INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
CONTROL, OPTIMIZATION AND SIMULATION OF INTELLIGENT TRANSPORTATION SYSTEMS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Jia Lei, B.S.E.E., M.S.

* * * * *

The Ohio State University

2001

Dissertation Committee:
Professor Ümit Özgüner, Adviser
Professor Vadim I. Utkin
Professor Benjamin Coifman

Approved by

Adviser
Department of Electrical Engineering
ABSTRACT

This dissertation explores issues in the control and optimization of Intelligent Transportation Systems, with particular emphasis on the design of dynamic routing and traffic signal control strategies in local and freeway traffic networks.

To formulate the control and optimization problems in traffic networks, representations of static and dynamic information of traffic networks are provided. We use a set of topology matrices to represent the traffic static information and give two dynamic models, a point-queuing model and a hydrodynamic model, to represent the characteristics of traffic in local and freeway networks.

Based on the point-queuing model, we first analyze the characteristics of dynamic routing and traffic signal control in local traffic networks. First, a decentralized multi-destination dynamic routing strategy is proposed, in which the values of the cost-to-go of every link to different destinations in a traffic network are calculated in real time to direct traffic. Then, a decentralized hybrid intersection controller with five states is designed to address the different combinations of traffic situations at an intersection, the downstream and upstream intersections in a network and the problem that queues may overrun an upstream intersection.

Based on the hydrodynamic model, we analyze the dynamic routing and signal control in freeway traffic networks. Using the sliding mode design concept, we
propose a dynamic two-freeway routing method and an on-ramp controller. This on-ramp controller can iteratively search for the optimal operating point under the given conditions and at the same time meter inflow to achieve this desired state.
To my husband, Zhang Yan
ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Ümit Özgüner, for his advise, guidance, encouragement, and support throughout my Doctoral program of study. It has always been educational, stimulating, and fun. I would also like to thank Dr. Vadim Utkin, Dr. Benjamin Coifman, and Dr. Ashok Krishnamurthy for serving on my candidacy examination and dissertation committees.

I also wish to thank Keith Redmill, Tankut Acarman and other control students for the friendship we shared and the assistance they gave me in many ways.

Finally, I sincerely appreciate my family for their endless support at all stages of my life.
VITA

September, 1972 ................................................. Born - Luzhou, Sichuan P.R.China

1994 ............................................................... B.S.E.E. Electrical Engineering,
Zhejiang University

1997 ............................................................... M.S. Electrical Engineering,
The Ohio State University

1997-1998 ............................................................... University Fellowship,
The Ohio State University

1998-1999 ............................................................... Graduate Research Associate,
The Ohio State University

1999-2000 ............................................................... ITS Fellowship

PUBLICATIONS


Jia Lei and Ümit Özgüner, "Integration of dynamic routing and intersection control in Intelligent Transportation System," *Proceedings of IEEE Conference on Intelligent Transportation Systems*, 2000, p.137-142


Jia Lei and Ümit Özgüner, "Highway on-ramp and speed signaling control using sliding modes," *Submitted to IEEE Transactions on Intelligent Transportation Systems*

Jia Lei, Keith Redmill and Ümit Özgüner, "VATSIM: a simulator for vehicles and traffic," *Submitted to ITSC 2001*

Jia Lei and Ümit Özgüner, "Decentralized hybrid intersection control," *Submitted to the 40th IEEE Conference on Decision and Control, 2001*

Jia Lei and Ümit Özgüner, "Control problems in transportation networks," *Submitted to the 40th IEEE Conference on Decision and Control, 2001*

Jia Lei, Keith Redmill and Ümit Özgüner, "A vehicle and traffic simulator: VATSIM," *Submitted to IEEE Transactions on Intelligent Transportation Systems*

**FIELDS OF STUDY**

Major Field: Electrical Engineering

Studies in:
- Control Systems
- Communications and Signal Processing
- Computational Mathematics
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiv</td>
</tr>
<tr>
<td>Chapters:</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Control and Optimization Problems in Traffic Networks</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Literature Survey</td>
<td>5</td>
</tr>
<tr>
<td>1.2.1 Routing and Signal Control in Local Traffic Networks</td>
<td>5</td>
</tr>
<tr>
<td>1.2.2 Routing and Signal Control in Freeway Networks</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Outline of Contents</td>
<td>11</td>
</tr>
<tr>
<td>2. Notation and Traffic Network Modeling</td>
<td>14</td>
</tr>
<tr>
<td>2.1 Overview</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Notation</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Static Information</td>
<td>16</td>
</tr>
<tr>
<td>2.3.1 Representation</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2 Examples</td>
<td>19</td>
</tr>
<tr>
<td>2.4 Dynamic Information</td>
<td>21</td>
</tr>
<tr>
<td>2.4.1 Point-queuing Model</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>viii</td>
</tr>
</tbody>
</table>
### Contents

#### 2.4.2 Hydrodynamic Model

2.5 Summary

#### 3. Decentralized Multi-destination Dynamic Routing Strategy for Congested Local Traffic Networks

3.1 Overview

3.2 Problem Statement

3.3 Routing for a Simple Network with Single-destination

3.4 Extension to Large Networks with Single-Destination

3.5 Extension to Networks with Multi-destination

3.6 Extension to Networks with Traffic Lights

3.7 Simulation Studies

3.7.1 A Single Destination Case

3.7.2 A Multi-Destination Case

3.7.3 The Ohio State University Traffic Simulation

3.8 Summary

#### 4. Decentralized Hybrid Intersection Control

4.1 Model of an Isolated Intersection

4.2 Control for an Isolated Intersection

4.2.1 Saturated Intersection

4.2.2 Unsaturated Intersection

4.2.3 Hybrid Control

4.3 Multi-intersection Control

4.3.1 Saturated Intersection Control

4.3.2 Unsaturated Two-intersection Control

4.3.3 Hybrid Multi-intersection Control

4.4 Simulation Example

4.4.1 Five Intersections

4.4.2 Four Intersections

4.4.3 Six Intersections

4.4.4 Nine Intersections

4.5 Conclusion

#### 5. Two-freeway Routing Using Sliding Modes

5.1 Problem Statement

5.2 Brief Introduction of Sliding Mode Theory

5.3 The User Equilibrium Dynamic Two-freeway Routing Using Sliding Modes
5.4 System Optimal Dynamic Two-freeway Routing .................................. 95
5.5 Simulation Example ................................................................. 96
5.6 Summary .................................................................................. 96

6. Freeway On-ramp Control Using Sliding Modes .............................. 99
   6.1 Overview ................................................................................ 99
   6.2 On-ramp Control Problem Formulation ................................ 100
   6.3 Sliding Mode On-Ramp Control .............................................. 102
   6.4 Simulation Studies ................................................................. 106
   6.5 Summary ................................................................................ 110

7. Applications in VATSIM .............................................................. 111
   7.1 Overview of VATSIM ............................................................... 111
   7.2 Applications in VATSIM ......................................................... 113
      7.2.1 Intersection Control ......................................................... 113
      7.2.2 Dynamic Routing ............................................................. 115
   7.3 Summary ................................................................................ 116

8. Conclusion and Future Directions .................................................. 117
   8.1 Contribution ........................................................................... 117
   8.2 Future Research Directions ..................................................... 119

Appendices:

A. VATSIM: A Simulator for Vehicles and Traffic ................................ 121
   A.1 Overview ............................................................................... 121
   A.2 History of VATSIM ............................................................... 122
   A.3 Simulation Framework ........................................................... 123
   A.4 Network ................................................................................ 125
      A.4.1 Segments .......................................................................... 125
      A.4.2 Nodes .............................................................................. 127
      A.4.3 Road Map Database ......................................................... 127
   A.5 Vehicles ................................................................................ 128
      A.5.1 Model .............................................................................. 129
      A.5.2 Sensors ........................................................................... 130
      A.5.3 Controllers ..................................................................... 132
      A.5.4 Vehicle Generation Database ........................................... 134
      A.5.5 Vehicle Source and Sink Database .................................... 136
   A.6 Transit Vehicles ..................................................................... 136
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Comparison of the simulation results of morning rush hour with routing and without routing</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of the simulation results of afternoon rush hour with routing and without routing</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of two intersection control methods in five intersection simulation when $T = 30s$</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of two intersection control methods in five intersection simulation when $T = 30s$ and $\lambda_{x1} = 0.75 \text{ veh/s}$</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of two intersection control methods in five intersection simulation when $T = 40s$</td>
</tr>
<tr>
<td>4.4</td>
<td>Comparison of two intersection control methods in four intersection simulation</td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of two intersection control methods in six intersection simulation</td>
</tr>
<tr>
<td>4.6</td>
<td>Comparison of two intersection control methods in nine intersection simulation</td>
</tr>
<tr>
<td>A.1</td>
<td>Road Map Input Database Specification</td>
</tr>
<tr>
<td>A.2</td>
<td>Vehicle Generation Input Database Specification</td>
</tr>
<tr>
<td>A.3</td>
<td>Source and Sink Database Specification</td>
</tr>
</tbody>
</table>

xii
A.4 Bus Path Database Specification ......................................................... 137
A.5 Bus Stop Database Specification ...................................................... 138
A.6 Traffic Light Database Specification .............................................. 140
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Illustration of ITS</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Example network 1</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Example network 2</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>A simple link</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>A direction of a node</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>A node</td>
<td>24</td>
</tr>
<tr>
<td>2.6</td>
<td>A road</td>
<td>25</td>
</tr>
<tr>
<td>2.7</td>
<td>Flow-density characteristics curve</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>Network with $L$ parallel links</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>A single destination network</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Simulation results of the single destination network</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>A multi-destination network</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Simulation results of multi-destination network</td>
<td>43</td>
</tr>
<tr>
<td>3.6</td>
<td>Illustration of the simplified OSU map</td>
<td>45</td>
</tr>
<tr>
<td>3.7</td>
<td>Simulation results for morning rush hour (&quot;-&quot; with routing, &quot;...&quot;) without routing)</td>
<td>46</td>
</tr>
</tbody>
</table>

xiv
CHAPTER 1

INTRODUCTION

The USA has one of the best surface transportation systems in the world. The transportation of people and goods in the USA is mainly through its road transportation systems. However, because the travel demand keeps growing while in many areas the building additional roadways to meet the increasing demand is not feasible due to the lack of suitable land to build on, limited resources, and environmental impact issues. As a result the mobility of the transportation system is threatened.

First, the quality of our life and the environment is threatened by congestion in the transportation system, particularly the Nation's highway and urbanized traffic networks. In a report by the Texas Transportation Institute on urban mobility [1], the total cost (delay plus wasted fuel) of 50 urbanized areas in the USA due to congestion was $48 billion in 1992. Secondly, safety on the surface transportation system is also a concern. Estimates show that approximately 40,115 people were killed and another 3 million were injured in traffic accidents involving automobiles in 1993.

There is no simple solution to this set of complex transportation problems. One way to help find a solution is the development and implementation of Intelligent Transportation Systems. Intelligent Transportation Systems (ITS) refers to the application of emerging hard and soft information systems technologies to address and
alleviate transportation problems. Figure 1.1 illustrates the main concepts of ITS as viewed by the author. As can be seen, there are communications between vehicles and from vehicles to traffic centers. It is believed that the improvement of four kinds of technologies effects the development of ITS technologies. They are:

- Advanced dynamic traffic control
- Advanced sensing technology
- Intelligence in vehicles
- Advanced communicated gathering and distribution

where, advanced dynamic traffic control technology refers to the optimization of traffic networks, intelligence in vehicles is the set of technologies to improve the automation of vehicles, and the advanced sensing technology and advanced communicated gathering and distribution are related to the information collection and communication of both traffic networks and vehicles.

In this dissertation, we are going to explore some issues in the advanced dynamic traffic control technology, i.e., the control and optimization problems in traffic networks.

1.1 Control and Optimization Problems in Traffic Networks

In this dissertation, traffic networks will be considered as two different classes. One is called local traffic networks, in which the speed limits of roads are normally less than 25m/s, the distances between intersections are short, and there are traffic lights at intersections to regulate the traffic from different directions. The other is freeway traffic networks, in which the speed limits of roads are normally higher than
Figure 1.1: Illustration of ITS
25m/s, no traffic light on the main line, and vehicles have limited access, whose entering/exiting is through on/off ramps.

In local traffic networks, the first control and optimization problem is the traffic signal control. This problem is one of optimization, to properly choose the traffic signal control parameters such that traffic can move safely and efficiently. The fundamental signal control parameter definitions [2] are given as follows:

- **Cycle time**, the total time for a signal to complete one sequence of signal indication
- **Phase**, the portion of a signal cycle allocated to any combination of traffic movements receiving simultaneous right-of-way
- **Green time**, the time within a given phase during which the green indication is shown
- **Lost time**, the time during which the intersection is not effectively used by any traffic movement
- **Split rate**, the ratio of green time to the cycle time
- **Offset**, the time difference between the start of the green indication at two successive intersections

All these parameters should be decided in traffic signal control. In this dissertation, we will fix some of parameters such as the cycle time as a constant to allow coordination and change the split rate in the real time to obtain an optimal performance.

At the same time, since there are so many possible routes from one place to another in a local traffic system, another control and optimization problem is the traffic
assignment or traffic routing problem. In this problem, we are concerned about how
to direct traffic efficiently from its source to its destination.

For freeway traffic networks, we also have two kinds of control and optimization
problems: traffic routing and traffic signal control, e.g. on-ramp control. Here, since
there are fewer freeways than local roads, the traffic routing of freeway systems is
conceptually simpler. But because of the differences between the characteristics of
freeways and local roads, there are some topics of traffic routing in freeway systems
which still need to be studied. For example, a typical routing problem in freeway
networks is called two-freeway routing. It studies how to direct traffic to an alternate
freeway when a freeway is congested. For the traffic signal control in freeway systems,
although there are no traffic lights on on the main lines, the traffic entering the free­
way systems still needs to be controlled. To achieve a high throughput of the main
line of a freeway system, on-ramp meters are used to control the traffic entering the
freeway. When the on-ramp meter is green, traffic can enter; when it is red, traffic
can not. How to control the ramp meters such that the highest throughput of the
main line is obtained is a important signal control problem in freeway networks.

In the next section, previous studies of the above control and optimization prob­
lems are discussed.

1.2 Literature Survey

1.2.1 Routing and Signal Control in Local Traffic Networks

Traffic routing problems have been studied extensively in the literature since
Wardrop's definition of the traffic equilibrium [3]. Traditionally, transportation re­
searchers have used static traffic assignment to estimate and predict traffic patterns
for traffic networks [4]. In the middle of 80s, people began to address the unrealistic assumptions of the static traffic routing [5]. At the same time, with the recent development of sensing and communication technologies, it has become possible to apply the dynamic traffic routing methods to the real traffic systems. Therefore, there has been a significant effort for the dynamic traffic routing of traffic networks [6][7].

Basically, there are two fundamental solutions to the dynamic routing problem: from the system view point, i.e., the system optimum; and from the user view point, which is called user equilibrium problem. The system optimum routing problem is to direct traffic such that the system optimal performance can be achieved. To achieve the system optimal, the average user can not ask for more. There are often some users who are forced to sacrifice for the good of the other users. So the system optimal routing strategies are usually difficult to be applied. However, the system optimum traffic pattern represents the efficient allocation of scarce network capacities and provides us with a lower bound on the total cost of travel for a given network, a given number of trips, and a given set of traffic management strategies. The user equilibrium is achieved when no driver can unilaterally reduce his/her travel costs by moving to another route. The user equilibrium traffic pattern may be viewed as a reasonable approximation of the noncooperative decisions made by commuters, which will actually result from the implementation of a particular routing policy. However, it is well known that the user equilibrium pattern is not likely to be efficient. The most common example of the inefficiency of the user equilibrium approach is Braess' paradox, which holds that total system cost of the system using the user equilibrium routing strategy may increase with the addition of a link to the network, for certain ranges of demand volumes. Extensive research has been reported for the dynamic
user equilibrium routing [8][9][10][11]. In [12] Bell presents a stochastic user equilib-
rium algorithm for steady state store-and-forward networks. The links of the network
have constant travel time and the links or nodes have finite capacities. Bell forms a
convex nonlinear programming problem and gives an iterative algorithm to solve the
nonlinear programming problem. The proof of convergence is provided. Kaufman,
Smith and Wunderlich [13] decompose the unified traffic dynamics and route choice
model into an assignment mapping and a routing mapping. They provide an iterative
routing mapping for continuous-time multipath routing, which adjust routing policies
more incrementally. In [14], Lam and Huang attempt to solve a multiple origins and
single destination user equilibrium problem. A continuous time optimal control model
is used in the paper, which can be regarded as a simplified discrete hydrodynamic
model. In order to avoid solving the complicated two-point boundary-value problem,
the authors develop a steady state-costate solution algorithm to generate an approx-
imate solution. At the same time, there are many researchers who have investigated
the system optimal routing problems [15]. As we know, the system optimal approach
is difficult to be implemented since it requires that all the drivers in the system follow
its directives. However, the research on system optimal dynamic routing problem is
still needed since it is the “base-line”. Although some individuals do not accept all
the directives given by system optimal routing strategies, we can study the effects
of certain percentage of acceptance, which implies the sub-optimum with respect to
the system optimum. Kufman, Nonis and Smith [16] use a mixed integer linear pro-
gramming formulation to solve the multi-destination dynamic system optimal route
guidance problem. This method has two stages: first, the selection of time depen-
dent link volumes, and second, the assignment of routes so that those volumes are
optimally utilized. In [17], a decentralized single-destination dynamic system optimal routing strategy is given by Sarachik and Ozguner. It is based on a linear model of links and nodes and uses the optimal control concept to find an optimal routing strategy for a network. This method is decentralized since the routing of the traffic is decided locally, i.e., every node gives the routing directives based on the value of the cost-to-go of its downstream links. The cost-to-go of a link includes the traffic information of its downstream nodes and links to the destination. The approach proposed in this earlier paper will be used to develop our decentralized multi-destination routing method in Chapter 3.

The comparison between the user equilibrium and system optimum solutions has long been studied in the literature [19], since it is one way to estimate the potential benefits of new traffic control and management system. In [18], Wie and the other authors compare the the time-varying traffic flow patterns under the system optimum and user equilibrium principles when drivers choose the departure time and routes simultaneously on a general traffic network with many origins and destinations at different levels of traffic congestion. These comparisons have important implications for the design and implementation of advanced traffic information and management systems.

The quality of a signal control system is generally defined in terms of safety and efficiency. Many methods have been developed to solve the intersection signal control problem [20][21][22][23]. A commonly used signal timing model is provided by Webster[24], who develops a detailed procedure to calculate cycle length and green times. One traffic signal control problem is to find an optimal split rate between green and red phases. In [25], Papageorgiou presents a centralized urban control
strategy. He designs a linear quadratic regulator off-line based on the simplified saturated intersection model proposed by Gazis [26]. This controller is used on-line for traffic-responsive coordinated network-wide signal control. Since the design of the regulator is based on the saturated traffic model, the analysis of traffic networks with this controller has shown that it cannot achieve the best performance when some of the intersections are unsaturated. At the same time, this method is centralized, which may not be feasible for a large traffic network. Han [27] develops a sequential optimization technique to minimize the total intersection delay over the successive periods by searching for the optimal signal timings and the time-shifts subject to certain queue length consideration. In [28], Davison and Ozguner apply a robust decentralized servomechanism method to balance the queue lengths when the intersections are saturated. A decentralized real-time intersection control scheme based on dynamic programming principle is proposed by Proche and Lafortune [29]. It requires the information regarding the future vehicle arrivals to compute the optimal signal switching sequence in order to minimize the delay time.

As the two main control and optimization problems in local traffic networks, the dynamic routing and traffic signal control have to be integrated together. Chen [2][32] proposes a three-level integration of dynamic intersection control and assignment, and applies game theory to solve this integration problem. The three levels of the integrations are the Cournot, Stackelberg and Monopoly games. The Cournot game is a duopoly game in which two players, traffic authority and the drivers, choose their strategies simultaneously. The Stackelberg game is a leader-follower game in which the traffic authority, who decides the traffic signal control based on the estimation of drivers' reaction, is the leader and the drivers are the follower. The monopoly game
is a system optimal control and assignment problem, in which the traffic authority controls both the signal settings and traffic flows, i.e. the drivers have to follow the routing of traffic authority. It is shown that the third level, a monopoly game, is the best one. In [33], Yang and Yagar formulate the combination of routing and signal control as a bi-level programming problem. The lower-level predicts how drivers will react to any given signal control pattern; the upper-level determines signal splits to optimize a system objective function, considering driver's route choice and route behavior in response to signal split changes. The gradient descent algorithm is used to optimize the signal splits.

1.2.2 Routing and Signal Control in Freeway Networks

Usually, people do not regard the routing problem of freeway networks as an individual routing problem. However, since the characteristics of the traffic dynamics in the freeway are different from those of local traffic networks, the dynamic routing in freeway networks needs to be studied. In [34], Kachroo and Ozbay propose a PI controller for the typical two-freeway routing problem based on the hydrodynamic model developed by Papageorgiou [35]. Then, Kachroo and Ozgay [36] present a the feedback linearization solution for this two-road routing problem. Their control objective is defined from users’ point of view, i.e., to balance the travel times of two roads. Since the hydrodynamic model is nonlinear, the feedback linearization method is applied to cancel the nonlinearity of the system dynamics and find the routing strategy.

Freeway traffic can be influenced by many kinds of traffic signal control, such as on-ramp control, variable speed limit signs, variable route recommendations, information provided to the drivers and further variable traffic signs. Inoue, Iida and
Hasegawa [37] use the linear programming on-ramp control method to reduce the highway traffic congestion. In [38] Papageorgiou proposes a feedback on-ramp control based on Linear-Quadratic methodology and an open-loop dynamic nonlinear optimal on-ramp control. In [40], a standard parameterized closed-loop regulator for on-ramp control has been developed by using Powell's method. Here, the ramp control limits and regulates the entrance of vehicles according to the time-varying traffic demand and condition. However, some researchers find that on-ramp control cannot always improve the performance of a traffic network, which includes highways and local roads. The results of Zhang and Recker [39] show that when an optimal on-ramp control is applied to a network with local roads and highways, it will never improve the performance, unless drivers have at least some propensity to divert their paths from the freeway based on the queue at the entry ramp. This problem does not exist in dynamic routing systems, because the dynamic traffic information is available for the vehicles and the vehicles are directed by certain optimal routing strategy which will divert vehicles based on the queue lengths at the entry ramps.

1.3 Outline of Contents

In this dissertation, all of our discussion is based on an assumption that the vehicles are not fully automated since we are going to analyze the signal control problem but all of the vehicles will follow the directives given by our routing strategies.

Chapter 2 presents the two models of traffic networks. Since every traffic network contains static and dynamic traffic information, the representations of the two kinds of traffic information are given. To represent the static information of traffic networks, we define a set of topology matrices and an operator of them. Using the
first-order topology matrix and the operator, all the possible routes from any node to another in the network and the distances of the routes can be found. For the dynamic information, due to the different characteristic of traffic in local and freeway traffic system, two different models are used. First, we use the point-queueing model with the fixed travel time along links to represent the traffic in local traffic network. Then the hydrodynamic model is given, in which the dynamics are characterized by traffic flow, traffic density and traffic velocity, and the travel time of a link varies with time and the changes of traffic density and flow. The representation of the static and dynamic information of traffic networks is the basis of the analysis of control and optimization problems in the later chapters.

In Chapter 3 and 4, the control and optimization problems in local traffic networks are investigated. Chapter 3 is focused on the dynamic routing problem of local networks. Here, we use the simple-to-complex approach to solve the routing problem of congested local traffic networks with multi-destination. First, we apply the optimal control method to find the routing strategy of a single-destination network with parallel links. Then the routing strategy is extended to multi-node single-destination networks and multi-destination networks via the decentralized approach. In the end, a decentralized multi-destination dynamic routing strategy for congested local traffic networks with traffic lights is obtained. In Chapter 4, a decentralized hybrid intersection control method is proposed. We first analyze the characteristics of an isolated intersection based on the point-queueing model of traffic in local networks. Under the assumption, the dynamics of an intersection have only two states: unsaturated and saturated. Based on the different dynamics of the two states, we design two intersection controllers and then combine them into one hybrid controller. This hybrid
controller is extended to multi-intersection case and becomes a decentralized hybrid intersection controller.

In Chapter 5 and 6, the dynamic routing and traffic signal control of freeway networks are explored. Since the dynamics of freeway traffic can be modeled by a set of nonlinear equations, we apply the sliding mode approach to both the routing problem and the signal control. In Chapter 5, a typical routing problem in freeway systems, the two-freeway routing, is investigated. We first formulate a user equilibrium two-freeway routing problem and solve it via sliding mode. It is shown that under this sliding mode routing method, the user equilibrium problem is equivalent to the system optimal problem. In Chapter 6, a signal control in freeway networks, on-ramp control, is discussed. Here we apply the sliding mode extremum seeking method to the on-ramp control problem such that the point with the highest throughput of the freeway system is searched and controlled in real-time.

