into consideration. This has to lead to the urgent need on developing a fuel optimization scheme for the HEVs which uses V2V/V2I information and considers exhaust system effects.

**Traffic Signal Control.** Traffic congestion caused by intersections leads to unnecessary travel delays, reduced traffic safety, and increased energy consumption and environmental pollution [8, 9]. To relieve traffic network congestion in large cities, efficient traffic signal control methods have always been highly demanded, especially with the rapid advances in communication and computing technologies [10, 11]. Traffic signal control includes four primary components: phase specification (sequence of the phases), signal timing/split control (relative green duration of each phase), cycle duration, and offset of cycles for coordination among multiple intersections that are spatially close. Among these components, signal timing/split control can have the most profound impact on traffic operations [10, 12]; hence, this study investigated this research topic.

### 1.1.2 Adaptive Learning Techniques

The transportation system is generally nonlinear and highly integrated. Thus, it is difficult to directly figure out a dynamic model based on the actual physic model. To accurately represent different systems (i.e., HEV system and traffic network system), one needs to develop a system identification scheme to estimates the original system with a well structured linear or non-linear model. As a result, in this work, we use adaptive control and system identification techniques to estimate original plants.

**Adaptive Control.** Adaptive control provides adaptation mechanisms that adjust a controller for a system with parametric, structural, and environmental uncertainties to achieve desired system performance. Payload variation or component aging causes parametric un-

certainties. Component failure leads to structural uncertainties, and the external noises are typical environmental uncertainties. Such uncertainties often appear in automobile engines, electronic devices, and other industrial processes. Adaptive control has experienced many successes in new challenging problems and their encouraging solutions.

Unlike other controllers using PID, pole placement, optimal, robust, or nonlinear control methods, whose designs are based on the certain knowledge of the system parameters, adaptive controllers do not need such knowledge of the system parameters; they are adapted to parameter uncertainties by using performance error information on-line [13]. There exist mainly two types of adaptive control: direct adaptive control and indirect adaptive control.

**System Identification.** System identification is a methodology for building mathematical models of dynamic systems using the measurements of the system inputs and outputs. To identify the system dynamics, one could build a white-box model based on first principles, e.g., a model for a physical process from the Newton equations. However, in many cases, such models are overly complicated and possibly even impossible to obtain in a reasonable time due to the complex nature of many systems and processes. A much more common approach is, therefore, to start from measurements of the behavior of the system and the external influences (inputs to the system) and try to determine a mathematical relation between them without going into the details of what is actually happening inside the system. This approach is called system identification. Adaptive parameter estimation is a frequently used system identification algorithm, which is a dynamic estimation procedure that makes use of updated system signals to produce updated parameter estimates on-line [13].

The definition of system identification can be included in the area of machine learning. With the fast development of recurrent neural networks and reinforcement learning, solving the dynamic problem has become a new objective for machine learning.

### 1.1.3 Optimization and Control

With the system being accurately identified, one needs to develop different advanced control strategies to achieve the objectives. In this work, our primary control objectives are

*energy consumption minimization* and *achieve smooth traffic*. An optimal control design can be used to achieve minimal energy consumption. To achieve smooth traffic, the goal is to have the traffic flow distribution of every intersection follows a given distribution, where this problem solved using stochastic control.

**Optimal Control.** Classic control design can satisfy some prescribed performance requirements, but the performance can not be not optimized. The modern optimal control theory, on the other hand, relies on designing techniques to maximize or minimize a performance index, resulting in a designed system that is optimal in the prescribed sense. Optimal control is an extension of the calculus of variations and is a mathematical optimization method for deriving control policies [14, 15]. To minimize fuel consumption for HEVs and minimize traffic delays in a traffic network, using optimal control is a natural approach.

**Stochastic Control.** Stochastic control has been an important subject in control theory. Most of the practical systems in control engineering are subjected to random inputs from either sensor noises, or random disturbances or random parameter changes [16]. Unlike optimal traffic signal control using a dynamic traffic network model whose objective is to minimize traffic delay directly, the objective for stochastic traffic control is to achieve "smooth" traffic. The definition of smooth traffic in terms of stochastic control is to have every traffic intersection achieve balanced and low wait-time.

## 1.2 Literature Review

This section presents a brief review of the research on HEV power management and traffic signal control. Then, a summary of the problems studied in this research is given.

### 1.2.1 HEV Power Management

Typical power management methods for HEVs are either *optimization-based methods* or *rule-based methods*. In [17–19], novel power management strategies were developed using dynamic programming-based methods and particle swarm optimization approaches. In

[20–23], the authors designed some direct optimal control strategies to optimize HEV power management. In recent studies, many researchers have incorporated traffic information into the fuel optimization design. In [24], the authors designed a look ahead prediction-based real-time optimal energy management method to reduce fuel consumption using CV information. In [25], the authors developed a stochastic optimal energy control strategy under different traffic scenarios. Because most optimal power management strategies result in the increase of harmful gas emissions, the trend is to consider the thermal response of the exhaust system as an additional constraint in the optimization. In [26–29], the authors considered the thermal response of the exhaust system in their optimal power management strategy design. In [30], the authors aimed at reducing pollution while considering mainly the effect of fuel consumption. In [31], the authors analyzed the effect of the thermal response to a connected HEV system, and they optimized the vehicle speed and power split of the system.

Rule-based power management methods have been widely used for vehicle power management, which is built based on "if-then" rules, making it suitable for fast real-time implementations. For example in [32], the authors developed a rule-based power management method using a power follower control strategy. In that paper, the Toyota Prius Hybrid model is studied, which uses a rule-based power management method.

Some research has focused on optimizing rule-based power management. For example, in [33] and [34], the authors optimized some control parameters using genetic algorithms for the HEVs to maximize fuel efficiency. In [34], a particle swarm optimization strategy is described that determines a proper driving torque demand. These two works constitute useful approaches to optimizing some parameters in the powertrain to save fuel. However, traffic information is not considered in these methods. To make use of V2I for better power management, [35], a fuzzy rule-based technique was formulated to maximize fuel efficiency.

While the above methods can achieve optimal fuel efficiency, the effects of such power management strategies on vehicle emissions have not been explored, and this has become an outstanding issue for the implementation of the above power management methods to

vehicles on the market. The exhaust emission system uses a three-way catalyst system to convert harmful emission gases such as  $\text{NO}_x$ , HC, and CO to  $\text{N}_2$ ,  $\text{H}_2\text{O}$ , and  $\text{CO}_2$ . The efficiency of these catalyst reactions depends on temperature. Thus, optimizing fuel efficiency by increasing the engine-off time would significantly increase the emission of harmful gases. At present, this effect is seldom considered in the optimization problem for HEVs. In [36], the authors described a rule-based power management strategy, and to that achieves a near-optimal fuel economy and emission performance with the use of the dynamic programming (DP) methodology. However, this work primarily aimed at emission reduction instead of saving fuel, which results in a significant emission reduction at a small loss of fuel economy.

### 1.2.2 Traffic Signal Control

The early investigation of urban traffic signal control dates back to the 20th century when traffic lights were first invented. Since then, a variety of signal timing control methods have been developed. Generally, these methods can be divided into two categories: pre-timed and traffic-responsive [8]. Pre-timed methods adopt constant green time splits based on historical traffic demands over the considered signalized urban area. One of the representative pre-timed traffic signal control systems is TRANSYT (TRAffic Network Study Tool) [37], which tries to minimize the overall travel time, delays, and the number of stops. With a fixed green time duration, nevertheless, pre-timed control methods may not be able to handle the dynamics of real-time traffic conditions.

Aiming at addressing the limitation of pre-timed control, traffic-responsive methods optimize signal timing based on real-time traffic data. The most common traffic-responsive signal control method is actuated control. It uses sensors (e.g., inductive loop detectors) located upstream of the stop line to sense the request of green time. The method also pre-defines minimum and maximum green times and passage time. On the basis of minimum green time, actuated control extends the green time by the amount of passage time once a vehicle is detected until the maximum green time is reached [8].

Although actuated control is responsive to traffic dynamics to some extent, it is ide-

ally suited for isolated intersections in which traffic demands and patterns vary widely throughout the day [38]. To improve traffic efficiency at a network-wide level, plenty of emerging traffic signal control methods have been proposed, such as evolutionary algorithms [39, 40], heuristics approaches (e.g., max pressure [41] and self-organizing traffic light controls (SOTL) [42]), fuzzy logic control [43, 44], neural networks control [45, 46], reinforcement learning based control [47, 48], and deep reinforcement learning control [49–52]. To consider multiple evaluation metrics (e.g., traffic throughput, traffic delay, safety, and energy consumption), multi-objective optimization algorithms have also been applied in traffic signal controls to optimize the system performances [11, 53, 54]. However, they are not widely deployed in real-world intersections because of their complex or black-box logics, high-performance variances, and special hardware requirements [55].

For network-level traffic signal control, SCATS (Sydney Coordinated Adaptive Traffic System) [56], SCOOT (Split Cycle Offset Optimization Technique) [57], OPAC (Optimization Policies for Adaptive Control) [58], and TUC (Traffic-responsive Urban Control) [59, 60] are the most widely used traffic-responsive signal control systems [8, 10]. Both SCOOT and OPAC incorporate a network model that uses real-time traffic data as an input and computes the corresponding traffic network performance indices such as the total number of vehicle stops. The primary difference between SCOOT and OPAC is that the former uses heuristic rules whereas the latter uses dynamic programming. For SCOOT, the central control computer repeatedly runs the network model and determines the effect of incremental changes of splits, offsets, and cycle lengths at individual intersections. The changes will be submitted to the local signal controllers if they are predicted to be beneficial in terms of performance indices. OPAC assumes the phase switching time as a discrete variable and calculates the optimal switching time with dynamic optimization [8]. TUC adopts a store-and-forward modeling of the urban network traffic and uses the linear-quadratic regulator (LQR) theory [61]. The design of TUC results in a multivariable regulator for traffic-responsive coordinated network-wide signal control [60]. Similar to pretimed control at isolated intersections, coordinated pretimed signal control at the network level is limited

because it uses of historical data rather than real-time traffic information. Coordinated actuated control, on the other hand, although being potentially more efficient than pretimed control, can also be computationally expensive.

### 1.2.3 Summary

From the discussion above, one can see that there are many urgent requests for developing advanced control schemes for the ITS. This motivated our research on using adaptive learning methods and optimal control to design new controllers to meet transportation system demands.

In this research, we study and develop some new solutions to the following problems:

Problem 1: Fuel optimization for hybrid electric vehicle systems.

Problem 2: Traffic signal control problem for traffic networks in the urban area.

## 1.3 Dissertation Outline

The dissertation is organized as follows:

- In Chapter 1, the basic introduction of this research is given, including the research background and the literature review about the related areas.
- In Chapter 2, the system identification problem and the fuel optimization problem for hybrid electric vehicles are solved.
- In Chapter 3, the traffic signal control problem for large traffic networks in urban areas is introduced, and the system identification based LQR controller is designed to meet the control objectives.
- In Chapter 4, the system identification and control problem for a traffic network with a non-linear equivalent model is presented. Online-learning based multiple neural network models are used to represent the nonlinearity of the traffic network system.

- In Chapter 5, a stochastic control approach toward traffic signal control for a traffic network system is designed to achieve a smooth traffic flow.
- In Chapter 6, some concluding remarks are given.

## Chapter 2

# System Identification and Fuel Optimization for HEV Systems

This chapter develops a parameter co-optimization scheme for HEV powertrain and exhaust system using V2V/V2I information.

Rule-based power management methods are widely used, but few reports describe approaches to design optimal rule-based power management schemes using V2V and V2I. Moreover, the existing methods for powertrain controls for HEVs have not considered the simultaneous use of V2V or V2I information and the influence of the exhaust emission systems on powertrain controls. To address these outstanding issues, a parameter co-optimization scheme is developed in this chapter for HEV powertrain and exhaust systems using V2V/V2I information.

The *main contributions* of this work are listed as follows:

- Formulate the co-optimization problem, the solution to which optimizes fuel consumption using traffic information and emission constraints.
- Build a simplified yet equivalent neural network (NN) model for the optimization problem based on the first-principles models for powertrain and exhaust system.
- Design an optimal powertrain control parameter tuning strategy to map the optimal

control parameter tuning with future traffic information.

## 2.1 Hybrid Electric Vehicle System Modeling

In this section, the dynamics of the HEV is first introduced. Then, the first-principles model based on Matlab/Simulink is validated using actual vehicle driving cycle testing data obtained from a 2017 Toyota Prius Hybrid vehicle. The model is used to obtain the equivalent NN model for the proposed co-optimization design in this work.

### 2.1.1 Dynamic Model of HEV

The Toyota hybrid systems (THS) dynamic model is derived based on previous studies [62, 63]. The THS uses a planetary gear as the power split device. The planetary gear system dynamics derived in [62] are summarized as follows:

$$(I_s + I_{MG1})\dot{\omega}_{MG1} = FS - T_{MG1} \quad (2.1)$$

$$(I_c + I_e)\dot{\omega}_e = T_e - FR - FS \quad (2.2)$$

$$\begin{aligned} \left(\frac{R_{tire}^2}{K}m + I_{MG2}K + I_rK\right)\dot{\omega}_r &= (T_{MG2} + FR)K \\ -T_f - mgf_r R_{tire} - 0.5\rho AC_d \left(\frac{\omega_r}{K}\right)^2 R_{tire}^3 & \end{aligned} \quad (2.3)$$

where  $I_s$ ,  $I_r$ , and  $I_c$  are inertia values of the sun gear, ring gear, and carrier gear, respectively;  $I_{MG1}$  and  $I_{MG2}$ , and  $I_e$  are the first/secondary motor/generator set inertia, and the engine inertia;  $T_{MG1}$ ,  $T_{MG2}$ , and  $T_e$  are the corresponding torques, and  $\omega_{MG1}$  and  $\omega_e$  are the corresponding rotary speeds. In the above equations,  $F$  is the internal force on the pinion gear,  $S$  and  $R$  are the number of teeth on the sun gear and ring gear, respectively,  $R_{tire}$  is the tire radius,  $m$  is the vehicle mass,  $K$  is the final drive ratio,  $\omega_r$  is the rotational speed of the ring gear,  $f_r$  is the rolling resistance coefficient,  $T_f$  is the brake torque, and  $0.5\rho AC_d$  is the aerodynamic drag resistance coefficient.

To consider the efficiencies of the motor and generator sets, the battery power is given

by

$$P_b = T_{MG1} \omega_{MG1} \eta_{MG1}^k \eta_{i1}^k + T_{MG2} \omega_{MG2} \eta_{MG2}^k \eta_{i2}^k, \quad (2.4)$$

where  $\eta_{MG1}$  and  $\eta_{MG2}$  are the efficiencies of the motor/generator sets,  $\eta_{i1}$  and  $\eta_{i2}$  are the corresponding inverter efficiencies. The exponent  $k = -1$  indicates that the battery is discharging, and  $k = 1$  means the battery is charging. Battery state of charge (SOC) dynamics are given as follows [62]:

$$\dot{SOC} = -\frac{V_o - \sqrt{V_o^2 - 4P_b R_b}}{C_{batt}}, \quad (2.5)$$

where  $V_o$  is the open circuit battery voltage,  $R_b$  is the battery internal resistance, and  $C_{batt}$  is the maximum battery capacity.

Equations (2.1) - (2.5) represent the dynamics of the Toyota hybrid system or the plant to be controlled, where equations (2.1) - (2.3) are the dynamics of the powertrain mechanic path, and (2.4) - (2.5) are the dynamics of the powertrain electric path. They will form the first-principles model together with the controls for the powertrain system. This model will be used to generate an equivalent NN model for the construction of the proposed co-optimization algorithm.

### 2.1.2 First-Principles Model Validation

A Simulink model was built as shown in Fig. 2.1. Equations (2.1) - (2.5) were used to model the existing control for the powertrain system and the dynamics of the plant to be controlled. In this figure, in addition to the models given in (2.1) - (2.5), it can be seen that the closed-loop system for the powertrain has functional blocks for a hybrid controller, a driver, and a driving cycle built into the Simulink model. In this context, the hybrid controller consists of several components, such as motor/generator speed controllers, an engine optimal controller, a battery charging controller, a power management controller. Fig. 2.2 shows how each controller is integrated inside the hybrid controller. The "Driver" function block in Fig. 2.1 is used to generate gas and brake signals to the hybrid controller

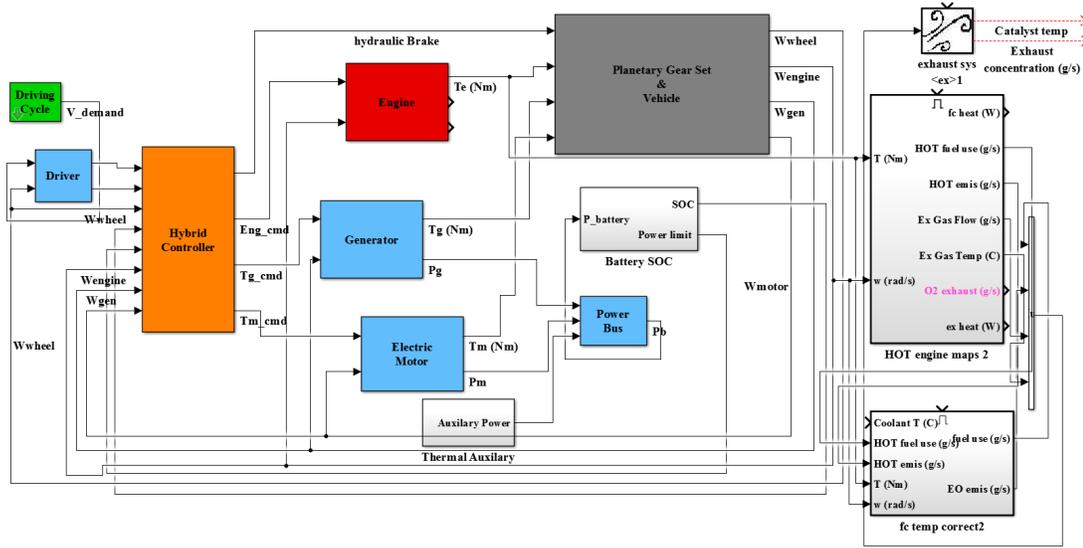


Figure 2.1: Integrated powertrain and exhaust Simulink model [1].

based upon the driving cycle and wheel speed information. Therefore, the Simulink system in Fig. 2.1 stands for a closed-loop control system for the powertrain operation. The Simulink model was built based upon [62, 63]. The exhaust system is built based on a MATLAB package for exhaust systems named ADVISOR, which was developed by the National Renewable Energy Laboratory (USA) with proper validations.

The main idea of this work is to use the closed-loop system model shown in Fig. 2.1 to establish a co-optimization algorithm that is capable of tuning some of the important parameters embedded in the hybrid controller using V2V and V2I information, so that the fuel consumption can be minimized with satisfactory exhaust emission. For this purpose, this closed-loop Simulink model will be validated first using real driving data.

**Driving cycle information.** Fig. 2.3 shows the vehicle driving route around the University of Michigan, from which driving cycle data were collected and used for the validation of the Simulink model. Data from different driving scenarios (weather, temperature, and traffic conditions) are used. This driving cycle information is used in the HEV model validation and the simulation study.

**Powertrain model validation.** In the validation, we updated multiple mapping functions (including engine torque and engine speed mapping, engine torque and engine fuel con-

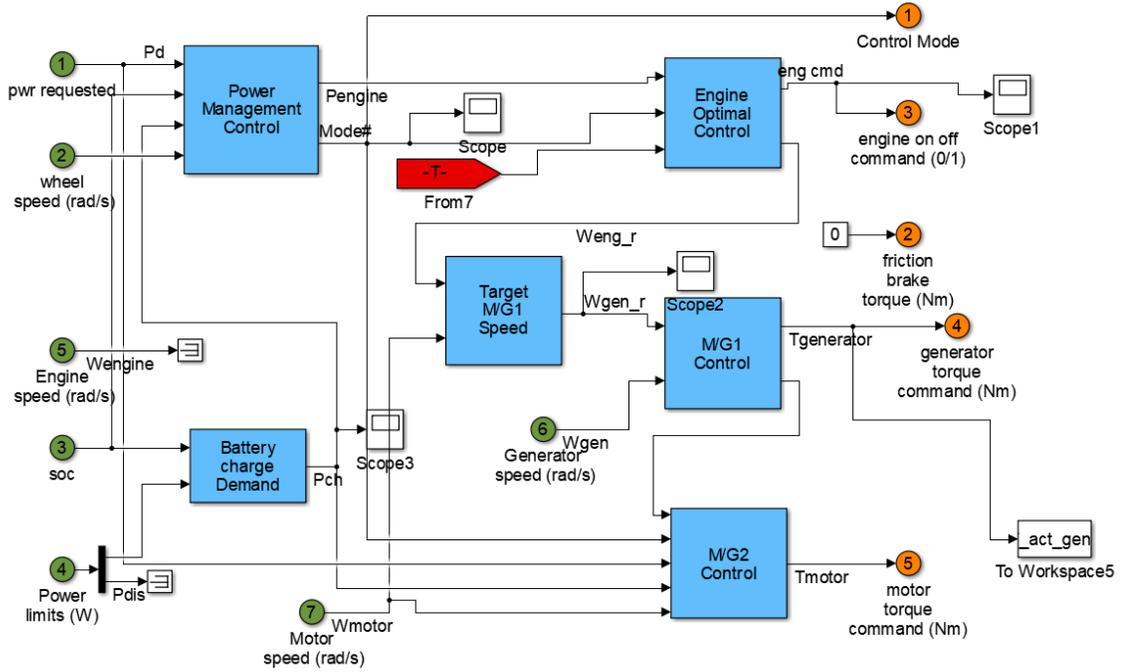


Figure 2.2: Integration of the hybrid controller.

sumption mapping) in the Simulink model using the measurements of the generator speed, engine speed, motor speed, generator torque, and engine torque in several test drives. Fig. 2.4 gives the powertrain validation results, where the red curves are the measured powertrain state data, and the blue curves are the simulation results generated from the Simulink model in Fig. 2.1. Based on the figure, one can see that the speed states are well validated, while the torque states can also indicate satisfactory performance from validation.

**Exhaust model validation.** To validate the exhaust system model, we measured the mass flow rate, catalyst temperature, and NO<sub>x</sub> emission rate. We constructed experiments to test the NO<sub>x</sub> emission and used the data for exhaust model validation. Since the NO<sub>x</sub> data before the catalyst were not available, we used the ratio between before-catalyst NO<sub>x</sub> data from the simulation and the catalyst temperature, and the ratio between the actual NO<sub>x</sub> data from the sensor installed in the exhaust pipe of the testing vehicle and the catalyst temperature to find the light-off temperature. Fig. 2.5 shows the NO<sub>x</sub> sensor used in this study. To validate NO<sub>x</sub> emission, we tuned the scaling parameters in the ADVISOR model.

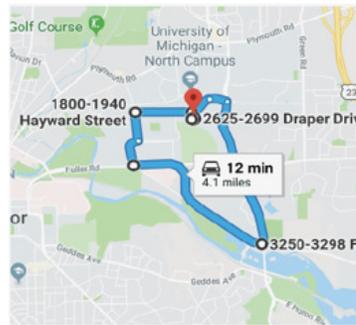


Figure 2.3: Driving path for powertrain model validation [1].

It can be seen that validating NOx emission could improve the simulation performance of HC and CO emission. Fig. 2.6 shows the validation results for the exhaust system, where one can see that the exhaust catalyst temperature and exhaust air flow rate and engine on/off status are well validated. However, some differences can be seen between the curves for NOx emission. One possible reason is that since the NOx sensor is attached at the end of the exhaust pipe, the NOx emission may have been diluted, which would cause the measured NOx emission rate to be lower than the simulation results.

## 2.2 Objective and Problem Statement

In this section, the objective of this work is first introduced. Then the co-optimization problem is formulated.

### 2.2.1 Control Objective

The objective of this work is to develop a co-optimization scheme to use V2V and V2I information to tune some important powertrain control parameters embedded in the hybrid controller block in Fig. 2.1. The selected control parameters are sensitive to fuel consumption and the co-optimization based tuning strategy is aimed at achieving the following goals:

- Maximize fuel saving by taking into account future driving conditions obtained from V2V/V2I information;
- Increase the speed for the catalyst temperature to reach the light-off temperature.

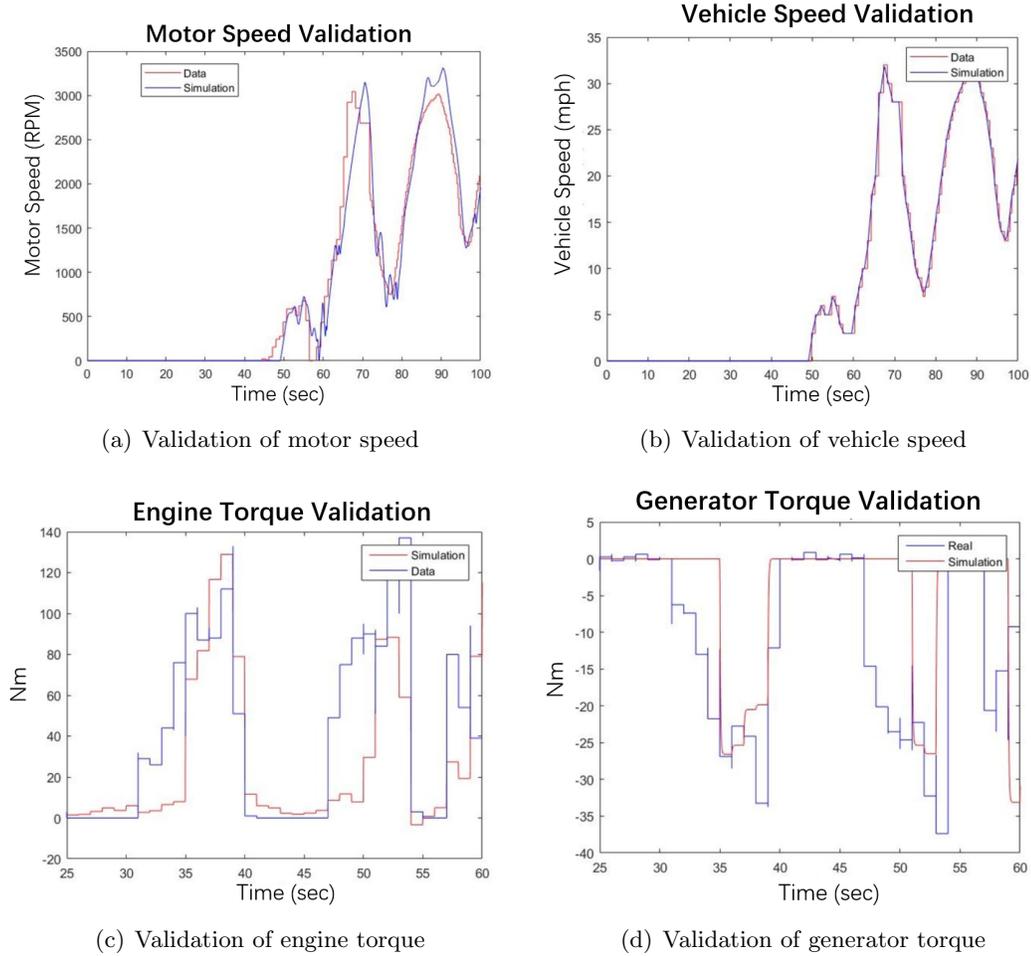


Figure 2.4: Powertrain validation results [1].

**Co-optimization design structure.** Fig. 2.7 shows the parameter co-optimization scheme. The powertrain controller parameters are tuned based on the co-optimization scheme, which is derived from the V2V/V2I inputs and uses the equivalent NN model as the HEV system model to formulate the optimal tuning of sensitive control parameters in a simple way. The optimally tuned controller parameters are used for vehicle control, which controls the HEV system to generate desired system outputs, such as engine torque/speed and catalyst temperature. For this purpose, the selection of key parameters embedded in the hybrid controller function block will be described first.

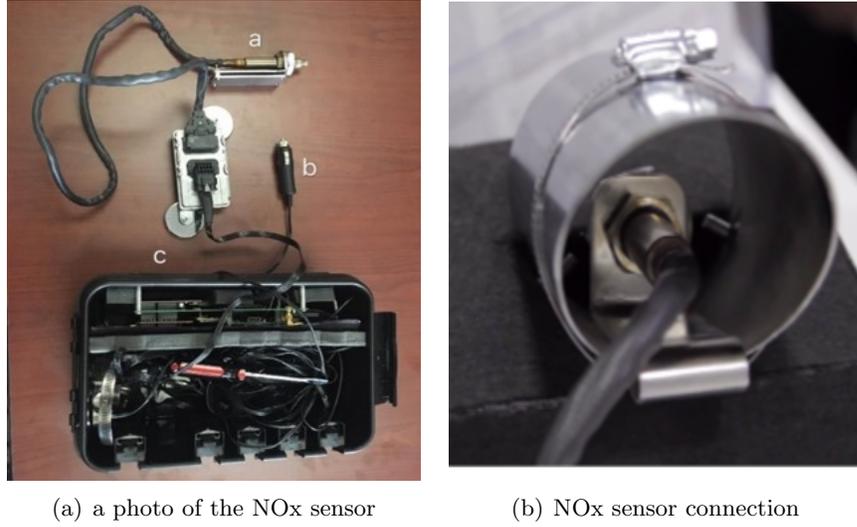


Figure 2.5: NOx sensor

### 2.2.2 Parameter Selection

For the powertrain model with hybrid controller in Fig. 2.2, there are many control parameters involved. Based on our tests, the tuning of those parameters may affect fuel consumption. Thus, this work first carries out a sensitivity study on those control parameters to choose the ones that are most sensitive to fuel consumption. In this section, the fundamentals of parameter sensitivity analysis are introduced.

