ROBUST-INTELLIGENT TRAFFIC SIGNAL CONTROL WITHIN
A VEHICLE-TO-INFRASTRUCTURE AND VEHICLE-TO-VEHICLE
COMMUNICATION ENVIRONMENT

By
Qing He

Copyright© Qing He 2010

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY

In the Graduate College
THE UNIVERSITY OF ARIZONA
2010
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Qing He entitled Robust-Intelligent Traffic Signal Control within a Vehicle-to-Infrastructure and Vehicle-to-Vehicle Communication Environment and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

___________________________________________________ Date: 07/23/2010
K. Larry Head

___________________________________________________ Date: 07/23/2010
Wei-Hua Lin

___________________________________________________ Date: 07/23/2010
Mark Hickman

___________________________________________________ Date: 07/23/2010
Yi-Chang Chiu

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College. I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

___________________________________________________ Date: 07/23/2010
Dissertation Director: K. Larry Head

___________________________________________________ Date: 07/23/2010
Dissertation Director: Wei-Hua Lin
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Qing He
ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Dr. K. Larry Head, for his guidance and support on this research in past several years. He provides me not only research insights and suggestions, but also the way to work with other people and the way to deal with different kinds of issues. He is a great advisor, as well as a great mentor.

Also I appreciate numerous help and support from my co-advisor Dr. Wei-Hua Lin. I learned so much from him on traffic flow theory, traffic models and logistics. I want to thank Dr. Mark Hickman for teaching me urban planning, transit and traffic assignment.

Also I want to thank Dr. Yi-Chang Chiu for teaching me traffic engineering, traffic assignment and providing lots of research sources. I am grateful to Professor Pitu Mirchandani for his long term support on my research and study in ATLAS center.

The research was supported by Arizona Department of Transportation (ADOT). I want to express my thanks to Faisal Saleem from Maricopa County of Transportation (MCDOT), Ravi Puvvala, Ramesh Siripurapu and Sreenivasulu Guduguntla from Savari Networks, Michael Hicks and Simon Ramos from City of Tucson Department of Transportation.

Special thanks to my fellow graduate student Jun Ding in the lab. Jun helps me solve the problems on the project, share the research experience, and build the simulation models in my dissertation.

Last but least, I wish to express my deepest gratitude to my wife Cheng Zhu and my parents who have been constant source of motivation, encouragement and inspiration.
DEDICATION

To my wife Cheng Zhu,

Thanks for your love, strength and patience in my life.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................... 11

LIST OF TABLES ............................................................................................................... 15

ABSTRACT ....................................................................................................................... 16

CHAPTER 1: INTRODUCTION ......................................................................................... 19

1.1 Current traffic control systems ............................................................................. 20

1.2 IntelliDrive<sup>SM</sup> - advanced vehicle communications and capabilities ............ 24

1.3 Research objectives ............................................................................................... 28

1.3.1 Pseudo-lane-level GPS positioning ................................................................. 28

1.3.2 Robust multiple priority control within a v2x environment ......................... 30

1.3.3 Multi-modal traffic control within a v2x environment ................................... 33

1.4 Summary of the dissertation ............................................................................... 35

CHAPTER 2: LITERATURE REVIEW ............................................................................ 39

2.1 U.S. IntelliDrive<sup>SM</sup> (VII) ................................................................................. 39

2.1.1 California ........................................................................................................... 40

2.1.2 Michigan .......................................................................................................... 40

2.1.3 New York ......................................................................................................... 42

2.1.4 Virginia ........................................................................................................... 42

2.1.5 Arizona ............................................................................................................ 43
TABLE OF CONTENTS – *Continued*

2.2 Traffic Signal Control ........................................................................................................ 44
   2.2.1 Fixed-time traffic signal control ................................................................................. 45
   2.2.2 Actuated traffic signal control .................................................................................. 49
   2.2.3 Adaptive traffic signal control ............................................................................... 56

2.3 Probe vehicle technology ............................................................................................... 59
   2.3.1 Traffic state and travel time estimation with probe data ........................................... 59
   2.3.2 Traffic control with probe data ................................................................................. 64