Chapter 7 introduces how the control methods proposed in this dissertation are applied in our traffic and vehicle simulator, VATSIM. A brief summary of the topics covered in this thesis as well as possible future research directions are given in Chapter 8, which concludes this dissertation.
CHAPTER 2

NOTATION AND TRAFFIC NETWORK MODELING

2.1 Overview

Before we analyze the control and optimization problems in traffic network systems, we will discuss the model of traffic in transportation networks. As we know, every traffic network contains static and dynamic traffic information: the topology of a network contains the main static traffic information of the network, e.g., the number and locations of nodes, the connections between nodes, etc.; and the dynamic traffic information of the network is characterized by the traffic densities, traffic flow, traffic speeds, queue lengths, etc. How to systematically represent the static and dynamic information of a traffic network is important for our study of the control and optimization of transportation systems in the later.

In this chapter, models of traffic network will be presented. After giving the notation of variables used throughout this dissertation, we will define a topology matrix to model the static traffic information. Then two dynamic traffic models will be discussed: the point-queuing model and hydrodynamic model, which will be used in the analysis of our traffic routing and signal control problems.
2.2 Notation

The notations used throughout the dissertation are defined below:

**Static Traffic Information Variables:**

- $A$: A traffic network;
- $N(A)$: Set of nodes (intersections and termination points of roads) in $A$;
- $L(A)$: Set of links (roads) in $A$;
- $N_n$: The $n$th node in $A$;
- $N$: Total number of nodes in $A$;
- $(x,y)$: A point in $A$ where $x$ and $y$ are the coordinates;
- $(N_m,N_n)$ or $(m,n)$: Link from node $m$ to node $n$;
- $d_{mn}$: Distance of link $(m,n)$;
- $\tau_{mn}$: Travel time of link $(m,n)$;
- $\theta_{ij}$: First order topology matrix of $A$;
- $\Theta = [\theta_{ij}]_{N\times N}$: First order topology matrix of $A$;
- $D$: First order distance matrix of $A$;
- $T$: First order travel time matrix of $A$;

**Dynamic Traffic Information Variables:**

- $q_n^j(t)$: Queue length of the $j$th direction of the $n$th node at time $t$;
- $q_{x,y,n}(t)$: Simplified notation of queue length at the $n$th node, when the node only has $x$ and $y$ directions;
- $r_n^j(t)$: Traffic flow coming from outside of the network at the $j$th direction of the $n$th node;
- $\varphi_n^j$: Estimated mean value of traffic from outside of the network at the $j$th direction of the $n$th node;
- $u_{mn}^i(t)$: Traffic flow from link $(m,n)$ at time $t$;
- $u_{nk}^i(t)$: Traffic flow into link $(n,k)$ at time $t$;
- $u_{nx,y}^i(t)$: Traffic flow coming to the $x$ or $y$ direction at the $n$th node;
- $\bar{u}_{nx,y}(t)$: Estimated mean value of traffic flow to the $x$ or $y$ direction at the $n$th node;
- $u_{nx,y}^o(t)$: Traffic flow leaving from the $x$ or $y$ direction at the $n$th node;
- $u_{mn}^m(t)$: Maximum traffic flow of the $x$ or $y$ direction at the $n$th node;
- $L_{jn}$: Index sets of links entering the node $n$ at the $j$th direction;
- $L_{jn}^o$: Index sets of links leaving the node $n$ at the $j$th direction;
- $S_{mn}^d$: Cost-to-go of link $(m,n)$ to destination $d$;
- $T$: Cycle time of a traffic light;
- $T_l$: Lost time per cycle time;
- $c_i$: Split rate of effective green times at $x$ and $y$ direction of the $i$th node;
$\rho_{mn}(x, t)$ Traffic density of link $(m, n)$ at position $x$ time $t$;
$f_{mn}(x, t)$ Traffic flow of link $(m, n)$ at position $x$ time $t$;
$v_{mn}(x, t)$ Traffic velocity of link $(m, n)$ at position $x$ time $t$;
$C_{mn}$ Capacity of link $(m, n)$;
$\rho^i_{mn}(t)$ Average traffic density of the $i$th segment of link $(m, n)$ at time $t$;
$f^i_{mn}(t)$ Average traffic flow of the $i$th segment of link $(m, n)$ at time $t$;
$v^i_{mn}(t)$ Average traffic velocity of the $i$th segment of link $(m, n)$ at time $t$;
$\delta^i_{mn}(t)$ Length of the $i$th segment of link $(m, n)$ at time $t$;
$\nu^i_{f, mn}(t)$ Free traffic speed of the $i$th segment of link $(m, n)$ at time $t$;
$s_i(t)$ Off-ramp traffic flow at the $i$th ramp;
$s(x, t)$ $\sum_{i=1}^{M} s_i(t) \delta(x - x_i)$, total off-ramp traffic of a freeway at time $t$;
$p_i(t)$ On-ramp traffic flow at the $i$th ramp;
$p(x, t)$ $\sum_{i=1}^{M} p_i(t) \delta(x - (x_i + \Delta_i))$, total off-ramp traffic of a freeway at time $t$;
$\Delta_i$ Distance between the $i$th on and off ramp;

2.3 Static Information

2.3.1 Representation

The topology of a traffic network $\mathcal{A}$ is defined to be a pair $(\mathcal{N}(\mathcal{A}), \mathcal{L}(\mathcal{A}))$, where $\mathcal{N}(\mathcal{A})$ is a non-empty finite set of elements called nodes which include intersections and termination points of roads in the traffic network, and $\mathcal{L}(\mathcal{A})$ is a finite family of elements called links which represent roads with different traffic directions.

Assuming that there are $N$ nodes in the traffic network and every node has its unique index number, then $\mathcal{N}(\mathcal{A}) = \{ \mathcal{N}_n(x_n, y_n) \}$, where $n$ is the index number of a node, which is an integer between 1 and $N$, and $x_n$ and $y_n$ are the coordinates of node $n$ in the traffic network.

A link from node $i$ to node $j$ is a pair of ordered nodes and a finite set of ordered points which is represented as $\{(\mathcal{N}_i, \mathcal{N}_j) | (x_1, y_1), \cdots, (x_l, y_l), \cdots, (x_L, y_L) \}$, where $\mathcal{N}_i$ and $\mathcal{N}_j$ are start and end nodes of the link, and $(x_l, y_l)$ is the coordinations of the $l$th
point of the link. Since there are total $L$ points in the link, $l = 1, \cdots, L$. Because sometimes we don’t need so much detail for a link, the representation can be abbreviated to an ordered pair of nodes as $(N_i, N_j)$ or just simply $(i, j)$. Here, we do not include the case that there is more than one link from node $i$ to node $j$ since it is unusual that there is more than one road from one place to anther without passing other intersections in a real traffic network. The length of a link can be obtained by a function called the distance function. If we assume that every two sequent points of a link, $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$, are connected with a straight line, the distance function is given as follows,

$$d(N_i, N_j) = \sum_{l=1}^{L-1} \sqrt{(x_l - x_{l+1})^2 + (y_l - y_{l+1})^2}.$$  \hspace{1cm} (2.1)

The delay of link $(N_i, N_j)$ is defined as the travel time between node $N_i$ and $N_j$, which is related to both the static information, i.e., the distance of a link, and the dynamic information, i.e., the speed of traffic. If we assume that the traffic speed at link $(N_i, N_j)$ is fixed as $v$, the travel time of the link can calculated as

$$\tau(N_i, N_j) = \frac{d(N_i, N_j)}{v}. \hspace{1cm} (2.2)$$

Now, let us represent the topology information of a traffic network by matrices. First, we define the first-order topology matrix as

$$\Theta = [\theta_{ij}]_{N \times N}, \hspace{1cm} (2.3)$$

where

$$\theta_{ij} = \begin{cases} (i, j) & \text{if there is a link between node } N_i \text{ and } N_j \\ \phi & \text{if } i = j \text{ or no link between node } N_i \text{ and } N_j \end{cases}. \hspace{1cm} (2.4)$$

Here $\phi$ denotes the empty set of links and it has a property as $\phi \cup (i, j) = (i, j)$. The reason why $\theta_{ii}, i = 1, \cdots, N$ is alway equal to $\phi$ is that we are not interested in the
loop information of a traffic network.

Now, the \( i \)th row of matrix \( \Theta \) contains the information of nodes reached by node \( j \) without passing other nodes. The \( j \)th column of \( \Theta \) represents the nodes which can reach the node \( i \) without passing other nodes. The distance from a node to another node can be obtained by applying the distance operator to the first-order matrix \( \Theta \), which is

\[
\mathcal{D} = d(\Theta) = [d(\theta_{ij})] = [d_{ij}] .
\]

Similarly, applying the function of travel time to the \( \Theta \) matrix, we can obtain the travel time matrix

\[
\mathcal{T} = \tau(\Theta) = [\tau(\theta_{ij})] = [\tau_{ij}] .
\]

Now, we define an operator \( \otimes \) such that the path from node \( i \) to node \( j \) through other nodes can be found. The rules for \( \otimes \) are

\[
\phi \otimes \phi = \phi , \quad (2.7)
\]

\[
\phi \otimes (i, j) = \phi , \quad (2.8)
\]

\[
(i, k) \otimes (k, j) = (i, k, j) , \quad \text{if } i \neq j , \quad (2.9)
\]

\[
(i, k) \otimes (k, j) = \phi , \quad \text{if } i = j , \quad (2.10)
\]

\[
(i, \ldots, k) \otimes (k, \ldots, j) = (i, \ldots, k, \ldots, j) , \quad \text{if } i \neq j , \quad (2.11)
\]

\[
(i, \ldots, k) \otimes (k, \ldots, j) = \phi , \quad \text{if } i = j , \quad (2.12)
\]

Then the second-order topology matrix

\[
\Theta^2 = [\theta^2_{ij}] = \Theta \otimes \Theta \quad (2.13)
\]
with
\[
\theta^2_{ij} = \bigcup_{k=1}^{N} \theta_{ik} \otimes \theta_{kj} .
\] (2.14)

According to the definition of \( \otimes \), we can find that \( \theta^2_{ij} \) is always equal to \( \phi \), when \( i = j \).

And \( \theta^2_{ij} \) represents the set of all possible paths from node \( i \) to node \( j \) through one other node \( k \) in the network. Similarly, we can obtain \( \Theta^i \) with \( i = 3, \cdots, N \). All these matrices contain the information of the paths from a node to another node through other \( i - 1 \) nodes. At the same time, the distance from one node to another node can be found by applying the distance operator to those matrices.

Therefore, the topology information of the network is contained by the topology matrices \( \Theta^i \), \( i = 1, \cdots, N \).

### 2.3.2 Examples

Here, we will give two examples to show how to use the above matrices to represent the topologies of networks.

For the following simple network, we can write the topology matrix \( \Theta \) as

![Example network 1](image_url)

\[ \Theta = \begin{bmatrix} \phi & (1,2) & (1,3) \\ (1,2) & \phi & (2,3) \\ (1,3) & (2,3) & \phi \end{bmatrix} \] (2.15)
since there are three links in the network: \((N_1, N_2), (N_1, N_3)\) and \((N_2, N_3)\). The distance and travel time matrix of the example network 1 are

\[
D = \begin{bmatrix}
\phi & d_{12} & d_{13} \\
\phi & \phi & d_{23} \\
\phi & \phi & \phi \\
\end{bmatrix}
\]

and

\[
T = \begin{bmatrix}
\phi & \tau_{12} & \tau_{13} \\
\phi & \phi & \tau_{23} \\
\phi & \phi & \phi \\
\end{bmatrix}.
\]

Based on our definition, the second order topology matrix is

\[
\Theta^2 = \Theta \otimes \Theta = \begin{bmatrix}
\phi & \phi & (1, 2, 3) \\
\phi & \phi & \phi \\
\phi & \phi & \phi \\
\end{bmatrix},
\]

where \((1, 2, 3) = (1, 2) \otimes (2, 3) = (N_1, N_2, N_3)\) which means that there is a path from node 1 through node 2 to node 3. And the third order topology matrix is an empty matrix, which means that there is no path from any node to any other node through two other nodes in the network. Then the topology information of the simple example network given in Figure 2.1 is properly represented by the topology matrices.

Now, Consider a more complex network given in Figure 2.2. The first-order topol-
ogy matrix is written as

$$\Theta = \begin{bmatrix}
\phi & (1, 2) & (1, 3) & \phi & \phi \\
\phi & \phi & (2, 3) & (2, 4) & (2, 5) \\
\phi & (3, 2) & \phi & (3, 4) & (3, 5) \\
\phi & \phi & \phi & \phi & (4, 5) \\
\phi & \phi & \phi & \phi & \phi
\end{bmatrix} .$$ (2.19)

Then, the second order topology matrix

$$\Theta^2 = \begin{bmatrix}
\phi & (1, 3, 2) & (1, 2, 3) & (1, 2, 4) \cup (1, 3, 4) & (1, 2, 5) \cup (1, 3, 5) \\
\phi & \phi & \phi & (2, 3, 4) \cup (2, 5, 4) & (2, 3, 5) \cup (2, 4, 5) \\
\phi & \phi & \phi & (3, 2, 4) \cup (3, 5, 4) & (3, 2, 5) \cup (3, 4, 5) \\
\phi & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi
\end{bmatrix} .$$ (2.20)

and third order topology matrix

$$\Theta^3 = \begin{bmatrix}
\phi & \phi & \phi & (1, 3, 2, 3, 4) \cup (1, 3, 2, 5, 4) & (1, 3, 2, 3, 5) \cup (1, 3, 2, 4, 5) \\
\phi & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi \\
\phi & \phi & \phi & \phi & \phi
\end{bmatrix} .$$ (2.21)

where any path passing a node twice means there is a loop in the path. For this network, the topology matrices with order higher than 3 are empty matrices.

From the above two examples, we can see that the definition and method proposed above can fully represent the topology information of networks.

### 2.4 Dynamic Information

Obtaining the topology information is not enough to formulate the control and optimization problems for a traffic network. To do that, we still need to model the dynamic traffic information. Now, two dynamic traffic models will be used in this dissertation. One is the point-queuing model and the other is the hydrodynamic model. The travel time along a link assumed to be a constant in the point-queuing
model, but in the hydrodynamic model, the travel time is decided by the average travel speed, which is a function of traffic density of the link. Based on the characteristics of the two different model, we find that the point-queuing model is convenient for describing the traffic in local networks and the hydrodynamic model can be used to represent the traffic in both local and freeway networks.

2.4.1 Point-queuing Model

For every traffic network, the traffic information is associated with links and nodes. A simple but reasonable model for a local traffic network is the point-queuing model [17]. This point-queuing model consists of two parts. The first part is a linear link model, and the other part is a node “tank” model.

The linear link model is a much simplified model, in which the influence of different traffic density on the travel speed of vehicles is neglected, i.e., vehicles are assumed to move along a link with a constant speed. So for a link \((m, n)\) with \(x \in [0, X]\), (Figure 2.3), the traffic dynamics are given as

![Figure 2.3: A simple link](image)

The linear link model is a much simplified model, in which the influence of different traffic density on the travel speed of vehicles is neglected, i.e., vehicles are assumed to move along a link with a constant speed. So for a link \((m, n)\) with \(x \in [0, X]\), (Figure 2.3), the traffic dynamics are given as
\[ 0 \leq \rho_{mn}(x, t) \leq \rho_{max} \]  
\[ u_{mn}(x, t) = v_0 \]  
\[ f_{mn}(x, t) = \rho_{mn}(x, t)u_{mn}(x, t) = \rho_{mn}(x, t)v_0 \]

where \( \rho_{mn}(x, t) \), \( u_{mn}(x, t) \) and \( f_{mn}(x, t) \) are the traffic density, velocity and traffic flow of link \((m, n)\) at position \(x\) time \(t\) respectively, \( \rho_{max} \) denotes the maximum traffic density, and the maximum traffic flow is usually called the capacity of the link denoted \( C_{mn} \). Based on the definition of traffic flow, we know it is equal to the product of traffic density and traffic velocity. Thus, the link capacity \( C_{mn} = \rho_{max}v_0 \), where \( v_0 \) is the constant speed.

The travel time on a link is fixed, which can be calculated as

\[ \tau_{mn} = \frac{d_{mn}}{v_0} \]  

So in the point-queuing model, link \((m, n)\) is represented as a time delay between node \(m\) and \(n\).

For the traffic at the \(j\)th direction of \(n\)th node in the network, which is shown in Figure 2.4, its "tank" model can be written as

\[ \text{Figure 2.4: A direction of a node} \]
where \( q^j_n(t) \) is the queue length of the \( j \)th direction of the \( n \)th node at time \( t \), \( r^j_n(t) \) denotes the traffic flow coming from outside of the network, \( u^i_{mn}(t) \) represent the flow from link \((m,n)\) at time \( t \), \( u^o_{nk}(t) \) is the flow into link \((n,k)\) at time \( t \), and \( L^i_{jn} \) and \( L^o_{jn} \) denote the index sets of links entering and leaving node \( n \) at the \( j \)th direction.

If the node \( n \) has two incoming traffic directions, its illustration is shown in Figure 2.5. As we can see, there are two queues, denoted as \( q^1_n \) and \( q^2_n \) related to direction 1 and 2. \( L^o_{1n} \) and \( L^o_{2n} \) represent the two sets of upstream links at the two directions, and \( L^i_{1n} \) and \( L^i_{2n} \) are the sets of the downstream links at the two directions. However, there might be overlap between the two sets of downstream links, i.e., some links can be the downstream links of both directions. Based on those notations, the "tank"
model for the traffic at the two directions of this node is given as

\[ \dot{q}_n^1(t) = r_n^1(t) + \sum_{(m,n) \in L_{1n}} u^i_{mn}(t) - \sum_{(n,k) \in L_{1n}} u^o_{nk}(t), \quad (2.27) \]

\[ \dot{q}_n^2(t) = r_n^2(t) + \sum_{(m,n) \in L_{2n}} u^i_{mn}(t) - \sum_{(n,k) \in L_{2n}} u^o_{nk}(t). \quad (2.28) \]

Combining the linear model of links and “tank” model of nodes, we can summarize the point-queuing model as

\[ \dot{q}_n^j(t) = r_n^j(t) + \sum_{\theta_{mn} \in L_{jn}} u_{mn}(t - \tau_{mn}) - g_j(t) \sum_{\theta_{nk} \in L_{jn}} u_{nk}(t), \quad (2.29) \]

\[ u_{nk}(t) = 0, \quad \theta_{nk} \in L_{jn} \quad \text{when } q_n^j = 0, \quad (2.30) \]

where \( u_{mn}(t) \) is the traffic entering link \((m,n)\) at time \(t\), and \( g_j(t) \) represents the phase of the traffic light at direction \(j\) of node \(n\), \( g_j(t) = \begin{cases} 1 & \text{light is green} \\ 0 & \text{otherwise} \end{cases} \).

### 2.4.2 Hydrodynamic Model

From the macroscopic perspective, the traffic flow in a freeway is considered analogous to a fluid flow, which is a distributed parameter system represented by partial differential equations [35][34]. A mass conservation model of a freeway \((m,n)\) (Figure 2.6), characterized by position \(x \in [0, X]\), is given by

\[ \frac{\partial}{\partial t} \rho_{mn}(x, t) = -\frac{\partial}{\partial x} f_{mn}(x, t). \quad (2.31) \]
According to the definition of traffic flow $f(x, t)$, we have

$$f_{mn}(x, t) = \rho_{mn}(x, t)\nu_{mn}(x, t)$$

(2.32)

From the theory of kinematic waves, we know the traffic flow and traffic density are related. The relationship is illustrated by the flow-density characteristics curve shown in Figure 2.7. As we can see, the curve is separated into two parts by the critical density $\rho_{cr}$. At point $\rho_{cr}, f_{max}$ the curve reaches its maximum; i.e., when $\rho = \rho_{cr}$, the highway system achieves the highest throughput.

The PDE model (2.31) can be discretized to continuous ODEs by dividing the
road \((m, n)\) into segments, i.e.,

\[
\dot{\rho}_{mn}^i(t) = \frac{1}{\delta_{mn}^i} [f_{mn}^{i-1}(t) - f_{mn}^i(t)] . \tag{2.33}
\]

where \(\rho_{mn}^i(t)\) and \(f_{mn}^i(t)\) are the traffic density and flow of the \(i^{th}\) segment of the road \((m, n)\), and \(\delta_{mn}^i\) is the length of the \(i^{th}\) segment. Based on the research of traffic engineers [35], the dynamics of traffic velocity at the \(i^{th}\) segment without neglecting the reaction time of drivers can be written as

\[
\dot{v}_{mn}^i(t) = (v_{mn}^{i-1}(t) - v_{mn}^i(t)) \frac{v_{mn}^i (t)}{\delta_{mn}^i} + \frac{1}{\kappa} (v_e(\rho_{mn}^i(t)) - v_{mn}^i(t)) - \frac{\mu}{\kappa \delta_{mn}^i} \frac{\rho_{mn}^{i+1}(t) - \rho_{mn}^i(t)}{\rho_{mn}^i(t)} + \frac{\chi}{\kappa} , \tag{2.34}
\]

where \(v_e(\cdot)\) is the mean speed of traffic. The mean speed of traffic is a function of the density of the \(i^{th}\) segment \(\rho_{mn}^i(t)\), which is written as

\[
v_e(\rho_{mn}^i) = v_{f_{mn}} \left[1 - \left(\frac{\rho_{mn}^i(t)}{\rho_{max}}\right)^l\right] . \tag{2.35}
\]

where \(v_{f_{mn}}\) is the free speed and \(\rho_{max}\) is the maximum traffic density of the road, and \(\kappa, \mu, \chi, l\) and \(m\) are constant, which are given in [35]. Actually, this mean speed function is modified from an assumption proposed by Greenshield [26], in which a linear relationship was assumed between speed and density.

In order to enable the formulation of an optimal control problem, the flow-density characteristics have to be expressed in terms of the space discretized variables. For this purpose, the traffic flow of the \(i^{th}\) segment is expressed as a weighted sum of the traffic flow corresponding to the densities of segments, i.e.,

\[
f_{mn}^i(t) = \alpha \rho_{mn}^i(t) v_{mn}^i(t) + (1 - \alpha) \rho_{mn}^{i+1}(t) v_{mn}^{i+1}(t) , \quad 0 \leq \alpha \leq 1 . \tag{2.36}
\]

This hydrodynamic model only coarsely represents reality since there are at least three traffic features which are not properly captured [41]. The three features
are driver differences, vehicular motion through a shock wave and traffic instability, where the shock wave, representing the discontinuities of traffic propagation, is a special kind of phenomenon in traffic systems. For the driver differences, we know that in the real world some drivers are driving faster than others during the light traffic. But in the hydrodynamic model, the state of the system at time $t$ is described by a single unstratified variable, $\rho_{mn}(x,t)$, which can not provide the information about the types of drivers near $x$ at $t$. Moreover, a passing intersection between fast and slow cars in the real world is very different from that in the hydrodynamic model. The second deficiency of the hydrodynamic model is that it implies the vehicles change their speed with infinite decelerations and accelerations, which is physically impossible. But this does not indicate that the predictions of the hydrodynamic model are erroneous away from shocks. For the third deficiency, the traffic system is always stable in the hydrodynamic model when the world is not. As a result, it can not explain the stop-and-go traffic behavior which is often observed in congested freeway sections.

Although the hydrodynamic model has the above deficiencies, it still can be used to analyze traffic problems because it does capture most of the characteristics of traffic. In Chapter 5 and 6, we will discuss the two-freeway routing and on-ramp control based on this hydrodynamic model.

2.5 Summary

In this chapter, we analyzed the representation of static information and models of dynamic traffic information. First, the topology matrices were defined to represent the static information of traffic networks. Then two dynamic models were provided
to describe the traffic dynamics in local and freeway traffic networks. In the later chapters, the control and optimization problem in traffic network, such as dynamic routing and traffic signal control, will be discussed based on these models.
CHAPTER 3

DECENTRALIZED MULTI-DESTINATION DYNAMIC ROUTING STRATEGY FOR CONGESTED LOCAL TRAFFIC NETWORKS

3.1 Overview

Many dynamic routing strategies reported in the literature are centralized where all the traffic conditions and traffic requirements of a traffic network are collected and the routing of traffic is decided by a traffic center. If these strategies can be implemented to real traffic networks, the performance should be good. However, traffic systems are usually very large. It is possible that there are hundreds of roads and intersections and thousands of vehicles in one traffic network. Using only one traffic center to handle the routing problem of all the vehicles in this network, i.e., centralized routing strategies, is not feasible. Therefore, the decentralized approach has to be used for the routing of large traffic networks.

In this chapter, a decentralized dynamic routing strategy for congested local traffic networks will be introduced. The routing of a simple network with only one node, one destination and parallel links will be analyzed first. Then, this method will be extended to multi-node networks with single destination and with multi-destination in a decentralized way. We will also provide a way to modify the decentralized routing
strategy such that it can consider the delay caused by traffic lights. The simulation results given in the end of this chapter show that this decentralized multi-destination dynamic routing strategy does alleviate the congestion of networks.

3.2 Problem Statement

In this chapter, we will first consider the traffic routing problem of local networks without the consideration of the effects of traffic lights.