**Bayesian definition of sensitivity.** For a system represented by input  $X$  and output  $Y$ , sensitivity in the Bayesian sense can be expressed as

$$\Gamma = \frac{\text{var}_X[E(Y|X)]}{\text{var}(Y)}. \quad (2.6)$$

In this work,  $Y$  is the fuel consumption and  $X$  is the powertrain control parameters inside the hybrid controller, the variance  $\text{var}_X$  is taken over all the possible values of  $X$ , and  $E(Y|X)$  is the expectation of  $Y$  that is conditional on an observed value on  $X$ .

In this work, the first-order sensitivity index to the total fuel consumption is calculated using the Fast Bayesian method [64]. Table 2.1 shows the selected control parameters, which are most sensitive to fuel consumption changes. One control parameter was chosen for the

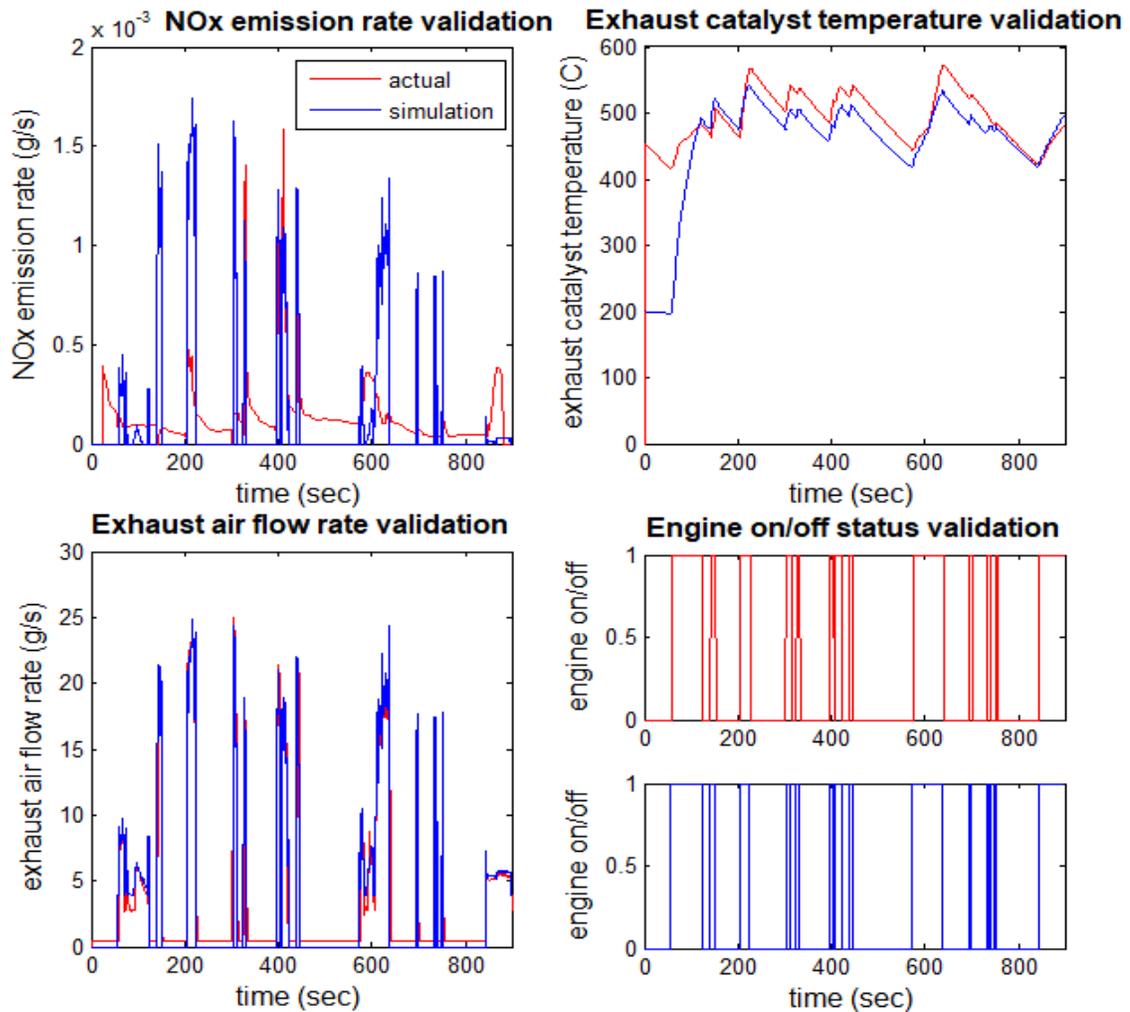


Figure 2.6: Exhaust system validation results

battery charging controller; two were chosen for the generator speed controller of the first motor/generator set (MG1), denoted as  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , respectively. Other control parameters are sensitive to fuel consumption changes. However, to simplify the scale of calculation in the optimization problem, only these three parameters, which are most sensitive to fuel consumption, are tuned in this study.

The purpose is, therefore, to establish an optimal tuning rule for  $\{\theta_1, \theta_2, \theta_3\}$  using V2V/V2I information so that the fuel consumption is minimized with satisfactory exhaust emission.

**Battery charging controller.** The battery charging controller uses saturated propor-

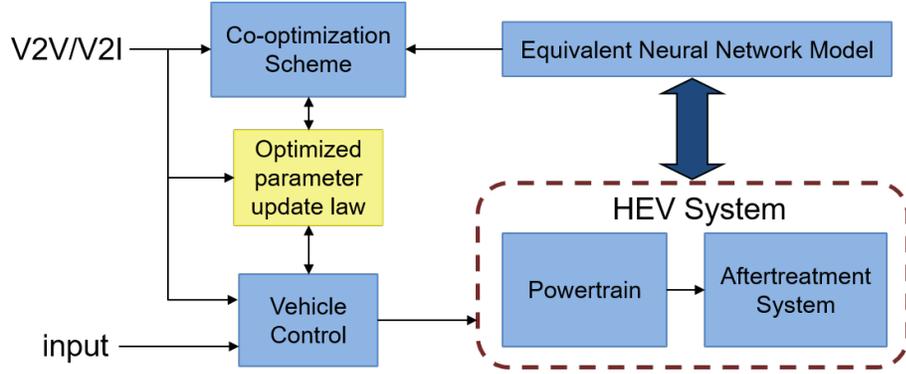


Figure 2.7: Parameter co-optimization scheme structure.

Table 2.1: Three control parameters selected for parameter tuning

Parameter Name	Physical Meaning	Nominal Value	Range
$\theta_1$	MG1 controller proportional gain	0.9	[0.5,1]
$\theta_2$	MG1 controller integral gain	0.005	[0.0001,0.05]
$\theta_3$	Battery charging controller gain	0.005	[13000,17000]

tional control. The objective of the battery charging controller is to have the SOC track a given optimal SOC as follows:

$$u_p(t) = \min(K_p(SOC^*(t) - SOC(t)), u_{pmax}), \quad (2.7)$$

where the control input  $u_p = P_b$  is the battery charging power,  $u_{pmax}$  is the upper bound of charging power, and  $SOC^*$  is the reference optimal state of charge. For this part of the system,  $K_p = \theta_1$  is the proportional gain of the controller that is to be tuned by the co-optimization algorithm.

**MG1 generator speed controller.** The MG1 speed controller uses a classic proportional-integral controller to have the generator speed track an optimal reference speed as follows

$$u_{MG1}(t) = K_{p1}(\omega^*(t) - \omega_{MG1}(t)) + K_{i1} \int_0^t (\omega^*(\tau) - \omega_{MG1}(\tau)) d\tau, \quad (2.8)$$

where the control input  $u_{MG1} = T_{MG1}$  is the generator torque,  $\omega^*$  is the referenced optimal

generator speed,  $\omega_{MG1}$  is the actual generator speed,  $K_{p1} = \theta_2$ ,  $K_{i1} = 0.05\theta_3$  are the proportional and integral gains of the controller that are selected as two key parameters to be tuned by the co-optimization algorithm.

### 2.2.3 Use of V2V/V2I Information

To utilize traffic information for the optimization framework for the tuning of  $\{\theta_1, \theta_2, \theta_3\}$ , we need to characterize V2V/V2I information. The goal is to use V2V/V2I information as extra tuning optimization input and to solve for desired controller parameter tuning with given traffic information represented by V2V/V2I to achieve optimal fuel-saving (i.e., to find the cost function with controller parameter tuning and traffic information being the inputs). Thus, we need to characterize V2V/V2I information in mathematical terms. The components of V2V information are the speed and acceleration of the preceding vehicle and the left/right front vehicles, denoted as  $v_p, a_p, v_l, a_l, v_r$ , and  $a_r$  respectively. To capture V2I information, we assume the signal light to be the primary infrastructure that is communicating with the vehicle. As a result, we use the traffic light signal  $S_T$  to represent V2I information. In this work, we assume that with the V2V/V2I information given, one can accurately predict the vehicle future speed  $v_f$  with

$$v_f(t) = W(S_T, v_p, a_p, v_l, a_l, v_r, a_r), \quad t \in [t_p, t_p + t_{max}], \quad (2.9)$$

where  $t_p$  is the present time, and  $t_{max}$  is the maximum prediction length,  $W$  represents the future speed prediction function [65, 66]. With given V2V/V2I information, one can make  $t_{max}$  seconds of future vehicle speed prediction.

For this study, we assume that the future traffic prediction is available for the control design. That is, the V2V/V2I information is used for optimization. In simulation tests, we can obtain the driving cycle information. Adding some white noise to the driving cycle information, one can be used to mimic the prediction of the future driving information.

### 2.2.4 Co-Optimization Problem Formulation

To design parameter co-optimization schemes, one can see that it is difficult to directly formulate an optimization problem based on the dynamic system in (2.1) - (2.5) and the controllers in (2.7) - (2.8). Thus, it is necessary to directly formulate input-output model for optimization study.

**Cost function.** The following cost function defined for the co-optimization problem indicates the relationship between the fuel consumption of the HEV with the selected three control parameters and the future driving information obtained using V2V/V2I, which can be derived in the discrete time as

$$J = fuel(nT) = \sum_{k=1}^n \Delta f(\theta(k), cyc(k)), \quad (2.10)$$

where  $J$  is the fuel consumption,  $\theta = [\theta_1, \theta_2, \theta_3]^T$  are the three chosen powertrain control parameters,  $cyc(k) = [v_f(k+1), v_f(k+2) \dots, v_f(k+k_{max})]^T$  indicates  $k_{max}$  steps of available future vehicle speed prediction,  $T$  is the step size,  $n$  is the number of total step, and  $\Delta f$  is the fuel consumption in one time step  $T$ , which is a measurable signal from simulation experiments. This cost function indicates the total fuel consumption function with the change of system parameter and future driving information.

**State constraints.** The HEV dynamics are considered as a constraint to the co-optimization problem to guarantee the correctness of optimization results. The discrete-time state constraint is defined as

$$x(k+1) = g(x(k), \theta(k), cyc(k)), \quad (2.11)$$

where  $x = [T_e, T_g, T_m, \omega_e, \omega_g, \omega_m]^T$  indicates the engine, generator, motor torques, and engine, generator, motor speeds respectively. Function  $g(x(k), \theta(k), cyc(k))$  represents the discrete-time dynamics of the powertrain control system in (2.1) - (2.5) and (2.7) - (2.8).

**Exhaust constraints.** To reduce harmful emissions, the effect of the exhaust system

should also be considered as an additional constraint for fuel optimization. In this work, the catalyst temperature dynamics are considered to be the exhaust constraint because the emission rates of NOx, HC, and CO are significantly reduced when the catalyst temperature reaches the light-off temperature. In this case, the discrete-time catalyst temperature dynamics can be expressed as

$$\Delta T_{cat}(k+1) = h(T_{cat}(k), z(k), \theta(k), cyc(k)), \quad (2.12)$$

where  $\Delta T_{cat}(k+1) = T_{cat}(k+1) - T_{cat}(k)$  is the finite difference,  $T_{cat}$  is the catalyst temperature, and  $z = [T_e, \omega_e, C_{on}, r_{air}, T_{cool}]^T$  indicates the engine torque, engine speed, vehicle on/off command, exhaust airflow rate and coolant temperature, respectively. Function  $h(T_{cat}(k), z(k), \theta(k), cyc(k))$  is the dynamics of the catalyst temperature. The purpose of imposing an exhaust constraint is to guarantee that the speed for catalyst temperature to reach the light-off temperature with optimal parameters is faster than the speed with nominal parameters, which is given by

$$\Delta T_{cat}(i) \geq \Delta T_{nom}(i), \quad i = 1, \dots, i_o, \quad (2.13)$$

where  $i_o$  is the time step needed for the catalyst temperature to exceed the light-off temperature and  $\Delta T_{nom}(k)$  indicates the catalyst temperature difference with nominal controller parameters.

From the discussion above, one can see that there exists a relationship between the controller parameters and fuel efficiency. Moreover, as mentioned in the previous section, V2V/V2I information can be characterized to predict future vehicle speed. It is a natural approach to use this information to pre-organize power management. Thus, in this study, we propose the following co-optimization problem.

**Co-optimization problem.** Minimize the fuel consumption represented by the cost function  $J$  in (2.10) of the HEV with the selected controller parameters  $\{\theta_1, \theta_2, \theta_3\}$  as a set of decision variables and the future driving speed  $cyc$  as an additional input to the

cost function (2.10). At the same time, consider the catalyst temperature described in (2.12) - (2.13) and the HEV dynamics in (2.11) as optimization constraints to guarantee the emission reduction and proper HEV operation.

**Trade-off between fuel saving and emission reduction.** As mentioned in Section I, most methods to minimize fuel consumption will result in the increase of harmful gas emissions. It is difficult to reduce fuel consumption and harmful gas emission at the same time. In this work, our primary goal is to achieve fuel saving. In the mean time, the effect of the exhaust system is considered as an additional constraint to the fuel optimization problem.

## 2.3 Co-Optimization Scheme Design

In this section, the equivalent optimization model using an NN is derived first. Then the optimization problem is solved to obtain the optimal parameter tuning strategy.

### 2.3.1 Equivalent NN Models

For the optimization problem formulated in (2.10) - (2.13), the relationship between fuel consumption and powertrain control parameters is highly nonlinear as it can be seen from (2.1) - (2.5) and (2.7) - (2.8). Although one can solve the co-optimization problem directly from these first-principles models, the calculation is significant because these three parameters must be tuned simultaneously. Thus, this work develops a simplified yet equivalent NN model to represent the optimization framework. This allows the optimization problem to be much more simply solvable based on the equivalent NN model [1, 67].

Artificial Neural Networks (ANN) have been successfully used in nonlinear complex function approximations and many areas of artificial intelligence studies [68]. There have been many neural network structures for learning algorithms. Multi-layer feed-forward (MLF) neural networks are the most commonly used neural networks, and they have been applied to a wide variety of scientific problems [69]. Fig. 2.8 is an example of an MLF neural network. A neural network consists of many neurons in each layer, where every neuron has

a bias, weight coefficient and activation function. Each neuron receives signals from other connected neurons and sends signals to the next layer by passing through its activation function. The neural network is trained using back-propagation training algorithms.

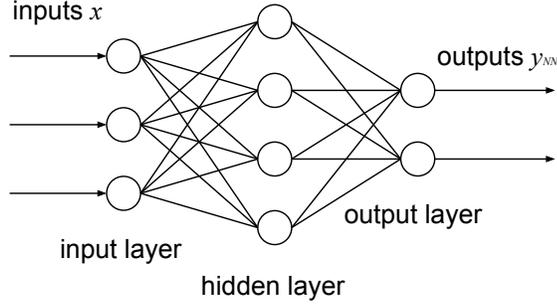


Figure 2.8: Neural network example.

**NN approximation for nonlinear systems.** Consider the nonlinear system given as

$$y(k) = f(x(k)) \quad (2.14)$$

where  $x(k)$  is the system input and  $y(k)$  is the system output. the above nonlinear system can be approximated with a neural network model as

$$y_{NN}(k) = w_2 \phi(k) + b_2 \quad (2.15)$$

$$\phi(k) = L(\psi(k)) \quad (2.16)$$

$$\psi(k) = w_1 x(k) + b_1 \quad (2.17)$$

where,  $x(k) \in R^n$  are the neural network inputs,  $y(k) \in R^m$  are the neural network outputs,  $w_1 \in R^{n \times m}$ ,  $w_2 \in R^m$  are the weights,  $b_1 \in R^n$ ,  $b_2 \in R$  are the bias,  $\phi(k)$  corresponds to the output of the hidden layer,  $L$  is the activation function (such as 'Relu' function) of the hidden layer,  $\psi(k)$  corresponds to the output of the input layer. The weight and bias update rule for the neural network is a gradient based adaptation given by:

$$w_N^{i+1} = w_N^i + \Delta w_N^i \quad (2.18)$$

$$b_N^{i+1} = b_N^i + \Delta b_N^i \quad (2.19)$$

$$\Delta w_N^i = -\alpha \frac{\partial E(k)}{\partial w_N^i} \quad (2.20)$$

$$\Delta b_N^i = -\alpha \frac{\partial E(k)}{\partial b_N^i}, \quad N = 1, 2 \quad (2.21)$$

where  $\alpha > 0$  is the learning rate of the neural network,  $i$  is the iteration number,  $\varepsilon(k)$  is the loss function to be minimized by updating the weights. The loss function for the NN training is defined as

$$E(k) = \frac{1}{2} \varepsilon^2(k) \quad (2.22)$$

$$\varepsilon(k) = y_{NN}(k) - y(k), \quad (2.23)$$

where  $y(k)$  is the output obtained from the original system,  $\varepsilon(k)$  is the estimation error between the actual output and the predicted output. With the use of neural network, one can represent the co-optimization problem formulated in (2.10) - (2.13) with neural networks.

**Training procedure** One can use the following procedure to train the NN model described above: (a) use random sampling methods to generate  $N$  different choices of controller parameter dynamics; (b) run the simulation model with different choices of controller parameter dynamics and different driving cycles; (c) collect the time-series-based input/output data for the training of the equivalent NN-based cost function model, state dynamic model, and catalyst temperature dynamic model; (d) train the NN model based on the input/output data collected from the simulation. Using this training procedure, one can obtain the equivalent NN model for the co-optimization problem as follows.

**Cost function approximation.** The NN equivalent expression of the cost function (2.10) is given by

$$\hat{J} = \sum_{k=1}^n \Delta \hat{f}(\theta(k), cyc(k)), \quad (2.24)$$

where  $\Delta \hat{f}$  is expressed by the NN as

$$\Delta \hat{f}(k) = w_{c2}L_1(w_{c1}y_f(k) + b_{c1}) + b_{c2}, \quad (2.25)$$

which is an NN with one input layer:  $y_f(k) = [\theta^T, cy_c^T]^T \in R^{3+k_{max}}$ . The inputs in  $y_f(k)$  are the controller parameters (i.e.,  $\theta = [\theta_1, \theta_2, \theta_3]^T$ ) and the number of available future speed steps, one hidden layer with twenty neurons, one output layer with the output as the fuel consumption;  $w_{c1} \in R^{20 \times (3+k_{max})}$ ,  $w_{c2} \in R^{20 \times 1}$  are the weights;  $b_{c1} \in R^{20 \times 1}$ ,  $b_{c2} \in R$  are the biases; and  $L_1$  is the activation function, which is a ‘ReLU’ function in this case. The prediction results with NN in (2.24)-(2.25) are shown in Fig. 2.9, where the blue curves are the actual fuel consumption and the red curves are the fuel consumption from NN model. Based on the prediction results, one can conclude that the NN model in (2.24) - (2.25) is a good equivalent expression of the cost function (2.10).

**State constraint function approximation.** The NN equivalent model of the state constraints (2.11) is

$$\hat{g} = w_{s2}L_2(w_{s1}y_s(k) + b_{s1}) + b_{s2}, \quad (2.26)$$

which is an NN with one input layer:  $y_s = [x^T, \theta^T, cy_c^T]^T \in R^{9+k_{max}}$ , one hidden layer with twenty neurons, and one output layer;  $\hat{g}$  is the equivalent expression of  $g$  in (2.11);  $w_{s1} \in R^{20 \times (9+k_{max})}$ ,  $w_{s2} \in R^{20 \times 6}$  are the weights;  $b_{s1} \in R^{20 \times 1}$ ,  $b_{s2} \in R^{6 \times 1}$  are the biases; and  $L_2$  is the ‘ReLU’ activation function. The state prediction results from NN are shown in Fig. 2.11. Based on the prediction results, one can conclude that the NN model in (2.26) is a good equivalent expression of the state constraints (2.11).

**Exhaust constraint function approximation.** Similarly, the NN equivalent model for the exhaust constraint (2.12) is given by

$$\Delta \hat{T}_{cat}(k+1) = w_{t2}(L_3(w_{t1}y_t(k) + b_{t1}) + b_{t2}), \quad (2.27)$$

which indicates an NN with one input layer  $y_t = [T_{cat}^T, z^T, \theta^T, cy_c^T]^T \in R^{9+k_{max}}$ , one hidden layer again with twenty neurons, one output layer with its output representing the catalyst

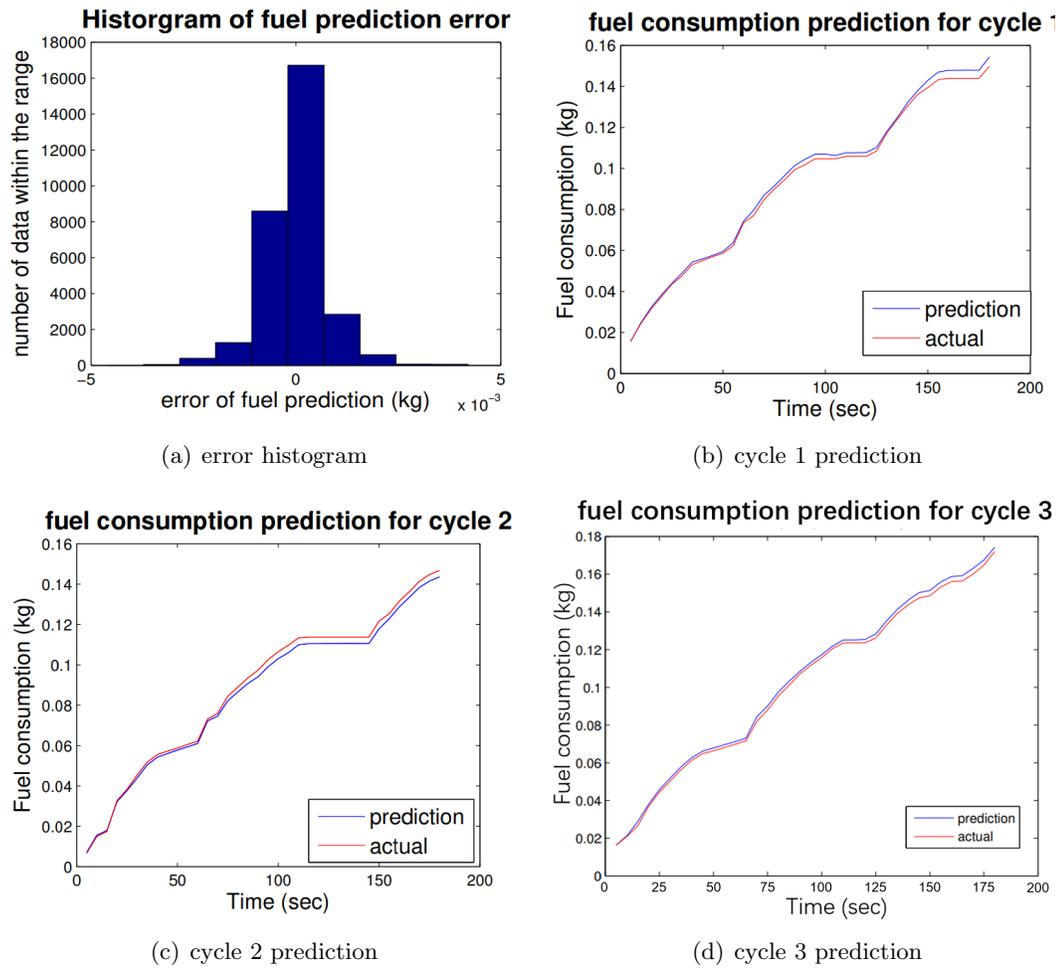


Figure 2.9: Neural network fuel prediction results.

temperature difference. In equation (2.27),  $w_{t1} \in R^{20 \times (9+k_{max})}$ ,  $w_{t2} \in R^{1 \times 20}$  are the weights;  $b_{t1} \in R^{20 \times 1}$ ,  $b_{t2} \in R^{1 \times 1}$  are the biases; and  $L_3$  is the 'ReLU' activation function. The results of prediction are displayed in Fig. 2.11. It can be concluded that the NN represented by (2.27) is a good equivalent expression of the exhaust constraints (2.12).

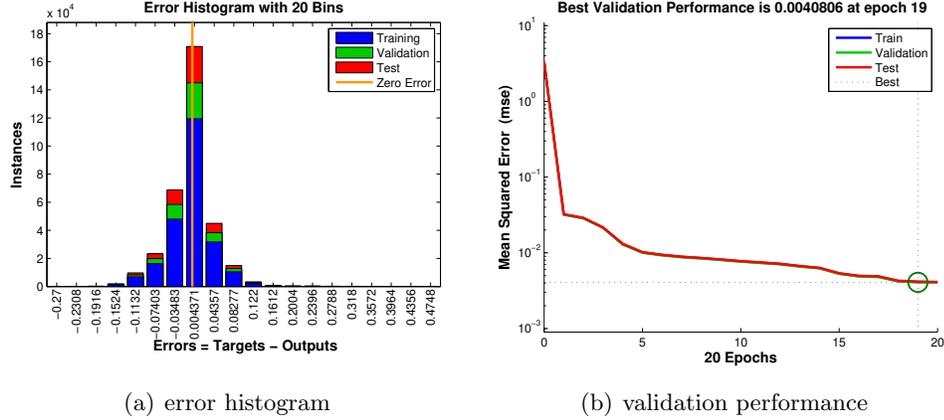


Figure 2.10: Neural network training performances for fuel consumption predictions.

### 2.3.2 Equivalent Optimization Problem Description

Once all the equivalent NN models are obtained, the equivalent optimization problem is to solve the tuning of  $\theta$  to minimize fuel consumption using the equivalent cost function

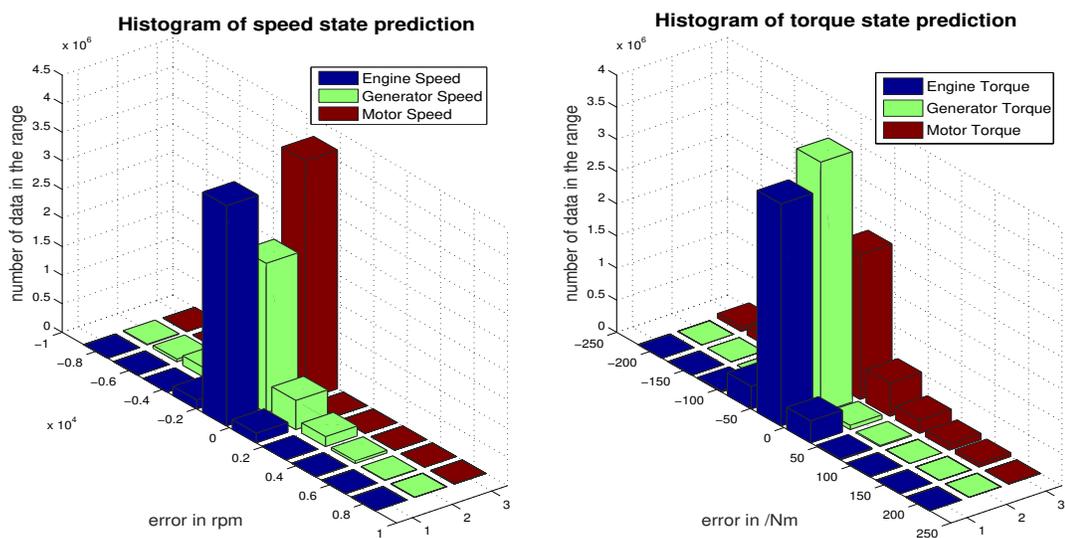
$$\hat{J} = \sum_{k=1}^n \Delta \hat{f}(\theta(k), cyc(k)), \quad (2.28)$$

and satisfy the following equivalent NN represented constraints:

$$x(k+1) = \hat{g}(x(k), \theta(k), cyc(k)) + \epsilon_1 \quad (2.29)$$

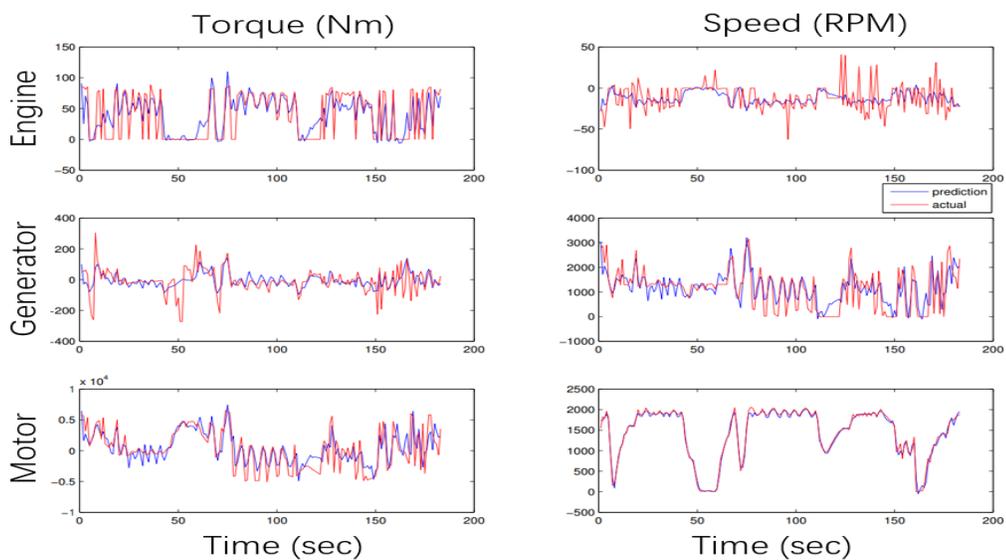
$$\Delta T_{nom} \leq \hat{h}(T_{cat}(i), z(i), \theta(i), cyc(i)) + \epsilon_2, \quad (2.30)$$

where  $\epsilon_1$  and  $\epsilon_2$  are bounded constants that indicate the estimation error upper bound for the state constraint and exhaust constraint respectively,  $i = 1, \dots, i_o$  indicates the time step required for the catalyst temperature to exceed the light-off temperature, and  $\Delta T_{nom}(k)$  indicates the catalyst temperature difference between step  $k$  and  $k - 1$ . Equation (2.30) guarantees that with the choice of optimal controller parameters, the catalyst temperature reaches the light-off temperature no more slowly than the system with a nominal controller parameters.