2.4 Summary ....................................................................................................................... 65

CHAPTER 3: PSEUDO-LANE-LEVEL, LOW-COST GPS POSITIONING WITH
VEHICLE-TO-INFRASTRUCTURE COMMUNICATION AND DRIVING EVENT
DETECTION ......................................................................................................................... 66

3.1 Introduction .................................................................................................................... 66

3.2 The V2I positioning environment .................................................................................. 69

3.3 Extended Kalman filter ............................................................................................... 75

3.4 Lane status monitor – EWMA SPC control chart ......................................................... 81

3.5 Turn event-driven lane alignment optimization .............................................................. 91

3.6 Field data results ......................................................................................................... 96

3.7 Summary ....................................................................................................................... 99
TABLE OF CONTENTS – Continued

CHAPTER 4: ROBUST ACTUATED PRIORITY TRAFFIC SIGNAL CONTROL
WITH VEHICLE-TO-INFRASTRUCTURE COMMUNICATIONS......................... 101

4.1 Introduction ........................................................................................................... 102

4.2 Phase-time diagram: a new tool to model traffic signal controller logic .......... 108

4.3 Deterministic mixed integer linear program (MILP) formulation ................. 113

4.4 Robust mixed integer linear program (MILP) formulation ............................ 115

4.5 Integration of robust MILP formulation and actuated control ....................... 119

4.5.1 Green extension representation in MILP...................................................... 119

4.5.2 Green extension group (GEG) and $a_{pk}$ reassignment .............................. 122

4.6 Robust signal coordination with priority......................................................... 127

4.7 Numerical experiments ...................................................................................... 131

4.8 Summary ............................................................................................................ 142

CHAPTER 5: A HEURISTIC ALGORITHM TO IMPLEMENT MULTIPLE PRIORITY
CONTROL FOR EMBEDDED CONTROL AT A SINGLE INTERSECTION.......... 144

5.1 Introduction ........................................................................................................... 144

5.2 Simplification of multiple priority control problems ...................................... 145

5.2.1 Simplification to a polynomial solvable problem........................................ 145

5.2.2 Revised exhaustive search algorithm and delay evaluation ......................... 149
### TABLE OF CONTENTS – *Continued*

5.2.3 Simulation results ................................................................. 153

5.3 Field implementation ............................................................... 157

5.3.1 System structure ................................................................. 157

5.3.2 RSE and OBE introductions ................................................... 160

5.3.3 NTCIP implementation ......................................................... 165

5.3.4 DSRC field experiences ....................................................... 167

5.3.5 Field tests ........................................................................ 169

5.4 Summary ............................................................................. 178

CHAPTER 6: PAMSCOD: PLATOON-BASED ARTERIAL MULTI-MODAL SIGNAL CONTROL WITH ONLINE DATA ......................................................... 180

6.1 Introduction ........................................................................ 180

6.2 Platoon-based signal optimization ........................................... 182

6.2.1 Hierarchical platoon recognition ......................................... 183

6.3 Mixed integer linear program (MILP) in PAMSCOD ................. 188

6.3.1 Unified precedence constraints ........................................... 195

6.3.2 Platoon delay categorization and evaluation ......................... 201

6.3.3 Multi-modal dynamical progression .................................... 212

6.3.4 Formulation enhancement ............................................... 216
TABLE OF CONTENTS – Continued

6.3.5 Objective and summary of PAMSCOD ......................................................... 218
6.4 Design of experiment ............................................................................................ 220
  6.4.1 Illustration of solutions from PAMSCOD ...................................................... 222
  6.4.2 Experiment results .......................................................................................... 232
6.5 Summary ............................................................................................................... 234

CHAPTER 7: CONCLUSIONS AND FUTURE RESEARCH ................................. 236
7.1 Summary of the Research ..................................................................................... 236
  7.1.1 Pseudo lane-level GPS positioning in a v2x environment ............................. 237
  7.1.2 Multiple priority control and field implementation in a v2x environment..... 238
  7.1.3 Multi-modal traffic control in a v2x environment........................................... 239
7.2 Future Research Topics ......................................................................................... 241