Now, consider a complex traffic network with $D$ destinations, $N$ other nodes. All the nodes are connected by $L$ links and queues of vehicles can be formed at every non-destination node. To simplify the notations of queues and links in networks, we only index nodes and links by numbers in this chapter. The point-queuing model can be written as

$$q_n^j(t) = r_n^j(t) + \sum_{l \in L_n^j} u_l(t - r_l) - \sum_{l \in L_n^j} u_l(t) .$$

(3.1)

If we add another super-script to index the destination of the traffic, the queue of the $n$th node in the $j$th direction $q_n^j(t) = \sum_{d=1}^D q_n^{jd}(t)$, and $u_l(t) = \sum_{d=1}^D u_l^d(t)$. There are constraints for queue length and traffic flow, which are

$$q_n^j(t) \geq 0 , \quad 0 \leq u_l(t) \leq C_l .$$

(3.2)

Similarly, the traffic from the outside of the network of the $j$th directions at node $n$ $r_n^j(t) = \sum_{d=1}^D r_n^{jd}(t)$. It varies over the time. But the summation of the mean values $\bar{r}_n^j$ at all the directions of node $n$ must be in the range that the network can handle. If we assume that the $n$th node has $B_n$ directions, the constraint for the traffic from
outside of the network can be expressed as

\[ 0 \leq \sum_{j}^{B_n} \hat{r}_n^j \leq \sum_{l \in L_n} C_l . \]  

Otherwise some state variables of this system will tend to infinity, i.e., the system is unstable, whatever the routing strategy is.

Since the performance of the routing is measured by the delay time caused by the network, which includes the travel delay along links and queuing delay at nodes, we define the cost function as

\[ J\{u\{t\}\} = \int_{0}^{\infty} \left[ \sum_{n=1}^{N} \sum_{j=1}^{B_n} q_n^j(t) + \sum_{l=1}^{L} t_l u_l(t) \right] dt . \]  

where \( B_n \) denotes the total number of directions in node \( n \) where queues can formed.

Now, our problem is how to find a feedback routing strategy, i.e. \( u_l(t) \), to minimize the cost. Since there are the delay and all kinds of constraints in the dynamics of the system, the problem is not easy to solve. We will analyze a simple routing problem first and then extend it to complicated situations.

### 3.3 Routing for a Simple Network with Single-destination

Consider a network shown in Figure 3.1, which has a single source node and \( L \) parallel links leading to the destination. The dynamics of the queue is

\[ \dot{q}(t) = r(t) - \sum_{l=1}^{L} u_l(t) . \]  

Let \( S_l \) denote the delay or cost to reach the destination per unit of flow along link \( l \) and number the links according to the ordering

\[ S_1 \leq S_2 \leq \cdots \leq S_L . \]  

32
In this simple case, $S_i$ is equal to the travel time of link $i$. However, in a more complex network, $S_i$ is not simply equal to the travel time any more. In the next sections, we will discuss how to calculate $S_i$ for different complex networks, but now let us focus on this simple network.

The performance to be optimized becomes

$$J\{u(t)\} = \int_0^\infty \left[ q(t) + \sum_{i=1}^L S_i u_i(t) \right] dt .$$  \hfill (3.7)

Write the Hamilton function as

$$H(q, u, \lambda, t) = q(t) + \sum_{i=1}^L S_i u_i(t) + \lambda(t)(r(t) - \sum_{i=1}^L u_i(t)) ,$$ \hfill (3.8)

$$= q(t) + \sum_{i=1}^L (S_i - \lambda)u_i + \lambda r(t) .$$

According to the Maximum principle, the control should be

$$u_i^* = \begin{cases} C_i & \text{if } \lambda > S_i , \\ 0 & \text{if } \lambda \leq S_i . \end{cases} \hfill (3.9)$$

Since in this problem, the final time is free and the Hamilton function is not the explicit function of time $t$, then

$$H(q^*, u^*, \lambda^*) \equiv 0 ,$$ \hfill (3.10)
and Eqn. (3.8) becomes

\[ q(t) + \sum_{i=1}^{L} (S_i - \lambda) u_i^* + \lambda r(t) = 0 . \]  

(3.11)

Assume \( K \) is the largest index \( l \) satisfying \( S_l < \lambda \). The control becomes

\[ u_i^* = \begin{cases} 
C_l & l \leq K \\
0 & \text{otherwise} 
\end{cases} . \]  

(3.12)

Substituting this control into Eqn. (3.11), we obtain

\[ q(t) + \sum_{i=1}^{K} (S_i - \lambda) C_l + \lambda r(t) = 0 . \]  

(3.13)

Then

\[ \lambda^* = \frac{q + \sum_{i=1}^{K} S_i C_l}{\sum_{i=1}^{K} C_l - r} , \]  

(3.14)

and the condition \( \lambda > S_K \) becomes

\[ q > \sum_{i=1}^{K} C_l (S_K - S_l) - S_K r . \]  

(3.15)

With the condition \( S_1 \leq S_2 \leq \cdots \leq S_K \), the links \( i = 1, 2, \cdots, K \) satisfy Eqn. (3.15), i.e.,

\[ q > \sum_{i=1}^{i} C_i (S_i - S_l) - S_i r \quad i = 1, 2, \cdots, K , \]  

(3.16)

Therefore, the dynamic routing strategy is

\[ u_i^* = \begin{cases} 
C_i & q > \sum_{i=1}^{i} C_i (S_i - S_l) - S_i r \\
0 & q \leq \sum_{i=1}^{i} C_i (S_i - S_l) - S_i r \end{cases} . \]  

(3.17)

**Theorem 3.1** Consider the network in Figure 3.1. Under the condition (3.6) and dynamic routing strategy (3.17), the total time to clear the congestion of the network
can be approximately calculated as

\[
\psi = \frac{q_0 + \sum_{i=1}^{K} C_i S_i}{\sum_{i=1}^{K} C_i - \hat{\bar{f}}},
\]

(3.18)

where \(q_0\) is the initial queue length and \(\hat{\bar{f}}\) is the estimated mean value of traffic flow from the outside of the network.

3.1 Proof: Let \(T_K, T_{K-1}, \ldots, T_i, \ldots\) be the time at which the links close and assume the queue is clear at time \(T_K\). Then the time for the congestion in the network to be clear is \(T_K + S_K\). Obviously, we have

\[
T_K < T_{K-1} < \cdots < T_i < \cdots
\]

(3.19)

From the routing strategy, we know initially links \(i = 1, \ldots, K\) carry traffic since

\[
q > \sum_{i=1}^{K} C_i (S_K - S_i) - S_K\hat{\bar{f}} \text{ holds.}
\]

The dynamics of queue can be rewritten as

\[
\dot{q} = \hat{\bar{f}} - \sum_{i=1}^{K} C_i, \quad 0 \leq t \leq T_K
\]

(3.20)

Because \(\hat{\bar{f}}\) is fixed during the time period \(0 \leq t \leq T_K\),

\[
q(t) = q_0 - \left(\sum_{i=1}^{K} C_i - \hat{\bar{f}}\right)t, \quad 0 \leq t \leq T_K
\]

(3.21)

Since link \(K\) becomes closed at time \(T_K\), the queue length at this time

\[
q(T_K) = \sum_{i=1}^{K} C_i (S_K - S_i) - S_K\hat{\bar{f}}
\]

(3.22)

From Eqn. (3.21) and (3.22), we have

\[
T_K = \frac{q_0 - \sum_{i=1}^{K} C_i (S_K - S_i) + S_i\hat{\bar{f}}} \sum_{i=1}^{K} C_i - \hat{\bar{f}}
\]

(3.23)

\[
= \frac{q_0 + \sum_{i=1}^{K} C_i S_K}{\sum_{i=1}^{K} C_i - \hat{\bar{f}}} - S_K
\]
Similarly, the queue length at later time

\[ q(t) = q(T_i) - \left( \sum_{l=1}^{i-1} C_l - \hat{\tau} \right) (t - T_i), \quad T_i \leq t \leq T_{i-1} . \quad (3.24) \]

Then

\[ q(T_{i-1}) = q(T_i) - \left( \sum_{l=1}^{i-1} C_l - \hat{\tau} \right) (T_{i-1} - T_i), \quad (3.25) \]

\[ q(T_i) - q(T_{i-1}) = \left( \sum_{l=1}^{i-1} C_l - \hat{\tau} \right) (T_i - T_{i-1}). \quad (3.26) \]

At the same time, from

\[ q(T_i) = \sum_{l=1}^{i} C_l (S_l - S_i) - S_i \hat{\tau}, \quad (3.27) \]

\[ q(T_{i-1}) = \sum_{l=1}^{i-1} C_l (S_{l-1} - S_i) - S_{i-1} \hat{\tau}, \quad (3.28) \]

we have

\[ q(T_i) - q(T_{i-1}) = \sum_{l=1}^{i} C_l (S_l - S_i) - S_i \hat{\tau} - \sum_{l=1}^{i-1} C_l (S_{l-1} - S_i) - S_{i-1} \hat{\tau} \]

\[ = \sum_{l=1}^{i-1} C_l (S_l - S_{l-1}) - (S_i - S_{i-1}) \hat{\tau} \]

\[ = \left( \sum_{l=1}^{i-1} C_l - \hat{\tau} \right) (S_i - S_{i-1}). \quad (3.29) \]

Comparing (3.26) and (3.29), we obtain

\[ T_{i-1} - T_i = S_i - S_{i-1}, \quad (3.30) \]

\[ T_{i-1} + S_{i-1} = T_i + S_i. \quad (3.31) \]

Therefore, the time for the congestion of the network to be clear is

\[ S_{K*} + T_{K*} = T_K + S_K, \]

\[ = \frac{\psi + \sum_{l=1}^{K} C_l S_K}{\sum_{l=1}^{K} C_l - \hat{\tau}} \]

\[ = \psi. \quad (3.32) \]
3.4 Extension to Large Networks with Single-Destination

In the above section, we have proved that $\psi$ is the congestion clear time of the simple network. For a large network, in which many queues are formed at nodes by the congested traffic, we can extend the concept of $\psi$ to every node in the network. Denote the average cost of the $n$th node as $\psi_n$, which can be thought of as the average travel time from node $n$ to the destination. Similar to Eqn.(3.18), we can calculate $\psi_n$ as

$$\psi_n = \frac{\sum_{j=1}^{B_n} q^j_n + \sum_{l=1}^{L^n} C_l S_l}{\sum_{l=1}^{L^n} C_l - \sum_{j=1}^{B_n} q^j_n}.$$  \hfill (3.33)

and the cost to go the destination of link $l$

$$S_l = t_l + \psi_n \quad l \in L^i_n .$$  \hfill (3.34)

From these two equations, we can find that in this routing strategy the cost-to-go of every link is calculated based on the average cost of its downstream node and every link sends its cost-to-go to its upstream node to calculate the average cost. There are communications between nodes. Therefore, this routing strategy which is suggested originally in [17] by Sarachik and Ozguner is partially decentralized. It was also shown in [17] that because the cost-to-go of a link is the sum of the traveling time of the link and the cost of its downstream node, the cost-to-go of a loop should be higher than that without the loop. This routing strategy will not direct traffic to a loop.
3.5 Extension to Networks with Multi-destination

For the multi-destination routing problem, the performance Eqn. (3.4) can be rewritten as follows:

\[
J\{u(t)\} = \int_0^\infty \sum_{d=1}^D \left[ \sum_{j=1}^{B_n} q_n^{jd}(t) + \sum_{i=1}^L t_i u_i^d \right] dt,
\]

\[
= \sum_{d=1}^D \int_0^\infty \left[ \sum_{j=1}^{B_n} q_n^{jd}(t) + \sum_{i=1}^L t_i u_i^d \right] dt,
\]

\[
= \sum_{d=1}^D J^d,
\]

where \( J^d = \int_0^\infty \left[ \sum_{j=1}^{B_n} q_n^{jd}(t) + \sum_{i=1}^L t_i u_i^d \right] dt. \)

Now, we will use the single destination routing strategy discussed in the above section to minimize \( J^d \) and then minimize the whole \( J \). However, for this multi-destination routing problem, we can not use traffic flow as the routing unit any more, since vehicles may have different destinations. So, in our multi-destination routing strategy, vehicle is the routing unit, i.e., every vehicle chooses its route based on the cost-to-go of its destination. It can be expressed as follows:

If we denote the cost of the \( l \)th link to the \( d \)th destination is \( S_l^d \) and assume \( S_1^d < S_2^d < \cdots < S_L^d \) and the vehicle at the \( j \)th direction of node \( n \) time \( t \) has the destination \( d \) and with the right to pass the node, it will

\[
\left\{ \begin{array}{ll}
\text{enter link } l & \text{if } \sum_{j=1}^{B_n} q_n^j(t) > \sum_{i=1}^{l-1} C_i (S_l^d - S_i^d) - C_l \sum_{j=1}^{B_n} r_j^i, \\
& u_i(t) = C_i, \quad i = 1, \cdots, l - 1, \\
& \text{and } u_i(t) < C_l \\
\text{stay at node } n & \text{if } \sum_{j=1}^{B_n} q_n^j(t) \leq \sum_{i=1}^{l-1} C_i (S_l^d - S_i^d) - C_l \sum_{j=1}^{B_n} r_j^i, \\
& u_i(t) = C_i, \quad i = 1, \cdots, l \\
\end{array} \right.
\]

for \( l = 1, 2, \cdots, L \),

(3.36)
where $u_I(t)$ is not the rate of traffic flow but the number of cars entering the link $l$ at time $t$, then the corresponding traffic of link $l$ becomes

$$u_I(t) = \begin{cases} 
  u_I(t) + 1 & \text{if the car enters link } l \\
  u_I(t) & \text{if the car stays at node } n
\end{cases}
$$

for $l = 1, 2, \ldots, L$. (3.37)

$S^d_I$ is the cost-to-go of link $l$ to destination $d$, which can be calculated by

$$S^d_I = t_l + \psi^d_n .$$

and $\psi^d_n$ is the average cost of node $n$ to destination $d$, which is calculated as

$$\psi^d_n = \frac{\sum_{j=1}^{B_n} q^j_i + \sum_{l=1}^{L^d_{n}} C_l S^d_l}{\sum_{l=1}^{L^d_{n}} C_l - \sum_{j=1}^{B_n} r^j_n} ,$$

(3.39)

where $L^d_{n}$ denotes the links leaving from node $n$ directly and indirectly to the destination $d$.

### 3.6 Extension to Networks with Traffic Lights

In the decentralized dynamic routing strategy given above, the delay caused by traffic lights is not considered. However, in local traffic networks, especially in downtown areas, the delay caused by traffic lights, as one of the main reasons for traffic delay, can not be neglected. Now, if we assume the control of traffic lights is known, we can estimate the delay of every route caused by traffic lights. If we denote the delay of the $l$th link caused by traffic lights as $\tau_l$, the cost-to-go of the $l$th link to the $d$th destination $\bar{S}^d_l$ can be modified as

$$\bar{S}^d_l = S^d_l + \tau_l .$$

(3.40)
Then by modifying the definition of cost-to-go of every link, we include the delay caused by traffic lights in our decentralized dynamic routing strategy.

3.7 Simulation Studies

To illustrate the behavior of the dynamic routing strategy, the simulation of two networks, one of which has single destination and the other has two destinations, are first given. Then we simulate the morning and afternoon rush hour around the campus of The Ohio State University in Columbus, in which traffic lights are included.

3.7.1 A Single Destination Case

For the network shown in Figure 3.2, the parameters were chosen to be $C_1 = \ldots$

![Figure 3.2: A single destination network](image)

$C_2 = C_4 = C_6 = C_7 = C_8 = C_{11} = 2, C_5 = C_9 = C_{10} = 1, C_3 = 3, C_{12} = 4$ and $\tau_1 = \tau_3 = \tau_9 = \tau_{11} = 1, \tau_2 = \tau_4 = \tau_5 = \tau_6 = \tau_{10} = \tau_{12} = 2, \tau_7 = \tau_8 = 3$. Traffic enters the network through node 1, 2 and 3. $r_1(t)$ is a random number uniformly distributed
Figure 3.3: Simulation results of the single destination network
in the range between 0 to 4; \( r_2(t) \) is distributed in the range between 0 to 6; and \( r_3(t) \) is distributed in the range between 0 to 1. Since we assume every node only has one queue in this simulation, there are total six queues in this network. Their initial queue lengths are \( q_1(0) = 20, q_2(0) = 25, q_3(0) = 15, q_4(0) = 20, q_5(0) = 20 \) and \( q_6(0) = 0 \). The simulation results of the queue lengths are shown in Figure 3.3. As we can see, under this decentralized dynamic routing strategy the queue lengths in the network decrease when time increases. At the end of the simulation, all the queues are less than two. The network is not congested any more.

### 3.7.2 A Multi-Destination Case

For the network shown in Figure 3.4, the parameters were chosen to be \( C_1 = \)

![Diagram of a multi-destination network](image)

\( C_2 = C_4 = C_6 = C_7 = C_8 = C_{11} = C_{13} = 2, C_5 = C_9 = C_{10} = C_{12} = C_{14} = 1, C_3 = 3, \) and \( \tau_1 = \tau_3 = \tau_9 = \tau_{11} = 1, \tau_2 = \tau_4 = \tau_5 = \tau_6 = \tau_{10} = \tau_{12} = \tau_{13} = \tau_{14} = 2, \tau_7 = \tau_8 = 3. \) The amount of the traffic from outside of the network is the same as
Figure 3.5: Simulation results of multi-destination network
that in the single-destination simulation. Since we also assume every node only has one queue in this simulation, there are total seven queues in this network. The initial queue lengths are \( q_1(0) = 20, q_2(0) = 25, q_3(0) = 15, q_4(0) = 20, q_5(0) = 20, q_6(0) = 0 \) and \( q_7(0) = 0 \). The destination of a vehicle is randomly given, when it enters the network. The queue lengths under our multi-destination routing strategy are shown in Figure 3.3. As can be seen, all of the queue lengths decrease, which means our routing strategy can alleviate the congestion of traffic networks.

### 3.7.3 The Ohio State University Traffic Simulation

Using the strategy in Section 3.6, we simulated the rush hour traffic on the campus of The Ohio State University at Columbus in the morning and afternoon. Figure 3.6 is the simplified map of the OSU campus in Columbus. In this map, 315 is a North-South highway, which has two ramps near campus. High street is a N-S main road running through the campus. And Summit is a one-way street near the campus. Here node 1 and 2 represent two main living areas, called uppertown and Upper Arlington. In the morning, node 1 and 2 have large incoming traffic from outside the network. They are the source nodes. Node 7 represents the OSU campus and Node 13 denotes downtown Columbus. They are the destination nodes. In the afternoon, the traffic condition is the opposite: node 1 and 2 are destinations, and node 7 and 13 are sources. As can be seen, there are total 13 nodes in the map, all of which except node 1, 2, 10 and 13 have traffic lights. The reason why we choose to simulate the morning and afternoon rush hours around the OSU campus is that they are typical traffic situations in big cities.

**Morning Rush Hour** In this case, some of the queues have nonzero initial values,
Figure 3.6: Illustration of the simplified OSU map
which are $q_1(0) = 20$, $q_2(0) = 20$, $q_3(0) = 20$, $q_4(0) = q_5(0) = q_6(0) = q_7(0) = q_{11}(0) = q_{12}(0) = q_{12}(0) = 10$, $q_3(0) = q_8(0) = q_9(0) = q_{10}(0) = q_{13}(0) = 5$. There are traffic running into the network through node 1 and 2, whose mean values are given as $r_1(t) = \begin{cases} 7 & t \leq 400 \\ 6 & 400 < t \leq 600 \end{cases}$ and $r_2(t) = \begin{cases} 3 & t \leq 400 \\ 1 & 400 < t \leq 600 \\ 0 & t > 600 \end{cases}$. Destinations of vehicles are randomly given. The ratio between the number of vehicles with destination 7 and 13 is $\frac{1}{4}$.

![Figure 3.7](image)

Figure 3.7: Simulation results for morning rush hour ("-" with routing, "..." without routing)

Figure 3.7 shows the simulated queue lengths in the OSU morning rush hour, in which the solid line represented the routing with the consideration of delay caused by
<table>
<thead>
<tr>
<th>Destination</th>
<th>Number of Vehicles Reaching the Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Routing</td>
</tr>
<tr>
<td>Node 7 (OSU Compus)</td>
<td>1615</td>
</tr>
<tr>
<td>Node 13 (Downtown Columbus)</td>
<td>6780</td>
</tr>
<tr>
<td>Total</td>
<td>8395</td>
</tr>
</tbody>
</table>

Table 3.1: Comparison of the simulation results of morning rush hour with routing and without routing

traffic light, while the dot line shows the results without the routing. Since node 1 and 2 are source nodes and the incoming traffic from outside of the network is given to be larger than the traffic that both nodes can handle during certain period, it is reasonable that their queue lengths are large. Comparing the queue lengths of \( q_3 \), \( q_4 \), and \( q_5 \) with and without the routing, we can find that although the queue length \( q_4 \) without the routing is smaller than that with our decentralized dynamic routing, \( q_3 \) and \( q_5 \) with the routing are much smaller than those without it. Table 3.1 gives the numbers of vehicles reaching their destination during the simulation time when the vehicles in the network is with routing or without routing. As we can see, the total number of vehicles reaching the destinations for the system with routing is much higher than that without routing. Thus, we conclude that our routing strategy can improve the performance of the system.

**Afternoon Rush Hour** In this case, the initial values for the queue lengths are

\[
q_3(0) = q_3(0) = q_5(0) = q_5(0) = q_{10}(0) = q_{12}(0) = 10, \quad q_3(0) = q_3(0) = q_4(0) = q_4(0) = q_5(0) = q_5(0) = 5, \quad q_7(0) = 20 \quad \text{and} \quad q_1(0) = 25.
\]

The mean values of traffic flow from outside of the network is \( r_1(t) = \begin{cases} 3 & t \leq 600 \\ 1 & t > 600 \end{cases} \) and \( r_{13}(t) = \begin{cases} 4 & t \leq 600 \\ 2 & t > 600 \end{cases} \). The ratio between the number of vehicles to node 1 and 2 is \( \frac{1}{4} \). The simulation results
Figure 3.8: Simulation results for afternoon rush hour (— with routing, … without routing)

<table>
<thead>
<tr>
<th>Destination</th>
<th>Number of Vehicles Reaching the Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Routing</td>
</tr>
<tr>
<td>Node 1 (Uppertown)</td>
<td>7452</td>
</tr>
<tr>
<td>Node 2 (Upper Arlington)</td>
<td>1532</td>
</tr>
<tr>
<td>Total</td>
<td>8984</td>
</tr>
</tbody>
</table>

Table 3.2: Comparison of the simulation results of afternoon rush hour with routing and without routing
of afternoon rush hour are illustrated in Figure 3.8 and Table 3.2. As can be seen, these results are similar to those of the morning rush hour. Therefore, by analyzing the simulation results, we can conclude that our decentralized routing strategy can improve the performance of a congested traffic system.

3.8 Summary

In this chapter, we presented a multi-destination decentralized dynamics routing strategy for congested local traffic networks based on the point-queuing model given in Chapter 2. We first provided a routing method for a simple network with single destination, then extended it to complex networks with single and multi-destination. To consider the delay caused by traffic lights, we modify the calculation of cost-to-go. Three simulation examples conclude this chapter and show that this decentralized dynamics routing strategy can improve the traffic conditions of networks.

In the next chapter, we will discuss another important traffic problem, intersection control.
DECENTRALIZED HYBRID INTERSECTION CONTROL

After discussing the routing strategy for local traffic networks in Chapter 3, we now analyze another important control and optimization problem in local traffic networks: intersection control. In this chapter, we will first propose a hybrid intersection controller for an isolated intersection, then apply the control design method to the multi-intersection case and develop a decentralized hybrid intersection controller.

4.1 Model of an Isolated Intersection

Consider an isolated intersection with one way roads along x and y directions respectively, which is shown in Figure 4.1. Based on the point-queuing model, the dynamics of queues in x and y direction at the intersection can be described as

\[
\begin{align*}
\dot{q}_x(t) &= u^i_x(t) - u^o_x(t) & \text{when the light in x direction is Green} \\
\dot{q}_y(t) &= u^i_y(t) \\
\dot{q}_x(t) &= u^i_x(t) \\
\dot{q}_y(t) &= u^i_y(t) - u^o_y(t) & \text{when the light in y direction is Green}
\end{align*}
\]

(4.1)

where \( u^i_x \) and \( u^i_y \) are traffic coming to the intersection from x and y direction respectively, and \( u^o_x \) and \( u^o_y \) are traffic leaving the intersection.

If the cycle time, phase and offset of the traffic light are fixed, and the individual dynamics of vehicles are not considered, the only control variable \( c \) is the split rate
between the effective green times at x and y direction, which can be changed only at the beginning of every cycle time. Assuming the lost time per cycle is fixed as $T_l$, we simplify the model as

$$\dot{q}_x(t) = u^i_x(t) - sat_x(q_x(t) + u^i_x(t))g_x(c, t),$$

$$\dot{q}_y(t) = u^i_y(t) - sat_y(q_y(t) + u^i_y(t))g_y(c, t),$$

where $g_x$ and $g_y$ represent whether the traffic light at x or y direction is Green, i.e.,

$$g_x(c, t) = \begin{cases} 1 & 0 \leq \text{mod}(t - T) \leq c\left(\left\lfloor \frac{T}{T_l} \right\rfloor \right)(T - T_l) \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_y(c, t) = \begin{cases} 1 & c\left(\left\lfloor \frac{T}{T_l} \right\rfloor \right)(T - T_l) \leq \text{mod}(t - T) < T - \frac{T_l}{2} \\ 0 & \text{otherwise} \end{cases}.$$
respectively, in which $u^m_x$ and $u^m_y$ are the traffic capacities in $x$ and $y$ direction.