(a) error histogram for speed states

(b) error histogram for torque states



(c) state prediction

Figure 2.11: Neural network state constraint prediction results.

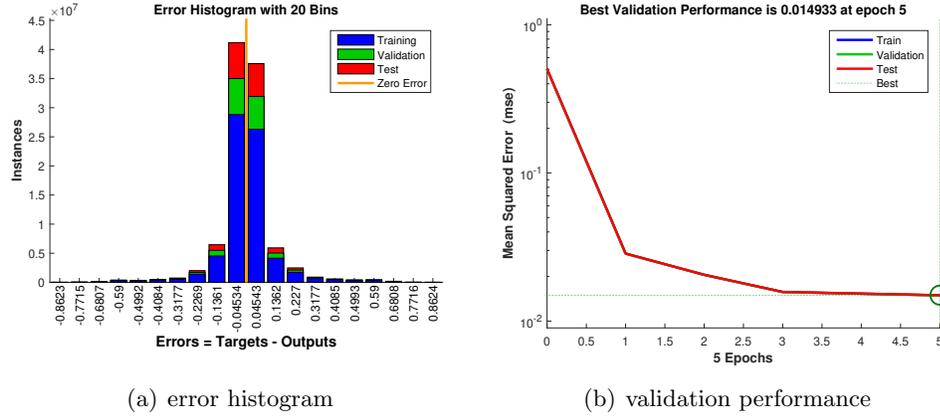


Figure 2.12: Neural network training performances for state constraint predictions.

### 2.3.3 Co-Optimization Based Parameter Tuning Strategy

To derive the parameter tuning law, we need to solve the equivalent co-optimization problem described in (2.28) - (2.30) to obtain tuning of optimal controller parameter  $\theta(k)$  as a function of  $\{x(k), z(k), cyc(k)\}$ . Since  $cyc(k)$  is obtained using V2V/V2I information, such an optimal tuning of  $\theta(k)$  will actually depend on V2V/V2I information.

**Parameter tuning law.** The purpose of solving the optimization problem (2.28) - (2.30) is to find out the optimal relationship between  $\theta(k)$  and V2V and V2I information embedded in  $cyc(k)$  so that the fuel consumption  $J$  is minimized.

Since the cost function is in the accumulation form, we only need to solve  $\theta$  for  $\hat{J}_k = \Delta \hat{f}(k)$  at each time frame  $(k-1)T, kT]$ , which is equivalent to solving  $\partial \hat{J}_k / \partial \theta = 0$ . With the expression of  $\Delta \hat{f}(k)$  in (2.25), the optimization problem is summarized to solve  $\theta$  for

$$\frac{\partial \hat{J}_k}{\partial \theta} = 0 \quad (2.31)$$

$$\Delta T_{nom} \leq \hat{h}(T_{cat}(i), z(i), \theta(i), cyc(i)) + \epsilon_2 \quad (2.32)$$

$$x(k+1) = \hat{g}(x(k), \theta(k), cyc(k)) + \epsilon_1, \quad i = 1, \dots, i_o, \quad (2.33)$$

where  $w_{c1} = [w_\theta, w_{cyc}]$  is the corresponding weights with controller parameter vector  $\theta(k)$  and the future driving information  $cyc(k)$ . Equations (2.31) - (2.33) can be solved using

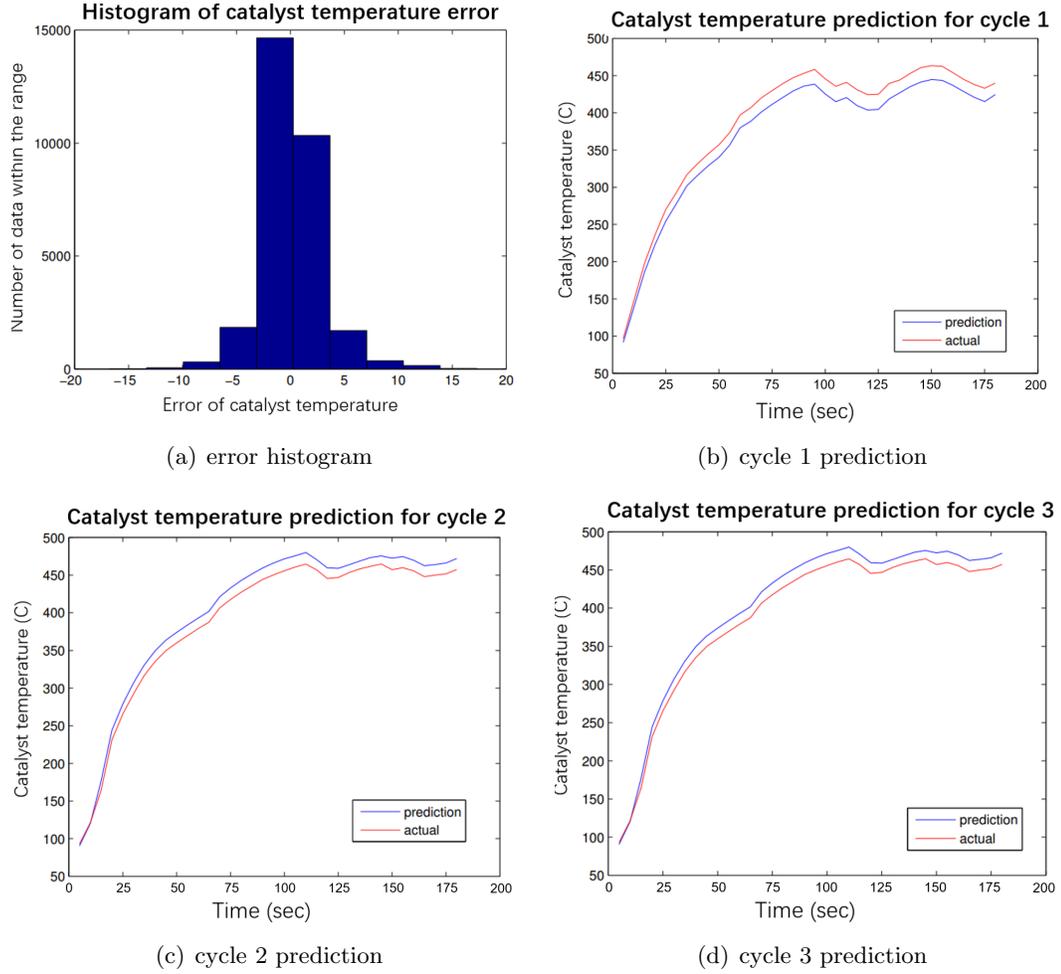


Figure 2.13: NN catalyst temperature prediction results.

the Matlab build-in optimizer “fmincon.” The solution to (2.31) - (2.33) can be written in the following form:

$$\theta(k) = \Psi(x(k), z(k), cyc(k)). \quad (2.34)$$

This optimal parameter tuning law for  $\{\theta_1, \theta_2, \theta_3\}$  indicates that the dynamics of powertrain parameters depend on powertrain states, exhaust states, and future driving conditions represented by V2V/V2I information.

**Co-optimization control scheme.** With the parameter tuning strategy obtained in (2.34), the co-optimization control scheme designed in this study can be summarized in the following steps:

**Step 1:** Use the data generated from the system in (2.1) - (2.5) and (2.7) - (2.8) to train the NN for the co-optimization problem in (2.25) - (2.27) to obtain the equivalent cost function and constraints;

**Step 2:** Solve the equivalent co-optimization problem described in (2.28) - (2.30) for  $\theta(k)$ , and obtain the parameter tuning law (2.34) for  $\theta(k) = \Psi(x(k), z(k), cyc(k))$ ;

**Step 3:** Since at each time step  $k$ ,  $x(k)$ ,  $z(k)$ ,  $cyc(k)$  are known, we can calculate the optimal controller parameter vector  $\theta(k)$  according to the parameter update law. Thus, one can set the controller parameter values in the controller (2.7) and (2.8) to achieve fuel consumption minimization.

The implementation on the actual vehicle of the above optimization requires minor changes to the HEV powertrain control because only a little tuning is required for the control parameters embedded in the hybrid controller functional block in Fig. 2.1.

## 2.4 Simulation Study

This simulation study is based on the integrated Matlab/Simulink model for the HEV powertrain and exhaust system described in Section 2.1.2. The co-optimization problem is to solve the tuning of  $\theta$  in line with V2V and V2I information to minimize fuel consumption.

The simulation results are shown in this section. First, the optimization result for fuel saving is given, and then the optimal parameter trajectory is calculated based on the optimization result. Then, the algorithm was tested under conditions in which the availability of future driving information in terms of V2V/V2I varied. The purpose of such tests is to analyze the effect of V2V/V2I on fuel savings. Finally, we tested our algorithm on different driving cycles to obtain the average fuel saving for the proposed method.

**Optimization results.** The co-optimization scheme has been first tested with two steps of future driving information ( $cyc(k) = [v(k), v(k+1), v(k+2)]$ ) with step size  $T = 2s$ . Fig. ?? shows the optimization results, where the blue curves are the nominal system performance with a nominal optimal choices of controller parameters, and the red curves are the optimal system performance with parameter tuning. Fig. ?? shows the responses of

the fuel saving, catalyst temperature, battery SOC and the driving cycle information. One can see that a 9.8% fuel saving has been achieved on this driving cycle, and the speed for catalyst temperature to reach the light-off temperature ( $200^{\circ}\text{C}$ ) is similar for both nominal and optimal cases. The battery SOC plots show that both cases will have similar battery charging performance, indicating that the fuel saving is not from the increased use of the battery. The optimal parameter tuning trajectory is shown in Fig. ?? . The blue curves are the optimal parameter trajectories, the red curves are the nominal optimal controller parameter values, and, the green curves are the driving cycles. From this figure, one can see that the parameter tuning is sensitive to change in vehicle speed, indicating the effect of V2V/V2I embedded in the future driving information on fuel saving.

1

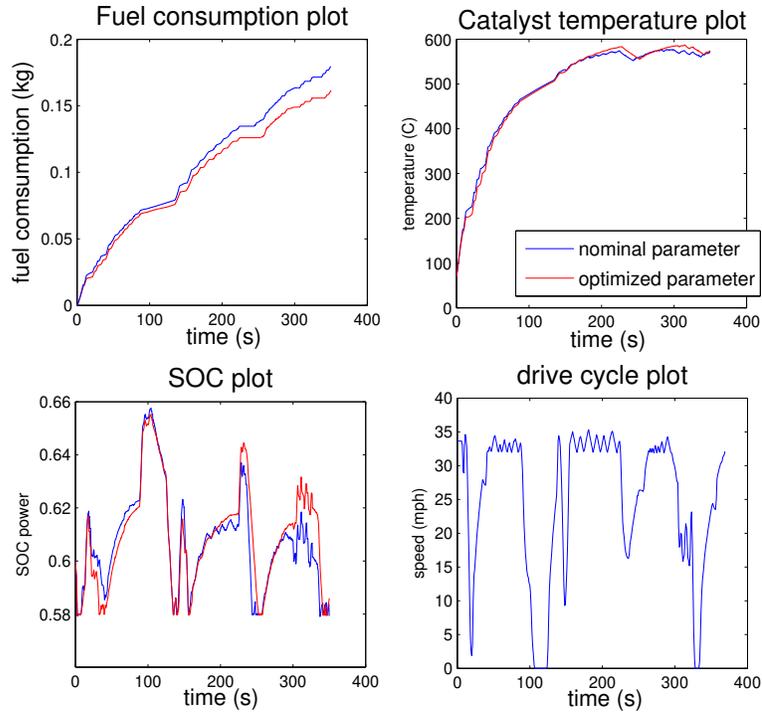


Figure 2.14: Optimal system performance.

**Influence of V2V/V2I information.** The availability of V2V/V2I embedded in future driving information will change the co-optimization model. This might influence the fuel saving results. To test the influence of future driving cycle information to parameter

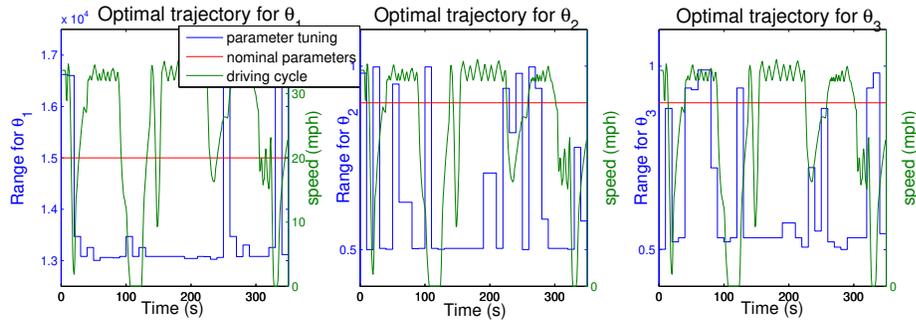
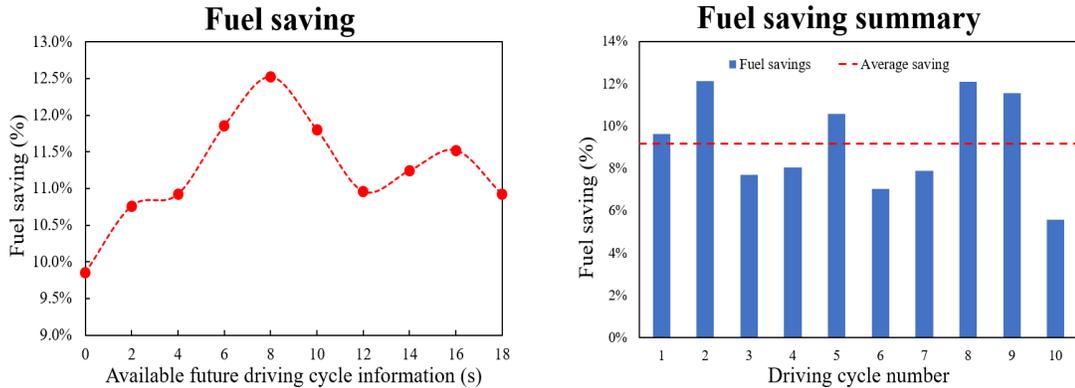


Figure 2.15: Optimal parameter trajectory.

tuning strategy and fuel savings, we tested our algorithm with different numbers of available future steps used in the optimization. For this purpose, we did this test for several different driving cycles. Fig. 2.16(a) shows the relationship between future driving cycle information availability and fuel saving. We chose step size  $T = 2s$ . In this simulation study, when the future driving information is unavailable, we still apply our co-optimization algorithm to generate optimal parameter trajectory instead of directly using the nominal optimal controller parameter. Based on our result, one can observe that without V2V/V2I information, less fuel saving can be achieved with our co-optimization algorithm. Moreover, a maximum fuel saving is reached when 8 seconds (4 steps) of future driving information is available. What is interesting is that longer future driving prediction does not necessarily improve fuel saving.



(a) Fuel saving with different future driving information

(b) Fuel saving in 10 driving cycles

Figure 2.16: Fuel saving results.

**Fuel saving summary.** To test the fuel-saving ability of the proposed parameter co-optimization strategy, we used our algorithm to test fuel saving for 10 different driving cycles, with future speed information fixed to 4 steps ahead. An average of 9.22% of fuel saving is achieved. Fig. 2.16(b) shows the fuel saving of the individual driving cycle tested.

**Discussion.** Based on the simulation results, we can see that our co-optimization design can achieve an average fuel saving of 9.22% for the 2017 Toyota Prius hybrid. From the optimization results, it can be seen that there exists a relationship between V2V/V2I information and optimal control parameter value, which can be derived by solving the optimization problem.

## 2.5 Conclusions and Future Work

In this work, a parameter co-optimization scheme for the HEV powertrain and exhaust system designed to minimize fuel consumption by using V2V/V2I information is described. An optimization framework was formulated, and an equivalent NN optimization model (including both the equivalent cost function model and the equivalent constraints model) was built. Then the optimization problem was solved, and the parameter tuning strategy was obtained. The simulation study described in this study verified the parameter co-optimization design. From the simulation result, it can be concluded that a significant fuel saving can be achieved.

The significance of this work is to take future driving conditions and the effect of the exhaust system into consideration in fuel consumption minimization design. This method can be easily applied for vehicles on the market to save fuel. However, the work is based on the simulation using real-system data, and the goal of future work would be to test the effect on the fuel savings on actual vehicles.

## Chapter 3

# System Identification and Traffic Signal Control for Large Traffic Networks in Urban Areas

This chapter developed a system identification scheme and an LQR traffic signal controller for large traffic networks in urban areas. Compared with conventional signal control methods, the proposed research is a novel approach to implement on-line parameter estimation based LQR control in traffic signal controls for large traffic networks.

Although the methods addressed in the literature in Section 1.2.2 are widely used, the aforementioned traffic-responsive control systems have several limitations and challenges:

- **Relying on off-line traffic-network models:** Although using real-time traffic data as an input, the network models used by SCOOT and OPAC are built based on historical data and are not updated during their real-time operations. Because historical data may not accurately reflect current traffic conditions [43], these control strategies may potentially result in poor performance.
- **Exponential complexity for global minimization:** Similar to OPAC, because of the presence of discrete variables that require exponential-complexity algorithms for a

global minimization, the control strategies may not be real-time feasible for large-scale traffic networks [70].

- **Ignoring interactions between intersections:** In TUC, traffic systems are modeled at intersection level, and the impacts of the traffic states in neighborhood intersection and control inputs on the travel delays of investigated intersection are ignored.

To address these limitations, this study proposes a new multi-input and multi-output (MIMO) traffic signal control method that can not only improve network-wide traffic operations in terms of reduced delay and energy consumption, but also is more computationally feasible than existing centralized signal control methods. Considering intersection interactions, a linear dynamic traffic system model was built and updated online to reflect how signal control inputs at each intersection would affect network-wide vehicle delays. Based on the system model, an LQR was built to minimize both traffic delay and control-input changes [71]. We select LQR because it has the following advantages over other control strategies from the control design perspectives: 1) LQR is a simple optimal control that can be made adaptive in combination with system parameter estimation; 2) LQR is robust with respect to model uncertainties and errors as well as unexpected disturbances; and 3) LQR is easy to be implemented as a MIMO control strategy that produces a feedback control for a large networked intersection whilst automatically takes care of the handling of intersection interactions. Indeed, compared with the existing methods, the proposed adaptive LQR traffic control algorithm does not rely on large historic datasets to identify the unknown traffic model. Moreover, using LQR control with a MIMO linearized model can achieve minimal traffic delay while considering neighboring intersection interactions. The proposed control method was implemented and evaluated in a microscopic traffic simulation environment with a 35-intersection network of Bellevue City, Washington. Simulation results show that the proposed method had shorter average travel delays in the network when compared with the delays controlled by the methods proposed in the state-of-the-art max-pressure [41], self-organizing traffic lights (SOTL, [42]), and independent deep Q network (IDQN) controls, [72].

The major contributions of this chapter include:

- Proposed a globalized modeling framework for network-wide traffic signal control using calibrated VISSIM model via real traffic flow data. In this framework, intersection interactions were explicitly modeled by a MIMO linear time-varying traffic system model that reflects how the signal control input at each intersection affects network-wide vehicle delay measurements.
- Established an adaptive LQR based traffic signal control method for traffic networks. The method can identify unknown system dynamics online, thus being able to handle system uncertainties caused by traffic and environmental randomness.
- Cross-compared the performances of several traffic signal control methods based on a real-world data-based microscopic traffic simulation model.

## 3.1 Urban Traffic Network and Simulation Model

### 3.1.1 Traffic Network

This study focuses on urban road networks with signalized intersections. Specifically, a grid road network from downtown Bellevue, Washington was selected as the area of the networked intersections in this study. The study area covered from Main Street (the south end) to NE 12th Street (the north end) and from Bellevue Way NE (the west end) to 112th Ave NE (the east end). It included 35 intersections and 57 major bi-directional road links, with the average link length being 664.4 ft.

To replicate real-world traffic conditions, traffic count data by movement were collected for each intersection in the midday off-peak period (i.e., 1-2 p.m.). Fig. 3.1a shows the traffic counts of the northwest corner intersection (i.e., NE 12th Street and Bellevue Way NE) of the study area as an example. Link traffic volumes were calculated by aggregating traffic movement counts in the same direction as shown in Fig. 3.1b [2].

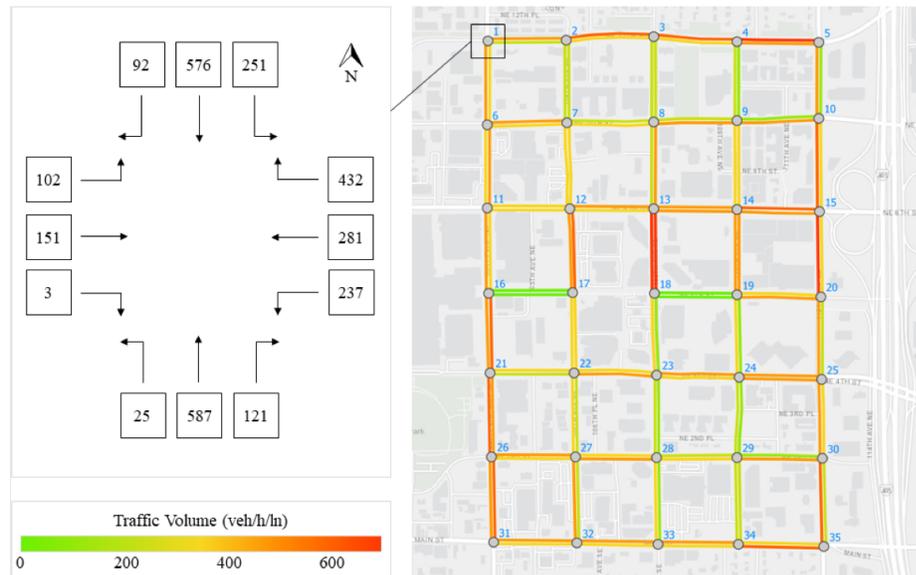


Figure 3.1: Illustration of (a) traffic count by movement data (left) and (b) traffic volume data in the road network [2].

### 3.1.2 Microscopic Traffic Simulation Model

In this study, PTV VISSIM [73], a commonly used microscopic software for traffic simulation and signal controls, was used to facilitate the development and testing of different traffic signal control methods [6,7,9,10]. VISSIM uses Wiedemann car-following and lane-changing models [74,75] to model the movements and interactions of vehicles. The VISSIM traffic model shown in Fig. 3.2 was developed based on actual road geometries of the study area. This microscopic simulation model has been calibrated by the City of Bellevue with actual traffic data and has been used for planning and management purposes in downtown Bellevue [2]. The calibration process involved adjusting parameters including vehicle composition, speed distributions, conflict areas, priority rules, reduced speed areas, and car-following and lane-change behaviors to ensure that the simulation was consistent with the traffic characteristics of the real-world.

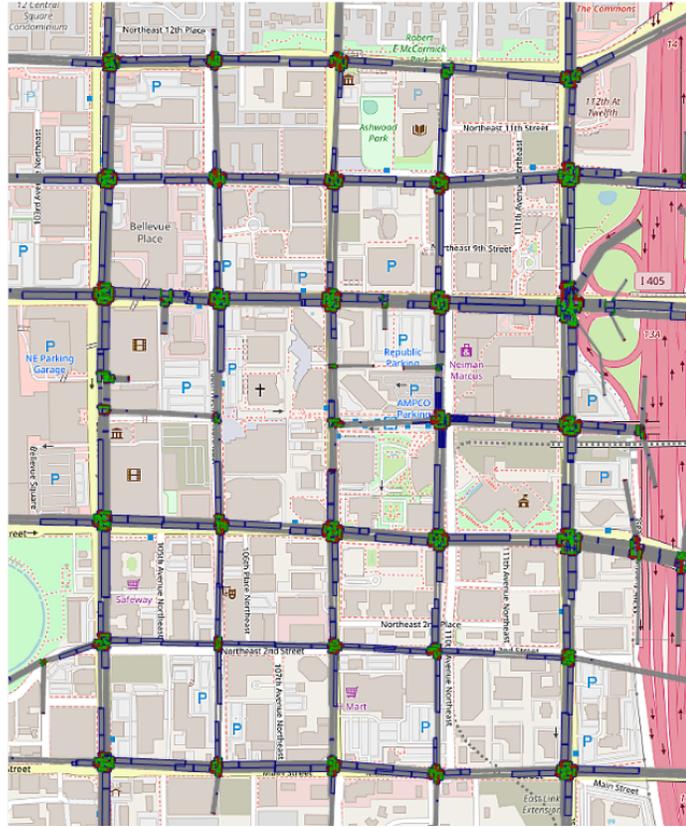


Figure 3.2: Illustration of the VISSIM simulation model for the investigated urban traffic network [2].

## 3.2 Traffic Signal Controller Design

### 3.2.1 Traffic System Modeling

The subject traffic network comprises 35 intersections, and the green time for each intersection could affect the traffic flow performance of the whole system. In this study, we assume a fixed signal cycle length (90 s) with two phases: the east and west (E-W) approaches share one phase, and the north and south (N-S) approaches share the other phase so as to simplify the control algorithm formulation. As shown in Fig. 3.3, the N-S direction green time of intersection  $i$  is denoted as  $v_i$  with  $i = 1, 2, \dots, 35$ . In practice, both the signal cycle length and the initial green time can be derived based on the widely used Webster's method [76], which considers both the saturation flow rate and the flow ratio for

each lane group. It is assumed that each intersection had two delay measurements: one for the N-S direction and the other for the E-W direction. Therefore, there are a total of 70 delay measurements for the 35 intersections. Mathematically, the traffic network described in Section 3.1.1 can be expressed as a discrete-time input-output and real-data calibrated model in VISSIM [73] as:

$$z(k+1) = F(z(k), v(k)) \quad (3.1)$$

$$z(k) = [z_1, z_2, \dots, z_{70}]^T \quad (3.2)$$

where  $z(k) \in R^{70}$  are the system outputs, which are the measurable traffic delays of N-S and E-W direction vehicle flows at each node;  $v \in [v_{min}, v_{max}] \in R^{35}$  are the system control inputs comprised of all  $v_i$ , indicating the N-S direction green times of the intersections; and  $k$  denotes the time step (i.e., the cycle number).  $F$  stands for the nonlinear relationship between traffic delays and the green signal period of the N-S direction as represented by VISSIM simulation [73].