REFERENCES ............................................................................................................... 243
LIST OF FIGURES

Figure 1.1. Typical intersection with detectors................................................................. 22
Figure 1.2 IntelliDrive\textsuperscript{SM} (VII) architecture data flow (Faradyne 2005)............... 26
Figure 1.3 Intersection layout with multiple priority requests................................. 31
Figure 2.1 Actuated phase timing diagram (Federal Highway Administration 2008) ..... 50
Figure 2.2 Coordination on a time-space diagram (Sunkari et al. 2004)............... 52
Figure 2.3 (a) Pre-defined splits; (b) Fixed force-offs implementation; (c) Floating force-offs implementation. ................................................................. 54
Figure 3.1 Intersection layout defined in a MAP....................................................... 69
Figure 3.2 The GPS error in the test site in Tucson, AZ ........................................ 71
Figure 3.3 A driving event of making a turning maneuver and the actual and the uncorrected measured position before and after the event......................... 73
Figure 3.4 GPS positioning System Structure ............................................... 74
Figure 3.5 A lane changing model ........................................................................ 84
Figure 3.6 The relationship between $\sigma_{\text{lane}}$ and $T$, given $\lambda = 0.4$, $l_a = 3.2m$ and $v_k = 13.33m/s$ ...................................................................................... 87
Figure 3.7 (a) An example of vehicle events detection; (b) Events detection with EWMA control chart .......................................................... 89
Figure 3.8 (a) GID map with inbound-outbound trajectory of two type turns; (b) Line approximations for inbound-outbound trajectories of right turn ............... 93
Figure 3.9 The test intersection in Tucson, AZ ......................................................... 96
LIST OF FIGURES – Continued

Figure 3.10 GPS noncommon-mode error fixed by lane alignment optimization for the left turn event ................................................................. 97

Figure 3.11 Lane change detection by SPC control chart .......................................... 98

Figure 4.1 Dual-ring, eight-phase controller ................................................................. 109

Figure 4.2 Precedence graph representation of a dual-ring controller ..................... 110

Figure 4.3 Phase-time diagram representation of a dual-ring controller ................. 110

Figure 4.4 (a) A cyclic serving rectangle (CSR) for an interval priority request; (b) Four possible service cases for a CSR ........................................ 118

Figure 4.5 Representation of flexible signal plan for actuated control .................... 120

Figure 4.6 The procedure for generating GEG in a ring ............................................ 124

Figure 4.7 (a) An example of green extension group and reassignment; (b) Green extension reassignment in GEG2 .............................................. 125

Figure 4.8 Priority requests with coordination ......................................................... 131

Figure 4.9 Evaluation platform ............................................................................... 132

Figure 4.10 Layouts of a two-intersection arterial ..................................................... 133

Figure 4.11 Overall vehicle delay and bus delay under “ASC-TSP coord.” and “Robust priority coord.” ................................................................. 140

Figure 4.12 Side bus delay and left-turn delay under “ASC-TSP coord.” and “Robust priority coord.” ................................................................. 140

Figure 4.13 Overall vehicle delay and bus delay with or without priority control ....... 141

Figure 5.1 Illustration of separation of \(|J_\rho|\) requests into \(|K|\) cycles ......................... 148
LIST OF FIGURES – Continued

Figure 5.2 Priority request assignment table for phase $p$ in two cycles....................... 148

Figure 5.3 Revised exhaustive algorithm to find the near-optimal solutions .................. 151

Figure 5.4 (a) CSRs on the phase-time diagram (b) Phase-time diagram evaluation of an assignment of serving $R_{22}$ in the second cycle................................................................. 153

Figure 5.5 Performance comparisons of five different methods (a) Average total vehicle delay; (b) Average bus delay; ........................................................................................................... 155

Figure 5.6 Test intersection layouts at Southern Ave. & 67 Ave. Phoenix, AZ (Google Earth)............................................................................................................................... 158

Figure 5.7 System structure of field implementation...................................................... 159

Figure 5.8 RSE installations (a) Inside cabinet (for demonstration purposes); (b) On a pole.................................................................................................................................. 161

Figure 5.9 OBE installation ............................................................................................ 163