When traffic demand approaching an intersection is larger than its capacity, i.e., traffic in $x$ and $y$ direction can not be cleared in one cycle time, the intersection is saturated. The dynamics of a saturated intersection can be simplified further from Eqn. (4.2) and (4.3), which are

$$
\dot{q}_x = u_x - \frac{T - T_i}{T} c \left\lfloor \frac{t}{T} \right\rfloor u^m_x,
$$

$$
\dot{q}_y = u_y - \frac{T - T_i}{T} \left(1 - c \left\lfloor \frac{t}{T} \right\rfloor \right) u^m_y.
$$

Let $\gamma = \frac{T - T_i}{T}$ and the above equations become

$$
\dot{q}_x = u_x - c \left\lfloor \frac{t}{T} \right\rfloor \gamma u^m_x,
$$

$$
\dot{q}_y = u_y - \left(1 - c \left\lfloor \frac{t}{T} \right\rfloor \right) \gamma u^m_y.
$$

This simplified model has been proposed by Gazis [26]. Its most important contribution is making $c$ the input variable of the system dynamics so that the optimal control method can be applied.

### 4.2 Control for an Isolated Intersection

For an isolated intersection, two controllers are designed based on the two different traffic situations at the intersection: saturated or unsaturated.

#### 4.2.1 Saturated Intersection

When an intersection is saturated, the traffic control can be modeled as an optimal control problem since the control variable $c$ is an input of the system in the simplified dynamics Eqn. (4.6) and (4.7).
In [42], the Maximum principle is used to solve this saturated intersection control problem, in which an optimization criterion is defined as

$$\min_c J = \int_0^\infty (q_x(t) + q_y(t))dt ,$$

and a constraint of the control variable $c$ is given as follows,

$$c_{\text{min}} \leq c \leq c_{\text{max}} .$$

This criterion is to minimize the queue lengths at $x$ and $y$ direction, which is equivalent to maximizing the throughput of the intersection. Since the split rate $c$ can only be changed once every cycle and is limited between the two constants, the result of this optimal problem is a “bang-bang” controller, i.e., the direction with higher capacity will obtain the maximum green time, i.e., $c_{\text{max}}(T - T_i)$ or $(1 - c_{\text{min}}(T - T_i))$, and the other direction will receive the minimum green time, $c_{\text{min}}(T - T_i)$ or $(1 - c_{\text{max}}(T - T_i))$; thereby, achieving the highest throughput. However, this is unfair for the traffic in the direction with smaller capacity since it has to wait for longer time.

To include some amount of fairness, the term $|q_x(t) - q_y(t)|$ has to be added to the optimization criterion. With the term inherited from Eqn. (4.8), we have

$$\min_c J = \int_0^T [(q_x(t) + q_y(t)) + |q_x(t) - q_y(t)|]dt$$

The first term of the criterion represents the total number of vehicles queuing at the intersection to achieve a high throughput. The second term is the queue length difference between $x$ and $y$ direction to consider the fairness between the traffic waiting at the two directions. Notice that we choose the cycle time of the traffic light as our optimization interval. This means that we try to find a split rate to minimize the queue lengths and their difference in one traffic light cycle.
Since the square function is easier to be handled mathematically than the absolute function and it also can keep the value of \( q_x(t) - q_y(t) \) positive, our criterion changes to

\[
\min_c J = \int_0^T ((q_x(t) + q_y(t))^2 + (q_x(t) - q_y(t))^2) dt = 2 \int_0^T (q_x^2(t) + q_y^2(t)) dt . 
\] (4.11)

Now, because the value of split rate \( c \) is fixed in a cycle time and only can be changed when a cycle time is complete, this optimal control problem becomes an optimization problem. To solve it, we need to discretize the system first. If we choose the sample time as \( \Delta t \), the dynamics of the intersection and the criterion can be rewritten as

\[
\min_{c(0)} J = \frac{1}{2} \sum_{k=0}^K (q_x^2(k) + q_y^2(k)) 
\] (4.12)

and

\[
q_x(k+1) = q_x(k) + u_x^i(k) - c(0)\gamma u_x^m, \quad (4.13)
\]
\[
q_y(k+1) = q_y(k) + u_y^i(k) - (1 - c(0))\gamma u_y^m, \quad (4.14)
\]

where \( c(0) \) is the variable to be decided and \( c_{min} \leq c(0) \leq c_{max} \). \( T = K\Delta t \) and \( 0 \leq k \leq K \).

Assuming the traffic flow rates coming to the intersection from \( x \) and \( y \) direction are random variables and their mean values in the next cycle time can be estimated, which are denoted as \( \hat{u}_x \) and \( \hat{u}_y \), we rewrite the dynamics of the intersection as

\[
\begin{bmatrix}
q_x(k+1) \\
q_y(k+1)
\end{bmatrix} =
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
q_x(k) \\
q_y(k)
\end{bmatrix} +
\begin{bmatrix}
-\gamma u_x^m \\
\gamma u_y^m
\end{bmatrix} c(0) +
\begin{bmatrix}
\hat{u}_x \\
\hat{u}_y - \gamma u_y^m
\end{bmatrix} . 
\] (4.15)

Let \( X(k) = \begin{bmatrix} q_x(k) \\ q_y(k) \end{bmatrix} \), \( A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \), \( B = \begin{bmatrix} -\gamma u_x^m \\ \gamma u_y^m \end{bmatrix} \), \( W = \begin{bmatrix} \hat{u}_x \\ \hat{u}_y - \gamma u_y^m \end{bmatrix} \), Eqn. (4.15) becomes

\[
X(k+1) = AX(k) + Bc(0) + W . 
\] (4.16)
Assume the value of $X$ at $k = 0$, i.e., $X(0)$ is known, then

$$X(k) = A^k X(0) + (A^{k-1} + \cdots + I) B c(0) + (A^{k-1} + \cdots + I) W .$$ \hspace{1cm} (4.17)

Substituting this equation into Eqn. (4.12), we obtain that

$$
\begin{align*}
\min_{c(0)} J &= \frac{1}{2} X(0)^T Q X(0) \\
&+ \frac{1}{2} \sum_{k=1}^{K} (A^k X(0) + (A^{k-1} + \cdots + I) B c(0) + (A^{k-1} + \cdots + I) W)^T Q (A^k X(0) + (A^{k-1} + \cdots + I) B c(0) + (A^{k-1} + \cdots + I) W) \, .
\end{align*}
$$ \hspace{1cm} (4.18)

Then set $\frac{dJ}{dc(0)} = 0$, i.e.,

$$
\begin{align*}
\frac{dJ}{dc(0)} &= \sum_{k=1}^{K} (A^{k-1} B + \cdots + B)^T Q (A^{k-1} B + \cdots + B) c(0) \\
&+ \sum_{k=1}^{K} (A^{k-1} B + \cdots + B)^T Q (A^k X(0) + (A^{k-1} + \cdots + I) W) \, ,
\end{align*}
$$ \hspace{1cm} (4.19)

We obtain the split rate which minimizes the cost $J$ as

$$\tilde{c}(0) = - \left[ \sum_{k=1}^{K} (A^{k-1} B + \cdots + B)^T Q (A^{k-1} B + \cdots + B) \right]^{-1} \left[ \sum_{k=1}^{K} (A^{k-1} B + \cdots + B)^T Q (A^k X(0) + (A^{k-1} + \cdots + I) W) \right] .$$ \hspace{1cm} (4.20)

Considering the constraints on $c$, we have

$$c(0) = \begin{cases} 
\tilde{c}(0) < c_{\text{min}} \\
\tilde{c}(0) \leq c(0) \leq c_{\text{max}} \\
\tilde{c}(0) > c_{\text{max}}
\end{cases} .$$ \hspace{1cm} (4.21)

In this control method, the queue lengths at time 0 and the mean value of the arrival traffic are needed to calculate $\tilde{c}(0)$. The first term in Eqn. (4.20) is a constant since it is only related to $A$, $B$ and $Q$. Here, $Q$ is a weighted matrix chosen by control engineers, and $A$ and $B$ represent the characteristics of the intersection, which do not change in a relative long time. For an isolated intersection, matrix $A$ actually is an identity. Notice that the matrix will include the capacity parameters of intersections.
In this section, we have discussed the signal control of a saturated intersection in which traffic cannot be cleared in one cycle time. This includes the case that traffic can be cleared over multiple cycles. How to handle the change between the saturated and unsaturated state of an intersection will be discussed later. In the next section, we will analyze the unsaturated intersection control.

4.2.2 Unsaturated Intersection

When an intersection is not saturated, the traffic situation at the intersection is complex and time-varying. We cannot model the problem as an optimal control problem any more. A simple solution for the unsaturated intersection control will be explained in the following.

Assume that the queues at the intersection could be cleared in a cycle time with the maximum flow capacities in $x$-direction and $y$-direction, $u_x^m$ and $u_y^m$, i.e.,

\[
q_x(0) + \int_0^T u_x^i(t)dt - \int_0^{c(T-T_l)} u_x^m dt = 0 ,
\]

\[
q_y(0) + \int_0^T u_y^i(t)dt - \int_{c(T-T_l)+\frac{T_l}{2}}^{T-T_l} u_y^m dt = 0 .
\]

Since $T - T_l = \gamma T$, the above equations become

\[
q_x(0) + \int_0^T u_x^i(t)dt = c\gamma T u_x^m ,
\]

\[
q_y(0) + \int_0^T u_y^i(t)dt = (1-c)\gamma T u_y^m .
\]

Dividing Eqn. (4.24) by (4.25), we obtain

\[
\frac{cu_x^m}{(1-c)u_y^m} = \frac{q_x(0) + \int_0^T u_x^i(t)dt}{q_y(0) + \int_0^T u_y^i(t)dt} .
\]
Then
\[ \bar{c}(0) = \frac{(q_x(0) + \int_0^T u_x(t)dt)u_y^m}{(q_x(0) + \int_0^T u_x(t)dt)u_y^m + (q_y(0) + \int_0^T u_y(t)dt)u_x^m}. \]  

(4.27)

If only the mean values of arrival can be estimated, the solution can be modified as
\[ \bar{c}(0) = \frac{(q_x(0) + T\hat{u}_x)u_y^m}{(q_x(0) + T\hat{u}_x)u_y^m + (q_y(0) + T\hat{u}_y)u_x^m}. \]

(4.28)

where \( \hat{u}_x \) and \( \hat{u}_y \) are the estimated average vehicle arrival rates at \( x \) and \( y \) direction.

Considering the constraints on \( c \), we have the control of an isolated unsaturated intersection as
\[ c(0) = \begin{cases} 
\ c_{\min} & \bar{c}(0) < c_{\min} \\
\ \bar{c}(0) & c_{\min} \leq \bar{c}(0) \leq c_{\max} \\
\ c_{\max} & \bar{c}(0) > c_{\max} 
\end{cases} \]

(4.29)

4.2.3 Hybrid Control

For an isolated intersection, the hybrid control is simple because it only has two states: one for the saturated and the other for the unsaturated situation. There is a problem: how to decide an intersection is saturated or unsaturated. Here, according to our unsaturated intersection control, we define
\[ \omega = \frac{q_x(0) + T\hat{u}_x}{Tu_x^m} + \frac{q_y(0) + T\hat{u}_y}{Tu_y^m}, \]

(4.30)

and
\[ \begin{cases} 
\omega \leq 1 & \text{Unsaturated} \\
\omega > 1 & \text{Saturated} 
\end{cases}. \]

(4.31)

Then the finite state machine defined for the isolated intersection control is shown in Figure 4.2.

As we know, the isolated intersection control problem is the simplest one in the intersection control problems since there is no coupling and delay between intersections. In the following, the multi-intersection control problem will be discussed.
4.3 Multi-intersection Control

In this section, using the same analysis approach of isolated intersections, we will first discuss the saturated and unsaturated multi-intersection control, and then propose a hybrid control method to include both of them.

In the multi-intersection control, the two-intersection control problem is the simplest case. Now, we are going to analyze this two-intersection control.

Consider a network with two intersections of one-way roads shown in Figure 4.3. Assume that the travel time between the two intersections is $\tau$. We choose the synchronization time difference between the two intersections is a constant called $\zeta$. To keep the synchronization between the two intersections fixed, the cycle times of the two intersections have to be the same, i.e., $T_1 = T_2 = T$. In the following discussion, we do not allow turning traffic in the intersections or entering/exiting traffic between the intersections.
4.3.1 Saturated Intersection Control

When both of the intersections are saturated, the dynamics are

\[ q_{x1} = u_{x1} - \gamma c_1 \left( \left\lceil \frac{t}{T} \right\rceil T \right) u_{x1}^m, \]  \hspace{1cm} (4.32)\]

\[ q_{y1} = u_{y1} - \gamma \left( 1 - c_1 \left( \left\lceil \frac{t}{T} \right\rceil T \right) \right) u_{y1}^m, \]  \hspace{1cm} (4.33)\]

\[ q_{x2} = \gamma c_1 (t - \tau) u_{x1}^m - \gamma c_2 \left( \left\lceil \frac{t + T - \zeta}{T} \right\rceil T \right) u_{x2}^m, \]  \hspace{1cm} (4.34)\]

\[ q_{y2} = u_{y2} - \gamma \left( 1 - c_2 \left( \left\lceil \frac{t + T - \zeta}{T} \right\rceil T \right) \right) u_{y2}^m, \]  \hspace{1cm} (4.35)\]

where \( q_{x1}, q_{y1}, q_{x2} \) and \( q_{y2} \) are the queue lengths at \( x \) and \( y \) direction of intersection 1 and 2 respectively, the split rates of two intersections are \( c_1 \) and \( c_2 \), and \( u_{x1}^m, u_{y1}^m, u_{x2}^m \) and \( u_{y2}^m \) are the capacities of intersections at \( x \) and \( y \) directions.
Then the discretized dynamics are expressed as

\[ q_x(k+1) = q_x(k) + u_{x_1}^i(k) - \gamma c_1 \left( \left[ \frac{k}{K} \right] K \right) u_x^m, \]
\[ q_y(k+1) = q_y(k) + u_{y_1}^i(k) - \gamma \left( 1 - c_1 \left( \left[ \frac{k}{K} \right] K \right) \right) u_y^m, \]
\[ q_{x_2}(k+1) = q_{x_2}(k) + \gamma c_1(k-n)u_{x_1}^m - \gamma c_2 \left( \left[ \frac{k+K-j}{K} \right] K \right) u_{x_2}^m, \]
\[ q_{y_2}(k+1) = q_{y_2}(k) + \gamma c_1(k-n)u_{y_1}^m - \gamma c_2 \left( \left[ \frac{k+K-j}{K} \right] K \right) u_{y_2}^m. \]

(4.36) (4.37) (4.38) (4.39)

where \( T = K \Delta t, \tau = n \Delta t \) and \( \zeta = j \Delta t \).

As we know, there is a synchronization time offset between the two intersections, which means that the split rates of the two intersections are decided at different time instances. A decentralized control method will be applied to exploit these characteristics.

Here, we use the “over-lapped decomposition" method to decompose the system. For the intersection 1, due to the coupling between the two intersections, effects of its control on the intersection 2 need to be considered. So the subsystem for the intersection 1 has to include the dynamics from the intersection 2, i.e., the dynamics for the intersection 1 subsystem are

\[ q_x(k+1) = q_x(k) + u_{x_1}^i(k) - c_1 \left( \left[ \frac{k}{K} \right] K \right) \gamma u_x^m, \]
\[ q_y(k+1) = q_y(k) + u_{y_1}^i(k) - \left( 1 - c_1 \left( \left[ \frac{k}{K} \right] K \right) \right) \gamma u_y^m, \]
\[ q_{x_2}(k+1) = q_{x_2}(k) + c_1(k-n)\gamma u_{x_1}^m - c_2 \left( \left[ \frac{k+K-j}{K} \right] K \right) \gamma u_{x_2}^m. \]

(4.40) (4.41) (4.42)

Over-lapped decomposition is an approach to decompose a large scale system into several subsystems such that decentralized control method can be used. The mathematical framework for decentralized control using overlapping decomposition is the Inclusion Principle. Here, overlapping subsystems are expanded into larger state space and control space, such that the subsystems appear to be decoupled in state, input and output. Using certain contraction conditions, the subsystems and controllers are contracted to the original space for implementation, and if all of the appropriate conditions are met, the closed loop properties that were derived for the expanded decoupled subsystems are intact when the system is contracted.
Notice that the dynamics are similar to those of an isolated intersection, except the delay term. Usually, delay is difficult to be handled in control problems. However in discrete-time, we can define new states to avoid this problem. Define \( x_1(k) = q_{x1}(k) \), \( x_2(k) = q_{y1}(k) \), \( x_3(k) = q_{x2}(k) \), and \( x_4(k) = c_1(k - 1) \), \( x_5(k) = x_4(k - 1) = c_1(k - 2) \), \( \ldots \), \( x_{n+3} = c_1(k - n) \). Since the value of \( c_2 \left( \left\lfloor \frac{k+K-1}{K} \right\rfloor \right) \) is unknown, we use the current intersection 2 control, denoted as \( \hat{c}_2 \), to substitute it. The dynamics of the intersection 1 subsystem are written as

\[
X(k + 1) = \begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & \gamma u^m_{x1} \\
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1
\end{bmatrix}X(k) + \begin{bmatrix}
-\gamma u^m_{x1} \\
\gamma u^m_{y1} \\
0 \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix} + \begin{bmatrix}
\hat{u}_{x1} \\
\hat{u}_{y1} - \gamma u^m_{y1} \\
-\hat{c}_2 \gamma u^m_{x2} \\
\hat{c}_1(0) + \hat{u}_{x2} \\
\hat{c}_1(0) - \gamma u^m_{y1} \\
-\hat{c}_2 \gamma u^m_{x2}
\end{bmatrix},
\]

Defining \( A = \begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & \gamma u^m_{x1} \\
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 1
\end{bmatrix} \), \( B = \begin{bmatrix}
-\gamma u^m_{x1} \\
\gamma u^m_{y1} \\
0 \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix} \), \( W = \begin{bmatrix}
\hat{u}_{x1} \\
\hat{u}_{y1} - \gamma u^m_{y1} \\
-\hat{c}_2 \gamma u^m_{x2} \\
\hat{c}_1(0) + \hat{u}_{x2} \\
\hat{c}_1(0) - \gamma u^m_{y1} \\
-\hat{c}_2 \gamma u^m_{x2}
\end{bmatrix} \),

we have

\[
X(k + 1) = AX(k) + Bc_1(0) + W,
\]

whose format is the same as that of the isolated saturated intersection. Therefore, the isolated saturated intersection control method, i.e. Eqn. (4.20) and (4.21), can be used.

In the above, we extend the saturated control method to multi-intersection. However, there is a problem which does not exist in isolated intersection but has to be
considered in multi-saturated-intersection control analysis: the queuing vehicles of intersection 2 overrun intersection 1. This means that the queue length between intersection 1 and 2 is constrained by a maximum value, denoted as \( q_{z2}^m \). When \( q_{x2} = q_{z2}^m \), there is no more space for vehicles to enter and the method proposed above is not valid any more. Under this condition, we should give the \( y \) direction the largest green time to achieve a bigger throughput, i.e.,

\[
c_1(0) = c_{\text{min}}, \quad \text{if} \quad q_{x2} = q_{z2}^m .
\]  

For the saturated intersection 2, since there are no other intersections in its downstream, its control will not affect other intersection's performance. So it can be considered as an isolated intersection and the saturated isolated intersection control method is used.

For a saturated intersection with one-way streets, a more complex situation is to consider effects of its control on both \( x \) and \( y \) downstream intersections. When all the three intersections are saturated, the intersection subsystem has to include the dynamics of the two downstream intersections. The control method for two intersections still can be used, except that the subsystem is bigger. The order of the subsystem dynamics is \( 4 + \max(n, m) \), where \( n\Delta t \) and \( m\Delta t \) are the travel times to the \( x \) and \( y \) direction downstream intersections.

### 4.3.2 Unsaturated Two-intersection Control

When the intersection under our analysis, which has downstream and upstream intersections, is unsaturated, the situation is different: the effects of its intersection control on the downstream intersections are not as important as when the intersections are saturated; but the traffic from its upstream intersections has to be
estimated because the coming traffic is deterministic due to the upstream intersection control.

For intersection 1 in Figure 4.3, we can use the isolated unsaturated intersection control because there is no upstream intersection. But for intersection 2, if the control and traffic conditions of intersection 1 are available, we can estimate the coming traffic of intersection 2 during the next cycle time. Based on the estimated traffic information, the unsaturated intersection 2 control is decided.

As we know, for an unsaturated intersection, when the traffic signal turns green, traffic waiting in the queue will leave the intersection at the maximum flow rate. When the traffic in the queue is cleared, the traffic rate passing through the intersection is equal to the incoming traffic rate. Now, assume that at time \( t_0 \) intersection 2 finishes its cycle and needs to decide the split rate for next cycle time \( c_2(t_0) \), and it is unsaturated. Then, at time \( t_0 - \zeta \) intersection 1 began a new cycle with a split rate \( \bar{c}_1 = c_1(t_0 - \zeta) \). The queue length at its \( x \)-direction at this time is \( q_{x1}(t_0 - \zeta) \).

If intersection 1 is unsaturated, its queue in \( x \) direction is cleared with the maximum flow \( u_{x1}^m \). Assume \( t_c \) is the time interval between the time when the traffic light at \( x \) direction of intersection 1 turns green to the time when the queue is cleared. We have

\[
q_{x1}(t_0 - \zeta) + \hat{u}_{x1} t_c = u_{x1}^m t_c ,
\]

where \( \hat{u}_{x1} \) is the estimated average value of the incoming traffic rate at \( x \) direction of intersection 1. Then

\[
t_c = \frac{q_{x1}(t_0 - \zeta)}{u_{x1}^m - \hat{u}_{x1}} .
\]
Therefore, when intersection 1 is unsaturated, the departing traffic at \( x \) direction

\[
u^x_1(t) = \begin{cases} 
u^m_{x1} & t_0 - \zeta \leq t \leq t_0 - \zeta + t_c \\ \bar{u}_{x1} & t_0 - \zeta + t_c < t \leq t_0 - \zeta + \bar{c}_1(T - T_t) \end{cases}, \tag{4.48}
\]

where \( t_c \leq \bar{c}_1(T - T_t) = \bar{c}_1\gamma T \). Since there is no entering/exiting traffic between intersection 1 and 2, the traffic reaching intersection 2 at \( x \) direction is

\[
u^x_{x2}(t) = u^x_1(t - \tau) = \begin{cases} 
u^m_{x1} & t_0 - \zeta + \tau \leq t \leq t_0 - \zeta + t_c + \tau \\ \bar{u}_{x1} & t_0 - \zeta + t_c + \tau < t \leq t_0 - \zeta + \bar{c}_1\gamma T + \tau \end{cases}, \tag{4.49}
\]

If \( 0 \leq \tau - \zeta \leq T \), the traffic from the queue of intersection 1 will reach intersection 2 during the next traffic light cycle of intersection 2. The total traffic reaching intersection 2 in the \( x \) direction during its next cycle can be estimated as

\[
\int_{t_0}^{t_0 + T} u^i_{x2} \, dt = t_c u^m_{x1} + (\bar{c}_1\gamma T - t_c) \bar{u}_{x1} = t_c (u^m_{x1} - \bar{u}_{x1}) + \bar{c}_1\gamma T \bar{u}_{x1}. \tag{4.50}
\]

If \( 0 < \zeta - \tau < t_c \), some amount of traffic from intersection 1 will reach intersection 2 before the next cycle begins, i.e., the total traffic reaching intersection 2 in the \( x \) direction during its next cycle

\[
\int_{t_0}^{t_0 + T} u^i_{x2} \, dt = (t_c + \tau - \zeta) u^m_{x1} + (\bar{c}_1\gamma T - t_c) \bar{u}_{x1} = t_c (u^m_{x1} - \bar{u}_{x1}) + (\tau - \zeta) u^m_{x1} + \bar{c}_1\gamma T \bar{u}_{x1}. \tag{4.51}
\]

At the same time, since there is no upstream intersection in the \( y \) direction of intersection 2, the total incoming traffic in the \( y \) direction of intersection 2 is

\[
\int_{t_0}^{t_0 + T} u^i_{y2} \, dt = T \hat{u}_{y2} \tag{4.52}
\]

where \( \hat{u}_{y2} \) is the estimated mean traffic flow coming to the \( y \) direction of intersection 2.