Assuming that the nonlinearity of the traffic system is linearizable, one can linearize the system (3.1) to the state-space form as

$$\Delta z(k+1) = A\Delta z(k) + B\Delta v(k) + w(k), \quad (3.3)$$

where

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,70} \\ \vdots & \ddots & \\ a_{70,1} & \dots & a_{70,70} \end{bmatrix} \in R^{70 \times 70}, \quad B = \begin{bmatrix} b_{1,1} & \dots & b_{1,35} \\ \vdots & \ddots & \\ b_{70,1} & \dots & b_{70,35} \end{bmatrix} \in R^{70 \times 35}, \quad (3.4)$$

are the system parameters, and  $\Delta z(k) = z(k) - z(k-1)$ ,  $\Delta v(k) = v(k) - v(k-1)$  are the increment of  $z$  and  $v$  at time step  $k$ .  $w(k)$  is the bounded linearization error, which satisfies  $\|w(k)\|_2 \leq w_b$  for  $k = 1, 2, \dots$ , and  $w_b$  is the bound for the error  $w(k)$ . Based upon our comprehensive linearization around all the possible operating points of model (3.1),

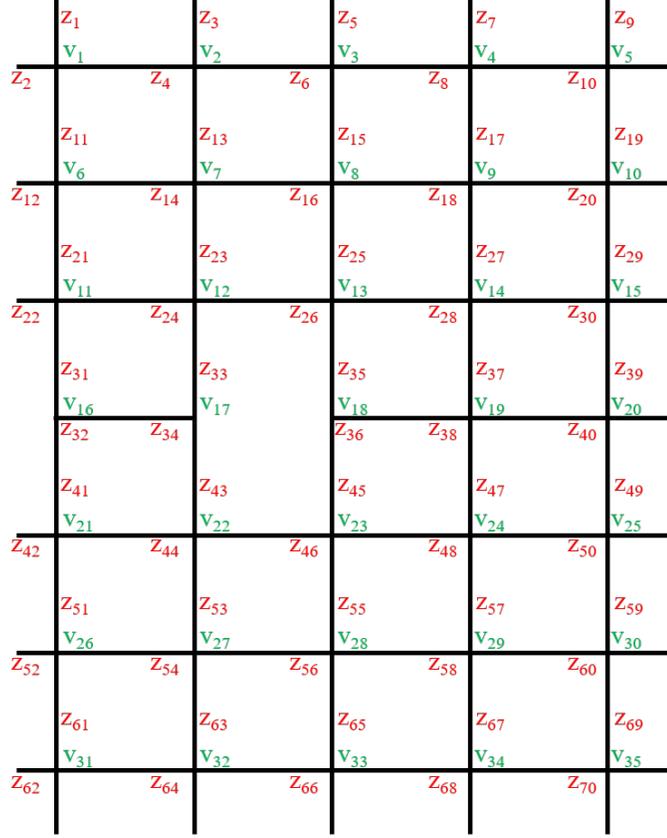


Figure 3.3: Illustration of the notations for signal control inputs and traffic delay measurements.

the value of  $w_b = 4.5$  s in terms of travel delay. This will be used as a threshold in the normalized least squares algorithm to be described for estimating  $A$  and  $B$  matrices. To simplify notation, we denote  $\Delta z = y$  and  $\Delta v = u$ , and rewrite the linearized state-space form as

$$y(k+1) = Ay(k) + Bu(k) + w(k). \quad (3.5)$$

Because the direct mathematical relationship between traffic delay and green signal period is unknown, the linearized parameter matrices  $A$  and  $B$  are unknown and possibly time-varying, and we need to design a parameter estimation scheme to estimate the system parameter matrices  $A$  and  $B$  online.

**Control Objectives.** The control objective is to design an online estimation based LQR controller with unknown system parameter matrices  $A$  and  $B$  to minimize the traffic

delay characterized by  $y(k)$  with low control energy. This constitutes the minimization of the following cost function by selecting an optimal signal timing strategy  $u(k)$ :

$$J = \sum_{k=0}^{\infty} [y^T(k)Qy(k) + u^T(k)Ru(k)], \quad (3.6)$$

where  $Q = Q^T > 0$  and  $R = R^T > 0$  are positive definite matrices with relevant dimensions. They denote the pre-specified state-cost weighting matrix and input-cost weighting matrix, respectively. This is the standard LQR cost function widely used in the literature [77].

### 3.2.2 Online Parameter Estimation Scheme

Before starting the LQR controller design, one needs to identify the system parameter matrices  $A$  and  $B$  using system identification techniques [13, 78–80]. To this end, the linearized system (3.5) can be parameterized as follows:

$$y(k+1) = \Theta\Phi(k) + w(k), \quad (3.7)$$

where

$$\Theta = \begin{bmatrix} A & B \end{bmatrix} = \begin{bmatrix} a_{1,1} & \dots & a_{1,70} & b_{1,1} & \dots & b_{1,35} \\ \vdots & \ddots & & \vdots & \ddots & \\ a_{70,1} & \dots & a_{70,70} & b_{70,1} & \dots & b_{70,35} \end{bmatrix} \in R^{70 \times 105} \quad (3.8)$$

$$\Phi = \begin{bmatrix} y^T & u^T \end{bmatrix}^T = \begin{bmatrix} y_1 & y_2 & \dots & y_{70} & u_1 & u_2 & \dots & u_{35} \end{bmatrix}^T \in R^{105}, \quad (3.9)$$

where  $\Phi$  groups all the measurable inputs and outputs and is the information vector used in the estimation of  $\Theta$ .

**Estimation Error.** To estimate the system parameter matrices  $A$  and  $B$ , we denote  $\Theta(k) \in R^{70 \times 105}$  as the estimate of  $\Theta$  at cycle  $k$ . Then the estimation error is defined as

$$\varepsilon(k) = \Theta(k-1)\Phi(k-1) - y(k) \quad (3.10)$$

$$= \Theta(k-1)\Phi(k-1) - \Theta\Phi(k-1) - w(k-1) \quad (3.11)$$

In this work, we assume that the bound for the linearization error term  $w(k)$  is small, and the effect of the term  $w(k)$  can be neglected in the control design due to the inherent strong robustness property of LQR control strategy and the use of the following normalized least squares estimation algorithm, [81].

**Parameter Update Law.** As it has been shown that the normalized least squares identification is robust to possible modeling uncertainties such as the linearization errors in (6) [81], it is used here to provide the required estimates for parameter matrices  $A$  and  $B$ . The estimation updating rule for  $k = 0, 1, 2, \dots$  is therefore given by:

$$\Theta(k+1) = \begin{cases} \Theta(k) - \frac{P(k-1)\Phi(k)\varepsilon(k)}{m^2(k)} & \|\varepsilon(k)\| > w_b \\ \Theta(k) & \|\varepsilon(k)\| \leq w_b \end{cases} \quad (3.12)$$

$$P(k) = P(k-1) - \frac{P(k-1)\Phi(k)\Phi^T(k)P(k-1)}{m^2(k)} \quad (3.13)$$

$$m(k) = \sqrt{\kappa + \Phi^T(k)P(k-1)\Phi(k)}, \quad (3.14)$$

where  $\kappa > 0$  is the pre-specified design parameter normally less than 0.05.  $P(k)$  is a positive definite variance matrix and its initial value is  $P(0) = I$  with  $I$  being an identity matrix.  $\Theta(0) = \Theta_0$  is the chosen initial estimates of the parameter matrices  $A$  and  $B$ . The above parameter estimation can be regarded as an online learning process where the parameter matrices  $A$  and  $B$  are learned using input and output data grouped in  $\Phi$  when the cycle number  $k$  increases.

It can be seen that (3.12) has a switching function, when the error  $\varepsilon(k)$  is smaller than or equal to the lower bound of  $w(k)$ , the parameter updating is stopped and the  $k+1$  cycle estimate is its estimate at  $k$ . This functionality enhances the robustness of the estimation algorithm so as to make it robust with respect to linearization error  $w(k)$  [81].

### 3.2.3 LQR Controller Design

When system parameter matrices  $A$  and  $B$  are known, one can choose the following LQR state feedback control law as

$$u(k) = -Ky(k), \quad (3.15)$$

where the state feedback gain matrix  $K$  is obtained by solving the following Riccati equation for matrices  $S = S^T > 0$  and  $K$  simultaneously [77]:

$$A^T S A - S - A^T S B K + Q = 0 \quad (3.16)$$

$$K = (B^T S B + R)^{-1} B^T S A. \quad (3.17)$$

**Adaptive LQR Controller.** Because system parameter matrices  $A$  and  $B$  are estimated according to the parameter update law (3.12)-(3.14), one can obtain  $A(k)$  and  $B(k)$  from  $\Theta(k) = [A(k), B(k)]$  and (3.12) at each cycle  $k$ . Therefore, the adaptive LQR controller can be obtained by replacing  $A$  and  $B$  in (3.16) and (3.17) with their estimates at each cycle  $k$ . This leads to the following adaptive LQR control law

$$u(k) = -K(k)y(k) \quad (3.18)$$

where adaptive gain matrix  $K(k)$  is obtained by solving the Riccati equation for matrices  $S(k) = S^T(k) > 0$  and  $K(k)$  at each cycle  $k$  as follows:

$$0 = A^T(k)S(k)A(k) - S(k) - A^T(k)S(k)B(k)K(k) + Q \quad (3.19)$$

$$K(k) = (B^T(k)S(k)B(k) + R)^{-1}B^T(k)S(k)A(k). \quad (3.20)$$

### 3.2.4 Adaptive LQR Algorithm Summary

With the real-data calibrated system model (3.1), the above adaptive LQR control algorithm is realized in the following steps:

1. Calibrate VISSIM model (1) using real traffic data so that system model (1) gives a desired reflection of the actual system dynamics;
2. At time  $k$ , use input and output data from VISSIM nonlinear system (3.1) to estimate parameters  $\{A(k), B(k)\}$  by (3.12) – (3.14);
3. For the estimated  $\{A(k), B(k)\}$ , solve the Riccati equation (3.19) – (3.20) for adaptive control gain  $K(k)$ ;
4. Apply control input (3.18) to the signal timing to all intersections in Figs. 3.1 and 3.3 as represented by system nonlinear model (3.1); and
5. Set  $k = k + 1$  and go to Step 2.

Fig. 3.4 shows the closed-loop control structure and the information flow chart of the above adaptive LQR control algorithm.

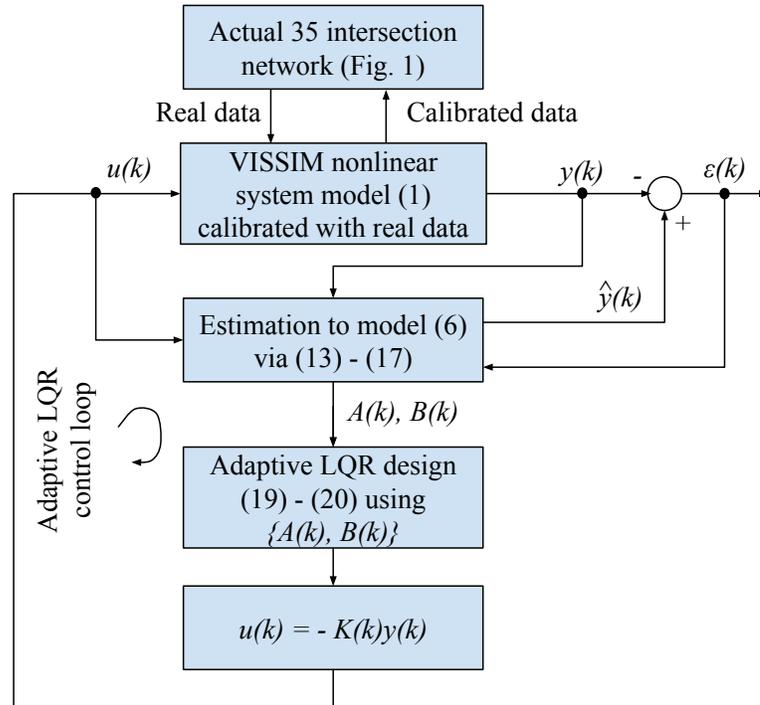


Figure 3.4: The adaptive LQR closed loop control structure and the information flow chart.

The above algorithm shows that as travel delay state vector  $y(k)$  is assumed measurable, it is used as a state feedback information in the construction of the closed loop control.

### 3.2.5 Robustness of the Proposed Algorithm

It is worth noting that the control signal calculated from (3.18) is directly applied to the nonlinear system (3.1) which had been calibrated using real-traffic flow data, rather than to linearized system (3.5). The simulation results in the following sections would show the desired robustness of the proposed algorithm with respect to the linearization error  $w(k)$  in (3.5). Indeed, as it has been shown that LQR has a good robustness with respect to model uncertainties in terms of providing infinite gain margin [77], the proposed control algorithm is also robust with regard to the model uncertainties. In addition, the normalized least squares with switching functionality as shown in (3.12) is robust with respect to modelling error  $w(k)$  of the system. As a result, the proposed algorithm is robust because of the use of LQR. together with the normalized least squares algorithm for the estimation of parameter matrices  $\{A(k), B(k)\}$ .

Moreover, looking at the system model (3.1), it can be seen that adding more signal phases at each intersection will not constitute difficulties in the control design as this would just increase the dimensionality of input  $u(k)$  and  $y(k)$  in (3.5), while the control design methods remains the same as those in (3.18) – (3.20).

### 3.2.6 Coordination and Scalability Issues

Since the proposed adaptive LQR control is obtained using the MIMO model in (3.5), it is a multivariable control that automatically takes into account of the interactions among all the intersections as the off-diagonal elements in matrix  $A$  are normally non-zeros, where the coupling feature among intersections are taken care of when control input vector (3.18) is applied to the system (3.1). This is the advantage of using multivariable control to deal with the signal timing for the networked intersection area as shown in Fig. 3.1.

Moreover, if more intersections in the traffic network need to be added, instead of

rebuilding the entire traffic model (3.5), we can build subsystems relating only to the neighboring intersections and use our MIMO control algorithm in a decentralized way, which does not alter the core LQR design as represented in (3.18) – (3.20). In this context, our adaptive LQR control algorithm can be made scalable in practice. Indeed, the same method can also be made decentralized by using sub-blocks in matrices  $A$  and  $B$ , which leads to de-centralized design equations of the same form as expressed in LQR control gain solution (see (3.15) – (3.20)).

### 3.3 Baseline Methods and Ablation Study

The performance of the proposed method (adaptive LQR control) was compared with that of three baseline methods: max-pressure, SOTL, and independent deep Q network (IDQN) control methods. Additionally, an ablation study was conducted to test the effectiveness of the proposed method in modeling the system (compared with linear feedback control) and updating system parameters online (compared with offline LQR control). All of these signal control methods were implemented using the VISSIM COM interface.

#### 3.3.1 Baseline Methods

**Max Pressure.** The max-pressure traffic signal control method was inspired by the max pressure algorithm [82] in the field of communication network, which considers the routing and scheduling of packet transmission in a wireless network. Specifically, the max-pressure traffic signal control method models traffic flows as substances in a pipe and optimizes traffic signals to maximize the relief of pressure between incoming and outgoing lanes [83]. The pressure for a certain phase  $p$  is defined as:

$$\text{Pressure}(p) = \sum_{l \in L_{p,inc}} |q_l| - \sum_{l \in L_{p,out}} |q_l| \quad (3.21)$$

where  $l$  stands for traffic lanes;  $q_l$  is the average vehicle queue length in lane  $l$  during the last updating time interval;  $L_{p,inc}$  is the set of incoming lanes that have green lights in phase  $p$ ;

and  $L_{p,out}$  is the set of outgoing lanes from all incoming lanes in  $L_{p,inc}$ .

The max-pressure method does not have a fixed cycle length. Instead, the algorithm predefines an updating time interval (40 s in this study). At the beginning of each time interval, the algorithm examines the pressures of all signal phases and chooses phases with the maximum pressure as green phases in the incoming time interval. According to an open-source evaluation conducted by [83], the max-pressure method has the best performance compared with Webster’s method [76], SOTL [84], DQN [72] and deep deterministic policy gradient (DDPG) [85].

**SOTL.** Similar to the max-pressure method, SOTL [42] does not have a fixed cycle length. Each red-light phase counts the cumulative number of vehicles that arrived during its red light period ( $k_i$ ). When  $k_i$  is greater than a predefined threshold, the current green-light phase switches to red with  $k_i = 0$ , while the red light that counted turns green. During implementation, minimum green phase duration is applied.

**IDQN.** The DQN-based traffic signal control has a predefined updating time interval, similar to that of max-pressure control. At the beginning of each time interval, the DQN agent outputs an action based on the current state. This action determines which phase will turn green for the upcoming time interval. We have implemented the independent DQN-based traffic signal control using the algorithm in [83], where each intersection had a local DQN agent that took local state as input and determined actions for that intersection. The Q networks for these agents were updated independently. The state, action, reward, and Q network architecture were defined as follows.

- State: queue and delay of incoming lanes, and the most recent green phase at each intersection. The queue and delay are continuous variables, while the most recent green phase is encoded as a one-hot vector;
- Action: which phase should turn green for the upcoming time interval? In this study, two phases (N-S and E-W) were considered;
- Reward: global reward was used, i.e., the negative average vehicle delay for the whole

traffic network was used. Since the reward function is used only during offline training, we can assume that the average network delay is available;

- Q network: two hidden layers of  $3|S|$  fully connected neurons with the ReLU activation function, where  $|S| = 6$  is the dimension of the state. The output layer has two neurons corresponding to the two phases.

### 3.3.2 Ablation Study

An ablation study was conducted to further assess the performance of the proposed adaptive LQR control method. Two variants of the proposed method were considered in this ablation study: 1) a linear feedback control, which models the network system in a simple linearization form between delay and green time; and 2) an offline LQR method that uses the same initial  $A, B$  matrices as those used in the proposed method. Considering the performance difference between the proposed method and the offline LQR method might be insignificant under normal off-peak volumes since the  $A, B$  matrices were estimated based on off-peak traffic conditions, the ablation study was conducted using 150% of the off-peak traffic volumes in VISSIM.

**Linear Feedback Control.** For the linear feedback control [86], system model (3.1) is linearized to a simple form:

$$\Delta z(k) = H\Delta v(k), \quad (3.22)$$

where  $z \in R^{70}$  are the traffic delays of N-S and E-W direction vehicle flows at each node,  $v \in [v_{min}, v_{max}] \in R^{35}$  are N-S direction green times of the intersections, and  $H$  is the input-output gain matrix. The feedback control strategy is designed as:

$$\Delta z(k) = -\Gamma z(k), \quad (3.23)$$

where  $\Gamma > 0$  is a design parameter. Applying (3.23) to (3.22), one can obtain the controller as:

$$\Delta v(k) = -(H^T H)^{-1} H^T \Gamma z(k). \quad (3.24)$$

This variant uses a simplified system modeling approach and can be used to determine how system modeling contributes to the performance of the proposed method.

**Offline LQR.** For the offline LQR method, the system matrices  $A$  and  $B$  defined in (??) and (3.4) were estimated from historical data and were not updated during the control. This variant will help show how much gain the adaptive parameter updating design provides.

### 3.4 Results

This simulation study is in collaboration with Professor Yinhai Wang’s group from the University of Washington. The proposed method was evaluated in the VISSIM simulation environment with the 35-intersection network of Bellevue, Washington, as mentioned in Section 5.1. VISSIM traffic simulation software is a widely used simulation tool especially in the transportation engineering field. VISSIM uses Wiedemann car-following and lane-changing models to model the dynamic movements and interactions of vehicles. Each simulation test lasted for 5,000 seconds, and multiple (around 30) runs with different random seeds were conducted for each test case. The parameter initial estimates  $\Theta(0)$  are chosen within the range  $[-10, 10]$  based upon the linearization tests. The random seed initialized a random number generator. With various random seeds, different value sequences were assigned to the stochastic functions in VISSIM, and the simulation of stochastic variations of vehicle arrivals in the network was achieved [71]. This will affect the following aspect of traffic simulation.

- Traffic volume: Given an input traffic volume, the actual traffic volume will be a stochastic variable with its expectation being the set one.
- Vehicle arrival: Vehicles arrive with their time headways following a Poisson distribution. With different random seeds, vehicle arrival sequences will be different.
- Turn movements: With turn ratio predefined, random seeds will affect which vehicles will turn right, go straight, or left at an intersection.

- Driving behavior: The car-following and lane-change behavioral models used in VIS-SIM contain several random parts (e.g., at steady-state car following, drivers will have stochastic accelerations around zero).

The average vehicle delays across the whole simulation period were used as the primary evaluation metric. To investigate how the proposed method can handle various initial traffic states, different initial N-S green times (20 s, 40 s, and 60 s) were chosen. The idea is that an unbalanced initial green time (e.g., 60 s) will result in a congested initial traffic state, whereas a balanced initial green time (e.g., 40 s) may lead to a less congested initial traffic state. In practice, both the signal cycle length and the initial green time can be derived based on Webster’s method [35], which considers both the saturation flow rate and the flow ratio for each lane group. For max-pressure, SOTL, and IDQN controls, there is no initial green time because they do not have a fixed cycle length.

### 3.4.1 Comparison with Baseline Methods

TABLE 3.1 presents the performance of the proposed method against the baseline methods. Given different initial green times, it can be seen that the LQR method outperformed max-pressure, SOTL, and IDQN methods, especially when the initial N-S green time was 40 s. The performance of the IDQN method was much worse than the other methods, and this result is consistent with the results presented in [49]. The poor performance of IDQN might be caused by the independence of the reinforcement learning (RL) agents in different traffic intersections, where the control algorithm tries to optimize local traffic situations but fail to cooperate with each other.

Fig. 3.5 presents the changing of average vehicle delay along with the progression of the simulation time, with initial green time = 20 s, 40 s, and 60 s. With any initial green times, the proposed LQR method outperformed other baseline methods.

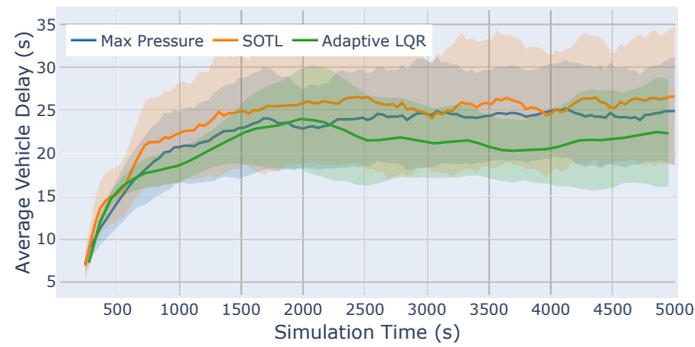
Fig. 3.6 shows the distribution (violin plot) of average vehicle delays across the whole simulation period. The violin plot shows the probability density of the data at different values. Based on the interquartile range (IQR) inside the violin plots, it can be observed

Table 3.1: Travel delays with different initial green times and control methods, under normal off-peak traffic volume

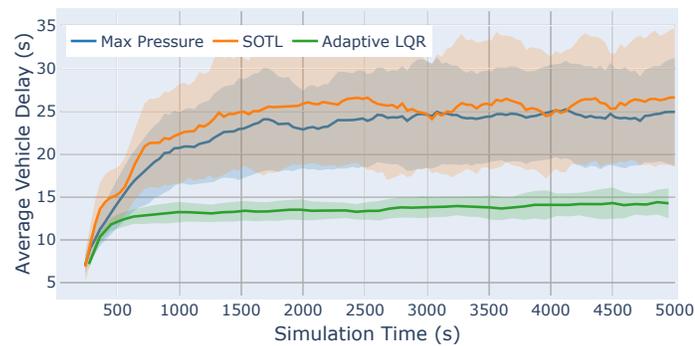
Init. Green Time (s)	Control Method	Avg. Veh. Delay (s)	Std. Dev.
20	Adaptive LQR	20.34	5.84
40	Adaptive LQR	<b>13.24</b>	2.16
60	Adaptive LQR	16.01	3.59
N/A	SOTL	23.73	8.46
N/A	Max pressure	22.24	6.93
N/A	IDQN	96.83	107.81

that the LQR method constantly maintained smaller values of lower (25%), median (50%), and upper (75%) quartiles, indicating that its overall performance was consistently better than other methods compared.

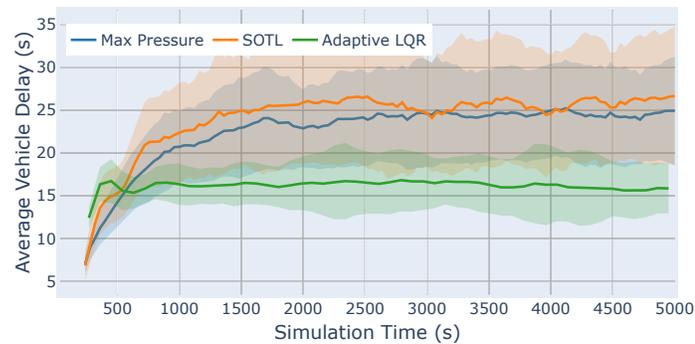
Fig. 3.7 depicts how the LQR method optimized signal splits at each intersection cycle-by-cycle over the simulation period. In the beginning, all the intersections shared the same initial N-S green time (e.g., 20 s, 40 s, or 60 s). As time progressed, the N-S green time at each intersection was updated every cycle (90 s) to reflect the control output designed to minimize the average travel delay. It can be observed that, at the end of the simulation, the N-S green times of many intersections centered around 40 s, which is the balanced green time between N-S and E-W directions (5 s of yellow and red time between green phases), indicating that traffic volumes in both N-S and E-W directions at these intersections were similar. The N-S green time in some other intersections, nevertheless, continued to change throughout the simulation period. For example, with the initial N-S green time being 20 s, almost all intersections' N-S green time continued to increase throughout the simulation. This was caused by insufficient N-S green time from the beginning and, since the control was adjusted cycle-by-cycle, traffic in the N-S direction was never cleared at those intersections. As a result, the proposed LQR method continued to increase the N-S green time at these intersections to help push traffic through and minimize average delays.



(a) Initial N-S green time = 20 s



(b) Initial N-S green time = 40 s

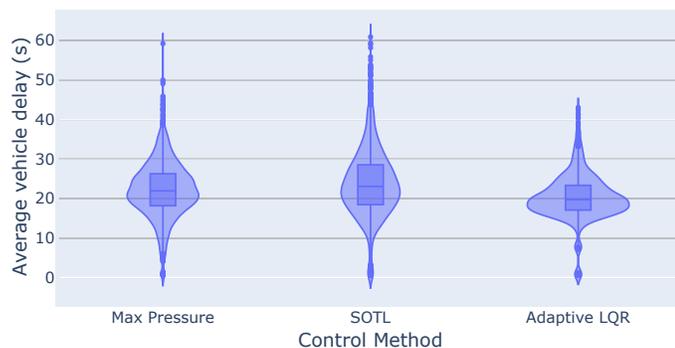


(c) Initial N-S green time = 60 s

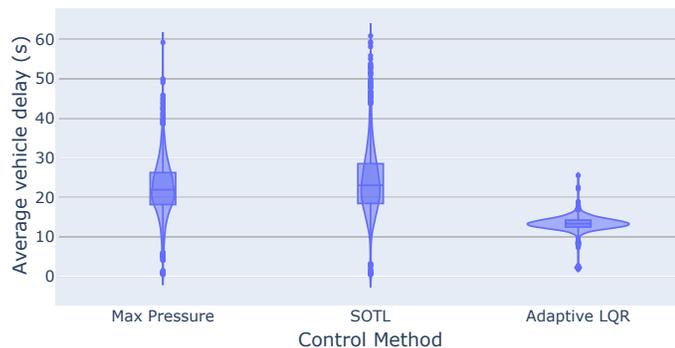
Figure 3.5: Changing of average vehicle delay during the simulation with normal off-peak traffic volume. Solid colored lines represent the mean, and shaded areas represent the mean  $\pm$  one standard deviation intervals. Results of the IDQN method were not visualized because of its large delays.

### 3.4.2 Ablation Study

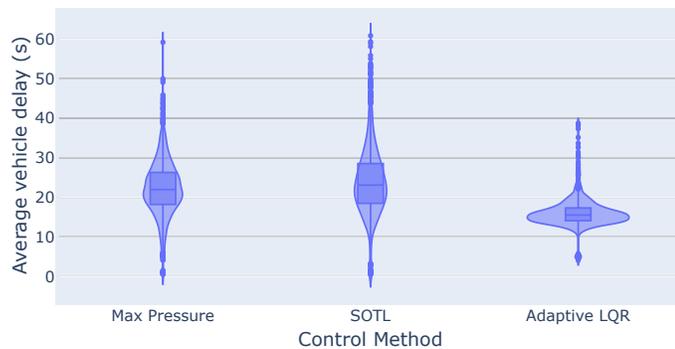
Table 3.2 and Fig. 3.8 present the results of the ablation study. For different initial N-S green times, the proposed adaptive LQR method outperformed both the linear feedback



(a) Initial N-S green time = 20 s



(b) Initial N-S green time = 40 s



(c) Initial N-S green time = 60 s

Figure 3.6: Distribution of average vehicle delays over the simulation test period under normal off-peak traffic volume for comparison among different initial green times. Results of IDQN were not visualized due to its much larger delays.

and the offline LQR control methods. The results demonstrate that the system modeling approach and the online parameter updating strategy of the proposed adaptive LQR method have resulted in improved control performance.