Because intersection 2 is unsaturated, based on our definition for unsaturated intersection, Eqn. (4.22) and (4.23) still hold. Using these two equations, we can obtain
our unsaturated intersection control with the consideration of effects of unsaturated upstream intersection $1$, which is

\[
\tilde{c}_2(t_0) = \begin{cases} 
\frac{q_{x2}(t_0) + q_{x1}(t_0 - \zeta) + c_{11} \gamma T \tilde{u}_{x1}}{(q_{x2}(t_0) + q_{x1}(t_0 - \zeta) + c_{11} \gamma T \tilde{u}_{x1})u_{y2}^m + (q_{y2}(0) + T \tilde{u}_{y2})u_{x2}^m} & 0 \leq \tau - \zeta \leq T \\
\frac{q_{x2}(t_0) + q_{x1}(t_0 - \zeta) + c_{11} \gamma T \tilde{u}_{x1} + (\tau - \zeta)u_{x1}^m}{(q_{x2}(t_0) + q_{x1}(t_0 - \zeta) + c_{11} \gamma T \tilde{u}_{x1} + (\tau - \zeta)u_{x1}^m)u_{y2}^m + (q_{y2}(0) + T \tilde{u}_{y2})u_{x2}^m} & 0 < \zeta - \tau < t_c 
\end{cases}
\] (4.53)

When intersection $1$ is saturated, the traffic from intersection $1$ to $2$ is easy to estimated, i.e.

\[
u_{x21}^i(t) = u_{x1}(t - \tau) = u_{x1}^m, \quad t_0 - \zeta + \tau \leq t \leq t_0 - \zeta + \tilde{c}_1 \gamma T + \tau .
\] (4.54)

Then

\[
\int_{t_0}^{t_0 + T} u_{x21}^i(t)dt = u_{x1}^m \tilde{c}_1 \gamma T
\] (4.55)

So with the consideration of effects of saturated upstream intersection $1$, the control of unsaturated intersection $2$ becomes

\[
\tilde{c}_2(t_0) = \frac{(q_{x2}(t_0) + u_{x1}^m \tilde{c}_1 \gamma T)u_{y2}^m}{(q_{x2}(t_0) + u_{x1}^m \tilde{c}_1 \gamma T)u_{y2}^m + (q_{y2}(0) + T \tilde{u}_{y2})u_{x2}^m}
\] (4.56)

Including the constraint of the split rate, we have

\[
c_2(0) = \begin{cases} 
c_{\text{min}} & \tilde{c}_2(t_0) < c_{\text{min}} \\
\tilde{c}_2(t_0) & c_{\text{min}} \leq \tilde{c}_2(t_0) \leq c_{\text{max}} \\
c_{\text{max}} & \tilde{c}_2(t_0) > c_{\text{max}}
\end{cases}
\] (4.57)

If there is an upstream intersection in the $y$ direction of intersection $2$, we can use the same method to estimate the traffic from the $y$ direction and include it in the control.

In the following section, we will combine the saturated and unsaturated multi-intersection control methods together and propose a hybrid multi-intersection controller.
4.3.3 Hybrid Multi-intersection Control

After analyzing the control of two or three intersections, we are ready to explain our control strategy for multi-intersections. First, this control is decentralized. Every intersection decides its split rate itself, based on its traffic condition and the traffic information from its upstream and downstream intersections.

For an intersection of one-way streets, e.g., intersection 0 in Figure 4.4, there are usually two upstream and two downstream intersections. Considering the control of intersection 0, we define a finite state machine with five states, which is illustrated in Figure 4.4: A network with five intersections.
State 1:
Intersection 0 is unsaturated, i.e., \( \omega_0 \leq 1 \). Depending on the traffic situations of its upstream intersections, either Eqn. (4.51) or (4.54) can be used to estimate the traffic from intersection 1 and 2. Then the estimated values can be substituted into Eqn. (4.27) to decide the unsaturated intersection control.

State 2:
Intersection 0 is saturated and its downstream intersections are unsaturated, i.e., \( \omega_0 > 1 \), \( \omega_3 \leq 1 \) and \( \omega_4 \leq 1 \). Since the two downstream intersections are unsaturated, their traffic conditions do not need to be considered. The saturated isolated intersection control (Eqn. (4.20) and (4.21)) is used.

State 3:
Intersection 0 and its \( x \) direction downstream intersection are saturated, i.e., \( \omega_0 > 1 \), \( \omega_3 > 1 \) and \( \omega_4 \leq 1 \). The control for two saturated intersections is used. If the queue length at \( x \) direction of intersection 3 does not overrun intersection 0, the order of the sub-system is \( 3 + n \) and Eqn. (4.20) and (4.21) are used. If does, \( c_0 = c_{\text{min}} \).

**State 4:**

Intersection 0 and its \( y \) direction downstream intersection are saturated, i.e., \( \omega_0 > 1 \), \( \omega_3 \leq 1 \) and \( \omega_4 > 1 \). The control for two saturated intersections is used. If the queue length at \( y \) direction of intersection 4 does not overrun intersection 0, the order of the sub-system is \( 3 + m \) and Eqn. (4.20) and (4.21) are used. If does, \( c_0 = c_{\text{max}} \).

**State 5:**

Intersection 0 and both of its downstream intersections are saturated, i.e., \( \omega_0 > 1 \), \( \omega_3 > 1 \) and \( \omega_4 > 1 \). The traffic conditions of both its downstream intersections need to be considered. If \( q_{x3} \) overruns intersection 0, \( c_0 = c_{\text{min}} \); If \( q_{y4} \) overruns intersection 0, \( c_0 = c_{\text{max}} \); otherwise, Eqn. (4.20) and (4.21) are used, and the order of the system is \( 4 + \max(m, n) \).

## 4.4 Simulation Example

In this section, we will apply the above decentralized hybrid intersection controller to four different intersection topologies.

### 4.4.1 Five Intersections

First, we will simulate the traffic conditions in a network with five intersections shown in Figure 4.4. The travel times between intersections are chosen as \( \tau_{10} = 10s \), \( \tau_{20} = 15s \), \( \tau_{03} = 20s \) and \( \tau_{04} = 15s \). The maximum number of vehicles can be contained between the intersections are \( q_{x0}^m = 50 \text{ veh} \), \( q_{y0}^m = 65 \text{ veh} \), \( q_{x3}^m = 85 \text{ veh} \).
and \( q_{y4}^m = 65 \text{ veh} \). The maximum flow rates of all the intersections at \( x \) and \( y \) directions are chosen to be the same respectively, i.e. \( u_{x0}^m = u_{x1}^m = u_{x2}^m = u_{x3}^m = u_{x4}^m = 1.5 \text{ veh/s} \) and \( u_{y0}^m = u_{y1}^m = u_{y2}^m = u_{y3}^m = u_{y4}^m = 1 \text{ veh/s} \). Traffic from outside of the network is modeled as the random walk process, i.e., the time interval between two arrival vehicles is exponentially distributed with rate \( \lambda \). We select \( \lambda_{x1} = 0.7 \text{ veh/s} \), \( \lambda_{y1} = 0.3 \text{ veh/s} \), \( \lambda_{x2} = 0.7 \text{ veh/s} \), \( \lambda_{y2} = 0.3 \text{ veh/s} \), \( \lambda_{y3} = 0.3 \text{ veh/s} \) and \( \lambda_{x4} = 0.7 \text{ veh/s} \). Assume the synchronization time differences between the traffic lights at the intersection 1 and 0 \( \zeta_{10} = 15s \), 2 and 0 \( \zeta_{20} = 15s \), 0 and 3 \( \zeta_{03} = 20s \), 0 and 4 \( \zeta_{04} = 15s \), and the lost time of every traffic light is the same, i.e., \( T_i = 5s \).

To demonstrate the performance of our decentralized hybrid controller, we set up a fixed time control simulation, in which the fixed split rates of the five intersections are chosen to be the optimal values based on the static traffic flow rates of every intersections, which are \( c_0 = 0.58 \), \( c_1 = 0.625 \), \( c_2 = 0.6087 \), \( c_3 = 0.53 \) and \( c_4 = 0.56 \).

When the cycle time \( T = 30 s \), the simulation results are given in Figure 4.6, where the solid lines indicate the results under the decentralized hybrid control and the dot lines indicate the results under the fixed time control. As can be seen, the performance of the system under the decentralized hybrid control and the fixed time control do not differ by much, although under the fixed time control the queue length at \( x \) direction of intersection 3 is larger than that of the decentralized controller.

Table 4.1 gives the average throughput and mean values of delay of different paths in the network, where the value of the delay of every path only includes the delay caused by traffic lights since the travel times along links are fixed. It is shown that the total values of the throughput and delay of the paths are almost equal. If we compare the paths containing three intersections (path 1 and 2), we can find under
Figure 4.6: Queue lengths of five intersections when $T = 30s$
<table>
<thead>
<tr>
<th>Paths</th>
<th>Decentralized Control</th>
<th>Fixed Time Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Throughput(s⁻¹)</td>
<td>Mean Value of Delay(sec)</td>
</tr>
<tr>
<td>1: x direction of intersection 1,0,3</td>
<td>0.70</td>
<td>23.5</td>
</tr>
<tr>
<td>2: y direction of intersection 2,0,4</td>
<td>0.29</td>
<td>14.9</td>
</tr>
<tr>
<td>3: y direction of intersection 1</td>
<td>0.29</td>
<td>8.9</td>
</tr>
<tr>
<td>4: x direction of intersection 2</td>
<td>0.69</td>
<td>7.5</td>
</tr>
<tr>
<td>5: y direction of intersection 3</td>
<td>0.35</td>
<td>9.0</td>
</tr>
<tr>
<td>6: x direction of intersection 4</td>
<td>0.71</td>
<td>5.3</td>
</tr>
<tr>
<td>Total</td>
<td>3.03</td>
<td>69.1</td>
</tr>
</tbody>
</table>

Table 4.1: Comparison of two intersection control methods in five intersection simulation when T = 30s

the decentralized control the values of delay are close to each other but there is big difference between the delay of path 1 and 2 under the fixed time control.

At the same time, please remember that to find the fixed optimal split rates, we have to know the mean values of the traffic from outside of the network. If those mean values are varying or not known, the fixed time control may not achieve a good performance. In Figure 4.6 and Table 4.2, we give the simulation results when only λₓ1 is changed to 0.75 veh/s. From the figure and table, we can find although the change in λₓ1 is only 0.05, the performance of the system under the fixed time control becomes much worse than that of the decentralized hybrid control. Not only the queue lengths at x directions of intersection 0 and y direction of intersection 1 under the fixed time control are lager than those of the decentralized control, but also
Figure 4.7: Queue lengths of intersections when $\lambda_{x1} = 0.75$ \textit{veh/s} and $T = 30$s
Table 4.2: Comparison of two intersection control methods in five intersection simulation when $T = 30s$ and $\lambda_{x_1} = 0.75 \text{ veh/s}$

<table>
<thead>
<tr>
<th>Paths</th>
<th>Decentralized Control</th>
<th>Fixed Time Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Throughput(s$^{-1}$)</td>
<td>Mean Value of Delay(sec)</td>
</tr>
<tr>
<td>1: $x$ direction of intersection 1,0,3</td>
<td>0.74</td>
<td>33.0</td>
</tr>
<tr>
<td>2: $y$ direction of intersection 2,0,4</td>
<td>0.30</td>
<td>18.0</td>
</tr>
<tr>
<td>3: $y$ direction of intersection 1</td>
<td>0.31</td>
<td>9.0</td>
</tr>
<tr>
<td>4: $x$ direction of intersection 2</td>
<td>0.69</td>
<td>7.9</td>
</tr>
<tr>
<td>5: $y$ direction of intersection 3</td>
<td>0.34</td>
<td>11.8</td>
</tr>
<tr>
<td>6: $x$ direction of intersection 4</td>
<td>0.74</td>
<td>6.3</td>
</tr>
<tr>
<td>Total</td>
<td>3.12</td>
<td>86.0</td>
</tr>
</tbody>
</table>

To show the effects of different cycle times, we simulate the system with $T = 40s$. The simulation results of this case are given in Figure 4.8 and Table 4.3. Comparing Figure 4.7 and 4.8, we find that there is no big differences between the queue lengths of the intersections under the decentralized control when $T = 30s$ and $T = 40s$. But from Table 4.2 and 4.3, we can clearly see that both the total throughput and the total delay of the paths when $T = 40s$ are better than those when $T = 30s$. This is because the effective green time of the traffic lights increases as the cycle time increases. In this simulation, since the lost time $T_l$ is a constant, the value of $\gamma$ increases with the increase of the cycle time. This value of $\gamma$ represents the effective green time of
Figure 4.8: Queue lengths of intersections when $\lambda_{x1} = 0.75 \text{ veh/s}$ and $T = 40s$
Table 4.3: Comparison of two intersection control methods in five intersection simulation when $T = 40s$

<table>
<thead>
<tr>
<th>Paths</th>
<th>Decentralized Control</th>
<th>Fixed Time Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Throughput($s^{-1}$)</td>
<td>Mean Value of Delay(sec)</td>
</tr>
<tr>
<td>1: $x$ direction of intersection 1,0,3</td>
<td>0.74</td>
<td>28.1</td>
</tr>
<tr>
<td>2: $y$ direction of intersection 2,0,4</td>
<td>0.30</td>
<td>18.0</td>
</tr>
<tr>
<td>3: $y$ direction of intersection 1</td>
<td>0.30</td>
<td>9.2</td>
</tr>
<tr>
<td>4: $x$ direction of intersection 2</td>
<td>0.73</td>
<td>10.7</td>
</tr>
<tr>
<td>5: $y$ direction of intersection 3</td>
<td>0.36</td>
<td>10.2</td>
</tr>
<tr>
<td>6: $x$ direction of intersection 4</td>
<td>0.71</td>
<td>5.9</td>
</tr>
<tr>
<td>Total</td>
<td>3.14</td>
<td>82.1</td>
</tr>
</tbody>
</table>

the traffic lights. At the same time, the performance of the system under the fixed time control when $T = 40s$ is much better than that of $T = 30s$. As we know, when $T = 30s$ and $\lambda_{x1} = 0.75$ veh/s, the intersection 1 is almost saturated. But when the cycle time changes to 40s, the increase of the cycle time attenuates the near saturated situation of intersection 1, which makes the traffic conditions of the five intersection network close to those when we choose the fixed time control. This explains why the performance of the fixed intersection control when $T = 40s$ is much better than that of $T = 30s$. From this analysis, we also can see that the decentralized control method can work well in different traffic situations but the fixed time control can not.
Figure 4.9: A network with four intersections

4.4.2 Four Intersections

After we discussed the simulation for five intersections, we are going to analyze the four intersection case. Figure 4.9 shows a network with four intersections. Actually, this four intersection control problem is more difficult than that of the five intersections because the control of every intersection depends on that of another intersection, i.e., intersection 2 depends on intersection 1, intersection 3 depends on 2, intersection 4 depends on 3, and intersection 1 depends on 4. For this case, we choose the parameters of the system as follows: the delay between intersections
Figure 4.10: Queue lengths of four intersections
Table 4.4: Comparison of two intersection control methods in four intersection simulation.

<table>
<thead>
<tr>
<th>Paths</th>
<th>Decentralized Control</th>
<th>Fixed Time Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Throughput(s⁻¹)</td>
<td>Mean Value of Delay(sec)</td>
</tr>
<tr>
<td>1: x direction of intersection 1,2</td>
<td>0.53</td>
<td>14.9</td>
</tr>
<tr>
<td>2: y direction of intersection 2,3</td>
<td>0.45</td>
<td>15.0</td>
</tr>
<tr>
<td>3: x direction of intersection 3,4</td>
<td>0.46</td>
<td>16.1</td>
</tr>
<tr>
<td>4: y direction of intersection 4,1</td>
<td>0.47</td>
<td>18.8</td>
</tr>
<tr>
<td>Total</td>
<td>1.91</td>
<td>64.9</td>
</tr>
</tbody>
</table>

\( \tau_{12} = 15 \text{s}, \tau_{23} = 10 \text{s}, \tau_{34} = 15 \text{s}, \) and \( \tau_{41} = 10 \text{s}; \) the maximum number of vehicles between the intersections are \( q_{y1}^m = 50 \text{ veh}, q_{x2}^m = 65 \text{ veh}, q_{y3}^m = 50 \text{ veh} \) and \( q_{x4}^m = 65 \text{ veh}; \) the maximum flow rates at \( x \) and \( y \) directions \( u_{x1}^m = u_{x2}^m = u_{x3}^m = u_{x4}^m = 1.5 \text{ veh/s} \) and \( u_{y1}^m = u_{y2}^m = u_{y3}^m = u_{y4}^m = 1 \text{ veh/s}; \) the rate for traffic from outside of the network \( \lambda_{x1} = 0.5 \text{ veh/s}, \lambda_{y2} = 0.43 \text{ veh/s}, \lambda_{x3} = 0.5 \text{ veh/s}, \lambda_{y4} = 0.46 \text{ veh/s}; \) the synchronization time differences between the traffic lights \( \zeta_{12} = 15 \text{ s}, \zeta_{23} = 15 \text{ s}, \) \( \zeta_{34} = 15 \text{ s}, \zeta_{41} = 15 \text{ s}; \) the lost time \( T_l = 5 \text{ s}; \) and the cycle time \( T = 30 \text{ s}. \) Similarly to the five intersection case, we choose a set of split rates to simulate a fixed time control. Because of the dependency of the control between intersections, it is difficult to find a good set of fixed split rates. Through trial and error, we choose \( c_1 = 0.45, \) \( c_2 = 0.3486, c_3 = 0.45 \) and \( c_4 = 0.4045. \) In Figure 4.10 and Table 4.4, the comparison of the queue lengths of the network under the decentralized hybrid control and fixed time control is shown. As we can see, the queue length at \( x \) direction of intersection 2 under the fixed time control is larger than that of the decentralized control in Figure.
4.10. In Table 4.4, both the total average throughput and total delay of the paths under the decentralized control are better than those of the fixed time control. At the same time, the delays of the individual paths under the decentralized control are close to each other, but for the fixed control they have big differences. Therefore, we can conclude that the performance of the system under the decentralized hybrid controller is better than that of the fixed intersection control.

4.4.3 Six Intersections

![Diagram of a network with six intersections]

Figure 4.11: A network with six intersections

For the six intersection network shown in Figure 4.11, we choose the parameters as follows: the delay between intersections $\tau_{12} = 10 s$, $\tau_{23} = 10 s$, $\tau_{34} = 15 s$, $\tau_{45} = 10 s$, and $\tau_{56} = 10 s$; the maximum number of vehicles between the intersections are $q_{x1}^m = 50$ veh, $q_{x2}^m = 50$ veh, $q_{x3}^m = 65$ veh, $q_{x5}^m = 50$ veh, and $q_{x6}^m = 50$ veh; the maximum flow rates at $x$ and $y$ directions $u_{x1}^m = u_{x2}^m = u_{x3}^m = u_{x4}^m = u_{x5}^m = u_{x6}^m = 1.5$ veh/s and $u_{y1}^m = u_{y2}^m = u_{y3}^m = u_{y4}^m = u_{y5}^m = u_{y6}^m = 1$ veh/s; the rate for traffic from outside of the network $\lambda_{x1} = 0.6$ veh/s, $\lambda_{y1} = 0.4$ veh/s, $\lambda_{y2} = 0.45$ veh/s, $\lambda_{y3} = 0.48$ veh/s, $\lambda_{y4} = 0.48$ veh/s, $\lambda_{y5} = 0.5$ veh/s, and $\lambda_{y6} = 0.5$ veh/s; the synchronization time differences between the traffic lights $\zeta_{12} = 10 s$, $\zeta_{23} = 10 s$, $\zeta_{34} = 15 s$, $\zeta_{45} = 10 s$, and
Figure 4.12: Queue lengths of six intersections
<table>
<thead>
<tr>
<th>Paths</th>
<th>Decentralized Control</th>
<th>Fixed Time Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Throughput $s^{-1}$</td>
<td>Mean Value of Delay (sec)</td>
</tr>
<tr>
<td>1: x direction of intersections 1-6</td>
<td>0.52</td>
<td>50.8</td>
</tr>
<tr>
<td>2: y direction of intersection 1</td>
<td>0.41</td>
<td>10.3</td>
</tr>
<tr>
<td>3: y direction of intersection 2</td>
<td>0.45</td>
<td>12.0</td>
</tr>
<tr>
<td>4: y direction of intersection 3</td>
<td>0.46</td>
<td>10.7</td>
</tr>
<tr>
<td>5: y direction of intersection 4</td>
<td>0.48</td>
<td>14.0</td>
</tr>
<tr>
<td>6: y direction of intersection 5</td>
<td>0.48</td>
<td>14.9</td>
</tr>
<tr>
<td>7: y direction of intersection 6</td>
<td>0.46</td>
<td>11.5</td>
</tr>
<tr>
<td>Total</td>
<td>3.26</td>
<td>124.1</td>
</tr>
</tbody>
</table>

Table 4.5: Comparison of two intersection control methods in six intersection simulation
\( \zeta_{56} = 10 \text{ s}; \) the lost time \( T_l = 5 \text{ s}, \) and the cycle time \( T = 30 \text{ s}. \) Here, we choose the fixed split rates as \( c_1 = 0.4318, c_2 = 0.4185, c_3 = 0.3857, c_4 = 0.3665, c_5 = 0.3547 \) and \( c_6 = 0.3473. \) The simulation results shown in Figure 4.12 and Table 4.5 demonstrate that the performance of the system under the decentralized controller is better than that under the fixed intersection control.

### 4.4.4 Nine Intersections

A more complex network is the one with nine intersections shown in Figure 4.13. This case is worth to be studied because it almost includes all the different intersection topology combinations, e.g., four intersections, five intersections, and it is the unit of which a very complex network can be made up.

Here, the parameters of the system are chosen as: the delay between intersections \( \tau_{12} = 15 \text{ s}, \tau_{23} = 15 \text{ s}, \tau_{14} = 10 \text{ s}, \tau_{52} = 10 \text{ s}, \tau_{36} = 10 \text{ s}, \tau_{54} = 15 \text{ s}, \tau_{68} = 15 \text{ s}, \tau_{47} = 10 \text{ s}, \tau_{85} = 10 \text{ s}, \tau_{69} = 10 \text{ s}, \tau_{78} = 15 \text{ s}, \) and \( \tau_{89} = 15 \text{ s}; \) the maximum number of vehicles between the intersections are \( q_{x2}^m = 65 \text{ veh}, q_{y2}^m = 50 \text{ veh}, q_{x3}^m = 65 \text{ veh}, q_{x4}^m = 65 \text{ veh}, q_{y4}^m = 50 \text{ veh}, \) \( q_{x5}^m = 65 \text{ veh}, q_{y5}^m = 50 \text{ veh}, q_{x6}^m = 65 \text{ veh}, q_{y6}^m = 50 \text{ veh}, q_{x7}^m = 65 \text{ veh}, q_{x8}^m = 65 \text{ veh}, q_{y8}^m = 50 \text{ veh}. \) The maximum flow rates at \( x \) and \( y \) directions \( u_{x1}^m = u_{x2}^m = u_{x3}^m = u_{x4}^m = u_{x5}^m = u_{x6}^m = u_{x7}^m = u_{x8}^m = u_{x9}^m = 1.5 \text{ veh/s} \) and \( u_{y1}^m = u_{y2}^m = u_{y3}^m = u_{y4}^m = u_{y5}^m = u_{y6}^m = u_{y7}^m = u_{y8}^m = u_{y9}^m = 1 \text{ veh/s}; \)

the rate for traffic from outside of the network \( \lambda_{x1} = 0.6 \text{ veh/s}, \lambda_{y1} = 0.4 \text{ veh/s}, \lambda_{x3} = 0.47 \text{ veh/s}, \lambda_{x6} = 0.55 \text{ veh/s}, \lambda_{x7} = 0.55 \text{ veh/s}, \) and \( \lambda_{y8} = 0.4 \text{ veh/s}; \) the synchronization time differences between the traffic lights \( \zeta_{12} = 15 \text{ s}, \zeta_{23} = 15 \text{ s}, \zeta_{14} = 5 \text{ s}, \zeta_{45} = 15 \text{ s}, \zeta_{56} = 20 \text{ s}, \zeta_{47} = 10 \text{ s}, \zeta_{78} = 20 \text{ s}, \) and \( \zeta_{89} = 15 \text{ s}; \) the lost time
Figure 4.13: A network with nine intersections
Figure 4.14: Queue lengths of nine intersections
Figure 4.15: Queue lengths of nine intersections
### Table 4.6: Comparison of two intersection control methods in nine intersection simulation

<table>
<thead>
<tr>
<th>Paths</th>
<th>Decentralized Control</th>
<th>Fixed Time Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Throughput ($s^{-1}$)</td>
<td>Mean Value of Delay (sec)</td>
</tr>
<tr>
<td><strong>1:</strong> $x$ direction of intersections 1,2,3</td>
<td>0.59</td>
<td>25.1</td>
</tr>
<tr>
<td><strong>2:</strong> $x$ direction of intersections 6,5,4</td>
<td>0.51</td>
<td>23.3</td>
</tr>
<tr>
<td><strong>3:</strong> $x$ direction of intersections 7,8,9</td>
<td>0.51</td>
<td>17.1</td>
</tr>
<tr>
<td><strong>4:</strong> $y$ direction of intersection 1,4,7</td>
<td>0.40</td>
<td>19.9</td>
</tr>
<tr>
<td><strong>5:</strong> $y$ direction of intersection 8,5,2</td>
<td>0.41</td>
<td>21.9</td>
</tr>
<tr>
<td><strong>6:</strong> $y$ direction of intersection 3,6,9</td>
<td>0.43</td>
<td>22.2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>2.85</td>
<td>129.5</td>
</tr>
</tbody>
</table>

$T_i = 5$ s, and the cycle time $T = 30$ s. Through trial and error, we find a good set of the fixed split rates as $c_1 = 0.4706$, $c_2 = 0.52$, $c_3 = 0.4186$, $c_4 = 0.4286$, $c_5 = 0.5094$, $c_6 = 0.4431$, $c_7 = 0.4538$, $c_8 = 0.4839$ and $c_9 = 0.48$. The simulation results are shown in Figure 4.14, 4.15 and Table 4.6. As we can see, the decentralized hybrid controller still can work well when a network is as complex as this nine intersection network.

### 4.5 Conclusion

In this chapter, a decentralized hybrid intersection control method was proposed. We first analyzed traffic situations of an isolated intersection and found two states to model the intersection: saturated, and unsaturated. Based on different dynamics of
the two states, two controllers were designed. After defining the switching condition of the two controllers, a hybrid controller for an isolated intersection was provided. We extended the idea of the hybrid control to the multi-intersection case and obtained a decentralized hybrid intersection controller with five states, in which different traffic situations of the intersection and its downstream and upstream intersections are handled. We did a thorough simulation study for this decentralized hybrid controller, i.e., four, five, six and nine intersection networks are simulated. All the simulation results demonstrated that by using our decentralized hybrid intersection control method a good performance can be achieved.
CHAPTER 5

TWO-FREEWAY ROUTING USING SLIDING MODES

5.1 Problem Statement

In Chapter 3, we discussed dynamic routing problems in local traffic networks, in which the travel times along links are fixed and the delay caused by queues and traffic lights at intersections as two main delay factors have to be considered. However, for the routing problems in freeway systems, the situation is different. In a freeway network, nodes are the places with on/off ramps or merging of several freeways. Since there are usually no queues formed at the nodes, the queue length is not an important factor to be considered in the routing strategy any more. The travel times of freeways which change with the different traffic conditions in the freeways become the most important factor. In this chapter, we will analyze a dynamic routing problem in freeway systems based on the hydrodynamic model given in Chapter 2.

Typically, the number of roads in the freeway system of a city is much smaller than that in the local traffic system. That means that there are not as many possible paths from one place to another in the freeway system as those in the local traffic system. Normally, the freeway route from one place to another is fixed. Only when the route is in congestion, an alternate way may be taken. This alternate way can
be either a local route or a freeway. Here, we will discuss a routing problem between two freeways. In Figure 5.1, an illustration of this two-freeway routing problem is given. According to the PDE model of freeway traffic in Chapter 2, the dynamics of the system in Figure 5.1 are

\[
\dot{\rho}_k^i(t) = \frac{1}{\delta_k^i} [u_k(t) - f_k^i(t)] ,
\]

(5.1)

\[
\dot{\rho}_k^i(t) = \frac{1}{\delta_k^i} [f_k^{i-1}(t) - f_k^i(t)] ,
\]

(5.2)

\[
\dot{v}_k^i(t) = \left( v_k^{i-1}(t) - v_k^i(t) \right) \frac{v_k^i(t)}{\delta_k^i} + \frac{1}{\kappa} (v_e(\rho_k^i(t)) - v_k^i(t))
\]

(5.3)

\[
f_k^i(t) = \alpha \rho_k^i(t) v_k^i(t) + (1 - \alpha) \rho_k^{i+1}(t) v_k^{i+1}(t)
\]

(5.4)

\[
v_e(\rho_k^i) = u_f k \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{\text{max}}} \right)^7 \right] .
\]

(5.5)

where we use \( k = 1, 2 \) to index the freeways instead of two node numbers in Chapter 2, and \( u_k \) is the traffic into the \( k \)th road. \( i \) denotes the segment number of the freeways and \( i = 1, \ldots, n_k \) except \( i = 2, \ldots, n_k \) in Eqn. (5.2). \( n_k \) is the total segment number at freeway \( k \).

Assume the amount of traffic flow coming to the node is fixed as \( U \). The traffic
into the two freeways are constrained as

\[ u_1(t) + u_2(t) = U, \quad u_1(t) \geq 0, \quad \text{and} \quad u_2(t) \geq 0. \]  \hspace{1cm} (5.6)

Now, we choose the optimization criterion for the dynamic routing in freeway systems similar to that for local traffic networks, i.e., we try to minimize the aggregate travel time of traffic in the freeways, which is expressed as

\[
\min_{u_{1,2}(t)} J = \int_0^{t_f} [u_1(t)\Gamma_1(t) + u_2(t)\Gamma_2(t)]dt \]  \hspace{1cm} (5.7)

where \( t_f \) is the time interval to be optimized, \( \Gamma_1(t) \) and \( \Gamma_2(t) \) are the total travel times along freeway 1 and 2 at time \( t \).

Before we analyze this dynamic routing problem, we will give a brief introduction of sliding mode theory.

### 5.2 Brief Introduction of Sliding Mode Theory

There is a special kind of phenomenon which may appear when a dynamic system is governed by ordinary differential equations with discontinuous input. It may happen when the control is a discontinuous function of the system state and switches at high frequency.

For example, consider the simplest first-order tracking relay system \([43]\)

\[
\dot{x} = f(x) + u 
\]  \hspace{1cm} (5.8)

with the bounded function \( f(x) \) as \( |f(x)| < f_0 \), where \( f_0 \) is a constant. Select the control \( u \) as a relay function of the tracking error \( e = r(t) - x \) (Figure 5.2), i.e.,

\[
u = u_0 \text{sign}(e) = \begin{cases} u_0 & e > 0 \\ -u_0 & e < 0 \end{cases} \]  \hspace{1cm} (5.9)
where $r(t)$ is the reference input and $u_0$ is a constant. When $u_0 > f_0 + |\dot{r}|$, the values of $e$ and $\dot{e} = \dot{r} - f(x) - u_0 \text{sign}(e)$ have different signs. That means that the magnitude of the tracking error decays at finite rate and after a finite interval $T$ the error is ideally equal to zero, which is corresponding to the discontinuity point of the relay function. However, for real implementations, the imperfection of switching devices will result in the control switching at a high frequency around the discontinuity point. This motion is called Sliding Mode [43]. And the control which can cause the sliding mode motion is called sliding mode control.

In general, for a variable structure system governed by

$$\dot{x} = f(x, t) + B(x, t)u + h(x, t), \quad (5.10)$$

the sliding mode control is

$$u = \begin{cases} 
u^+(x, t), & \text{if } s(x) > 0 \text{ componentwise} \\ 
u^-(x, t), & \text{if } s(x) < 0 \text{ componentwise} \end{cases} \quad (5.11)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $s \in \mathbb{R}^m$ and $h(x, t) \in \mathbb{R}^n$ represents all the factors whose influence on the control process should be eliminated. The switching law (5.11) implies that each component of the control $u$ undergoes discontinuities in the surface $s_1 = 0,$

Figure 5.2: Relay control
The system with the discontinuous control is known to generate sliding modes with state trajectories running in the intersection of discontinuity surfaces.

Because sliding mode control can reject the effects of unknown disturbances and nonlinearities by using its switching control law, it is a powerful and robust nonlinear control method. Those characteristics make the sliding mode approach suitable to be applied to the transportation systems since the dynamics of traffic are nonlinear and with many unknown factors. In the next sections, we will use the sliding mode method to solve the two-freeway routing problem.

5.3 The User Equilibrium Dynamic Two-Freeway Routing Using Sliding Modes

Before analyzing the problem given in Section 5.1, we will first solve a user equilibrium dynamic two-freeway routing problem using sliding modes.

From the users' point view, the two-freeway routing problem is to choose the road with the smaller travel time. However, the travel time of a freeway changes with the number of vehicles on it. When more and more vehicles enter the road with the shorter travel time, its travel time increases. Thus, the result of choosing the road with the smaller travel time is balancing the travel times of the two freeways. In other words, the objective of the two-freeway user equilibrium routing problem is to balance the travel time of the two alternate roads, i.e.,

\[
S(u(t)) = \Gamma_1(t) - \Gamma_2(t) \rightarrow 0 . \tag{5.12}
\]

Now, to find an approximate expression of travel time of a freeway, we first neglect the dynamics of the traffic velocity and use its mean value to be the average speed of
segments, i.e.,
\[ v_k^i = v_{jk} \left[ 1 - (\frac{\rho_k^i(t)}{\rho_{\text{max}}})^{l_k} \right]^{m_k}. \]  \hspace{1cm} (5.13)

Then neglecting the effect of the traffic condition at the next segment to the traffic flow of the present segment, we have
\[ f_k^i(t) = \rho_k^i v_{ki}. \] \hspace{1cm} (5.14)

The dynamics of the two freeway system in Figure (5.1) can be rewritten as
\[ \dot{\rho}_k^i(t) = \frac{1}{\delta_k} [u_k(t) - f_k^i(t)], \quad k = 1, 2, \] \hspace{1cm} (5.15)
\[ \dot{f}_k^i(t) = \frac{1}{\delta_k} [f_k^{i-1}(t) - f_k^i(t)], \quad k = 1, 2, \quad i = 2, \cdots, n_k, \] \hspace{1cm} (5.16)
\[ f_k^i(t) = \rho_k^i v_k^i, \quad k = 1, 2, \quad i = 1, \cdots, n_k, \] \hspace{1cm} (5.17)
\[ v_k^i = v_{jk} \left[ 1 - (\frac{\rho_k^i(t)}{\rho_{\text{max}}})^{l_k} \right]^{m_k}, \quad k = 1, 2, \quad i = 1, \cdots, n_k. \] \hspace{1cm} (5.18)

Assuming that the average speed \( v_{ki} \) is measured, the average travel time of traffic in the \( i \)th segment of the \( k \)th road is calculated as
\[ \tau_k^i(t) = \frac{\delta_k^i}{v_k^i(t)}. \] \hspace{1cm} (5.19)

The total average travel time of the \( k \)th road
\[ \Gamma_k(t) = \sum_{i=1}^{n_k} \tau_k^i(t). \] \hspace{1cm} (5.20)

Now, to analyze the user equilibrium routing problem given in Eqn. (5.12), we select the following Lyapunov function
\[ V = \frac{1}{2} s^2 > 0, \] \hspace{1cm} (5.21)
then

\[ \dot{V} = S \dot{S}. \]  

(5.22)

Since

\[ \dot{\gamma}_k = \sum_{i=1}^{n_k} \frac{d}{dt} \left( \frac{\delta_k^i}{v_{fk}} \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{max}} \right)^{i_k m_k} \right] \right), \]

\[ = \sum_{i=1}^{n_k} m_k l_k \frac{\delta_k^i}{v_{fk}} \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{max}} \right)^{i_k m_k+1} \right] \hat{\rho}_k^i, \]

\[ = m_k l_k \frac{\delta_k^i}{v_{fk}} \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{max}} \right)^{i_k m_k+1} \right] \hat{\rho}_k^i + \sum_{i=2}^{n_k} m_k l_k \frac{\delta_k^i}{v_{fk}} \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{max}} \right)^{i_k m_k+1} \right] \hat{\rho}_k^i, \]

(5.23)

where

\[ A_k = \frac{m_k l_k}{v_{fk}} \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{max}} \right)^{i_k m_k+1} \right], \]

(5.24)

\[ B_k = -\sum_{i=2}^{n_k} m_k l_k \frac{\delta_k^i}{v_{fk}} \left[ 1 - \left( \frac{\rho_k^i(t)}{\rho_{max}} \right)^{i_k m_k+1} \right] \left( f_k^{i-1} - f_k^i \right), \]

(5.25)

we have

\[ \dot{S} = \dot{\gamma}_1 + \dot{\gamma}_2, \]

\[ = A_1 \gamma_1 + B_1 - (A_2 \gamma_2 + B_2), \]

\[ = A_1 \gamma_1 - A_2 \gamma_2 + B_1 - B_2, \]

\[ = A_1 \gamma_1 - A_2 (U - \gamma_1) + B_1 - B_2, \]

\[ = (A_1 + A_2) \gamma_1 - A_2 U + B_1 - B_2. \]

(5.26)

Because \( A_1 > 0, A_2 > 0 \) and the values of \( B_1 \) and \( B_2 \) are decided by the difference of traffic flows at two joined segments, which is small, the sign of \( \dot{S} \) is decided by the value of \( u_1 \). Assuming \( U > |B_1 - B_2| \), we have

\[ \dot{S} = \begin{cases} > 0 & u_1 = U, \\ < 0 & u_1 = 0. \end{cases} \]

(5.27)
Then
\[ u_1 = \begin{cases} U & \text{if } S < 0 \\ 0 & \text{if } S > 0 \end{cases}, \quad \hat{V} < 0; \]
\[ u_1 = 0 \Rightarrow \hat{V} < 0. \]

Therefore \( V \rightarrow 0 \), i.e. \( S = \Gamma_1 - \Gamma_2 \rightarrow 0 \).

Here, we use Theorem 5.1 to summarize the above analysis.

**Theorem 5.1** Consider the system in Figure 5.1 with dynamics given by Eqn. (5.1)-(5.5). Define \( S = \Gamma_1(t) - \Gamma_2(t) \) and use control
\[
u_1(t) = \begin{cases} U & \text{if } S < 0 \\ 0 & \text{if } S > 0 \end{cases}, \quad \nu_2(t) = \begin{cases} U & \text{if } S > 0 \\ 0 & \text{if } S < 0 \end{cases}, \quad (5.28)
\]
the travel times of the two roads are balanced, i.e., \( S \rightarrow 0 \).

### 5.4 System Optimal Dynamic Two-freeway Routing

Now, we claim that the above user equilibrium routing problem is equivalent to the system optimal routing problem given in Section 5.1 under our sliding mode control.

**Theorem 5.2:** Under the control Eqn. (5.28), the two-freeway user equilibrium problem is equivalent to the system optimal dynamic two-freeway routing problem given in Eqn. (5.7).

**Proof 5.2:**

Since
\[
\Gamma_1 u_1 + \Gamma_2 u_2 \geq 0 \quad \text{for } \forall t, \quad (5.29)
\]
we have
\[
\min J \leftrightarrow \min_{\forall t} (\Gamma_1 u_1 + \Gamma_2 u_2). \quad (5.30)
\]

We know when \( \Gamma_1 < \Gamma_2 \) i.e. \( S < 0 \), \( u_1 = U, u_2 = 0 \),
\[
\Gamma_1 u_1 + \Gamma_2 u_2 = \Gamma_1 U = \min_{\Gamma_1 < \Gamma_2, 0 \leq u_1 \leq U, 0 \leq u_2 \leq U} (\Gamma_1 u_1 + \Gamma_2 u_2); \quad (5.31)
\]
and when $\Gamma_1 > \Gamma_2$ i.e. $S > 0$, $u_1 = 0, u_2 = U$, 

$$\Gamma_1 u_1 + \Gamma_2 u_2 = \Gamma_2 U = \min_{\Gamma_1 > \Gamma_2, 0 \leq u_1 \leq U, 0 \leq u_2 \leq U} (\Gamma_1 u_1 + \Gamma_2 u_2). \tag{5.32}$$

Therefore,

$$\min J \iff S \to 0.$$ 

### 5.5 Simulation Example

The performance of the proposed dynamic two-freeway routing has been tested in the simulation of the freeway system given in Figure 5.1. The length of freeway 1 is $4\text{km}$ and that of freeway 2 is $3\text{km}$. Both of them have the same parameters, such as $l = 1$, $m = 3$, $v_f = 30\text{m/s}$, $\rho_{\text{max}} = 2$. Since the length of every segment is equal to $200\text{m}$, freeway 1 has 20 segments and freeway 2 has 15 segments. Here, we set the flow rate of traffic coming to the system $U = 5\text{s}^{-1}$. In Figure 5.3-5.5, the simulated travel time of freeway 1 and 2 and the difference of them are given. The simulation results show that the travel times of the two freeway are balanced by using our sliding mode dynamic routing method.

### 5.6 Summary

In this chapter, we discussed the dynamic two-freeway routing problem. Based on the discretized hydrodynamic model, we analyzed the system characteristics and proposed a user equilibrium dynamic two-freeway routing method using sliding modes. Then we proved that under this sliding mode control, the user equilibrium dynamic routing is equivalent to the system optimal dynamic routing. A simulation example was given and verified that our control can balance the travel times of two freeways.
Figure 5.3: The travel time of freeway 1

Figure 5.4: The travel time of freeway 2
Figure 5.5: The difference between the travel time of the two freeways
CHAPTER 6

FREEWAY ON-RAMP CONTROL USING SLIDING MODES

6.1 Overview

In a freeway system, on-ramp control is an important method to keep the system running smoothly when the traffic demand is higher than the capacity of the system. As a kind of signaling control, on-ramp control limits the amount of traffic entering the system by metering the inflow. Its objective is to achieve the highest throughput of the main line of the freeway system when it is congested. However, this highest throughput is not easy to obtain since the chosen model of the system is nonlinear. In this chapter, we apply a powerful nonlinear analysis method, sliding mode, to control on-ramp traffic such that the optimal state of the main line of the freeway is searched and achieved in real time.

This chapter is outlined as follows: In Section 6.2, the system characteristics analysis and problem formulation are given. Then an on-ramp control which directly applies the sliding mode method is proposed. Based on this method, we propose an on-ramp control using optimum search via sliding modes in Section 6.3. A numerical
example is given in Section 6.4 and the chapter concludes with a brief summary in Section 6.5.

6.2 On-ramp Control Problem Formulation

In this chapter, we assume that the positions of off-ramps along the traffic direction of a freeway come earlier than those of on-ramps and the distances between the off-ramps and their corresponding on-ramps are short. Consider a freeway with length $X$ and $M$ on/off ramps in Figure 6.1. Since the off-ramp traffic can be represented by

$$x = 0 \quad x = x_1 \quad x = x_1 + \Delta_i \quad x = x_1 \quad x = x_1 + \Delta_i \quad x = x_M \quad x = x_M + \Delta_M \quad x = X$$

![Figure 6.1: Freeway illustration at one direction](image)

point exits at \{x_1, \ldots, x_M\} and the on-ramp traffic is also point injections at $\Delta_i$ distance from corresponding exits, i.e., \{(x_1 + \Delta_1), \ldots, (x_i + \Delta_i), \ldots, (x_M + \Delta_M)\}; the on/off ramp traffic can be expressed easily using the special shifted impulse functions:

$$p(x, t) = \sum_{i=1}^{M} p_i(t) \delta(x - (x_i + \Delta_i)) \quad (6.1)$$

$$s(x, t) = \sum_{i=1}^{M} s_i(t) \delta(x - x_i) \quad (6.2)$$

100
where $p_i(t)$ and $s_i(t)$ are the on and off ramp traffic flow respectively. They have constraints as

\begin{align*}
0 &\leq p_i(t) \leq p_{\text{max}}, \quad i = 1, \cdots, M, \quad (6.3) \\
0 &\leq s_i(t) \leq s_{\text{max}}, \quad i = 1, \cdots, M. \quad (6.4)
\end{align*}

$p_{\text{max}}$ and $s_{\text{max}}$ are the maximum traffic flow of the on/off ramp. Based on the hydrodynamic model of traffic given by Eqn. (2.31) and (2.32), the dynamics of the highway with on/off ramp traffic are written as

\begin{equation}
\frac{\partial}{\partial t} \rho(x,t) + \frac{\partial}{\partial x} f(x,t) = p(x,t) - s(x,t), \quad (6.5)
\end{equation}

\begin{equation}
f(x,t) = \rho(x,t)v(x,t), \quad (6.6)
\end{equation}

In this chapter, since only one freeway is considered, we will neglect the subscribe $m,n$ in the equations of Chapter 2. The ODEs of the freeway are given as follows:

\begin{align*}
\rho^i(t) &= \frac{1}{\delta^i} \left[ f^{i-1}(t) - f^i(t) + p^i(t) - s^i(t) \right], \quad (6.7) \\
v^i(t) &= \left( v^{i-1}(t) - v^i(t) \right) \frac{v^i(t)}{\delta^i} + \frac{1}{\kappa} (v_e(\rho^i(t)) - v^i(t)) - \frac{\mu}{\kappa \delta^i} \frac{p^{i+1}(t) - p^i(t)}{\rho^i(t) + \chi}, \quad (6.8) \\
f^i(t) &= \alpha \rho^i(t) v^i(t) + (1 - \alpha) \rho^{i+1}(t) v^{i+1}(t), \quad 0 \leq \alpha \leq 1, \quad (6.9)
\end{align*}

where $\rho^i(t)$, $v^i(t)$ and $f^i(t)$ are the traffic density, speed and flow at the $i$th segment of the freeway. $r^i(t)$ and $s^i(t)$ denote the on/off ramp traffic flow at the $i$th segment. They are equal to zero when this is no on/off ramp at the segment.

Based on the analysis about the traffic flow and density characteristics curve in Chapter 2, we know that at point $(\rho_{\text{cr}}, f_{\text{max}})$ the curve reaches its maximum; i.e., when $\rho = \rho_{\text{cr}}$, the highway system is optimal, achieving the highest throughput. Since our objective of on-ramp control is to achieve the highest throughput, our problem
becomes how to make the state of the main line of the freeway system attain the point \((\rho_{cr}, f_{max})\) by controlling the on-ramp traffic. Moreover, from the flow-density characteristics curve, we can see that when \(\rho < \rho_{cr}\), traffic flow is approximately proportional to traffic density, i.e.

\[
f(x, t) \approx v_f \rho(x, t)
\]  

(6.10)

where \(v_f\) is the free speed. This approximation will be used in the next section.

### 6.3 Sliding Mode On-Ramp Control

From the discussion of the characteristics of traffic flow and density, we know that the dynamics of the freeway system are nonlinear and our objective for on-ramp control is to make its main line attain its optimal point. So, the sliding mode method might be a possible approach to solve this problem. In what follows, we present the design of two sliding mode on-ramp controllers.

**Theorem 6.1** For a freeway system with length \(L\) and \(M\) on/off ramps, if the traffic demand is higher than the capacity of the main line of the freeway and a boundary condition

\[
|\rho(0, t) - \rho_{cr}| \leq |\rho(L, t) - \rho_{cr}|
\]  

(6.11)

holds, using on-ramp sliding mode control

\[
p_n(t) = r_{max} \varphi(\rho(x_n, t) - \rho_{cr}), \quad n = 1, \cdots, M
\]  

(6.12)

where

\[
\varphi(\xi) = \begin{cases} 
0 & \xi > 0 \\
-1 & \xi < 0
\end{cases}
\]  

(6.13)
the main line can achieve the highest throughput, i.e.,

\[ f(x, t) \rightarrow f_{\text{max}} . \]  

(6.14)

6.1 Proof:

Since the system is stable initially, by substituting Eqn. (6.10) into Eqn. (6.5), we have

\[ \frac{\partial}{\partial t} \rho(x, t) = -v_f \frac{\partial}{\partial x} \rho(x, t) + r(x, t) - s(x, t) \]  

(6.15)

and because \( x \) is not a function of \( t \), Eqn. (6.15) can be rewritten as

\[ \dot{\rho}(x, t) = -v_f \rho(x, t)' + r(x, t) - s(x, t) \]  

(6.16)

where \( \rho' = \frac{\partial}{\partial x} \rho(x, t) \).

Select a Lyapunov function as

\[ V = \frac{1}{2} \int_0^L (\rho(x, t) - \rho_{cr})^2 dx \]  

(6.17)

then

\[ \dot{V} = \int_0^L (\rho - \rho_{cr}) \dot{\rho} dx 
= \int_0^L (\rho - \rho_{cr})(-v_f \rho' + p - s) dx 
= -v_f \int_0^L (\rho - \rho_{cr})(\rho - \rho_{cr})' dx + \int_0^L (\rho - \rho_{cr})(p - s) dx 
= -\frac{1}{2} v_f (\rho - \rho_{cr})^2|_0^L + \int_0^L (\rho - \rho_{cr})(p - s) dx . \]  

(6.18)

Substituting Eqn. (6.1)(6.2) in, we obtain

\[ \dot{V} = -\frac{1}{2} v_f (\rho - \rho_{cr})^2|_0^L 
+ \sum_{n=1}^3 \left[ (\rho(x_n + \Delta_n, t) - \rho_{cr})p_n(t) - (\rho(x_n, t) - \rho_{cr})s_n(t) \right] . \]  

(6.19)

Since usually the distance \( \Delta_i \) between off-ramp and on-ramp point is fairly short, the traffic densities of these two points can be thought to be equal. Therefore, Eqn. (6.19) becomes

\[ \dot{V} = -\frac{1}{2} v_f (\rho - \rho_{cr})^2|_0^L 
+ \sum_{n=1}^3 \{ [(\rho(x_n, t) - \rho_{cr})(p_n(t) - s_n(t))] \} . \]  

(6.20)
Using the boundary condition, we have
\[ -\frac{1}{2} v_f (\rho - \rho_{cr})^2 |_0^L \leq 0. \] (6.21)

Moreover, since we assume \( p_{max} > s_n(t) \) and the sliding mode control \( p_n(t) = p_{max} \varphi(\rho(x_n, t) - \rho_{cr}) \), the following equation holds.
\[ [\rho(x_n, t) - \rho_{cr}][p_n(t) - s_n(t)] < 0, \quad n = 1, \ldots, M. \] (6.22)

Thus, we conclude that
\[ \dot{V} < 0. \] (6.23)

Therefore, under the control strategy specified by Eqn. (6.12), the system is stable and the traffic density goes to the critical density \( \rho_{cr} \); simultaneously the traffic flow goes to the maximum \( \rho_{max} \).