From the above descriptions, it can be seen that the adaptive LQR control is actually

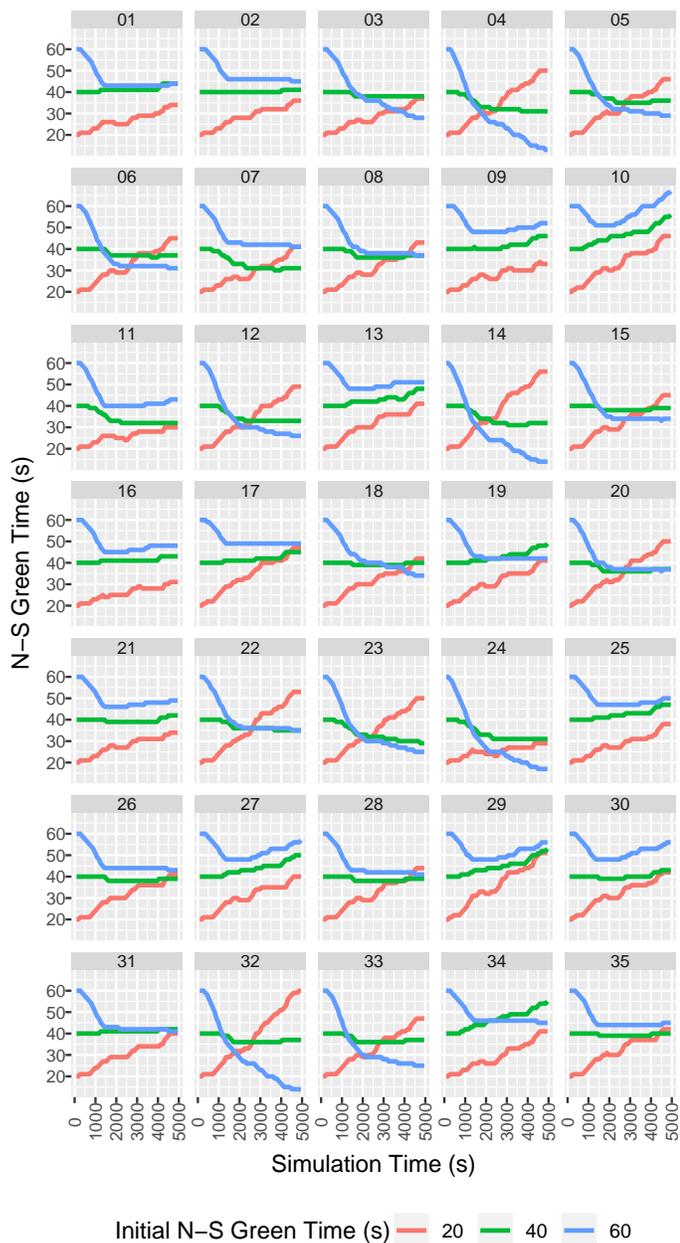


Figure 3.7: LQR control and changes of N-S green time of the 35 intersections during the simulation test period with different initial N-S green time under normal off-peak traffic volume.

performed for the nonlinear system in an adaptive way. It is used for the large-scale nonlinear model for 35 intersections as given in (3.1). To the best of our knowledge, this type of adaptive LQR has not been applied to such a large network. In the literature below, they

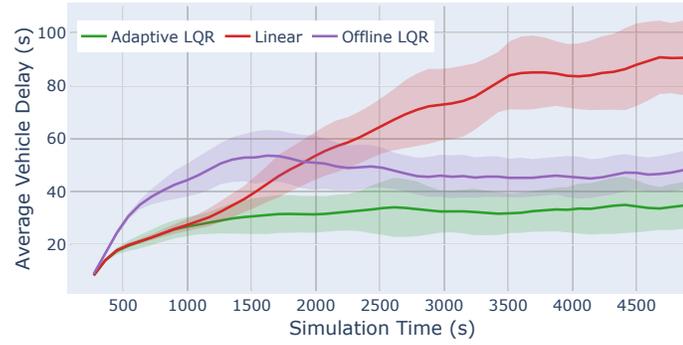
Table 3.2: Results of ablation study: travel delays with different initial green times and control methods, with 150% off-peak traffic volume.

Init. Green Time (s)	Control Method	Avg. Veh. Delay (s)	Std. Dev.
20	Adaptive LQR	<b>29.99</b>	10.07
	Linear Feedback	59.69	28.24
	Offline LQR	44.35	12.37
40	Adaptive LQR	<b>25.28</b>	9.38
	Linear Feedback	52.51	26.33
	Offline LQR	44.92	14.97
60	Adaptive LQR	<b>32.99</b>	11.70
	Linear Feedback	60.41	26.61
	Offline LQR	40.56	14.42
N/A	SOTL	32.09	10.11
N/A	Max pressure	41.70	27.52
N/A	IDQN	138.42	149.86

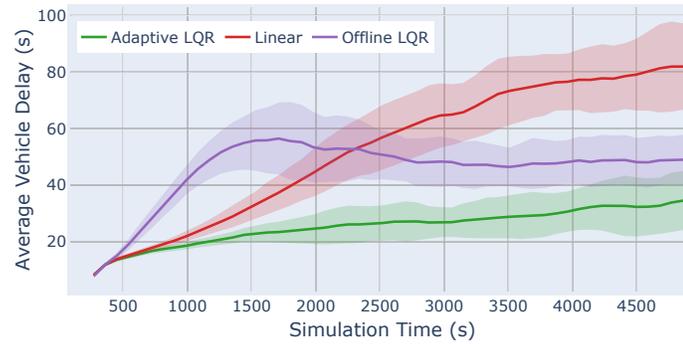
tested LQR-based perimeter control for 16-intersection network, [87].

### 3.5 Conclusion and future work

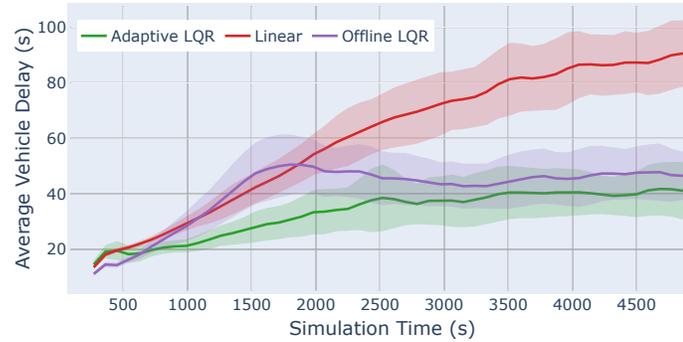
This study proposed an adaptive LQR based signal control method for an urban traffic network. A linear dynamic traffic system model was built and adaptively updated to reflect how each intersection’s signal control input affects network-wide vehicle delays. A linear-quadratic regulator was then built to minimize both traffic delays and control-input changes. With a predefined cycle length, the proposed algorithm adjusted traffic signal control strategies according to observed vehicle delays. The proposed control method was evaluated in the VISSIM traffic simulation environment with a 35-intersection network of Bellevue city, Washington. Traffic counts by approach and turn movement were collected for each intersection to replicate real-world traffic conditions. Simulation results indicate that the proposed method yielded shorter average travel delays in the network when compared with the state-of-the-art max-pressure, SOTL, and IDQN methods. Results of the ablation study also show that the proposed method outperformed the linear feedback control and the offline LQR control under 150% traffic volumes, indicating that the system



(a) Initial N-S green time = 20 s



(b) Initial N-S green time = 40 s



(c) Initial N-S green time = 60 s

Figure 3.8: Results of ablation study: changing of average vehicle delay during the simulation test period with 150% off-peak traffic volume. Solid colored lines represent the mean and shaded areas represent the mean  $\pm$  standard deviation interval.

modeling approach and the online parameter updating strategy of the proposed adaptive LQR method can effectively improve the performance.

Despite the demonstrated advantages, the proposed LQR method in this study has potential limitations, and future research in the following areas is needed.

- Currently, the proposed method only models a specific cycle length (90 s) with two phases to find optimal timing and phasing strategies, different cycle lengths and phasing combinations should be explored. However, looking at system model (3.1) or (3.5), it can be seen that adding more phases at each intersection will not constitute difficulties as this would merely increase the dimensionality of input  $u(k)$  and  $y(k)$  in (3.5), while the control design method remains the same as those in (3.18) – (3.20).
- The current system model assumes no noise in (3.1). Future work is needed to incorporate a stochastic noise term into the dynamic traffic system model to better account for the variability of the traffic system.
- The proposed model assumes that the system can be linearized as shown in (3.3) or (3.5). Future work will consider nonlinear traffic system modeling approaches, such as bi-linear and neural network models, aiming to better represent true characteristics of the traffic system.
- The proposed method has been tested in a traffic simulation environment. Implementing the proposed algorithm in a real-world traffic corridor/network is needed to test its effectiveness.

## Chapter 4

# Network Traffic Signal Control with Adaptive On-line Learning Scheme Using Multiple-Model Neural Networks

In this chapter, a network traffic signal control scheme is developed based on a non-linear approximation of the original traffic network model. The original traffic network model is approximated with a set of different neural networks, whose weight parameters are updated along with system operation. Then, a switching law is defined to find the most accurate approximation of the original traffic network model at each time step. This work is considered as an extension of the work in Chapter 3. The *main contributions* of this chapter include:

- Represented the unknown traffic network dynamic with a set of neural network approximation models.
- Derived the system identification scheme to update the weight parameters in the neural network model. Defined a switching law to find the most accurate neural

network approximation.

- Designed an optimal control scheme based on the neural network approximation model to minimize traffic delay.

## 4.1 System Model and Control Objective

In this section, the traffic network model is introduced. Then, the control objective is given in terms of total fuel consumption minimization.

### 4.1.1 Traffic Network Model

To model a traffic network with  $m$  intersections, we can make the following assumptions which can represent a majority of situations: 1) we assume that each intersection connects two roads, one from N-S direction and one from E-W direction; 2) we assume that the standard traffic signal control at each intersection shares a same a fixed signal length (100 s) with two phases: the E-W approaches share one phase, and the N-S approaches share the other phase. These assumptions are made to simplify the formulation of the control algorithm. Figure. 5.5 shows an example of the traffic network system, where Figure 4.1(a) is the traffic network example,  $N_{i,j}$  is the number of the intersection and  $l_1, l_2, l_3, l_4$  are the distance of  $N_{i,j}$  to its neighboring intersections, and Figure. 4.1(b) is the traffic signal sequence example, with  $T_{cyc} = 100$  being the fixed signal length.

The traffic network system described above with  $m$  intersection can be modeled with a discrete-time non-linear system as follows:

$$\Delta z_o(t+1) = f(\Delta z_o(t)) + g(\Delta z_o(t))\Delta v(t), \quad (4.1)$$

where  $z_o(t) = [z_{o1}(t), z_{o2}(t), \dots, z_{o2m}(t)]^T \in R^{2m}$  is the system state vector which is the average traffic delay of E-W direction and N-S direction at each intersection in the network,  $v(t) = [v_1(t), v_2(t), \dots, v_m(t)]^T \in R^m$  is the control input, where the components are the green signal percentage of the E-W direction at each intersection,  $v_i \in [0.2, 0.8]$ ,  $i = 1, \dots, m$

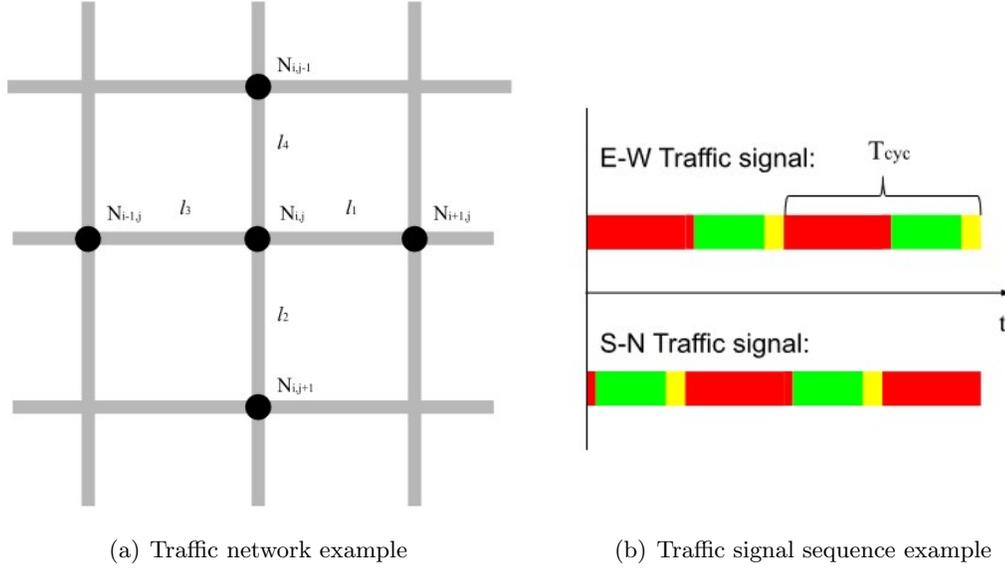


Figure 4.1: Traffic network system

indicates that the green signal length signal length should not be shorter than 20 section and should not exceed 80 second on each direction at every intersection.  $\Delta z_o(t) = z_o(t) - z_o(t-1)$ ,  $\Delta v(t) = v(t) - v(t-1)$  are the increment of  $z$  and  $v$  at time step  $t$ .  $f \in R^{2m}$ ,  $g \in R^{2m \times m}$  are the unknown non-linear function with respect to  $\Delta z_o$ ,  $t+1$  and  $t$  represent  $kT + T$  and  $kT$  with  $T$  being the sampling interval. For this problem, since the state is the average traffic delay at each intersection and the control signal  $v$  will not change within one signal cycle. The sampling interval  $T = 100s$  is chosen to be the same as the signal cycle length. To simplify notation, we denote  $\Delta z_o = y_o$  and  $\Delta v = u$  are we can rewrite the original non-linear model as:

$$x_o(t+1) = f(x_o(t)) + g(x_o(t))u(t). \quad (4.2)$$

Since the direct mathematical relationship between traffic delay and green signal period is unknown.  $f$  and  $g$  are unknown nonlinear functions and are very likely to be time varying. Thus, we need to design a parametrizable model which can accurately approximate the unknown non-linear relationship  $f$  and  $g$  online.

### 4.1.2 Control Objective

The control objective is to design an optimal control input  $u(t)$  within the range  $[u_{min}, u_{max}]$  for the system (4.2) to achieve minimal travel delay at  $t+1$  time step. This control objective can be summarized to solving the optimal control problem with cost function and constraint as:

$$J_o(t) = \mathbb{P}^T x_o(t+1) \quad (4.3)$$

$$u_j(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \dots, \quad j = 1, 2, \dots, m \quad (4.4)$$

where  $\mathbb{P} \in R^{2m}$  with every element greater equal to zero,  $u_{min}$ ,  $u_{max}$  are the lower and upper bound for the control signal  $u(t)$ , which indicates the maximum change in green signal period between signal cycles. This bound is used to improve system robustness. This cost function will change at each time step  $t$ , meaning that the optimal controller  $u(t)$  is calculated based on  $J_o(t)$  at each time step  $t$ . The cost function represent the weighted summation of the average traffic delay of all the intersections in the traffic network at the next time step. At each time step  $t$ , one can minimize  $x(t+1)$ , indicating an minimum traffic delay operation. Moreover, since the plant is a traffic system, the use of history information in the cost function is not necessary since the traffic system contains high randomness. In this work, we will design a one-step optimization scheme.

## 4.2 Nominal Neural-Network Based Design and Parameterization

In this section, the nominal neural-network based design is presented. First, the nominal neural network approximation is given. Then the nominal neural network based control law is derived.

### 4.2.1 Nominal Neural Network Approximation

Since the traffic system contains high randomness, it is difficult to capture all the system dynamics in (4.2) with a linear approximation. To develop an effective traffic signal control scheme, a better identification of the traffic network system is necessary. As mentioned in Chapter 2, neural network approximation has been commonly used for nonlinear complex function approximations and many areas of machine learning studies. More importantly, nowadays the research on machine learning to time series problem has become a key research focus [88]. This has inspired this research to use neural network structure in the approximation of the traffic system dynamic model. Using recurrent neural networks, a nominal approximation model of (4.2) can be constructed as:

$$x(t+1) = Ax(t) + W_1^* S_1(x(t)) + W_g^* \Phi_g(x(t), u(t)), \quad (4.5)$$

where  $x(t) \in R^{2m}$ ,  $u(t) \in R^m$  are the neural network approximation of system state, control input, respectively,  $A \in R^{n \times n}$  is a chosen stable matrix.  $W_1^*$ ,  $S_1$ ,  $W_g^*$  and  $\Phi_g$  are the parameterized neural network components:

$$W_g^* = [W_2^*, W_3^*, \dots, W_{m+1}^*] \quad (4.6)$$

$$\Phi_g(t) = [S_2(x(t))u_1(t), \dots, S_{m+1}(x(t))u_m(t)]^T, \quad (4.7)$$

where  $W_p^* \in R^{n \times l_p}$ ,  $p = 1, 2, \dots, m+1$ ,  $l_p$  is the number of layer for the  $p$ th neural network.

The neural network structure of the  $p$ th neural network is given as

$$S_p(x(t)) = [S_1^p, S_2^p, \dots, S_{l_p}^p] \in R^{l_p} \quad (4.8)$$

$$S_j^p(x) = \prod_{k=1, \dots, n} (s^p(x_k))^{d_k^p(j)}, \quad j = 1, \dots, l_p \quad (4.9)$$

$$s^p(x_k) = k_1 / (1 + e^{-x_k}) + k_2 \quad (4.10)$$

where with  $k_1, k_2$  being the design parameters.  $x$  is the estimates of the original state  $x_o$ . It has been shown in [89] that there exist ideal matrices  $W_1^*, \dots, W_{m+1}^*$  with appropriate  $S_1(x), \dots, S_{m+1}(x)$  such that (4.5) can approximate (4.1) to any degree of accuracy on any compact set.

**Parameterization of  $x_o(t)$ .** With the above neural network approximation, the original traffic system states  $x_o(t)$  can be expressed as

$$x_o(t) = x(t) + \delta(t), \quad (4.11)$$

where  $\delta(t)$  is the approximation error. With (4.5) - (4.7), one can further parameterize the system output  $y(t)$  is follows

$$x_o(t) = \theta^* \Phi(t-1) + \delta(t), \quad (4.12)$$

where

$$\theta^* = [A, W_1^*, W_2^*, \dots, W_{m+1}^*] \quad (4.13)$$

$$\Phi(t) = [x(t)^T, S_1(x(t))^T, (S_2(x(t))u_1(t))^T, \dots, (S_{m+1}(x(t))u_m(t))^T]^T, \quad (4.14)$$

are the system parameter vector and the regressor vector. It has been shown in [89] that there exist ideal matrices  $W_1^*, \dots, W_{m+1}^*$  with appropriate  $S_1(x), \dots, S_{m+1}(x)$  such that (4.5) can approximate (4.1) to any degree of accuracy on any compact set. As a result, one can make the assumption that is an exponential decay term, whose effect can be ignored. We assume  $\delta(t)$  satisfy

$$|\delta(t)| \leq \delta_1(t) \|\Phi(t)\|_2 + \delta_2(t) \quad (4.15)$$

where  $\delta_1(t) \in L^{1+\beta}$  and  $\delta_2(t) \in L^{1+\beta}$ ,  $1 \leq \beta < \infty$ , i.e.,

$$\sum_{t=0}^{\infty} |\delta_1(t)|^{1+\beta} < \infty, \quad \sum_{t=0}^{\infty} |\delta_2(t)|^{1+\beta} < \infty \quad (4.16)$$

### 4.2.2 Nominal Control Design

To achieve the control objectives stated in the previous section, we need to solve the optimization problem defined in (4.3) - (4.4). With the assumption in the previous section that (4.5) is a good approximation of (4.1), the optimization problem (4.3) - (4.4) can be reformulated with the neural network approximation model as

$$J(t) = \mathbb{P}^T x(t+1) \quad (4.17)$$

$$u_j^*(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \dots, \quad j = 1, 2, \dots, m \quad (4.18)$$

where  $x$  is the neural network approximation of the original traffic delay  $x_o$ . When the accurate approximation of the original traffic network system is known, we can design the nominal optimal controller by solving optimization problem, apply (4.5) to (4.17), one can obtain

$$J(t) = \mathbb{P}^T x(t+1) = \mathbb{P}^T Ax(t) + \mathbb{P}^T W_1^* S_1(x(t)) + \mathbb{P}^T W_g^* \Phi_g(x(t), u^*(t)). \quad (4.19)$$

Since at each time step  $t$  the traffic delay term  $x(t)$  is known, the above equation is linear with respect to  $u(t)$ . This optimization problem is a linear programming problem.

**Optimal Control Design.** To solve for the optimal control signal  $u^*(t)$ , by taking the derivative of the cost function with respect to  $u(t)$ , we have

$$\frac{\partial J(t)}{\partial u(t)} = \frac{\partial \mathbb{P}^T W_g^* \Phi_g(x(t), u(t))}{\partial u(t)} \quad (4.20)$$

$$= \frac{\partial \mathbb{P}^T (W_2^* S_2(x(t))u_1(t) + \dots + W_{m+1}^* S_{m+1}(x(t))u_m(t))}{\partial u(t)} = d^*(t) \quad (4.21)$$

where

$$d^*(t) = [d_1^*(t), d_2^*(t), \dots, d_m^*(t)]^T \in R^m \quad (4.22)$$

$$d_j^*(t) = \mathbb{P}W_{j+1}^*S_{j+1}(x(t)), \quad j = 1, 2, \dots, m \quad (4.23)$$

With  $x(t)$  known at time step  $t$ ,  $d_j^*(t)$  is a constant at every time step. Based on the sign of  $d_j^*(t)$ , the control input for minimal cost function  $J$  will lie on the boundary of  $u(t)$ . As a result, we come up with the following optimal control design:

$$u_j^*(t) = \begin{cases} u_{max}, & d_j^*(t) < 0 \\ 0, & d_j^*(t) = 0 \\ u_{min}, & d_j^*(t) > 0 \end{cases} \quad (4.24)$$

where for  $d_j^*(t) = 0, j = 1, \dots, m$ , the solution for  $u_j(t)$  is trivial. Thus, we manually set  $u_j(t) = 0$  for this situation.

**System Robustness.** For linear programming problem, the solution for the control signal  $u^*(t)$  will always lie on one of the boundaries. Since the traffic system is operation around  $z(t)$  and  $v(t)$  at time step  $t$ . To guarantee the system robustness, the maximum increment for the control signal  $u^*(t)$  should be relatively small, i.e.,  $|u^*(t)| < \bar{u}$ ,  $u_{max} = \bar{u}$ ,  $u_{min} = -\bar{u}$ , and  $\bar{u} < 0.1$  is the maximum amount of increment for  $u^*(t)$ .

### 4.3 Multiple-Model Neural Network Based Design and Parameterization

Although there exist ideal matrices  $W_1^*, \dots, W_{m+1}^*$  with appropriate  $S_1(x), \dots, S_{m+1}(x)$  such that (4.5) can accurately approximate (4.1). In reality, the exact value of  $W_1^*, \dots, W_{m+1}^*$  are unavailable, and we need to design appropriate parameter update law to estimate  $W_1^*, \dots, W_{m+1}^*$ . Moreover, unlike off-line neural network approximation, where we can test and compare the approximation accuracy with different neural network models, it is diffi-

cult to fine the ideal approximation model with only one choice of neural network structure in on-line approximation. Thus, in this section, we propose to use multiple-model based neural networks to approximate the traffic network system in (4.1).

### 4.3.1 Multiple-Model based Neural Network Approximation

To accurately approximate (4.1) using neural networks, we propose to use  $N$  multiple neural network structures. Using recurrent neural networks, the approximation models can be constructed as

$$x_{(i)}(t+1) = A_{(i)}x_{(i)}(t) + W_{(i)1}^* S_{(i)1}(x_{(i)}(t)) + W_{(i)g}^* \Phi_{(i)g}(x_{(i)}(t), u_{(i)}(t)) \quad (4.25)$$

where  $x_{(i)}(t) \in R^{2m}$ ,  $u_{(i)}(t) \in R^m$  are the system state, control input respectively,  $A_{(i)} \in R^{n \times n}$  is a chosen stable matrix,  $i = 1, 2, \dots, N$ , represents the  $i$ th neural network structure,  $W_{(i)1}^*$ ,  $S_{(i)1}$ ,  $W_{(i)g}^*$  and  $\Phi_{(i)g}$  are the parameterized neural network components:

$$W_{(i)g}^* = [W_{(i)2}^*, W_{(i)3}^*, \dots, W_{(i)m+1}^*] \quad (4.26)$$

$$\Phi_{(i)g}(t) = [S_{(i)2}(x_{(i)}(t))u_{(i)1}(t), \dots, S_{(i)m+1}(x_{(i)}(t))u_{(i)m}(t)]^T, \quad (4.27)$$

where  $W_{(i)p}^* \in R^{n \times l_{(i)p}}$ ,  $p = 1, 2, \dots, m+1$ ,  $l_{(i)p}$  is the number of layer for the  $p$ th neural network,  $S_{(i)p}(x(t)) = [S_{1p}^p, S_{2p}^p, \dots, S_{l_{(i)p}p}^p] \in R^{l_{(i)p}}$  are the basis functions.  $W_{(i)p}$  and  $S_{(i)p}$  are the different neural network structures used to approximate the original traffic network system. With this multiple model based online approximation design, it is more likely to find an ideal neural network structure to approximate traffic network system in (4.1).

**Parameterization of  $x_o(t)$ .** With the multiple-model based neural network approximation, the original traffic delay vector  $x_o(t)$  can be expressed as

$$x_o(t) = x_{(i)}(t) + \delta_{(i)}(t), \quad (4.28)$$

where  $\delta_{(i)}(t)$  is the approximation error of the  $i$ th neural network approximation. With

(4.25) - (4.27), one can further parameterize the system output  $x_o(t)$  as follows

$$x_o(t) = \theta_{(i)}^* \Phi_{(i)}(t-1) + \delta_{(i)}(t), \quad (4.29)$$

where

$$\theta_{(i)}^* = [A_{(i)}, W_{(i)1}^*, W_{(i)2}^*, \dots, W_{(i)m+1}^*] \quad (4.30)$$

$$\Phi_{(i)}(t) = [x_{(i)}(t)^T, S_{(i)1}(x_{(i)}(t))^T, (S_{(i)2}(x_{(i)}(t))u_{(i)}(t))^T, \dots, (S_{(i)m+1}(x_{(i)}(t))u_{(i)}(t))^T]^T, \quad (4.31)$$

are the system parameter vectors and the regressor vectors.

## 4.4 Adaptive Control Design

In this section, the adaptive control scheme is designed for the neural network approximation based traffic network systems. First, the design adaptive control for a single neural network based traffic network system approximation is presented. Then the multiple-model neural network based adaptive control design is presented.

### 4.4.1 Single Neural Network Based Adaptive Control Design

**Optimal Control Design.** The neural network approximation of the cost function is defined in (4.17). Since the neural network parameters are unknown, at time step  $t$ , assume that the system is observable, one can only obtain the estimate of  $x(t+1)$  as  $\hat{x}(t+1)$ . The estimate optimization problem is defined as:

$$\hat{J}(t) = \mathbb{P}^T \hat{x}(t+1) \quad (4.32)$$

$$u_j(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \dots, \quad j = 1, 2, \dots, m \quad (4.33)$$

where  $\hat{J}(t)$  is the estimate of the cost function  $J(t)$ . Define  $d_j(t) = \mathbb{P}W_{j+1}(t)S_{j+1}(x(t))$ , and based on the derivation on Section 4.2.2, we come up with the following optimal control

design:

$$u_j(t) = \begin{cases} u_{max}, & d_j(t) < 0 \\ 0, & d_j(t) = 0 \\ u_{min}, & d_j(t) > 0 \end{cases} \quad (4.34)$$

where  $W_{j+1}(t)$  are the estimates of the nominal parameter  $W_{j+1}^*$ .

**Parameter Estimation.** To obtain the state estimation  $\hat{x}(t)$ , we define the estimated model as:

$$\hat{x}(t+1) = \theta(t)\Phi(t), \quad (4.35)$$

where  $\theta(t)$  is the estimate of the system parameter  $\theta^*$ . To develop the adaptive update laws for  $\theta(t)$ , we introduce the estimation error

$$\varepsilon(t) = \theta(t-1)\Phi(t-1) - x_o(t) \quad (4.36)$$

$$\approx \tilde{\theta}(t-1)\Phi(t-1), \quad (4.37)$$

where  $\tilde{\theta} = \theta(t) - \theta^*$ , and we assume that the approximation error  $\delta(t)$  can be ignored. Based on a gradient algorithm, we choose the following adaptive parameter update laws as:

$$\theta(t+1) = \theta(t) - \frac{\Gamma\Phi(t)\varepsilon(t)}{m^2(t)}, \quad \theta(0) = \theta_0, \quad (4.38)$$

with  $\Gamma = \Gamma^T > 0$  is a gain matrix,  $m(t) = \sqrt{1 + \alpha\Phi^T(t)\Phi}$  with  $\alpha > 0$  being a design parameter, and  $\theta_0$  is the initial estimate of  $\theta^*$ .

The adaptive law has the desired properties:

*Lemma 1:* The adaptive parameter law (4.38) for the system (4.1) ensures that  $\theta(t) \in L^\infty$ ,  $\frac{\varepsilon(t)}{m(t)} \in L^2 \cap L^\infty$ ,  $\theta(t) - \theta(t-1) \in L^2$ .