Usually the critical density \( \rho_{cr} \) is an unknown parameter and in the real world the traffic density is very difficult to measure, but the above control method requires that the critical density is known and the traffic density \( \rho \) is measured. These requirements constrain the application of the above control algorithm. Thus, in Theorem 6.2, we give another sliding mode control which can search the optimal point automatically based on the traffic flow of the freeway and force the system to be optimal.

**Theorem 6.2** For a freeway system with length \( L \) and \( M \) on/off ramps, if the traffic demand is higher than the main line capacity of the freeway, using control
\[ p^i = p_{max} \psi(\sin(z^i)) \] (6.24)
where
\[ z^i = \Omega f^i + \beta t \quad \beta > 0 \] (6.25)
and

\[ \psi(\xi) = \begin{cases} 1 & \xi \geq 0 \\ 0 & \xi < 0 \end{cases} \]  

(6.26)

with properly choosing \( \Omega \) and \( \beta \), the main line can obtain the optimal throughput.

6.2 Proof:

Consider the ODEs model Eqn. (6.7), (6.8) and (6.9). As we know, the traffic flow at the \( i \)th segment is a function of time and traffic density, i.e.,

\[ f^i = F^i(t, \rho^i) \]  

(6.27)

Differentiating it with respect to time, we have

\[ f^i = \frac{\partial F^i}{\partial \rho^i} \frac{1}{\delta^i} (f^{i-1} - f^i + p^i - s^i) + \frac{\partial F^i}{\partial t} \]  

(6.28)

where

\[ A^i(t) = \frac{\partial F^i}{\partial \rho^i} \frac{1}{\delta^i} (f^{i-1} - f^i - s^i) + \frac{\partial F^i}{\partial t}, \]  

(6.29)

\[ B^i(t) = \frac{\partial F^i}{\partial \rho^i} \frac{1}{\delta^i}. \]  

(6.30)

Since

\[ z^i = \Omega f^i + \beta t, \quad \beta > 0 \]  

(6.31)

and

\[ p^i = p_{max} \psi(sin(z^i)) \]  

(6.32)

we have

\[ z^i = \Omega f^i + \beta = \beta + \Omega A^i + \Omega B^i \psi(sin(z^i)) \]  

(6.33)
If the conditions given by [44] are satisfied, i.e.,

$$|\Omega B^i| > |\Omega A^i + \beta|,$$  \hspace{1cm} (6.34)

$$\text{sign}(\beta + \Omega A^i) = -\text{sign}(\Omega B^i),$$ \hspace{1cm} (6.35)

the traffic density of the ith segment will go to the optimal value $\rho_{cr}$. Substituting Eqn. (6.29) and (6.30) into Eqn. (6.35), we obtain

$$\text{sign}(\beta + \Omega A^i) = -\text{sign}(\Omega B^i).$$ \hspace{1cm} (6.36)

When $\beta$ is small and $\Omega$ is large, Eqn. (6.36) becomes

$$\text{sign}(f^{i-1} - f^i - s^i) < 0.$$ \hspace{1cm} (6.37)

Since the on-ramp traffic is larger than the off-ramp traffic, we have $\rho_i > \rho_{i-1}$ and the initial condition of the system is stable, i.e., $f^i = v_f \rho_i$, Eqn. (6.37) holds. And the condition given by Eqn. (6.34) can be satisfied by appropriately selecting the value $\beta$ and $\Omega$. Therefore, we proved that using the control of Eqn. (6.24), the system would operate at the optimal point.

### 6.4 Simulation Studies

To demonstrate that the above control algorithm can make the system search and attain the optimal operation, specific simulations are done for the freeway shown in Figure 6.1. In simulations, the freeway is divided into four blocks. Every block except the fourth block is made up of 11 segments, in which only one segment has on-ramp and off-ramp traffic. The length of all these segments, $\delta_i$, equals 100m. In
this simulation, we use the mean speed Eqn. (2.35). Since the traffic flow can be approximated as

\[ f \approx \rho v_e \]  \hspace{1cm} (6.38)

With the mean speed function given in Eqn. (2.35), we obtain approximate flow-density characteristic curves shown in Figure 6.2, where \( l = 1.5, m = 3, v_f = 30m/s, \) and \( \rho_{max} = 0.2, 0.15. \) As can be seen, when \( \rho_{max} = 0.2m^{-1} \) and the traffic density \( \rho \)

\[ p_{\text{max}} = \frac{v_f}{1.5} \]

\[ \rho = \frac{v_f}{1.5} \]

\[ f_{\text{max}} \]

\[ \omega = \frac{\sqrt{2}}{\rho} \]

\[ \Omega = 4, \beta = 2.5 \]

\[ \tau_{\max} = 2s^{-1} \]

\[ \kappa = 0.3s^{-1}, \mu = 60m^2/s, \chi = 0.02m^{-1}, \alpha = 0.8, \]

\[ \] is around \( 0.12m^{-1}, \) the highway obtains its highest traffic flow value, which is around \( 3s^{-1}, \) and when \( \rho_{\text{max}} = 0.15m^{-1} \) and \( \rho = 0.9m^{-1}, f_{\text{max}} \) is around \( 2.2s^{-1}. \) Given \( \kappa = 0.3s^{-1}, \mu = 60m^2/s, \chi = 0.02m^{-1}, \alpha = 0.8, \Omega = 4, \beta = 2.5 \) and \( \tau_{\text{max}} = 2s^{-1}, \) we simulate the freeway system. Figure 6.3 and 6.4 demonstrate the simulation results.
of the traffic flow and traffic density of one segment, when the sliding mode on-ramp control is used. For the simulation with $\rho_{max} = 0.2 m^{-1}$, the stable value of the traffic density is between 0.11 and 0.125, which is approximately equal to the value of the critical density $0.12 m^{-1}$ estimated from Figure 6.2. At the same time, the stable value of the traffic flow is between 2.8 and 3.0, which is approximately equal to the maximum flow $2.9 s^{-1}$. This means that under the sliding mode on-ramp control given in Eqn. (6.24), the traffic density and flow of the main line converge to their optimal values. We also can obtain the same conclusion for the simulation with $\rho_{max} = 0.15 m^{-1}$. Thus, using the sliding mode on-ramp control, the main line of the freeway can achieve its highest throughput.

However, the sliding mode on-ramp control will cause the delay on the ramps

Figure 6.3: simulation results for on-ramp control when $\rho_{max} = 0.2 m^{-1}$
Figure 6.4: simulation results for on-ramp control when $\rho_{max} = 0.15m^{-1}$

since its basic concept is to force some traffic to wait at the on ramps such that the main line can achieve a high throughput. For example, in the simulation with $\rho_{max} = 0.2m^{-1}$, we find that the mean value of the traffic flow into the main line through the ramp of the 3rd block is $0.95s^{-1}$. If the traffic demand approaching the ramp is $2s^{-1}$, more than half of traffic is blocked from the main line of the freeway and has to be waited on the ramp. To compare the performance of the freeway system, we carry out another simulation in which the on-ramp control is disabled. We find that the simulation system is unstable. This means that the traffic in the main line is in the stop-and-go state, i.e., the throughput of the system is very low. Therefore, we can conclude that the sliding mode on-ramp control can improve the performance of the overall freeway system although it causes delay on the ramps.
6.5 Summary

In this chapter, we formulated two sliding mode on-ramp controllers for the distributed parameter setting of freeway systems. First, we used sliding mode control to force the main line of the system to achieve the highest throughput. Since the optimal point is usually unknown a priori, a method of using sliding mode to search for the optimum was presented. The simulation results showed that the sliding mode on-ramp control is a possible control method to improve the performance of freeway systems.

However, we have to notice that some of congestion problems in freeways are not considered in this chapter. For example, when in the freeway the on-ramps come earlier in the traffic direction than the off-ramps, or the distances between the on/off ramps are not negligible, or the congestion is caused by a bottleneck between two ramps, the on-ramp control algorithm proposed above can not efficiently solve them. How to deal with those problems needs future research.
CHAPTER 7

APPLICATIONS IN VATSIM

The Vehicle And Traffic SIMulator (VATSIM) is a vehicle and traffic simulator developed in The Ohio State University, in which some of the control methods proposed in this dissertation are applied.

7.1 Overview of VATSIM

VATSIM is a simulator which can simulate not only the movement of individual vehicles, such as lane following, lane changing, vehicle following and intersection stopping, but also the switching of traffic lights and the effects of different traffic routing strategies.

The simulation framework of VATSIM shown in Figure 7.1 includes two parts: the simulator and the input databases.

VATSIM is implemented as a modular simulation system, which is made up of four modules: network, vehicles, traffic lights and traffic management. Network includes the part of program related to the input, representation and analysis of the physical information about traffic network such as the road shape, number of lanes and speed limit. Vehicles represent the information of vehicles and drivers, such as the
static and dynamic information of different vehicles, the different driving characteristic, etc. Traffic lights save the information about the position, state, cycle time and split rate of traffic lights. Traffic management includes the traffic light control and traffic routing functions. Between these four modules there are information exchanges. The network module provides physical information of simulated traffic networks to the vehicles, traffic lights and traffic management since it is the basis of the simulator. Vehicles and traffic lights exchange their information through vehicle and traffic sensors. Vehicles communicate with traffic management module to obtain traffic routing information and feed back real-time traffic information. The information exchanges between traffic lights and traffic management make our simulator achieve real-time traffic control. The input databases save the user-specified information. There are five kinds of input database in this simulator, which are Road Map Database, Vehicle
Generation Database, Vehicle Source and Sink Database, Traffic Light Database, Bus Path and Stop Database. For more information of VATSIM, please refer to Appendix A.

7.2 Applications in VATSIM

There are two aspects in VATSIM adopted the control algorithms proposed in the previous chapters: intersection control and dynamic routing. But both of them are simplified to guarantee the computing speed of VATSIM. In the following, we will explain how the intersection control and dynamic routing methods are simplified and applied in VATSIM.

7.2.1 Intersection Control

VATSIM supports simulations of three types of traffic control devices: normal traffic lights, stop signs and ramp meters. Of them, the control of normal traffic lights is based on real-time traffic information.

In VATSIM, normal traffic light only has red and green phases. The yellow phase is not included to simplify the function of traffic lights. There are two parameters used to characterize its control. One is the cycle time and the other is the split rate between green and red time. In VATSIM, the cycle time is preselected, and the split rate is decided in the real-time. Here, every intersection is considered as an isolated intersection, and the saturated state of intersections is not considered. Therefore, the unsaturated isolated intersection control method is applied in VATSIM, i.e. Eqn. (4.28) and (4.29). Moreover, since only the queue lengths at intersections
are simulated to be measurable in VATSIM, we simplified the control further as

\[ c = \frac{q_x u^m_x}{q_x u^m_x + q_y u^m_x}. \]  

(7.1)

As we know, the above control is designed for intersections with one-way streets. But VATSIM supports intersections with two-way streets which are shown in Figure 7.2. The above equation needs to be modified. For an intersection with two-way streets, if

![Figure 7.2: An intersection with two-way streets](image)

we just consider the two queues at x or y direction to be one queue, i.e., \( q_x = q^1_x + q^2_x \) and \( q_y = q^1_y + q^2_y \), Eqn. (7.1) still can be used, which becomes

\[ c = \frac{(q^1_x + q^2_x) u^m_y}{(q^1_x + q^2_x) u^m_y + (q^1_y + q^2_y) u^m_x}. \]  

(7.2)

Based on this equation and Eqn. (4.29), the real-time intersection control in VATSIM is decided.
7.2.2 Dynamic Routing

In VATSIM, vehicles can be directed by a dynamic routing strategy. We first implemented the decentralized routing strategy proposed in Chapter 3, but it occupied so much CPU time that the simulation was very slow. So we simplified the calculation of cost-to-go of link \((m, n)\) as

\[
S_{mn}^d = \tau_{mn} + q_n \cdot w + \psi_n^d
\]  

(7.3)

where \(w\) is a given weight. Since VATSIM also supports freeway simulation, we have to define the cost-to-go of freeways. Since in Chapter 5 we directed freeway traffic based on the travel times of freeways, we decide to use the average travel time of a freeway as its cost-to-go. The average travel time can be obtained as

\[
\Gamma_{mn} = \frac{d_{mn}}{v_{mn}}
\]  

(7.4)

where \(v_{mn}\) is the average speed of freeway \((m, n)\), which is measured by simulated average speed sensors in VATSIM. Then the calculation of the cost-to-go of a link is concluded as

\[
S_{mn}^d = \begin{cases} 
\tau_{mn} + q_n \cdot w + \psi_n^d & \text{Local road} \\
\frac{d_{mn}}{v_{mn}} + \psi_n^d & \text{Highway}
\end{cases}
\]  

(7.5)

After defining the cost-to-go of links, the dynamic shortest route from a node to another is found by using the Dijkstra's algorithm [45] and saved in the node. When a vehicle arrives at a node, the node will direct the vehicle to take the route with the smallest cost-to-go to its destination. From the above explanation, we can find that the dynamic routing in VATSIM basically follows the algorithms we proposed in Chapter 3 and 5. The only difference is that the calculation of cost-to-go of links is simplified.
7.3 Summary

In this chapter, we provided some application examples of the algorithms proposed in this dissertation. VATSIM, a vehicle and traffic simulator, is developed in The Ohio State University. In this simulator, the simplified intersection control (Chapter 4) and decentralized dynamic routing strategy (Chapter 3 and 5) are applied. From these examples, we can see that the concepts and algorithms discussed in this dissertation are applied in a simulator now. We hope that they can also be applied in the real world in the future.
CHAPTER 8

CONCLUSION AND FUTURE DIRECTIONS

This chapter will summarize the contributions of this dissertation and provide suggestions for further research.

8.1 Contribution

This dissertation has primarily discussed the development of dynamic routing strategies and signal control algorithms for Intelligent Transportation Systems. Specifically,

- A decentralized multi-destination dynamic routing strategy for congested local traffic networks was proposed. This strategy is based on the cost-to-go, whose value includes the information about queue lengths at nodes and travel times along links and is transferred from down-stream nodes to up-stream nodes, to direct the traffic and improve the performance of whole traffic networks.

- A decentralized hybrid intersection control method was developed. Since an intersection with the consideration of the traffic situations at its down-stream intersections and up-stream intersections can be modeled as a finite state machine, a hybrid intersection controller was designed to deal with different states of intersections and improve the efficiency of intersections.
• A user equilibrium two-freeway dynamic routing method via sliding modes was presented. It was proved that under this sliding mode routing method the user equilibrium problem is equivalent to a system optimal routing problem.

• An on-ramp control using sliding mode extremum seeking method was proposed. Since the optimal point of a freeway, i.e. the state with highest throughput, is usually unknown, the sliding mode extremum seeking method is used to search the optimal point and control the system to attain it simultaneously.

This dissertation considers four different roadway network control problems. Although the various problems are presented independently, they have many common features. Among these commonalities is the fact that the applications have the same objective, i.e., to improve the operation of the transportation network by reducing delays and/or increasing throughput. For example, in the local and freeway routing problems, we are trying to minimize the aggregate travel time for all the traffic in the network. The optimization results will affect the efficiency of the whole transportation system. Another commonality is the fact that these problems are dynamic, and thus, that normal static optimization algorithms may not be able to solve them. Therefore, in this dissertation all these problems are analyzed and solved from the perspective of control engineering. We used dynamic models to represent the behaviors of the transportation systems. Then based on the models, we analyzed networks to select proper control methods. For example, in the two freeway applications, we apply the sliding mode control method since the problems are nonlinear.

More importantly, the problems are interrelated. Obviously, the dynamic local and freeway routing are two aspects of the whole transportation network routing.
problem. Likewise, the dynamic routing and control impact one another. In the local routing problem, we had assumed that the intersection control method is fixed; while in the intersection control problem, we had assumed the traffic is routed by a random arrival process. Similarly, the on-ramp control problem is related to the dynamic routing problems. As we know, traffic enters a freeway from local roads via on-ramps. The control of on-ramp meters will affect the traffic situations of both local traffic networks and freeway networks. Integrating the various approaches will be the subject of future research.

8.2 Future Research Directions

Based on our above discussion and the work we have done, there are some possible research directions:

• For the intersection control

  – We only considered very simple intersections in this dissertation, one-way roads with no turning traffic. But in the real world there are many intersections with two-way roads. One of the possible research directions for intersection control is to develop a real-time control method for the intersections with two-way roads and turning traffic.

  – We do not consider the coordination of intersections when we analyzed the intersection control problem in this dissertation. But a good coordination solution of traffic lights along main local roads can reduce the number of stops of vehicles moving on the main local roads and the delay time caused by traffic lights. Another possible future research direction for intersection control is to solve the coordination problem of traffic lights.
• Integration of the routing and intersection control for local traffic networks

• Integrate the routing of local networks and freeway network with considering the effects of on-ramp control

• how to integrate the four control problems discussed in the dissertation, dynamic routing of local traffic networks and freeway networks, the intersection control, and the on-ramp control, as one big problem and solve it
APPENDIX A

VATSIM: A SIMULATOR FOR VEHICLES AND TRAFFIC

A.1 Overview

In this appendix, we will introduce the structure and functions of our vehicle and traffic simulator, VATSIM. This simulator is based on a simple model of vehicle to simulate the movement of individual vehicles such as lane following, lane changing, vehicle following and intersection stopping. In it, different sensors are modeled such that the effects of different sensors on the movement of vehicles can be evaluated. So, VATSIM is basically a vehicle simulator. At the same time, because the models of vehicles and sensors are simple, and the simulator is developed on C in X-window environment, which has fast computing speed, VATSIM can handle hundreds of vehicles simultaneously. Thus, the movement of traffic flow can also be simulated in VATSIM. Furthermore, there is a traffic management module in VATSIM, which collects traffic information and provides traffic control to traffic lights and dynamic routing to vehicles. Different kinds of traffic control and routing strategies can be evaluated. So VATSIM can be used as a traffic simulator.

This chapter is organized as follows: the history of VATSIM is given first, then
the architecture and overall program execution flow of VATSIM are discussed in Section A.3, followed by detailed explanations about the structure, functions and input databases of every element in Section A.4-A.8. Three implementation examples are provided in Section A.9 to show that VATSIM can be used for different applications.

A.2 History of VATSIM

Basically, the process to develop VATSIM is divided into three stages: VATSIM I, II and III, respectively.

In 1998, Keith Redmill [46] finished the first developing stage of the simulator. In this stage, the fundamental structure of VATSIM was defined, such as segment and node structure to represent the geometric information of a traffic network, vehicle structure to represent the static and dynamic information of vehicles. And the overall simulator program execution flow and the format of input/output were determined [47]. VATSIM I is only a vehicle simulator, in which vehicles could be running on traffic networks but just moved around randomly without destination and there is no traffic lights simulated in the networks.

In VATSIM II, VATSIM became a real vehicle and traffic simulator by adding destinations to vehicles and traffic lights to traffic networks. As we know, adding destinations to vehicles is not simply adding a parameter to the vehicle structure. The routing strategies of vehicles to destinations are necessary. Thus, two structures related to routing were defined in this stage. One is for the static routing and the other is for the dynamic routing. To add the traffic lights, we defined a traffic light structure. These routing structure and the traffic light control functions are made up of the traffic management module in the following simulator framework.
In the third stage, we added more features into our simulator. First, the bus function is included. Adding buses to our simulator is not so difficult since the vehicle structure has been well defined. However, to make buses running as real buses, bus stations and people waiting at stations are needed. Therefore, a bus station structure and a bus path structure were defined to save all this information. After defining those new structures, buses can move along the given bus paths and stop at every bus station such that people reaching their destinations can get off and those waiting at the stop can get on. Second, we added a traffic accident detecting function into our simulator. In VATSIM I and II, vehicles were modeled as moving points without occupying any place on maps. Based on this vehicle model, traffic accidents will never happen. To make our simulator represent the real world better, we changed the vehicle model such that every vehicle has its width and length and occupies a specified place according to its position, yaw angle and direction. Then, our simulator can simulate the situation when a traffic accident happens. Now, the third stage have not been finished. We are still using all kinds of efforts to improve our simulator.

From the above explanation, we can see that our simulator, VATSIM, is becoming more and more complete and complex. In the following sections, we will discuss the architecture and functions of VATSIM in detail.

### A.3 Simulation Framework

The simulation framework of VATSIM shown in Figure A.1 includes two parts: the simulator and the input databases.

The Vehicle and Traffic Simulator (VATSIM) is implemented as a modular simulation system, which is made up of four modules: network, vehicles, traffic lights
Figure A.1: Simulation framework

and traffic management. Network includes the part of program related to the input, representation and analysis of the physical information about traffic network such as the road shape, number of lanes and speed limit. Vehicles represent the information of vehicles and drivers, such as the static and dynamic information of different vehicles, the different driving characteristic, etc. Traffic lights save the information about the position, state, cycle time and split rate of traffic lights. Traffic management includes the traffic light control and traffic routing functions. Between these four modules there are information exchanges. The network module provides physical information of simulated traffic networks to the vehicles, traffic lights and traffic management since it is the basis of the simulator. Vehicles and traffic lights exchange their information through different vehicle and traffic sensors. Vehicles communicate with traffic management element to obtain traffic routing information and transfer
back real-time traffic information to traffic management. The information exchanges between traffic lights and traffic management modules make our simulator achieve real-time traffic control.

The input databases save the user-specified information. There are five kinds of input database for this simulator, which are *Road Map Database*, *Vehicle Generation Database*, *Vehicle Source and Sink Database*, *Traffic Light Database*, *Bus Path and Stop Database*. The structures of these databases will be explained along their corresponding elements in VATSIM, in the next sections.

The main simulation structure is a loop over a group of modules at specified frequencies. At the same time, it can respond when certain events occur, such as traffic accidents. The execution flow of the simulator is summarized in Figure A.2.

In the above, the overall design of VATSIM is introduced. The detailed explanation of every module in the simulator is given next.

### A.4 Network

In VATSIM, road networks are represented by segments and nodes. These structures allow the simulation of vehicle and traffic operations in integrated networks of freeways and urban streets. The geometrical information about the simulated network is read from *Road Map Database*, whose specification is given in the following.

#### A.4.1 Segments

In VATSIM, we use only one structure called segment to contain all the information, including the shape of roads, number of lanes and speed limits of roads, etc. The main parameter of a segment is a group of points, which are specified either as latitude-longitude pairs or X and Y coordinates on arbitrary coordinate axes in units
Figure A.2: Simulation flow chart
of meters. The center line of a road is represented as a set of lines which sequentially connect those points. The value of a road curvature is not saved in the segment structure. We use more points to represent a curve road. When a road curvature is larger, the distance between points are smaller and the number of points are larger. And because the straight lines between those points are short, the road looks like having certain curvature. Since in segments the number of lanes in the forward and reverse direction is saved and we fixed the width of every lane as 4 meters, a road with different lanes can be simulated in VATSIM. The lane number is bigger as the lane is farther from the center of the road. Moreover, the information about the node number related to the head and tail of the segment and the bus stop number at the segment is also saved.

A.4.2 Nodes

A node is either an intersection of several local roads, a junction between local roads and freeways, or the start/end point of a road. The node structure contains information about the location of a node, the number of segments in and out of the node and the ID of segments which are in or out of the node. It also contains the index of the traffic light associated with the node.

A.4.3 Road Map Database

The geometry of the road map is given by the Road Map Database, the specification of which appears in Table A.1. The database is structured to accept data from the U. S. Geological Survey (USGS) digitized series of 7.5 minute quadrangle maps with only minor modifications, mainly consisting of stripping out redundant information, converting the data from degrees latitude and longitude to meters distance from an
arbitrary reference point, and the addition of information not present in the USGS databases such as speed limit and number of lanes in each direction.

<table>
<thead>
<tr>
<th>RecID</th>
<th>SegID</th>
<th>NumLaneF</th>
<th>NumLaneR</th>
<th>SpeedLim</th>
<th>NumPoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>20.0</td>
<td>3</td>
</tr>
<tr>
<td>40.006111</td>
<td>-83.014366</td>
<td>40.005932</td>
<td>-83.011002</td>
<td>40.005917</td>
<td>-83.010780</td>
</tr>
</tbody>
</table>

RecID Unique integer ID tag for this road segment
SegID Integer road ID tag
NumLaneF Number of lanes in forward direction
NumLaneR Number of lanes in reversed direction
SpeedLim Speed limit for this segment
NumPoints Number of coordinate pairs defining this segment
Lat$_i$ ($Y_i$) Latitude (degrees) or Y Position (meters) of point $i$
Long$_i$ ($X_i$) Longitude (degrees) or X Position (meters) of point $i$

A.5 Vehicles

For a vehicle and traffic simulator, how to represent vehicles is the most important issue. In VATSIM, a structure called Vehicle is defined. Basically, every vehicle associates with a vehicle structure when the vehicle is generated; and the structure will be deleted when the vehicle arrives at its destination. The data describing the simulated vehicles' generation and their source and sink are input from two input databases: *Vehicle Generation Database* and *Vehicle Source and Sink Database*. In the following, the vehicle structure, its functions and its input databases are explained.
A.5.1 Model

In this simulator, we use a very simple model to represent the dynamics of vehicles. This vehicle model is a highly idealized model of overall vehicle behavior under normal road driving conditions. Low-level models of the vehicle's powertrain, steering system and actuator, tires, and suspension are not considered. The potential for catastrophic failures, such as tire blow out or gross loss of traction, is likewise excluded from the model.