*Proof:* Define the positive definite function  $V(\tilde{\theta}) = \tilde{\theta}^T\Gamma^{-1}\tilde{\theta}$ , with adaptive law (4.38) and the estimation definition (4.36), we calculate the increment of  $V(\tilde{\theta})$  as

$$V(\tilde{\theta}(t+1)) - V(\tilde{\theta}(t)) \quad (4.39)$$

$$= - \left( 2 - \frac{\Phi^T(t)\Gamma\Phi(t)}{m^2(t)} \right) \frac{\varepsilon^2(t)}{m^2(t)} + 2 \frac{\varepsilon(t)\delta(t)}{m^2(t)} \quad (4.40)$$

$$\leq - \alpha_1 \frac{\varepsilon^2(t)}{m^2(t)} + 2 \frac{\varepsilon(t)\delta(t)}{m^2(t)} \quad (4.41)$$

$$\leq - \alpha_1 \frac{\varepsilon^2(t)}{m^2(t)} - \frac{\alpha_1}{2} \left( \frac{|\varepsilon(t)|}{m(t)} - \frac{|\delta(t)|}{\alpha_1 m(t)} \right) + 2 \frac{\delta^2(t)}{\alpha_1 m^2(t)} \quad (4.42)$$

for some constant  $\alpha_1 > 0$ . From (4.36), we have

$$\sum_{t=0}^{\infty} 2 \frac{\delta^2(t)}{\alpha_1 m^2(t)} \leq k_d \quad (4.43)$$

for some constant  $k_d > 0$ . Then, we have

$$V(\tilde{\theta}(t)) \leq V(\tilde{\theta}(0)) + \sum_{\tau=0}^{\infty} 2 \frac{\delta^2(\tau)}{\alpha_1 m^2(\tau)} + k_d, \quad \forall t \geq 0. \quad (4.44)$$

Thus, we have  $\theta(t) \in L^\infty$ ,  $\frac{\varepsilon(t)}{m(t)} \in L^2 \cap L^\infty$ ,  $\theta(t) - \theta(t-1) \in L^2$ .

#### 4.4.2 Multiple-Model Neural Network Based Adaptive Control Design

**Optimal Control Design.** The multiple-model based approximation of the cost function (4.3) - (4.4) using neural networks in (4.25) is defined as

$$J_{(i)}(t) = \mathbb{P}^T x_{(i)}(t+1) \quad (4.45)$$

$$u_{(i)j}(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \dots, \quad j = 1, 2, \dots, m \quad (4.46)$$

where  $J_{(i)}$ ,  $x_{(i)}$  are the  $i$ th neural network approximation of the cost function (4.3) - (4.4) and traffic delay vector  $x_o$ . The estimate of the above cost function is

$$\hat{J}_{(i)}(t) = \mathbb{P}^T \hat{x}_{(i)}(t+1) \quad (4.47)$$

$$u_{(i)j}(t) \in [u_{min}, u_{max}], \quad t = 1, 2, \dots, \quad j = 1, 2, \dots, m \quad (4.48)$$

where  $\hat{J}_{(i)}$ ,  $\hat{x}_{(i)}$  are the estimates of the  $i$ th neural network approximation the cost function (4.47) - (4.48) and traffic delay vector  $x_{(i)}$ . Define  $d_{(i)j}(t) = \mathbb{P}W_{(i)j+1}(t)S_{j(i)+1}(x_{(i)}(t))$ , and based on the derivation on Section 4.2.2, we come up with the following optimal control design:

$$u_{(i)j}(t) = \begin{cases} u_{max}, & d_{(i)j}(t) < 0 \\ 0, & d_{(i)j}(t) = 0 \\ u_{min}, & d_{(i)j}(t) > 0 \end{cases} \quad (4.49)$$

where  $W_{(i)j+1}(t)$  is the  $j$ th estimate of the nominal parameter  $W_{j+1}^*$ .

**Parameter Estimation.** To obtain the state estimation  $\hat{x}_{(i)}(t)$ , we define the estimated model as:

$$\hat{x}_{(i)}(t+1) = \theta_{(i)}(t)\Phi_{(i)}(t), \quad (4.50)$$

where  $\theta_{(i)}(t)$  is the  $i$ th NN estimate of the system parameter  $\theta^*$ . To develop the adaptive update laws for  $\theta_{(i)}(t)$ , we introduce the estimation error

$$\varepsilon_{(i)}(t) = \theta_{(i)}(t-1)\Phi_{(i)}(t-1) - y(t) \quad (4.51)$$

$$\approx \tilde{\theta}_{(i)}(t-1)\Phi_{(i)}(t-1), \quad (4.52)$$

where  $\tilde{\theta}_{(i)} = \theta_{(i)}(t) - \theta_{(i)}^*$ , and we assume that the approximation error  $\delta_{(i)}(t)$  can be ignored. Based on a gradient algorithm, we choose the following adaptive parameter update laws as:

$$\theta_{(i)}(t+1) = \theta_{(i)}(t) - \frac{\Gamma_{(i)}\Phi_{(i)}(t)\varepsilon_{(i)}(t)}{m^2(t)}, \quad \theta_{(i)}(0) = \theta_{(i)0}, \quad (4.53)$$

with  $\Gamma_{(i)} = \Gamma_{(i)}^T > 0$  is a gain matrix,  $m(t) = \sqrt{1 + \alpha\Phi_{(i)}^T(t)\Phi_{(i)}(t)}$  with  $\alpha > 0$  being a design parameter, and  $\theta_{(i)0}$  is the initial estimate of  $\theta_{(i)}^*$ .

Using the similar derivation as Lemma 1, the adaptive law (4.53) has the desired properties:

*Lemma 2:* The adaptive parameter law (4.53) for the system (4.1) ensures that  $\theta_{(i)}(t) \in L^\infty$ ,  $\frac{\varepsilon_{(i)}(t)}{m(t)} \in L^2 \cap L^\infty$ ,  $\theta_{(i)}(t) - \theta_{(i)}(t-1) \in L^2$ , for  $i = 1, 2, \dots, N$ .

**Control Switching Scheme.** According to the above design,  $N$  adaptive controllers are obtained, and a control switching scheme is needed to choose the current controller. The switching scheme in this study is designed based on the estimation cost with different neural network models. We define the estimation cost as

$$J_{(i)2}(t) = \sum_{j=1}^t \varepsilon_{(i)}^T(j) Q \varepsilon_{(i)}(j), \quad (4.54)$$

for  $i = 1, 2, \dots, N$ , where  $Q = Q^T$  is the constant design parameter matrix. The control signal  $u(t)$  is obtained as

$$u(t) = u_{(j)}(t), \quad J = \arg \min_{i=1,2,\dots,N} J_{(i)2}(t). \quad (4.55)$$

To prevent arbitrarily fast switching, we introduce a non-zero waiting time  $T_{min} > 0$  between two switches.

## 4.5 Simulation Study

In this section, the simulation study for the proposed multiple-model neural network based optimal control is presented. First, the simulation system is first introduced. Then the simulation results are presented. Finally, some discussion is given based on the simulation results.

### 4.5.1 Simulation System

In this section, the simulation system is introduced. In this simulation study, we consider a simple case, which is a two intersection traffic network. First, the traffic network model is introduced. Then, the neural network models used to approximated the original traffic network model are given.

### Traffic Network Model

For this study, we consider a two intersection traffic network. The traffic network structure is shown in Figure. 4.2. The road in this traffic network are "Road 1", "Road 2", and "Road 12".  $v_1 = 12.5\text{m/s}$ ,  $v_2 = 12.5\text{m/s}$ ,  $v_{12} = 16.6\text{m/s}$  are the average speed on each road,  $s_1 = 0.05\text{vehicle/s}$ ,  $s_2 = 0.1\text{vehicle/s}$ ,  $s_{12} = 1.13\text{vehicle/s}$  are the average traffic volume on each road,  $N_1$  and  $N_2$  are the two intersections,  $u_1$  and  $u_2$  are the traffic control signals on each intersection,  $l_{12}$  is the distance between  $N_1$  and  $N_2$ . Assume that the vehicle entrance follows a random distribution. Note that for this traffic network, we assume that the vehicle average speed and traffic volume on the two directions of each road are the same for simplicity.

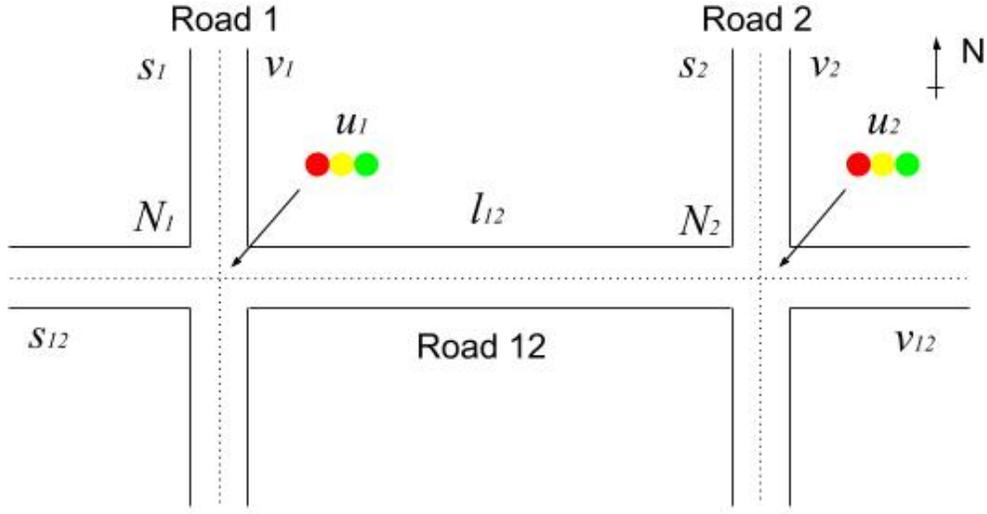


Figure 4.2: Two intersection traffic network example.

### Single Neural Network Approximation Model

For the traffic network system described above, it can be approximated with a neural network based dynamic model

$$x(t+1) = A^*x(t) + W_1^*S_1(x(t)) + W_g^*\Phi_g(x(t), u(t)) = \theta^*\Phi(t), \quad (4.56)$$

where  $x = [x_{EW1}, x_{NS1}, x_{EW2}, x_{NS2}]^T \in R^4$  are the traffic delay of both direction on each intersection,  $u = [u_1, u_2]^T$  are the control signal, which is the green cycle period of the E-W direction.  $A^*$ ,  $W_1^*$ ,  $W_g^*$  are the nominal system parameters, chosen as follows:

$$A^* = \begin{bmatrix} 0.9884 & -0.0467 & 0.0058 & 0.0017 \\ -0.0071 & 0.9643 & 0.0104 & -0.0017 \\ -0.0044 & -0.0341 & 1.0047 & -0.0073 \\ -0.0071 & -0.02937 & 0.0122 & 1.0060 \end{bmatrix}, \quad W_1^* = \begin{bmatrix} 0.9286 & 0.5130 & 0.2319 & 0.3595 \\ 0.7592 & 0.4136 & 0.7967 & 0.5036 \\ 0.6176 & 0.4858 & 0.8501 & 0.2002 \\ 0.5503 & 0.2894 & 0.3402 & 0.9055 \end{bmatrix}, \quad (4.57)$$

$$W_g^* = [W_2^*, W_3^*] \quad (4.58)$$

$$W_2^* = \begin{bmatrix} 0.1949 & 0.3273 & 0.3026 & 0.0873 \\ 0.6075 & 0.2799 & 0.9113 & 0.5158 \\ 0.8831 & 0.3655 & 0.2541 & 0.9572 \\ 0.4195 & 0.7796 & 0.5760 & 0.1937 \end{bmatrix}, \quad W_3^* = \begin{bmatrix} 0.8512 & 0.8501 & 0.1778 & 0.6718 \\ 0.5600 & 0.0988 & 0.3279 & 0.4857 \\ 0.4690 & 0.7370 & 0.2108 & 0.4138 \\ 0.5000 & 0.1745 & 0.0159 & 0.3580 \end{bmatrix}. \quad (4.59)$$

Based on (4.14), we have  $\Phi_g(t) = [S_2(x(t))u_1(t), S_3(x(t))u_2(t)]^T$ .  $S_1(x(t))$ ,  $S_2(x(t))$ ,  $S_3(x(t))$  are the activation functions which have the following structures:

$$sig(x) = \frac{1}{1 + e^{-x}} + 0.5 \quad (4.60)$$

$$S_1(x(t)) = \begin{bmatrix} sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{EW1}) \\ sig(x_{EW2}) * sig(x_{NS2}) \\ sig(x_{EW1}) * sig(x_{EW2}) \end{bmatrix}, \quad S_2(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{NS1}) * sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) * sig(x_{NS2}) \end{bmatrix}, \quad (4.61)$$

$$S_3(x(t)) = \begin{bmatrix} sig(x_{NS1}) * sig(x_{EW1}) \\ sig(x_{EW1}) \\ sig(x_{EW2}) * sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}. \quad (4.62)$$

#### 4.5.2 Simulation Results

In this section, the simulation results are presented. First, the single neural network based optimal control design results are shown. Then the multiple-model neural network based optimal control results are given.

##### Single Neural Network based Optimal Control Simulation

The single neural network estimate of the model (4.56) is

$$\hat{x}(t+1) = A(t)\hat{x}(t) + W_1(t)S_1(\hat{x}(t)) + W_g(t)\Phi_g(\hat{x}(t), u(t)) \quad (4.63)$$

$$= \theta(t)\Phi(t) \quad (4.64)$$

where  $\theta(t)$  is the estimate of the nominal parameter  $\theta^*$ .

**Estimation Result.** First, we tested the estimation of  $x_o(t)$ , which is obtained directly from the simulation model. With the initial estimate being  $0.8\theta^*$ . The estimation results for average traffic delay increment are shown in Figure. 4.3, and Figure. 4.4 is shows the estimation error of average traffic delay increment. From which we can see that the estimation model (4.63) can well approximate the original traffic network model. The estimation error can become very small after 5 signal cycles. Note that each step is one signal cycle.

**Optimal Control Result.** To test out the optimal control result, we compared our control algorithm with different fixed-time signal control. We choose the weight vector  $\mathbb{P} = [0.05, 0.13, 0.1, 0.13]^T$  which is determined based on the traffic flow on each road. We

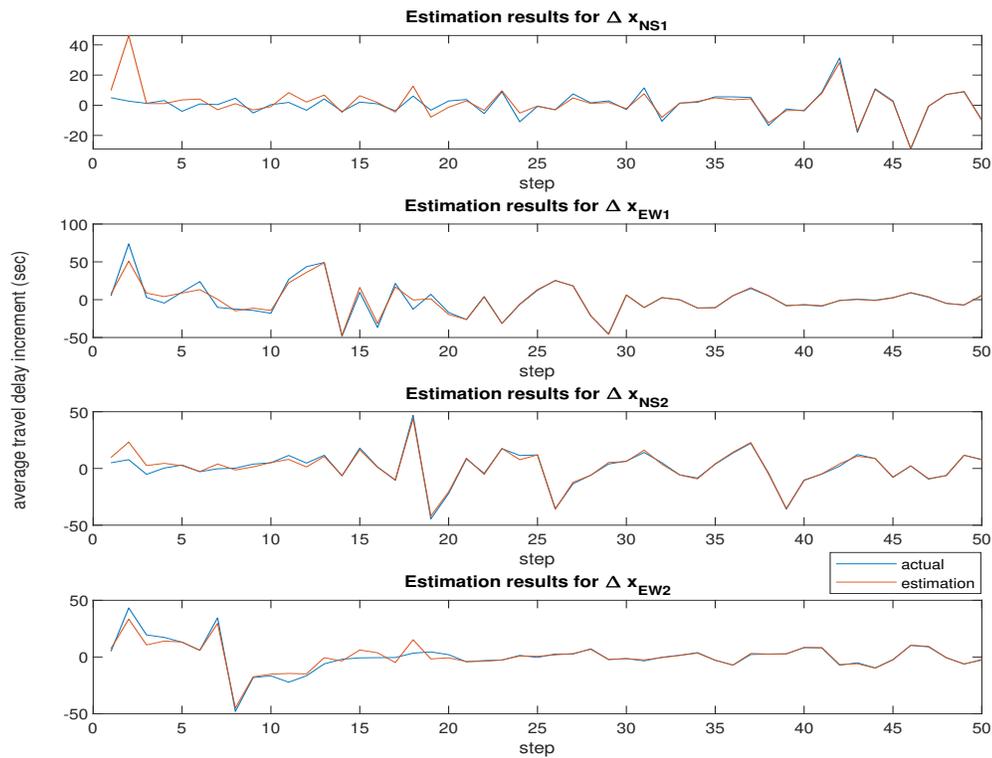


Figure 4.3: Estimation result for  $\Delta x$ .

compared our algorithm with the initial green signal on E-W direction being 20s, 50s, and 70s respectively. We ran each case for 20 runs to collect the average traffic delay data. Table. 4.1 shows the average traffic delay using a different method. From which we can see that using our optimal control method the average traffic delay is significantly reduced with all the initial green time options. Figure. 4.5 shows the optimal control result with initial green time on the E-W direction being 20 section. From which we can see that with our algorithm, we can make the average traffic delay significantly drop compared to fixed-time methods.

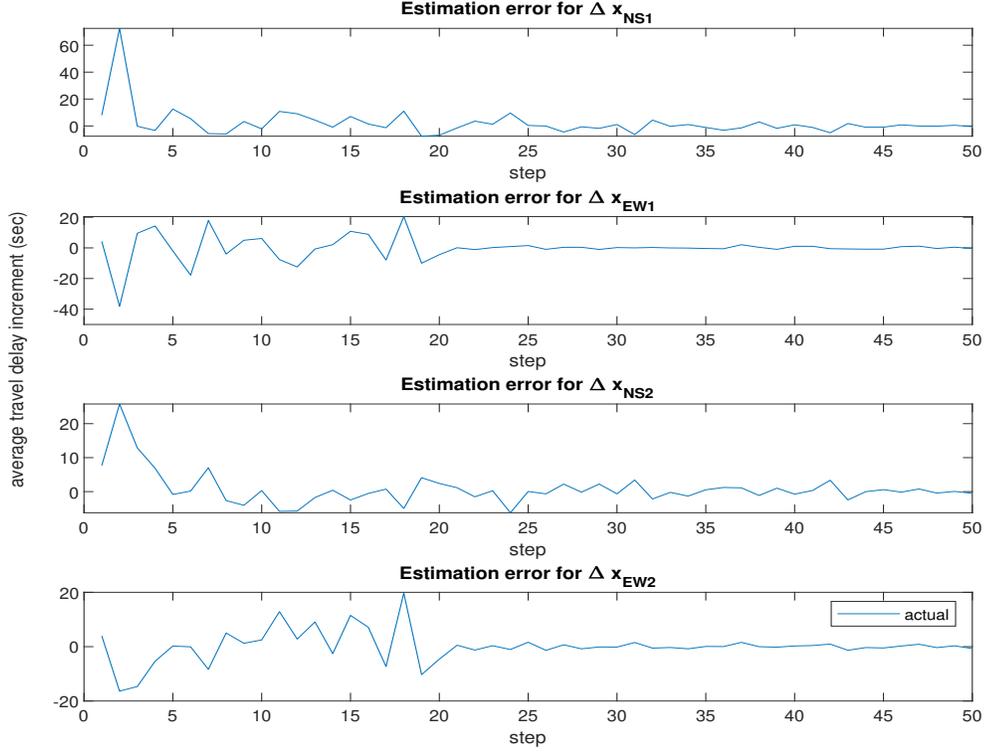


Figure 4.4: Estimation error for  $\Delta x$ .

### Multiple-Model Neural Network based Optimal Control Simulation

The multiple-model neural network estimate of the model (4.56) is

$$\hat{x}_{(i)}(t+1) = A_{(i)}(t)\hat{x}_{(i)}(t) + W_{(i)1}(t)S_{(i)1}(\hat{x}_{(i)}(t)) + W_{(i)g}(t)\Phi_{(i)g}(\hat{x}_{(i)}(t), u_{(i)}(t)) \quad (4.65)$$

$$= \theta_{(i)}(t)\Phi(t) \quad (4.66)$$

where  $\theta_{(i)}(t)$  is the  $i$ th neural network estimate of the nominal parameter  $\theta^*$ , and  $\theta_{(i)}(0)$  is chosen to be  $0.8\theta^*$  for  $i = 1, \dots, N$ . We set  $N = 3$ , and the activation functions  $S_{(i)1}(x(t))$ ,  $S_{(i)2}(x(t))$ ,  $S_{(i)3}(x(t))$  are chosen as follows:

$$sig(x) = \frac{1}{1 + e^{-x}} + 0.5 \quad (4.67)$$

Table 4.1: Travel delays with different initial green times and control methods

Init. Green Time (s)	Control Method	Avg. Veh. Delay (s)	Std. Dev.
20	Fixed-time	67.52	6.25
50	Fixed-time	19.82	3.56
70	Fixed-time	21.28	2.10
20	NN-based Optimal Control	30.16	12.25
50	NN-based Optimal Control	18.01	1.68
70	NN-based Optimal Control	19.33	4.36

$$S_{(1)1}(x(t)) = \begin{bmatrix} sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{EW1}) \\ sig(x_{EW2}) * sig(x_{NS2}) \\ sig(x_{EW1}) * sig(x_{EW2}) \end{bmatrix}, S_{(1)2}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{NS1}) * sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) * sig(x_{NS2}) \end{bmatrix}, \quad (4.68)$$

$$S_{(1)3}(x(t)) = \begin{bmatrix} sig(x_{NS1}) * sig(x_{EW1}) \\ sig(x_{EW1}) \\ sig(x_{EW2}) * sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}, \quad (4.69)$$

$$S_{(2)1}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}, S_{(2)2}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix}, S_{(2)3}(x(t)) = \begin{bmatrix} sig(x_{NS1}) \\ sig(x_{EW1}) \\ sig(x_{NS2}) \\ sig(x_{EW2}) \end{bmatrix} \quad (4.70)$$

$$S_{(3)1}(x(t)) = \begin{bmatrix} sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{EW1}) * sig(x_{EW2}) \\ sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{NS2}) * sig(x_{EW2}) \end{bmatrix}, S_{(3)2}(x(t)) = \begin{bmatrix} sig(x_{EW1}) * sig(x_{EW2}) \\ sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{NS2}) * sig(x_{EW2}) \\ sig(x_{NS1}) * sig(x_{EW2}) \end{bmatrix}, \quad (4.71)$$

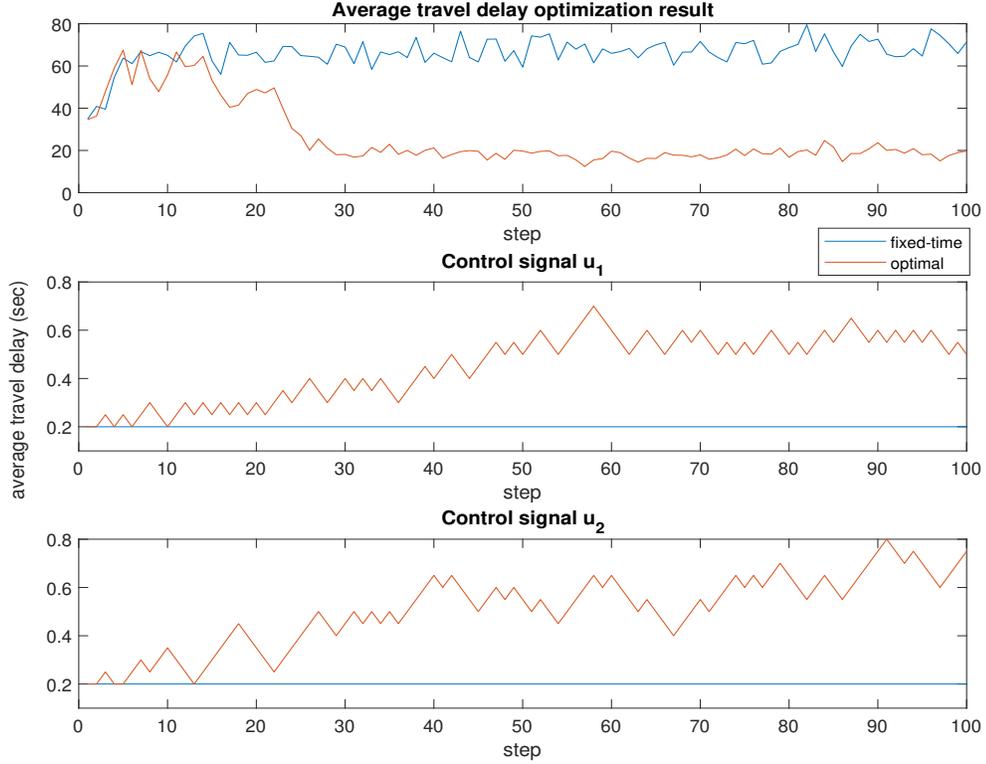


Figure 4.5: Optimization result with single Neural network.

$$S_{(3)3}(x(t)) = \begin{bmatrix} sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{NS1}) * sig(x_{NS2}) \\ sig(x_{EW1}) * sig(x_{NS2}) \\ sig(x_{NS1}) * sig(x_{EW2}) \end{bmatrix} \quad (4.72)$$

**Estimation Result.** First, we tested the estimation of  $x_o(t)$ , which is obtained directly from the simulation model. With the initial estimate being  $0.8\theta^*$ . The estimation results for average traffic delay increment are shown in Figure. 4.6, and Figure. 4.7 is shows the estimation error of average traffic delay increment. From which we can see that the estimation model (4.63) can well approximate the original traffic network model. Compared with estimation with single NN, multiple NN will give a better estimation of the original

traffic network system.

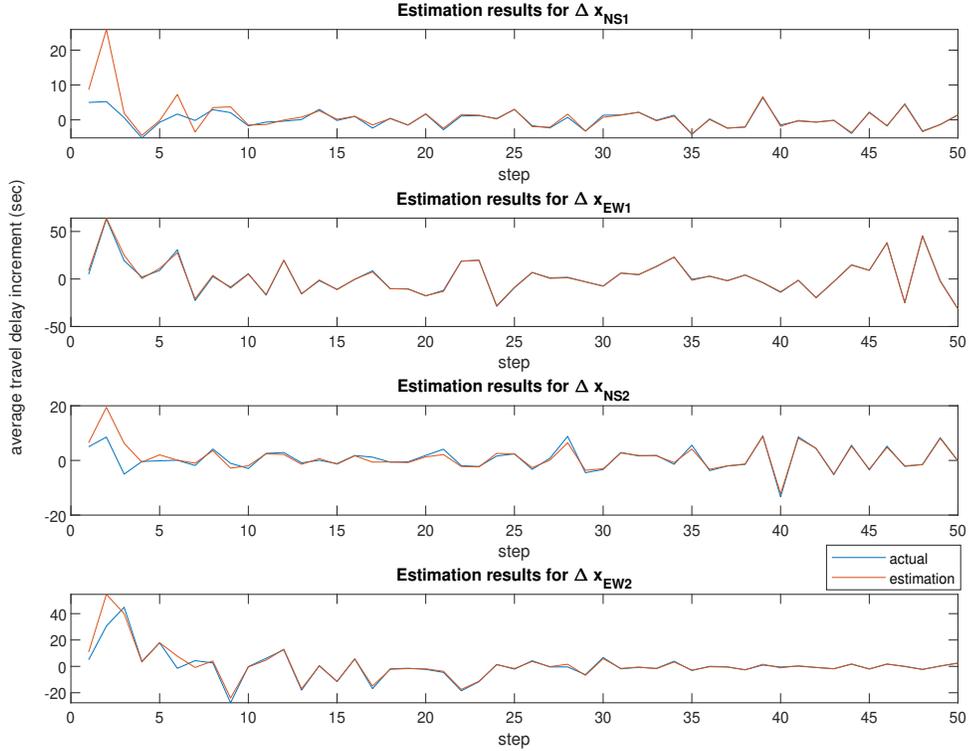


Figure 4.6: Estimation result for  $\Delta x$ .

**Optimal Control Result.** To test out the optimal control result, we compared our control algorithm with different fixed-time signal control. We choose the weight vector  $\mathbb{P} = [0.05, 0.13, 0.1, 0.13]^T$  which is determined based on the traffic flow on each road. We compared our algorithm with initial green signal on E-W direction being 20s, 50s, and 70s respectively. We ran each cases for 20 run to collect the average traffic delay data. Table. 4.2 shows the average traffic delay using different method. From which we can see that using our optimal control method the average traffic delay is significantly reduced with all the initial green time options. Figure. 4.8 shows the optimal control result with initial green time on E-W direction being 20 section. From which we can see that with our algorithm, we can make the average traffic delay significantly drop compared to fixed-time methods.

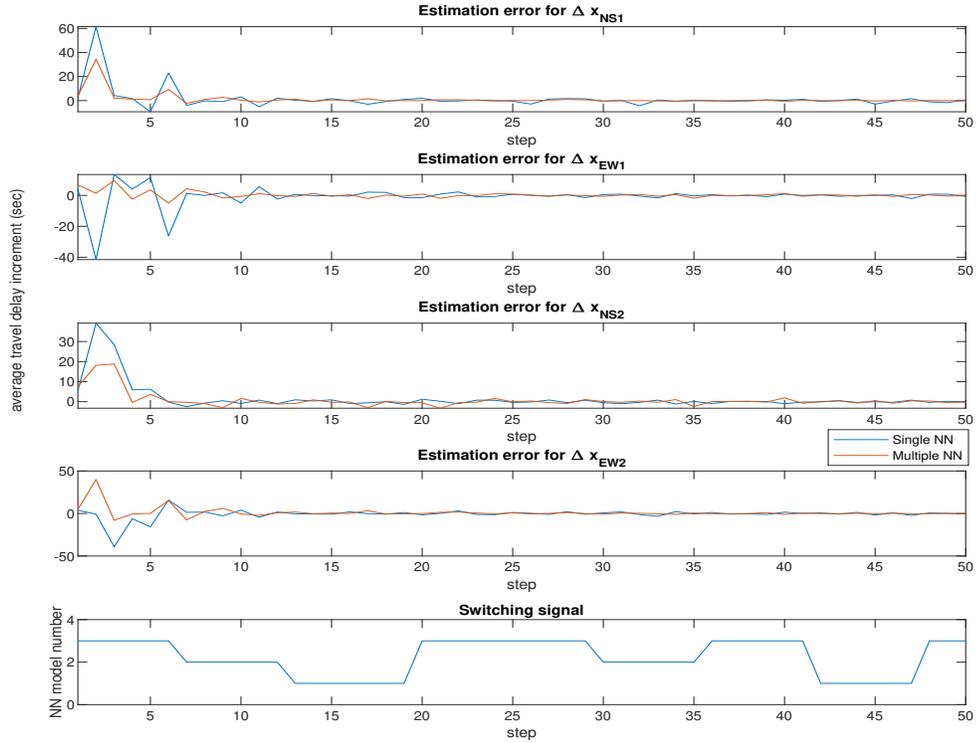


Figure 4.7: Estimation error for  $\Delta x$ .

Compared with single NN based optimal control scheme, the scheme with multiple neural network results in a better reduction on average traffic delay.

### 4.5.3 Discussion

From this simulation study, the effectiveness of the proposed adaptive on-line learning based optimal control design is verified. Based on the estimation results for the system state  $x_o$ , one can see that with both single neural network and multiple neural network model, a good estimation result can be obtained. However, with multiple neural network models, one would achieve a slightly better estimation of the original traffic network model. For the optimization results, one can see that with the proposed optimal control scheme, the average vehicle delay is significantly reduced compared with fixed-time traffic signal control.

Table 4.2: Travel delays with different initial green times and control methods

Init. Green Time (s)	Control Method	Avg. Veh. Delay (s)	Std. Dev.
20	Fixed-time	67.52	6.25
50	Fixed-time	19.82	3.56
70	Fixed-time	21.28	2.10
20	NN-based Optimal Control	26.06	13.25
50	NN-based Optimal Control	17.15	1.74
70	NN-based Optimal Control	16.61	2.39

## 4.6 Conclusions

In this chapter, a network traffic signal control scheme is designed with the use of an adaptive on-line learning scheme using multiple-model neural networks. First, NN based traffic network model is presented for both single NN and multiple NN. Then, the adaptive optimal control scheme is designed with an on-line learning scheme used to identify the traffic network model. Finally, a simulation study is presented to show the effectiveness of the proposed scheme. Based on the simulation results, one can conclude that the proposed multiple-model NN based on-line estimation scheme can well identify the original traffic network system. Moreover, the optimal control scheme designed based on the estimated traffic network model can achieve traffic delay minimization compared to fixed-time traffic signal control.

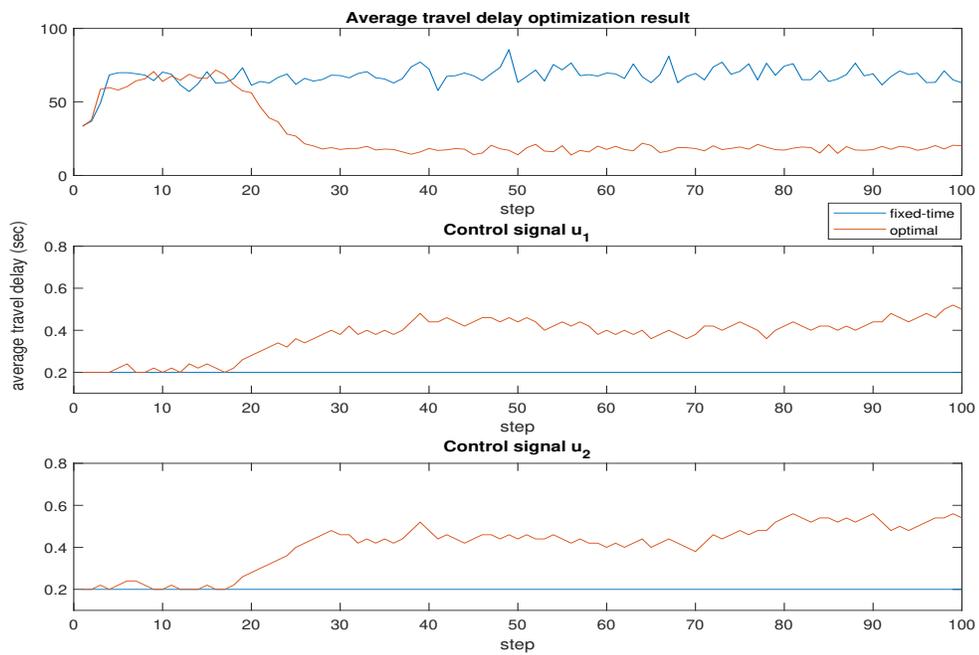


Figure 4.8: Optimization result with multiple Neural network.

## Chapter 5

# Stochastic Traffic Signal Control Design for Traffic Networks

Using a dynamic traffic network model as the control plant is a popular approach to solve traffic signal control problems. The traffic dynamic model is easy to obtain and the control design for the dynamic traffic network model is usually a direct solution to reduce vehicle travel time. Despite using a dynamic model to represent the traffic network, we come up with a statistic approach, which aimed at managing the traffic flow distribution to achieve smooth traffic. Since it is well known that the traffic flow in the network is randomly distributed in both space and time, and the randomness with traffic flow could be a challenge to traffic control with the dynamic traffic network model. Thus, one natural approach is to use a static model to represent the traffic network and solve the traffic control problem with stochastic control, which is a novel approach in the field of transportation research.

Unlike optimal traffic signal control using a dynamic traffic network model whose objective is to minimize traffic delay directly, the objective for stochastic traffic control is to achieve "smooth" traffic. The definition of smooth traffic in terms of stochastic control is to have every traffic intersection achieve balanced and low wait-time. Figure 5.1 gives an illustration of the unbalanced intersection, intersection with high traffic congestion, and

balanced low wait-time intersection.

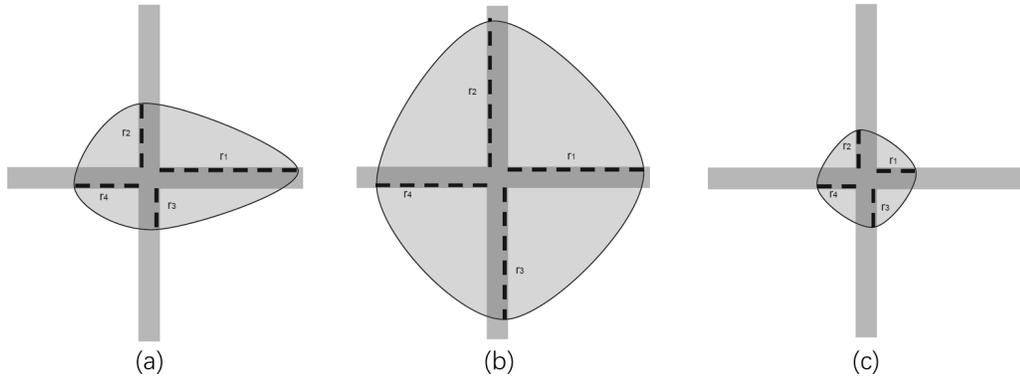


Figure 5.1: (a) unbalanced intersection; (b) highly congested intersection; (c) balanced and low wait-time intersection.

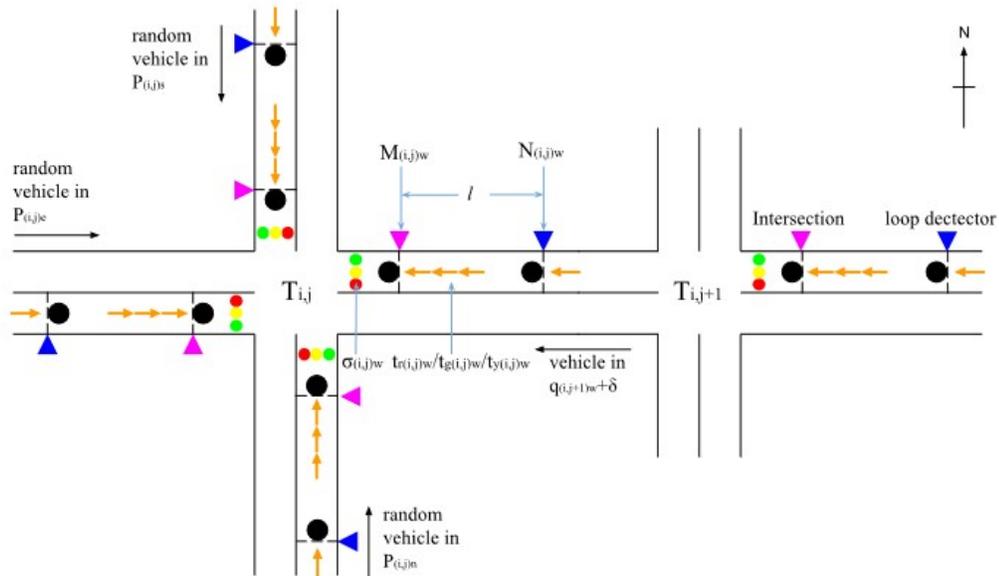


Figure 5.2: Single intersection example.

As a result, this study introduces a traffic signal control algorithm for traffic networks using stochastic control. We model the traffic network with a static model, where the relationship between the system output vehicle travel delay and the control input green signal period is represented by a probability density function (PDF). The objective is to have the travel time PDF track a given target PDF. With this objective achieved, we can

have the traffic flow follow a certain distribution, which can lead us closer to the goal of "smooth traffic". The main contributions of this study are given as follows:

1. Build a static traffic network model using B-spline functions to approximate the PDF and formulate such a model as the controlled plant;
2. Design a stochastic control scheme for a four intersection traffic network to move the actual travel time PDF to match a given PDF as much as possible;
3. Verify the control design with a simulation study.

## 5.1 Problem Statement

Before establishing the control design, the traffic control problem is stated in this section. The traffic delay time, which is used to represent the traffic flow, is first modeled based on the traffic queue modeling. Then the traffic system model is derived. Finally, the control objectives for stochastic traffic signal control is given.

### 5.1.1 Traffic Flow Modeling

For a single intersection shown in Figure 5.2, it is a combination of four one-way single-lane corridors with inter-dependency in each direction. A traffic delay model is used to represent the traffic flow in this study, which is directly equivalent to the traffic queue model. For each intersection, the total queue length  $q(k)$  is used to calculate the total travel delay, where the total queue length  $q(k)$  is the summation of queue length for all the directions in this intersection at time step  $k$ . To model the traffic queue, first, the traffic signal definition is introduced.

**Traffic control signal.** In this study, we ignore the left turn and right turn situation. The signal control for one intersection only contains one degree of freedom, denote as  $T_{green_{EW}}$  or  $T_{g_{EW}}$  in brief, which is the green cycle period of E-W direction. The traffic

control signal at node  $N_{i,j}$  is denoted as:

$$\sigma(k) = [\sigma_{EW}(k), \sigma_{NS}(k)] \quad (5.1)$$

$$\sigma_{EW}(k) = \begin{cases} \text{green} & kT \bmod T_{cyc} \in [0, T_{g_{EW}}] \\ \text{yellow} & kT \bmod T_{cyc} \in [T_{g_{EW}}, T_{g_{EW}} + T_y] \\ \text{red} & kT \bmod T_{cyc} \in [T_{cyc} - T_{r_{EW}}, T_{cyc}] \end{cases} \quad (5.2)$$

$$\sigma_{NS}(k) = \begin{cases} \text{red} & kT \bmod T_{cyc} \in [0, T_{r_{NS}}] \\ \text{green} & kT \bmod T_{cyc} \in [T_{r_{NS}}, T_{cyc} - T_y] \\ \text{yellow} & kT \bmod T_{cyc} \in [T_{cyc} - T_y, T_{cyc}] \end{cases} \quad (5.3)$$

where  $\sigma(k)$  is the traffic signal of both E-W and N-S direction, which decides the current signal light.  $T_{cyc}$  is the signal cycle length. In this work, we assume the signal cycle length is fixed for simplicity.  $T_{r_{EW}}$ ,  $T_{r_{NS}}$ ,  $T_{g_{EW}}$ ,  $T_{g_{NS}}$  are the red signal length and green signal length of the E-W and N-S directions respectively.  $T_y$  is the fixed yellow signal period. Figure 5.3 shows the traffic control signal at one intersection. Since at one intersection, the signal length on E-W direction and N-S direction depend on each other, there only exists one degree of freedom for the green cycle length. Assuming that the green signal length of E-W direction  $T_{green_{EW}}$  is an independent variable, we can derive the following relationship:

$$T_{g_{NS}} = T_{cyc} - T_{g_{EW}} - T_y \quad (5.4)$$

For a single direction, the red signal period can be calculated from

$$T_r = T_{cyc} - T_g - T_y. \quad (5.5)$$

**Traffic queue modeling.** To model the traffic queue for one intersection, we can first consider a single direction queue model [90]. We denote the vehicle detection counters for loop detectors  $N$  and  $M$  shown in Figure 5.2 by  $n$  and  $m$  respectively.  $n(k)$  and  $m(k)$

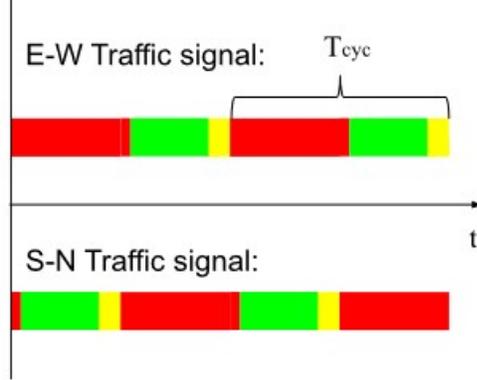


Figure 5.3: Traffic control signal.

indicates the  $n(k)$ th and  $m(k)$ th vehicle detected at loop detector N and M at time step  $k$  during the current signal light. The average vehicle length is denoted by  $l_v$ . The traffic queue model of an one-way single-lane corridor is given by:

$$q_i(k) = \begin{cases} r_i(k) & \sigma_i(k) = \text{red} \\ g_i(k) & \sigma_i(k) = \text{green} \\ y_i(k) & \sigma_i(k) = \text{yellow} \end{cases} \quad (5.6)$$

$$r_i(k) = \begin{cases} y_i(k_y) + n_i(k)l_v & \sigma_i(k) = \text{red} \\ r_i(k_r) & \sigma_i(k) \neq \text{red} \end{cases} \quad (5.7)$$

$$g_i(k) = \begin{cases} r_i(k_r) + (n_i(k) - m_i(k))l_v & \sigma_i(k) = \text{green} \\ g_i(k_g) & \sigma_i(k) \neq \text{green} \end{cases} \quad (5.8)$$

$$y_i(k) = \begin{cases} g_i(k_g) + (n_i(k) - m_i(k))l_v & \sigma_i(k) = \text{yellow} \\ g_i(k_g) & \sigma_i(k) \neq \text{yellow} \end{cases} \quad (5.9)$$

$$q(k) = \sum_{j=1}^{n_d} q_j(k) \quad (5.10)$$

where  $q_i(k)$ ,  $i = 1, 2, \dots, n_d$  is the queue length of  $i$ th direction at the intersection,  $k_r$ ,  $k_g$  and  $k_y$  are the last time step of the previous red, green and yellow signal, respectively. The

expression of red signal queue length  $r_i(k)$  indicates that the red queue length increases based on the vehicle entering the intersection  $n(k)$  when the signal  $\sigma_i = \text{'red'}$ , and it will remain the value  $r_i(k_r)$  (which is the red queue length at the time step before signal changes to green) for the coming green and yellow signal until the next red signal comes. Similar explanations hold for green signal queue  $g_i(k)$  and yellow signal queue  $y_i(k)$ . The total queue length at the intersection is represented by  $q(k)$ , which is the sum of the queue length at all directions in the intersection. In general,  $n_d = 4$  indicating the E-W-N-S four directions of the intersection. Figure 5.4 shows the simulation results for the traffic queue in different directions using the above traffic queue model. Figure. 5.9(b) is the queue traffic queue simulation of a two intersection network shown in Figure. 5.2, the red curves are the traffic queue length during the red signal, the green curves are the traffic queue length during the green signal, the yellow curves are the traffic queue length during the yellow signal. Figure. 5.9(c) shows the histogram and kernel density estimation (KDE) of the average vehicle travel time during the red signal period, green signal period and yellow signal period. The three rows represent the result for intersection  $T_{i,j}$  S direction, intersection  $T_{i,j}$  W direction and intersection  $T_{i,j+1}$  W direction.

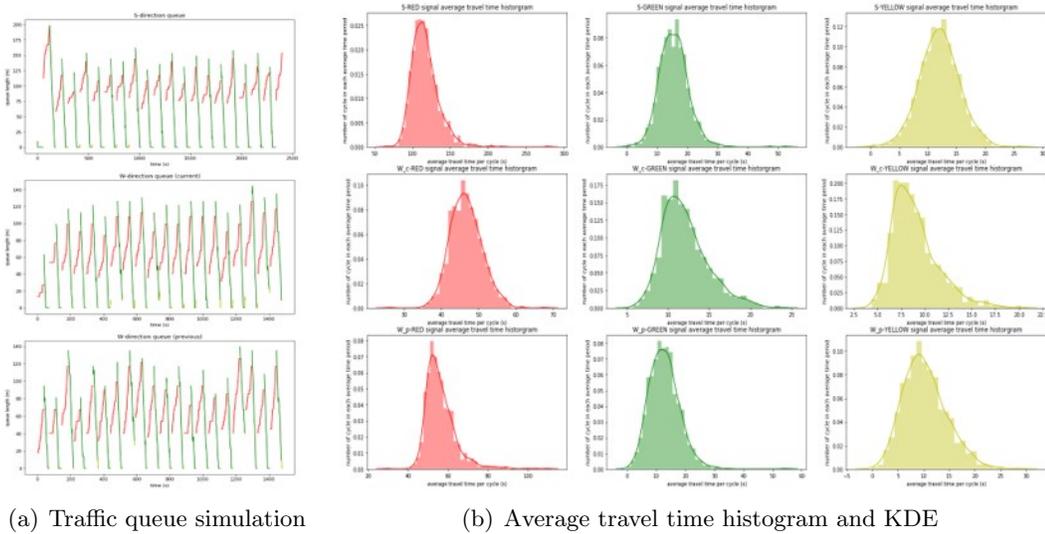


Figure 5.4: Traffic queue simulation results

**Traffic delay modeling.** To achieve the objective of reducing traffic delay, it is a nat-

ural approach to model the traffic delay model as well. With traffic queue model obtained, one can calculate the total traffic delay with the simplified equation

$$D(k) = \sum_{i=1}^{q(k)/l_v} (i * l_v / v_{delay} + t_{i_{wait}}) \quad (5.11)$$

where  $v_{delay}$  is the average delayed vehicle speed exiting the intersection,  $t_{i_{wait}}$  is the wait time till the green signal for the  $i$ th vehicle. The above equation indicates the total traffic delay is the summation of all the speed delay caused by the queue plus the total wait time till the green signal, which can be obtained from the previous signal cycle.

### 5.1.2 Traffic Network Modeling

This study is focused on the traffic networks in an urban area. Figure 5.5 gives an example of a simple traffic network. The traffic node  $N_{i,j}$  can be represented with a traffic system  $G_{i,j}$ . The traffic flow at each intersection is represented by  $y_{i,j} = \bar{D}_{i,j}$ , which is the average travel delay of all vehicles from all directions on  $N_{i,j}$  node. For a single node

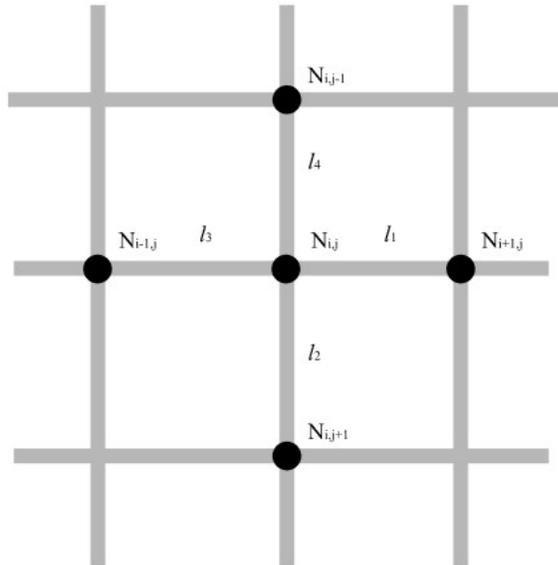


Figure 5.5: Traffic network example.

in the above traffic network, if the influence of neighboring intersections are neglected, the

discrete-time input-output model is:

$$y_{i,j}(k) = H_{i,j}(z)[u_{i,j}(k)] \quad (5.12)$$

where we have the notation  $\mathcal{Z}^{-1}(H_{i,j}(z)U_{i,j}(z)) \triangleq H_{i,j}(z)[u_{i,j}(k)]$ ,  $H_{i,j}(z)$  is the transfer function between the control input  $U_{i,j}(z)$  and the system output  $Y_{i,j}(z)$  in z-transform. To deal with the effect of the neighboring intersections as shown in Figure 5.5, one can build a discrete-time input-output model as:

$$y_{i,j}(k) = G_{i,j}(z)[u(k)] \quad (5.13)$$

$$G_{i,j}(z) = [H_{i,j}(z), z^{-\lambda_1} H_{i+1,j}(z), z^{-\lambda_2} H_{i,j+1}(z), \\ z^{-\lambda_3} H_{i-1,j}(z), z^{-\lambda_4} H_{i,j-1}(z)] \quad (5.14)$$

$$u_{i,j}(k) = [\sigma_{i,j}(k), \sigma_{i+1,j}(k), \sigma_{i,j+1}(k), \sigma_{i-1,j}(k), \sigma_{i,j-1}(k)]^T \quad (5.15)$$

$$H_{i,j}(z) = \frac{t_{ss_{i,j}}/\tau_{i,j}}{1 - e^{T/\tau_{i,j}} z^{-1}} \quad (5.16)$$

where  $\lambda_m = \lfloor l_m/\bar{v}T \rfloor$ ,  $m = 1, 2, 3, 4$  are the delay step between the node with its neighbors, calculated by the average time to travel between two nodes, is the traffic control signal at  $N_{i,j}$ ,  $T$  is the sampling time.  $\sigma_{i,j}(k)$  is the traffic signal at the node  $N_{i,j}$ ,  $t_{ss_{i,j}}$  is the steady state time delay, and  $\tau_{i,j}$  is the time constant, and these two parameter indicate the transient response of the Node  $N_{i,j}$ .

**Four-Node Traffic System.** In this study, a traffic network with four intersections will be considered as the testbed. The traffic system structure is shown in Figure 5.6. the input-output model is given by:

$$y(k) = G(z)[u(k)] \quad (5.17)$$

$$y(k) = [y_{1,1}^T(k), y_{1,2}^T(k), y_{2,1}^T(k), y_{2,2}^T(k)]^T \quad (5.18)$$

$$u(k) = [u_{1,1}^T(k), u_{1,2}^T(k), u_{2,1}^T(k), u_{2,2}^T(k)]^T \quad (5.19)$$

$$G(z) = \begin{bmatrix} H_{1,1}(z) & z^{-\lambda_1} H_{1,2}(z) & z^{-\lambda_2} H_{2,1}(z) & 0 \\ z^{-\lambda_1} H_{1,1}(z) & H_{1,2}(z) & 0 & z^{-\lambda_3} H_{2,2}(z) \\ z^{-\lambda_2} H_{1,1}(z) & 0 & H_{2,1}(z) & z^{-\lambda_4} H_{2,2}(z) \\ 0 & z^{-\lambda_3} H_{1,2}(z) & z^{-\lambda_4} H_{2,1}(z) & H_{2,2}(z) \end{bmatrix} \quad (5.20)$$

where  $\lambda_m$  is the average travel time step between different nodes associates with the distance between intersections  $l_m$ ,  $m = 1, 2, 3, 4$ .

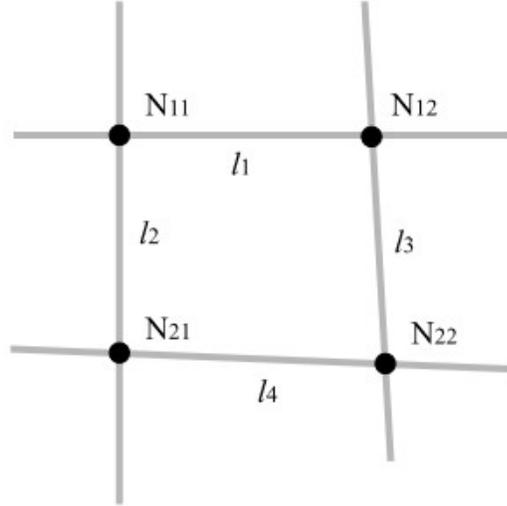


Figure 5.6: Traffic network with four intersections.

### 5.1.3 Static Model

**Probability Density Function.** To find the static model for the traffic network system, we assume that  $v(k)$  represent the output of a stochastic system.  $u(k)$  is the control input vector which controls the distribution of  $v(k)$ . At any time step  $k$ ,  $v(t)$  can be characterized by its cumulative distribution function defined as

$$F(y, u(k)) = P(v(k) < y, u(k)) \quad (5.21)$$

where  $P(v(k) < y, u(k))$  stands for the probability of variable  $v(k)$  less than the traffic network output  $y$  when the control signal  $u(k)$  is applied to the stochastic system. The

probability density function  $\gamma(y, u(k)) = \gamma_k(y, u)$  can be obtained from

$$\gamma_k(y, u) = \frac{dF(y, u(k))}{y} \quad (5.22)$$

where the control signal  $u(k)$  is a set of measurable inputs which controls the shape of the probability density function  $\gamma_k(y, u)$ .

**Static Model for Traffic Network System.** For the traffic network model in (5.17), there exist a static relationship between the system input  $u$  and the system output  $y$ . This static relationship can be represented with a probability density function of the output traffic delay  $\gamma_k(y, u)$ , which is

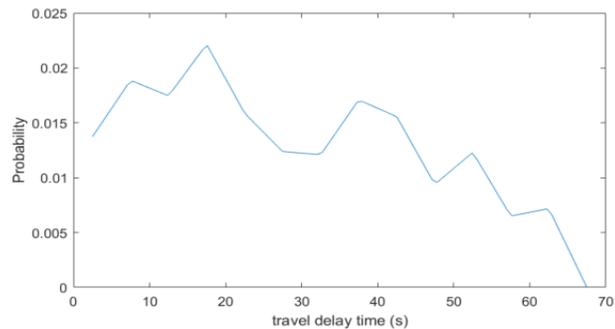
$$\gamma_k(y, u) = [\gamma_{k11}(y_{11}, u), \gamma_{k12}(y_{12}, u), \gamma_{k21}(y_{21}, u), \gamma_{k22}(y_{22}, u)]^T \quad (5.23)$$

where each row of (5.17) is considered as a multi-input single-output stochastic system. Since the PDF between traffic delay  $y(k)$  and the signal control  $u(k)$ , B-spline is used to approximate the PDF and it will be introduced in Section 5.2.

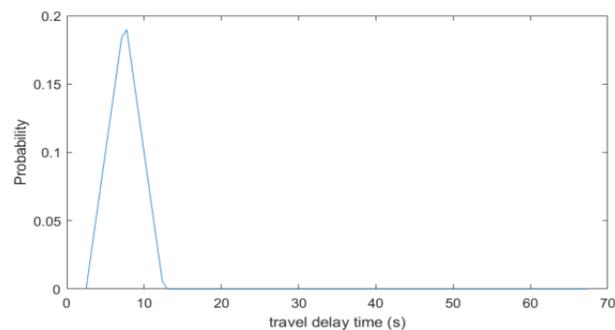
#### 5.1.4 Control Objective

The control objective is to design a control law for  $u(k)$  to have all the actual PDF  $\gamma_{i,j}(y, u)$  can approach to the target PDF  $\gamma_{i,j}^*(y)$ , shown in Figure 5.7, where Figure 5.7(a) is the actual PDF of the average vehicle travel delay of an intersection, and Figure 5.7(b) is the target PDF for average vehicle travel delay, which indicates that we want the average vehicle travel delay distribution to follow the target distribution with a mean of 60 second average travel delay.

*Remark 1.* Since the traffic flow contains a certain pattern and it contains high randomness and there will always be a red signal which effects the actual vehicle delay PDF, it is difficult to have the actual PDF totally matches the target PDF. What we can do is to have the actual PDF to approach the target PDF as much as possible.



(a) Actual PDF



(b) Target PDF

Figure 5.7: Actual PDF vs target PDF.

## 5.2 Static Traffic System Modeling

In this section, the static model of the traffic network is built. First, basic information about B-splines artificial neural networks is given. Then the static traffic network model is derived using B-splines approximation.