With these simplifications in mind, the longitudinal model of the vehicle is taken as a unity gain first order linear system with a time constant of 10 seconds coupled with an acceleration limit of $0.7 \text{ m/s} \ (0.07g)$ and a deceleration limit of $1.6 \text{ m/s} \ (0.16g)$. The acceleration limits are actually implemented in the vehicle speed regulator, but as they are hard limits we may consider them to be part of the vehicle model for this simulation. The acceleration and deceleration limits were chosen primarily for passenger comfort. The input to the longitudinal model is commanded vehicle velocity and the output is actual velocity.

For the lateral model of the vehicle, we assumed that vehicle yaw rate is linearly proportional to steering angle, where the gain is a function of the mass and dimensions of the vehicle, the approximate velocity, and the effective traction of the tires. This is a reasonable linear assumption for small steer angles, non-slippery roadways, tires with limited tread wear, and highway velocities. The lateral dynamics of the vehicle then become a single integration computing yaw angle (in an absolute fixed-frame ground coordinate system) from yaw rate (in the vehicle's body-fixed coordinate system).

Finally, the yaw angle and instantaneous vehicle velocity in the body frame is transformed onto the ground plane and integrated to obtain absolute position.
The overall vehicle model can be summarized as

\[
\dot{V} = \begin{cases} 
0.7, & (V_c - V) > 7.0 \\
-1.6, & (V_c - V) < -16.0 \\
0.1(V_c - V), & \text{else}
\end{cases}, \quad (A.1)
\]

\[
\dot{\psi} = \alpha \delta, \quad (A.2)
\]

\[
\dot{X} = V \sin \psi, \quad (A.3)
\]

\[
\dot{Y} = V \cos \psi, \quad (A.4)
\]

where \( V \) is the current velocity, \( V_c \) is the commanded velocity whose value is decided by the vehicle longitudinal controller introduced later, \( \delta \) is the steering angle, \( \psi \) is the yaw angle, and \( X \) and \( Y \) are the current vehicle position in the absolute ground reference frame.

**A.5.2 Sensors**

Every vehicle is equipped with five sensors, a forward-looking ranging device which can detect vehicles ahead, a forward-looking lane sensor which can measure vehicle offset from the center of a lane, a forward-looking sensor which can detect the distance to the next intersection (node) in the road map, a forward-looking traffic-light sensor which can detect the color of traffic light at the next intersection, and a lane-changing sensor which can measure the distance to the vehicles at the other lane.

The forward-looking ranging device simulates a scanning laser rangefinder. It will scan \( \pm 30 \) degrees from the current vehicle yaw angle orientation and detect any vehicle within 100 meters. Vehicles that are not in the same lane are then eliminated. The output of this sensor is the distance to the nearest vehicle ahead and its velocity, or 0 if there is no vehicle of interest within 100 meters.
The **forward-looking lane sensor** simulates a forward-looking lane marker detection based sensor with a look-ahead distance of 10 meters. Consider that a fixed camera is mounted in the center of the vehicle, aligned with the vehicle centerline, and pointed such that the center of its region of interest is 10 meters ahead of the vehicle’s center of mass. Based on the current position, yaw angle, and road segment of the vehicle, this sensor generates a position offset measurement from the center of the specified lane (at the look-ahead distance) to the center of the camera’s region of interest. This amounts to the distance from the center of mass of the vehicle projected 10 meters ahead in the current direction of travel to the center of a specified lane 10 meters further down the road. We allow our lane position sensor to function correctly even when it is directed to determine the offset distance from a lane other than the vehicle’s current lane.

The **forward-looking next intersection (node) sensor** determines the identity of the next node the vehicle will encounter as well as the travel distance to that node. In general, it could be implemented as a global positioning system receiver (GPS) coupled with a map database. Given the current accuracy limitations of reasonably priced GPS systems, this approach would need to be augmented with some kind of local stop sign or marker detection system to provide more accurate distance information at short distances from the intersection. For buses, this sensor is also used to detect the next bus-stop and the distance to it.

The **forward-looking traffic light sensor** simulates a camera with a look-ahead distance of 100 meters. This sensor returns the color (Red/Green) of the traffic light at the next intersection such that the vehicle can decide stop (decelerate)/go at the intersection.
The lane-changing sensor is made up of a group of scanning laser rangefinders and cameras to detect any vehicle within ±20 meters in the other lane. Vehicles that are not in the other lane are eliminated. The output of the sensor is the distance to the nearest vehicle in the other lane or 0 if there is no vehicle. This sensor is used to detect whether there is enough open space to make lane-changing.

A.5.3 Controllers

Because this is a micro-simulation with vehicle dynamics, low-level controllers must be provided for vehicle steering control and advanced cruise control. Some of these controllers are similar to the early control laws used on the actual vehicle. That is to say, with the correct gains they will control the actual vehicles, though not necessarily comfortably due to deleterious effects of unmodeled vehicle, road, and disturbance dynamics.

The longitudinal controller sets the commanded vehicle speed $V_c$ to be the desired vehicle velocity $V_{desired}$ unless the vehicle is having a vehicle ahead or approaching an intersection or a bus-station. The value of this desired vehicle velocity $V_{desired}$ is chosen according to the speed-limit of the current road and the driver's personality. In this simulator, three kinds of personalities are modeled, called Conservative, Normal and Aggressive. The desired vehicle speed is selected as

$$V_{desired} = \begin{cases} 
  \text{SpeedLimit} - 5 & \text{Conservative} \\
  \text{SpeedLimit} & \text{Normal} \\
  \text{SpeedLimit} + 5 & \text{Aggressive} 
\end{cases} \quad (A.5)$$

The personality of the drivers of every vehicle is randomly assigned but it can also be preselected according to the requirement of simulation.

When the vehicle ahead is a slower vehicle, the controller decelerates the vehicle to come to a specified following distance $D$ and match the speed of the lead vehicle
according to

\[ V_c = \max \{V_{\text{lead}} + 0.1(D_{\text{ahead}} - D), V_{\text{desired}}\}. \]  

When the vehicle ahead is a faster vehicle, the controller accelerates the vehicle to achieve the same speed, which is

\[ V_c = V_{\text{lead}} - V \]  

When the vehicle is approaching an intersection with a red traffic light, the controller decelerates the vehicle to speed zero at a specified distance \( D \) to the intersection according to

\[ V_c = V_{\text{desired}} * (D_{\text{nextnode}} - D)/100 \]  

The control for buses to arrive at a bus-stop is the same, except that the specified distance \( D = 0 \).

The lateral lane keeping controller is a fixed proportional feedback of the position offset from the lateral sensor.

The lateral changing controller will cause the vehicle to shift one lane to the left if it has been following a vehicle for longer than a prescribed amount of time or a vehicle has to turn right or left but it is in the left or right lane. After guaranteeing that there is enough space in the other lane to make lane-changing, the longitudinal controller accelerates the vehicle and the lateral controller moves the vehicle to the center of the other lane. The passing vehicle will, if possible, shift back after a while if there is enough space in the original lane. Because the lateral position sensor can see multiple lanes, it is sufficient to set the new desired lane and allow the lane keeping controller to move the vehicle into the center of the new lane. Many practical lateral
sensors, including those used on the actual vehicles, can not sense another lane, so a more complicated control strategy must be developed for the actual vehicles.

In this simulator, two different routing controllers are modeled since we want to compare the effect of different routing strategies on the traffic pattern. The first routing controller simulates a static routing strategy. In this strategy, the map information is available. Based on the map information, the routing controller finds the shortest route to the destination and makes the vehicle follow this route. Since the chosen route can not be changed when the vehicle is running and the route is chosen by static information, this routing controller is called the static routing controller. Another routing controller simulates a dynamic routing strategy. In this strategy, the routing controller has communication with every passing node. The node collects and transfers the dynamic traffic information to the routing controller such that the routing controller can choose the real-time optimal route from this node to the destination. Since this routing includes the dynamic traffic information and the chosen route can be changed at every intersection, this routing controller is called dynamic routing controller. How and what information is transferred from nodes to vehicles will be explained in the traffic management section. Basically, the type of the routing controller of every vehicle is randomly chosen, but it can also be given. For example, the routing controller of buses is set as the static routing controller.

A.5.4 Vehicle Generation Database

The method by which vehicles are generated from the simulation is specified in the Vehicle Generation Database, the specification of which appears in Table A.2. The database defines two types of vehicle insertion behavior (State). State 1 allows
Table A.2: Vehicle Generation Input Database Specification

<table>
<thead>
<tr>
<th>RecID</th>
<th>State</th>
<th>Time</th>
<th>Model</th>
<th>SLat (Y₁)</th>
<th>SLong (X₁)</th>
<th>DLong (Y₂)</th>
<th>DLat (X₂)</th>
<th>Velocity</th>
<th>Yaw Angle</th>
<th>Lane</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>15.0</td>
<td>1</td>
<td>39.9975</td>
<td>-82.9913</td>
<td>40.0069</td>
<td>-83.0095</td>
<td>30.0</td>
<td>90.0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

RecID: Unique integer ID tag for this record
State: 1: Insert single vehicle, 2: Insert at regular intervals,
Time: Simulated time or interval at which to insert vehicle
Model: 0: Start and destination point are given,
1: Start and destination point are randomly chosen
2: Start and destination are the same, vehicle is an obstacle
SLat (Y₁): Latitude (degrees) or Y Position (meters) of start point
SLong (X₁): Longitude (degrees) or X Position (meters) of start point
DLat (Y₂): Latitude (degrees) or Y Position (meters) of destination point
DLong (X₂): Longitude (degrees) or X Position (meters) of destination point
Velocity: Desired velocity for vehicle
Yaw Angle: Initial degree of yaw angle for vehicle
Lane: Initial Lane number for vehicle
Mode: 0: Normal Vehicle, 1: Bus

individual vehicle to be inserted at a specific time, whereas State 2 allows the insertion of vehicles at fixed regular intervals. The database specifies the model class number for the vehicle and the initial location, destination, and desired velocity. The model class number of a vehicle is associated with the determining method of its start point and destination. Model 0 means that the start and end point of a vehicle are specified by users. Model 1 means that the start and end point are randomly selected from the nodes in the source and sink database. Model 2 means that the start and end point are specified as the same value and the vehicle will stay at the point for a specified time, i.e., it is an obstacle. Based on the starting node and destination
node of a vehicle, the shortest route or the segment to enter is assigned to the vehicle depending on its routing controller. Initially, the vehicle travels on the road segment with the specified yaw angle at the specified lane. The database also defines two types of vehicles (Mode). Mode 0 means that the vehicle is a normal vehicle and Mode 1 means that it is a bus.

A.5.5 Vehicle Source and Sink Database

As mentioned above, the start and end point of vehicles can be randomly chosen from a user specified source and sink database. Actually, there are two databases: one for source and the other for sink. Since these source and sink databases have the same specification, we call them Vehicle Source and Sink Database. The specification is available in Table A.3.

<table>
<thead>
<tr>
<th>RecID</th>
<th>Lat (Y)</th>
<th>Long (X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>40.0064</td>
<td>-83.0202</td>
</tr>
</tbody>
</table>

RecID Unique integer ID tag for this record
Lat (Y) Latitude (degrees) or Y Position (meters) of node
Long (X) Longitude (degrees) or X Position (meters) of node

A.6 Transit Vehicles

Buses, as a special feature of VATSIM, are introduced in this section. We know that buses move along bus paths, stop at bus stations, and drop and pick up waiting people. Therefore, before adding buses into VATSIM, we have to define bus networks, which are presented by bus paths and stops.
A.6.1 Bus Paths and Stops

Bus paths actually are a special kind of routing table, which is characterized by a set of segment IDs. This set of segment IDs, which determines the roads that buses should take, is given by users in Bus Path Database. The specification of this database is available in Table A.4.

Bus stops are used to represent not only the geometric information such as the locations of bus stops, but also the information of people waiting at bus stops such as the number of waiting people and their destinations, etc. First, the location of a bus stop is corresponding to some point in a segment. And the bus stop has a parameter to show which direction of the segment this bus stop is in. Second, bus stops associate with people waiting for buses. People waiting at a bus stop form a queue and the number of people arriving at the stop within a period is modeled as a random walk process. That is, the period between two people arriving at the bus station is exponentially distributed on a rate $\lambda$. The destinations of those waiting people are uniformly chosen from the bus stops. All these data are read from the Bus Stop Database, whose specification is shown in Table A.5.

<table>
<thead>
<tr>
<th>RecID</th>
<th>NumSeg (n)</th>
<th>Seg1</th>
<th>\ldots</th>
<th>Segn</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>9</td>
<td>\ldots</td>
<td>25</td>
</tr>
</tbody>
</table>

Table A.4: Bus Path Database Specification

RecID | Unique integer ID tag for this record  
NumSeg (n) | Number of segments in path  
Seg1 | The first segment number in path  
Segn | The nth segment number in path
Table A.5: Bus Stop Database Specification

<table>
<thead>
<tr>
<th>RecID</th>
<th>Lat (Y)</th>
<th>Long (X)</th>
<th>Seg</th>
<th>Dir</th>
<th>ArrivalRate (λ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>40.0003</td>
<td>-83.0178</td>
<td>9</td>
<td>1</td>
<td>0.02</td>
</tr>
</tbody>
</table>

- **RecID**: Unique integer ID tag for this record
- **Lat (Y)**: Latitude (degrees) or Y Position (meters) of bus stop
- **Long (X)**: Longitude (degrees) or X Position (meters) of bus stop
- **Seg**: Segment number in which bus stop is
- **Dir**: Traffic direction of bus stop
- **ArrivalRate (λ)**: Random walk arrival rate of waiting people at stop

### A.6.2 Buses

VATSIM characterizes buses by all the information of vehicle structure and a parameter called capacity, which represents the number of people that the bus can contain. The mode of buses is fixed as 1 and their routing controller is always the static routing controller, which obtains the preselect routes from the bus path structure. Moreover, buses do not have destinations. They are running along a loop.

As a special kind of vehicles, the movement of a bus in the network is not only determined by the interaction with vehicles ahead and traffic lights, but also related to the number of people on board and waiting at bus stations. To show the movement related to bus stops more clearly, we define a graphical interface other than the main graphical interface and call it BusWin. In BusWin, an abstract network which is a mapping from the real traffic network is shown. This abstract network is made up of bus stations and simple connections between them which are corresponding to the real routes between stations. The movement of buses in BusWin is discrete since buses are jumping from one station to another corresponding to the positions of buses.
at the main interface. Here, the changes of waiting people at stations and how buses transfer people from stations to stations are emphasized. Therefore, the movements of buses are illustrated in two windows. One is the main graphical interface, which shows the movements of buses as normal vehicles. The other is BusWin, which shows the discrete movements of buses. The seat occupancy of a specified bus is also shown in a small window, called BusInfo. In this window, nonnegative number means that the seat is taken and -1 means that it is empty.

A.7 Traffic Lights

VATSIM supports simulations of several types of traffic control devices: traffic signals, stop signs and ramp metering. Those traffic control devices are characterized by their location, type and initial state. These information is specified by users and read from the Traffic Light Database.

A.7.1 Traffic Control Devices

Normal traffic lights is the kind of traffic light with Red/Green phases. Here the yellow phase is not included to simplify the function of traffic lights. For this traffic control device, two parameters are used to characterize its control. One is the cycle time and the other is the split rate between red and green. The state of the traffic signal is decided by the two parameters. That means, at any given time, the traffic signal is either red or green. In VATSIM, the cycle time and split rate can be preselected or changed in the real-time. This traffic control function will be explained in the traffic management section.

Stop signs include One-way stop sign and three/four way stop sign. Basically, when a vehicle notices that there is a stop sign in front, it has to stop and then go.
Ramp metering is used to control vehicles to enter highways. It also has Red and Green state. When a ramp meter is Green, traffic can enter the highway; when it is Red, traffic can not. In VATSIM, we use a fixed ramp control algorithm, but it can be easily changed to any other on-ramp control methods.

A.7.2 Traffic Light Database

The information related to traffic lights is input from Traffic Light Database, the specification of which appears in Table A.6.

<table>
<thead>
<tr>
<th>RecID</th>
<th>Lat (Y)</th>
<th>Long (X)</th>
<th>Type</th>
<th>Time</th>
<th>Rate</th>
<th>HighwaySegID</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>39.9901</td>
<td>-83.0062</td>
<td>0</td>
<td>100</td>
<td>0.6</td>
<td>-1</td>
</tr>
</tbody>
</table>

RecID: Unique integer ID tag for this record
Lat (Y): Latitude (degrees) or Y Position (meters) of location
Long (X): Longitude (degrees) or X Position (meters) of location
Type: Type 0: Normal Red/Green traffic light,
Type 1: three/four-way stop sign,
Type 2: One-way stop sign, Type 3: On-ramp meter
Time: Cycle time
Rate: Red/Green split rate
HighwaySegID: Highway SegID for On-ramp meter

A.8 Traffic Management

The traffic management element includes three functions: Traffic light control, dynamic routing, and traffic accident determination function.
A.8.1 Sensors

To obtain these traffic management functions, two kinds of sensors are needed. One is the queue length detecting sensor which is used to measure the number of vehicles in the queues formed at every direction of intersections. The other is the average speed sensor which is used to measure the average vehicle speed at highway segments.

The queue length detecting sensor is made up of a group of cameras to measure the number of vehicles in the 150m range of an intersection. The sensor returns the queue length at one direction of the intersection.

The average speed sensor is a set of cameras which are installed along a highway segment. They identify different vehicles and record the time when these vehicles pass by. Dividing the distance between cameras by the time difference between these vehicle passing them, we obtain the speeds of these vehicles. This sensor returns the average value of these speeds.

A.8.2 Traffic Light Control

Based on the method in [42], the real-time traffic light control needs the queue lengths of an intersection at x and y direction. Every cycle time, the time ratio of the intersection changes according to

\[ c = \frac{q_x u_y^m}{q_x u_y^m + q_y u_x^m} \]  \hspace{1cm} (A.9)

where \( c \) is the split rate of the traffic lights, and \( q_x \) and \( q_y \) are the queue lengths at x and y direction. When the intersection connects two-way roads, \( q_x = q_x^1 + q_x^2 \) and \( q_y = q_y^1 + q_y^2 \), where \( q_x^1 \) and \( q_x^2 \) are the two queues at the x or y direction. \( u_x^m \) and \( u_y^m \) are the maximum traffic flow at x and y direction of the intersection.
This is a very simple traffic light control strategy. It is easy to change it to other traffic control methods.

### A.8.3 Dynamic Routing

The dynamic routing needs all the dynamic traffic information of queue lengths and average speeds at highways. Basically, the Dijkstra’s algorithm [45] is used to find the shortest route from one node to another. But here the dynamic travel time is used instead of the lengths of links. This dynamic travel time is computed based on the following equation.

\[
Time_{travel} = \begin{cases} 
\frac{\text{Length}_{link}}{\text{Speed}_{limit}} + \text{queue} \cdot w & \text{Local road} \\
\frac{\text{Length}_{link}}{\text{Speed}_{average}} & \text{Highway}
\end{cases}
\]  

(A.10)

where \(w\) is the weight of queue length. When the travel time is available, the best route can be found for any two nodes in the map and saved in every node. When a vehicle is approaching a node, the best route information will be sent to the vehicle according to its destination such that the vehicle can follow the real-time optimal route.

### A.8.4 Traffic Accident Determination Function

The traffic accident determination function actually is not a function of traffic management, but since it is related to the traffic management, we explain it here.

Originally, our vehicle model is a point model. No traffic accident happens for this kind of vehicle. To make our simulator to simulate more traffic phenomena, we add a width and length to every vehicle such that every vehicle can occupy some space in the simulated network. To judge if two vehicles have accident, the only thing needs to do is to check if these two rectangles have overlap. If yes, an accident happens, and
then the speeds of the two vehicles will be set as zero. They will stay at the accident place for a while and then disappear. Here we try to simulate the police processing and towing time after an accident happens.

A.9 Use Case Example of VATSIM

In this section, we give three examples to show the applications of VATSIM.

A.9.1 Morning Rush Hour

By using VATSIM, many kinds of traffic situations can be simulated. In this example, we try to simulate the morning rush-hour around the campus of The Ohio State University at Columbus.

In Figure A.3, nodes in the upper part of the picture are called upper-town; the nodes in the center represent The Ohio State University (OSU); and those in the lower are downtown of Columbus. In the left side of the figure, there is a road labeled as Rt315, which is a highway. And in the right side, the 3rd avenue is a one-way street. The other roads in the figure are normal local roads. In the morning, traffic from upper-town goes to OSU and downtown. According to our vehicle input database, there are a great number of vehicles which try to merge to the highway at the junction between lane avenue and 315. During the simulation, the phenomenon of stop-and-go due to the amount of traffic merging into highway can be found. And it also can be found that the probability of traffic accidents at that junction is higher than other place.

From the above explanation, we can see that VATSIM is able to be used to simulate different kinds of traffic phenomena. The only thing that needs to do is to specify the input databases properly.
Figure A.3: Morning rush hour
A.9.2 Buses

In this example, the simulation of campus buses at OSU is given. In Figure A.4, the detailed OSU campus map is shown. There are six bus stops, which make the bus path as a loop. Buses, numbered as 1, 24, 43 and 58 in the figure, are running along the loop and picking up the people waiting at the bus stations.

Corresponding to the main interface, we have the BusWin given in Figure A.5. Now, in the abstract OSU bus network, Buses, shown as dark rectangles in the figure, are corresponding to the buses running in the main OSU map window. The passengers waiting at bus stations are represented as some small numbers following the bus stop number and the values of these small numbers are the index of bus stations where these passengers plan to go. When a bus arrives at a bus stop, how many people can get on depends on the number of available seats of the bus.

Using VATSIM, we can simulate vehicles running in different conditions, such as different traffic light setting and control strategy, if the input databases are properly provided. Specially for buses, different number of buses and different people arrival rate at bus stations can also be simulated. Those simulation results are very useful when we evaluate the performance of a traffic control method or a bus system.

A.9.3 Comparison of Vehicles with/without Communication

In this example, VATSIM is used to evaluate the effects of communication to traffic.

Dynamic/Static Routing

We know that two routing strategies, dynamic and static routing, are set up to vehicles. Dynamic routing means that vehicles have communication with intersections
Figure A.4: Buses in main interface
Figure A.5: Buses in BusWin

(nodes), while static routing does not. So one comparison is made between vehicles with dynamic routing and static routing.

Using our simulator VATSIM, we compare these two routing methods. Here, the morning rush hour simulation situation is used. That is, traffic comes from four sources in the north and goes to four sinks in the south. We pick one of the sink and try to compare the number of vehicles reaching the sinks and the time of those vehicles using in a fixed time period with different routing strategies. The mean results in 400 sim-time are given as follows:
<table>
<thead>
<tr>
<th>Static Routing</th>
<th>Dynamic Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Arrival</td>
<td>8</td>
</tr>
<tr>
<td>Using Time (sim-time)</td>
<td>286.7</td>
</tr>
</tbody>
</table>

As can be seen, more vehicles arrive at the sink, and shorter time is used in the dynamic routing than those in the static routing.

**Traffic Accidents**

In VATSIM, the communication between vehicles can also be simulated. Here, we assume if there is communication between vehicles, a vehicle can obtain precision information about position and speed of other vehicles, but if there is no communication between vehicles, those information will be obtained from sensors and have errors.

To compare automated vehicles with/without communication, we carry out the following simulation. In morning rush hour, we count the number of accident during a fixed time period under the situations of vehicles with/without communication. Given that the fixed 400 simulation time period and error ranges for the vehicles without communication as: look-forward distance ±5m, speed of the front vehicle ±2.5m/simtime and lane-changing distance ±5m, and set that an obstacle is in the highway and a vehicle will be broken after it has been running in the network for a while, we obtain the following results.

<table>
<thead>
<tr>
<th></th>
<th>With Communication</th>
<th>Without Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Accidents (mean)</td>
<td>5</td>
<td>12</td>
</tr>
</tbody>
</table>

As can be seen, automated vehicles with communication can decrease the number of accidents. By analyzing our simulation, we find that there are two main reasons why there are accidents when vehicles with communication. One is that now we only assume there is communication between vehicles on the same road. However, when a
vehicle merges to a highway, it needs the distance and speed information of vehicles in another road. Vehicles without those information may result in traffic accidents. The other reason for accidents is that vehicles do not have enough time and distance to reduce their speed when the vehicle ahead is suddenly broken down since in our simulator the brake force of vehicles is saturated. For vehicles without communication case, we can say that errors results in more accidents.

Concluding the results of the above comparisons, our evaluation result is that automated vehicles with communication improve the performance of the whole traffic network. Our simulator can be applied to evaluate the performance of automated vehicles with different controllers and sensors.

A.10 Summary

In this appendix, a vehicle and traffic simulator, VATSIM, supporting evaluation of advance vehicle and traffic control systems was presented. First, the developing history and architecture of VATSIM were provided. Then, the detailed explanation about every element of the simulator, including its structure, functions and input databases, was given. In the end, some application examples were provided to show that VATSIM can be used to evaluate vehicle and traffic control methods. In conclusion, VATSIM is a successful vehicle and traffic simulator. It is being developed more complete.
BIBLIOGRAPHY