### 5.2.1 B-splines Artificial Neural Networks

B-spline artificial neural networks are examples of associative memory networks (AMNs) [91]. It is a commonly used approximation algorithm that can approximate unknown functions with a linear combination of the B-splines [92].

A  $m$ th order spline is a piecewise polynomial with degree  $n - 1$  in input variable  $y$ . The input axes are divided in to several intervals by the values called knots, denoted as  $\lambda_{-m+1}, \dots, \lambda_0, \lambda_1, \lambda_2, \dots, \lambda_r, \dots, \lambda_{r+m}$  in a non-decreasing order. The knot vector contains

$r$  interior knots and  $m$  exterior knots, satisfying

$$\lambda_0 < y_{min} < \lambda_1 \quad (5.24)$$

$$\lambda_r < y_{max} < \lambda_{r+1} \quad (5.25)$$

where  $y_{min}$  and  $y_{max}$  are the upper bound and lower bound of the input variable  $y$ . These knots are used in the generation of basis functions of width  $m$ , which are close to the ends of the input space. The modeling performance of a B-spline network depends on the orders of the univariate B-splines. The order of a univariate B-spline determines the shape of the B-spline and the width of its *support*. The *support* is defined as the region of the input space for which the output of the B-spline is non-zero. Thus a univariate B-spline of order  $m$  has a support which is also  $m$  intervals wide. Each input is mapped to  $m$  non-zero B-splines, and the output becomes smoother as the order increases. The univariate B-splines are defined based on the recursive relationship:

$$B_{1,j}(y) = \begin{cases} 1 & \text{if } \lambda_j < y < \lambda_{j+1} \\ 0 & \text{otherwise} \end{cases} \quad (5.26)$$

$$B_{m,j}(y) = \left( \frac{y - \lambda_{j-m}}{\lambda_{j-1} - \lambda_{j-m}} \right) B_{m-1,j-1}(y) + \left( \frac{\lambda_j - y}{\lambda_j - \lambda_{j-m+1}} \right) B_{m-1,j}(y) \quad (5.27)$$

It has been shown in [] that the B-splines neural networks can be used to approximate any continuous functions defined on a compact set, they will be used to represent the coupled links between the control input  $u(k)$  and the output probability density function (5.23).

### 5.2.2 Traffic Network Static Modeling

The probability density function (PDF) between the the system output traffic delay  $y(k)$  and the signal control  $u(k)$  it difficult to calculate directly. With the order of the B-spline  $m$  chosen, we can denote  $B_{m,j}$  as  $B_j$ . Since the PDF  $\gamma(y, u)$  is a continuous function with

respect to  $y \in [0, q_{max}]$  with any fixed control input  $u(k)$ , the following inequality holds

$$|\gamma(y, u) - \sum_{i=1}^n w_i B_i(y)| \leq \delta \quad (5.28)$$

where  $\delta$  is a prespecified arbitrarily small positive number,  $w_i$  are the weights,  $B_i(y)$  are the pre-specified the basic functions on the interval  $y \in [0, q_{max}]$ .

Since the basis functions are fixed, a different value of  $u$  will result in different sets of weights  $w_i$ . As a result, one can further approximate the traffic delay PDF at time step  $k$  with

$$\gamma_k(y, u) = \sum_{i=1}^n w_i(u) B_i(y) + e_0(y, u), \quad (5.29)$$

where  $e_0$  is the approximation error which satisfies  $|e_0| \leq \delta$ . Figure 5.8 shows an example of approximated travel delay PDF using B-spline approximation.

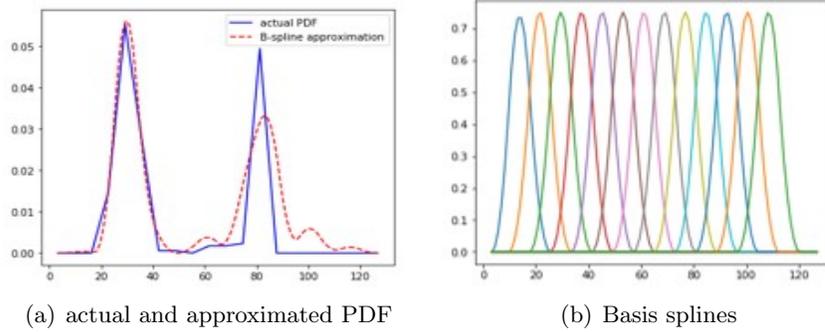


Figure 5.8: travel queue PDF using B-spline approximation

**Parametrization of the Static Model.** Since the static model  $\gamma(y, u)$  in (5.29) is a probability density function defined on  $[0, q_{max}]$ , it satisfies the flowing constraint:

$$\int_0^{q_{max}} \gamma(y, u) dy = \sum_{i=1}^n w_i(u) b_i = 1 \quad (5.30)$$

where

$$b_i = \int_0^{q_{max}} B_i(y) dy \quad (5.31)$$

are positive constants with selected basis functions. In most systems, there may exist a dynamic relationship between  $w_i$  and  $u$ . By solving the dynamic equation between  $w_i$  and  $u$ , one can reduce the number of independent weights in 5.29 to  $n - 1$  [16, 93]. The new stochastic system with  $n - 1$  degree of freedom is given by:

$$f(y, k) = C(y)V(k) + e(y, u(k)) \quad (5.32)$$

where

$$f(y, k) = \gamma(y, u(k)) - \frac{B_n(y)}{b_n} \quad (5.33)$$

$$b_i = \int_0^{q_{max}} B_i(y) dy \quad (5.34)$$

$$V(k) = [w_1(k), w_2(k), \dots, w_{n-1}(k)]^T \in R^{n-1} \quad (5.35)$$

$$C(y) = [B_1(y) - \frac{B_n(y)b_1}{b_n}, B_2(y) - \frac{B_n(y)b_2}{b_n}, \dots, B_{n-1}(y) - \frac{B_n(y)b_{n-1}}{b_n}] \in R^{1 \times (n-1)} \quad (5.36)$$

$$e(y, u(k)) = \frac{e_I(u(k))}{b_n} B_n(y) + e_0(y, u) \quad (5.37)$$

$$e_I(u(k)) = \int_0^{q_{max}} e_i(y, u(k)) dy \quad (5.38)$$

With the above derivation, one can separate the coupled  $y$  and  $u(k)$  in the PDF  $\gamma_k(y, u)$  and reduce the dimension of (5.29) to  $n - 1$ , which is needed for the stochastic control design.

### 5.3 Stochastic Control Design for Four Intersection Traffic Network

In this section, the stochastic control law is first derived. Then the simulation result is presented to verify the control design.

### 5.3.1 Stochastic Control Design

**Cost Function.** To achieve the control objective stated in Section 5.1.4, we choose the following performance function

$$J = \int_0^{q_{max}} (\gamma(y, u(k)) - \gamma^*(y))^2 dy \quad (5.39)$$

where  $\gamma^*(y)$  is the target PDF function, and  $u(k)$  is chosen to minimize  $J$ . From (5.34)-(5.38), one can obtain that

$$J = \int_0^{q_{max}} (C(y)V(u(k)) + L(y) - \gamma^*(y))^2 dy \quad (5.40)$$

where  $L(y) = \frac{B_n(y)}{b_n}$ , assuming

$$\Sigma = \int_0^{q_{max}} C(y)^T C(y) dy \in R^{(n-1) \times (n-1)} \quad (5.41)$$

$$\eta = \int_0^{q_{max}} (\gamma^*(y) - L(y)) C(y)^T dy \in R^{1 \times (n-1)} \quad (5.42)$$

$$\gamma_0 = \int_0^{q_{max}} (\gamma^*(y) - L(y))^2 dy \quad (5.43)$$

then, one can further obtain

$$J = V^T(u(k))\Sigma V(u(k)) - 2\eta V(u(k)) + \gamma_0 \quad (5.44)$$

**Control Law.** To minimize  $J$ ,  $u(k)$  is calculated from  $\frac{\partial J}{\partial u(k)} = 0$ . This leads to solving the following equation

$$(V^T(u(k))\Sigma - \eta) \frac{\partial V(u(k))}{\partial u(k)} = 0 \quad (5.45)$$

The above equation can be solved use the following gradient approach

$$u_k^{i+1} = u_k^i - 2\mu (V^T(u) \Sigma - \eta) \frac{\partial V(u)}{\partial u} \Big|_{u=u_k^i} \quad (5.46)$$

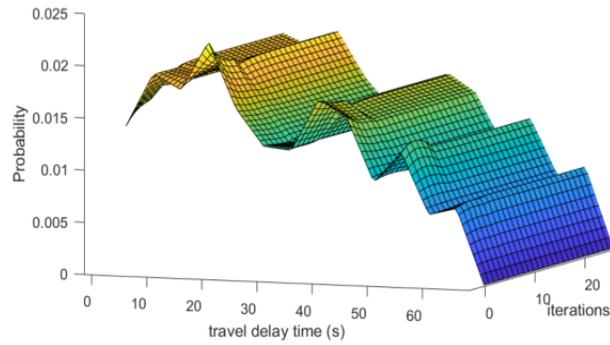
where  $\mu > 0$  is the pre-specified optimization step size, and  $i = 1, 2, \dots, K$ .

### 5.3.2 Simulation Results

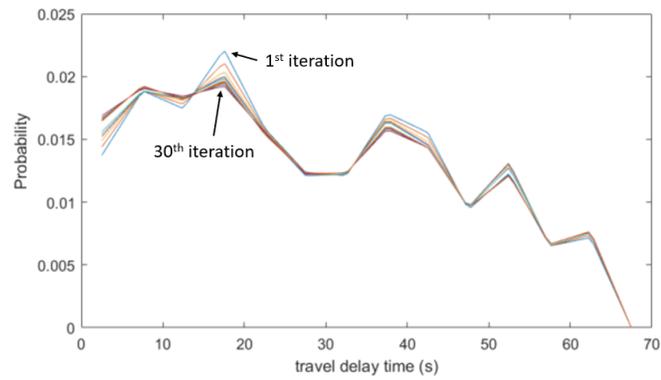
We tested the stochastic control design on a four intersection traffic network shown in Figure 5.6. Assume random number of vehicle entering the four two-direction road with averages of 500, 460, 200, 100 vehicles entering the road  $l_1$ ,  $l_2$ ,  $l_3$  and  $l_4$  per hour respectively, and the average speeds are 50, 50, 35, 35 miles per hour, respectively. The signal cycle length is chosen as 100 seconds. We choose number of B-spline  $n = 14$ , the iteration number  $K = 30$ , and optimization step size  $\mu = 1$ . The simulation results are shown in Figure 5.9. From the simulation results, one can see that with our control design, we can move the PDF closer to the target PDF.

## 5.4 Conclusion

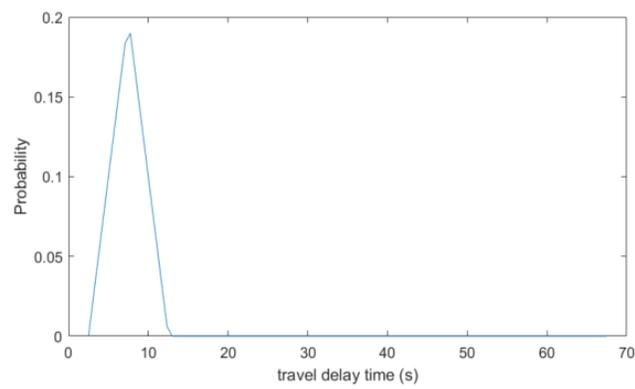
In this chapter, we first developed an input-output model for the traffic queue. Then we designed a stochastic control algorithm to move the actual traffic flow distribution towards a target distribution, and this result potentially indicates smooth traffic with minimized energy consumption. The significance of this work proposed a stochastic approach to solving the traffic congestion problem, which can effectively handle the randomness of traffic flow. The potential future work of this study is to quantify the energy-saving and implement the algorithm to more extensive traffic networks.



(a) Actual PDF variation in different iterations - 3D



(b) Actual PDF variation in different iterations



(c) Target PDF

Figure 5.9: Actual PDF vs target PDF.

## Chapter 6

# Conclusions and Future Work

In this chapter, we give concluding remarks to this dissertation and suggest some research topics for future study in this area.

**Conclusions.** Based on this dissertation, the use of adaptive learning and optimization schemes in intelligent transportation systems development can result in a better understanding of the unknown transportation systems and yet lead to energy consumption minimization with better identification of the system.

In Chapter 2, a co-optimization scheme is proposed to optimize fuel efficiency for HEVs. The proposed optimization scheme uses vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) information as the basis to optimally tune control parameters for the existing powertrain control system. Moreover, the speed of catalyst temperature to reach its light-off level in the exhaust emission system is also considered as an additional optimization constraint to reduce emission. It has been shown that a further 9.22% fuel savings can be achieved on average for a Toyota Prius test model.

In Chapter 3 to Chapter 5, three different approaches for traffic networks are developed. In Chapter 3, an adaptive linear-quadratic regulator (LQR) is designed to minimize both traffic delay and incremental changes in the control input, which is based on linear system approximation. In Chapter 4, a traffic signal optimal control scheme with an adaptive on-line learning scheme using multiple-model neural networks is designed to achieve traffic

delay minimization. In this work, a non-linear neural network model is used to represent the unknown traffic dynamics. In Chapter 5, the above problem is solved from another aspect, where the goal is to have the distributions of every intersection follows a target distribution. To this end, a stochastic traffic signal control algorithm is designed. The above three methods are not only practically meaningful for traffic signal control design but also have some theoretic contributions to the field of adaptive learning techniques.

The detailed research topics are summarized as follows:

- System identification and fuel optimization for HEV systems (Chapter 2).
- System identification and traffic signal control for large traffic networks in urban areas (Chapter 3).
- Network traffic signal control with adaptive on-line learning scheme using multiple-model neural networks (Chapter 4).
- Stochastic traffic signal control design for traffic networks (Chapter 5).

**Future Work.** The studies in this dissertation give some encouraging results in using adaptive learning and optimal control scheme in transportation systems. For future study, the following research topics are listed as some possible directions in this area:

- Use adaptive on-line learning strategy to identify the HEV system and design fuel consumption minimization scheme based on on-line estimation model;
- Design on-line learning strategy in stochastic traffic signal control design to achieve the identification of the static model on-line.

# Bibliography

- [1] W. Hong, I. Chakraborty, and H. Wang, “Parametric optimization problem formulation for connected hybrid electric vehicles using neural network based equivalent model,” Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), Tech. Rep., 2019.
- [2] W. Zhu, “A connected vehicle based coordinated adaptive navigation system,” Ph.D. dissertation, University of Washington, 2019.
- [3] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, “A license plate-recognition algorithm for intelligent transportation system applications,” *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp. 377–392, 2006.
- [4] T. C. Koopmans, “Optimum utilization of the transportation system,” *Econometrica: Journal of the Econometric Society*, pp. 136–146, 1949.
- [5] A. Altshuler, J. P. Womack, and J. R. Pucher, “The urban transportation system: Politics and policy innovation,” *Journal of Economic Literature*, vol. 18, no. 3, 1980.
- [6] M. Ehsani, Y. Gao, S. Longo, and K. Ebrahimi, *Modern electric, hybrid electric, and fuel cell vehicles*. CRC press, 2018.
- [7] B. Bilgin, P. Magne, P. Malysz, Y. Yang, V. Pantelic, M. Preindl, A. Korobkine, W. Jiang, M. Lawford, and A. Emadi, “Making the case for electrified transportation,” *IEEE Transactions on Transportation Electrification*, vol. 1, no. 1, pp. 4–17, 2015.

- [8] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, “Review of road traffic control strategies,” *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.
- [9] K. Triantis, S. Sarangi, D. Teodorović, and L. Razzolini, “Traffic congestion mitigation: combining engineering and economic perspectives,” *Transportation Planning and Technology*, vol. 34, no. 7, pp. 637–645, 2011.
- [10] L. B. De Oliveira and E. Camponogara, “Multi-agent model predictive control of signaling split in urban traffic networks,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 1, pp. 120–139, 2010.
- [11] X. Li and J.-Q. Sun, “Multi-objective optimal predictive control of signals in urban traffic network,” *Journal of Intelligent Transportation Systems*, vol. 23, no. 4, pp. 370–388, 2019.
- [12] A. Di Febbraro, D. Giglio, and N. Sacco, “A deterministic and stochastic petri net model for traffic-responsive signaling control in urban areas,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 510–524, 2015.
- [13] G. Tao, *Adaptive control design and analysis*. John Wiley & Sons, 2003, vol. 37.
- [14] R. Sargent, “Optimal control,” *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 361–371, 2000.
- [15] B. Kuo, “Digital control systems. 1992: Oxford university press.”
- [16] H. Wang, *Bounded dynamic stochastic systems: modelling and control*. Springer Science & Business Media, 2012.
- [17] C. K. Samanta, S. K. Padhy, S. P. Panigrahi, and B. K. Panigrahi, “Hybrid swarm intelligence methods for energy management in hybrid electric vehicles,” *IET Electrical Systems in Transportation*, vol. 3, no. 1, pp. 22–29, 2013.

- [18] A. Boyali, L. Güvenç *et al.*, “Real-time controller design for a parallel hybrid electric vehicle using neuro-dynamic programming method,” 2010.
- [19] Q. Gong, Y. Li, and Z.-R. Peng, “Trip-based optimal power management of plug-in hybrid electric vehicles,” *IEEE Transactions on vehicular technology*, vol. 57, no. 6, pp. 3393–3401, 2008.
- [20] B. Chen, S. A. Evangelou, and R. Lot, “Fuel efficiency optimization methodologies for series hybrid electric vehicles,” *2018 IEEE Vehicle Power and Propulsion Conference, VPPC 2018 - Proceedings*, vol. 1, pp. 1–6, 2019.
- [21] P. G. Anselma, A. Biswas, J. Roeleveld, G. Belingardi, and A. Emadi, “Multi-Fidelity Near-Optimal on-Line Control of a Parallel Hybrid Electric Vehicle Powertrain,” *ITEC 2019 - 2019 IEEE Transportation Electrification Conference and Expo*, pp. 2–7, 2019.
- [22] P. G. Anselma, Y. Huo, J. Roeleveld, G. Belingardi, and A. Emadi, “Integration of On-Line Control in Optimal Design of Multimode Power-Split Hybrid Electric Vehicle Powertrains,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3436–3445, 2019.
- [23] S. Uebel, N. Murgovski, C. Tempelhahn, and B. Baker, “Optimal Energy Management and Velocity Control of Hybrid Electric Vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 327–337, 2018.
- [24] F. Xu and T. Shen, “Look-Ahead Prediction-Based Real-Time Optimal Energy Management for Connected HEVs,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2537–2551, 2020.
- [25] X. Shen, J. Zhang, and T. Shen, “Real-time scenario-based stochastic optimal energy management strategy for HEVs,” *2016 European Control Conference, ECC 2016*, pp. 631–636, 2016.
- [26] T. J. Boehme, M. Schori, B. Frank, M. Schuitalbers, and B. Lampe, “Solution of a hybrid optimal control problem for parallel hybrid vehicles subject to thermal con-

- straints,” *Proceedings of the IEEE Conference on Decision and Control*, pp. 2220–2226, 2013.
- [27] C. Haupt, D. Bücherl, A. Engstle, H. G. Herzog, and G. Wachtmeister, “Energy management in hybrid vehicles considering thermal interactions,” *VPPC 2007 - Proceedings of the 2007 IEEE Vehicle Power and Propulsion Conference*, pp. 36–41, 2007.
- [28] E. D. Tate, J. W. Grizzle, and H. Peng, “SP-SDP for fuel consumption and tailpipe emissions minimization in an EVT hybrid,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 673–687, 2010.
- [29] A. Chasse, G. Corde, A. Del Mastro, and F. Perez, “Online optimal control of a parallel hybrid with after-treatment constraint integration,” *2010 IEEE Vehicle Power and Propulsion Conference, VPPC 2010*, pp. 1–6, 2010.
- [30] P. Michel, A. Charlet, G. Colin, Y. Chamailard, G. Bloch, and C. Nouillant, “Optimizing fuel consumption and pollutant emissions of gasoline-HEV with catalytic converter,” *Control Engineering Practice*, vol. 61, pp. 198–205, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.conengprac.2015.12.010>
- [31] M. R. Amini, Y. Feng, H. Wang, I. Kolmanovsky, and J. Sun, “Thermal Responses of Connected HEVs Engine and Aftertreatment Systems to Eco-Driving,” *CCTA 2019 - 3rd IEEE Conference on Control Technology and Applications*, pp. 724–729, 2019.
- [32] Y. Cheng, K. Chen, C. Chan, A. Bouscayrol, and S. Cui, “Global modeling and control strategy simulation for a hybrid electric vehicle using electrical variable transmission,” in *Vehicle Power and Propulsion Conference, 2008. VPPC’08. IEEE*. IEEE, 2008, pp. 1–5.
- [33] Y. Li, X. Lu, and N. C. Kar, “Rule-based control strategy with novel parameters optimization using nsga-ii for power-split phev operation cost minimization,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 3051–3061, 2014.

- [34] P. Shen, Z. Zhao, X. Zhan, and J. Li, "Particle swarm optimization of driving torque demand decision based on fuel economy for plug-in hybrid electric vehicle," *Energy*, vol. 123, pp. 89–107, 2017.
- [35] M. Montazeri-Gh and M. Mahmoodi-K, "Optimized predictive energy management of plug-in hybrid electric vehicle based on traffic condition," *Journal of cleaner production*, vol. 139, pp. 935–948, 2016.
- [36] D. Kum, H. Peng, and N. K. Bucknor, "Supervisory control of parallel hybrid electric vehicles for fuel and emission reduction," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 133, no. 6, pp. 1–10, 2011.
- [37] D. I. Robertson, "tansyt'method for area traffic control," *Traffic Engineering & Control*, vol. 8, no. 8, 1969.
- [38] P. Koonce and L. Rodegerdts, "Traffic signal timing manual." United States. Federal Highway Administration, Tech. Rep., 2008.
- [39] S. Mikami and Y. Kakazu, "Genetic reinforcement learning for cooperative traffic signal control," in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE, 1994, pp. 223–228.
- [40] L. Singh, S. Tripathi, and H. Arora, "Time optimization for traffic signal control using genetic algorithm," *International Journal of Recent Trends in Engineering*, vol. 2, no. 2, p. 4, 2009.
- [41] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
- [42] C. G. S. Cools and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in applied self-organizing systems*. Springer, 2013, pp. 45–55.
- [43] P. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intelligent Transport Systems*, vol. 4, no. 3, pp. 177–188, 2010.

- [44] Y. Bi, D. Srinivasan, X. Lu, Z. Sun, and W. Zeng, "Type-2 fuzzy multi-intersection traffic signal control with differential evolution optimization," *Expert systems with applications*, vol. 41, no. 16, pp. 7338–7349, 2014.
- [45] A. Nagare and S. Bhatia, "Traffic flow control using neural network," *Traffic*, vol. 1, no. 2, pp. 50–52, 2012.
- [46] H. Yin, S. Wong, J. Xu, and C. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 2, pp. 85–98, 2002.
- [47] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.
- [48] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.
- [49] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
- [50] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2496–2505.
- [51] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.
- [52] X. Wang, L. Ke, Z. Qiao, and X. Chai, "Large-scale traffic signal control using a novel multi-agent reinforcement learning," *arXiv preprint arXiv:1908.03761*, 2019.

- [53] X. Li and J.-Q. Sun, "Signal multiobjective optimization for urban traffic network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3529–3537, 2018.
- [54] X. Li and J. Sun, "Turning-lane and signal optimization at intersections with multiple objectives," *Engineering Optimization*, vol. 51, no. 3, pp. 484–502, 2019.
- [55] A. Stevanovic, *Adaptive traffic control systems: domestic and foreign state of practice*, 2010, no. Project 20-5 (Topic 40-03).
- [56] A. G. Sims, "The sydney coordinated adaptive traffic system," in *Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems, 1979, Pacific Grove, California, USA, 1979*.
- [57] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, "The scoot on-line traffic signal optimisation technique," *Traffic Engineering & Control*, vol. 23, no. 4, 1982.
- [58] N. H. Gartner, *OPAC: A demand-responsive strategy for traffic signal control*, 1983, no. 906.
- [59] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.
- [60] W. Kraus, F. A. de Souza, R. C. Carlson, M. Papageorgiou, L. D. Dantas, E. Camponogara, E. Kosmatopoulos, and K. Aboudolas, "Cost effective real-time traffic signal control using the TUC strategy," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 6–17, 2010.
- [61] H. Kwakernaak and R. Sivan, *Linear optimal control systems*. Wiley-interscience New York, 1972, vol. 1.
- [62] J. Liu, H. Peng, and Z. Filipi, "Modeling and analysis of the toyota hybrid system," *TIC*, vol. 200, p. 3, 2005.

- [63] J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *IEEE transactions on control systems technology*, vol. 16, no. 6, pp. 1242–1251, 2008.
- [64] G. J. McRae, J. W. Tilden, and J. H. Seinfeld, "Global sensitivity analysis—a computational implementation of the fourier amplitude sensitivity test (fast)," *Computers & Chemical Engineering*, vol. 6, no. 1, pp. 15–25, 1982.
- [65] B. Jiang and Y. Fei, "Vehicle Speed Prediction by Two-Level Data Driven Models in Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1793–1801, 2017.
- [66] J. Jing, A. Kurt, E. Ozatay, J. Michelini, D. Filev, and U. Ozguner, "Vehicle Speed Prediction in a Convoy Using V2V Communication," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2015-October, pp. 2861–2868, 2015.
- [67] W. Hong, I. Chakraborty, and H. Wang, "Parameter co-optimization for hybrid electric vehicles powertrain system leveraging v2v/v2i information," in *2019 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE, 2019, pp. 142–147.
- [68] S. Haykin, "A comprehensive foundation," *Neural networks*, vol. 2, no. 2004, p. 41, 2004.
- [69] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.
- [70] W. Hu, H. Wang, B. Du, and L. Yan, "A multi-intersection model and signal timing plan algorithm for urban traffic signal control," *Transport*, vol. 32, no. 4, pp. 368–378, 2017.
- [71] J. A. Laval and H. Zhou, "Large-scale traffic signal control using machine learning: some traffic flow considerations," *arXiv preprint arXiv:1908.02673*, 2019.

- [72] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.
- [73] *Vissim 9 User Manual*, PTV Group, Karlsruhe, Germany, 2016.
- [74] R. Wiedemann, *Simulation des Straßenverkehrsflusses. Schriftenreihe Heft 8*. Instituts für Verkehrswesen der Universität Karlsruhe, 1974.
- [75] M. Zhu, X. Wang, A. Tarko *et al.*, “Modeling car-following behavior on urban expressways in shanghai: A naturalistic driving study,” *Transportation research part C: emerging technologies*, vol. 93, pp. 425–445, 2018.
- [76] F. Webster, “Traffic signal settings. road research technical paper,(39), 1958,” *Great Britain Road Research Laboratory, London*.
- [77] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*. Prentice Hall, 1971.
- [78] S. Tu and B. Recht, “Least-squares temporal difference learning for the linear quadratic regulator,” *arXiv preprint arXiv:1712.08642*, 2017.
- [79] —, “The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint,” *arXiv preprint arXiv:1812.03565*, 2018.
- [80] S. T. H. Mania and B. Recht, “Certainty equivalent control of lqr is efficient,” *arXiv preprint arXiv:1902.07826*, 2019.
- [81] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Courier Corporation, 2014.
- [82] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” in *29th IEEE Conference on Decision and Control*. IEEE, 1990, pp. 2130–2132.
- [83] W. Genders and S. Razavi, “An open-source framework for adaptive traffic signal control,” *arXiv preprint arXiv:1909.00395*, 2019.

- [84] C. Gershenson, “Self-organizing traffic lights,” *arXiv preprint nlin/0411066*, 2004.
- [85] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.
- [86] H. Wang, C. Wang, M. Zhu, and W. Hong, “Globalized modeling and signal timing control for large-scale networked intersections,” in *Proceedings of The 2nd ACM/EIGSCC Symposium On Smart Cities and Communications (SCC 2019)*. Portland, OR, USA: ACM, 2019.
- [87] Z. H. H. Zhang, L. Zhang and M. Li, “LQR-based perimeter control approach for urban road traffic networks,” in *2017 6th Data Driven Control and Learning Systems (DDCLS)*. IEEE, 2017, pp. 745–749.
- [88] O. Corrigan, “An investigation into machine learning solutions involving time series across different problem domains,” Ph.D. dissertation, Dublin City University, 2018.
- [89] J. Sarangapani, *Neural network control of nonlinear discrete-time systems*. CRC press, 2018.
- [90] H. Wang, H. Aziz, S. E. Young, and S. Patil, “Control of networked traffic flow distribution: a stochastic distribution system perspective,” in *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*. ACM, 2017, p. 37.
- [91] M. Brown and C. J. Harris, *Neurofuzzy adaptive modelling and control*. Prentice Hall, 1994.
- [92] F. Girosi and T. Poggio, “Networks and the best approximation property,” *Biological cybernetics*, vol. 63, no. 3, pp. 169–176, 1990.
- [93] H. Wang, “Robust control of the output probability density functions for multivariable stochastic systems with guaranteed stability,” *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2103–2107, 1999.