Figure 5.10 Web page displays (a) OBE approaches a intersection; (b) OBE receives MAP and request table; (c) OBE sends a priority request; (d) OBE travels through the intersection. (GID is a former name of MAP in SAE J2735)................................. 165

Figure 5.11 NTCIP tree structure of reds group 1 in ASC controller......................... 167

Figure 5.12 DSRC communication ranges: (a) A urban intersection; (b) A suburban intersection................................................................................................................................. 169

Figure 5.13 Field tests (a) REACT vehicles ready to test; (b) A priority request is being served................................................................................................................................. 172
LIST OF FIGURES – Continued

Figure 5.14 (a) CSRs of two requests on the phase-time diagram (b) Phase-time diagram evaluation of an cut of serving both requests in the first cycle.........................177

Figure 6.1 (a) Original point vehicles on a link; (b) Platoon one-dimensional mapping and 2-level clustering with 2s and 1s headways respectively........................................185

Figure 6.2 (a) Intersection layout; (b) A dual-ring, eight-phase controller ..............196

Figure 6.3 Precedence graph representation of a dual-ring controller at intersection \( i \) . . 197

Figure 6.4 An example illustrate currently active phases at intersection \( i \) .................197

Figure 6.5 Platoon serving cycle selection ....................................................................203

Figure 6.6 (a) Queue delay when two platoons are served in the same cycle; (b) Queue delay and signal delay when two platoons are served in different cycles ...........209

Figure 6.7 Signal delay evaluation for the leading platoon .............................................211

Figure 6.8 Queue delay and signal delay at downstream intersection..........................214

Figure 6.9 Evaluation platform......................................................................................221

Figure 6.10 Speedway arterial in Tucson, AZ.................................................................224

Figure 6.11 Comparisons between optimized cycle length from SYNCHRO and average cycle length in PAMSCOD at two critical intersections. (a) Intersection at Euclid & Speedway; (b) Intersection at Campbell & Speedway. ..............................................226

Figure 6.12 Online optimized signal plan from PAMSCOD when SR=0.9 in set 3 (intersection 1 is in the bottom) ......................................................................................227

Figure 6.13 (a) Network throughput; (b) Average all vehicle delay on each phase at each intersection. (c) Average bus delay on each phase at each intersection. .........................230
LIST OF TABLES

Table 3.1 Probability of no lane drift for turns in NGSIM data ........................................ 94
Table 3.2 Experimental Results ........................................................................................ 99
Table 4.1 Symbol definition of decision variables and data ............................................. 106
Table 4.2 Traffic volume for speedway ........................................................................ 134
Table 4.3 Description of compared different methods ................................................... 134
Table 4.4 Basic optimal coordination timing plan from SYNCHRO ......................... 135
Table 4.5 Measured average delay under each scenario with eight different methods .. 138
Table 4.6 Delay changes from ASC-TSP to robust priority (%) ................................. 142
Table 5.1 Delay increment (%) from “Robust free” to “Heuristic” ............................ 156
Table 5.2 Field test scenarios and results ................................................................... 170
Table 5.3 Field test data with two conflicting priority requests on March 10, 2010 .... 174
Table 6.1 Notation definition of decision variables (lower case) and data (upper case) 190
Table 6.2 Four sets of stochastic traffic volume under seven different saturation rates 225
Table 6.3 Performance comparisons with four methods .............................................. 227
Table 6.4 Performance improvements of PAMSCOD compared with other methods ... 234
ABSTRACT

Modern traffic signal control systems have not changed significantly in the past 40-50 years. The most widely applied traffic signal control systems are still time-of-day, coordinated-actuated system, since many existing advanced adaptive signal control systems are too complicated and fathomless for most of people. Recent advances in communications standards and technologies provide the basis for significant improvements in traffic signal control capabilities. In the United States, the IntelliDrive\textsuperscript{SM} program (originally called Vehicle Infrastructure Integration - VII) has identified 5.9GHz Digital Short Range Communications (DSRC) as the primary communications mode for vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) safety based applications, denoted as v2x. The ability for vehicles and the infrastructure to communicate information is a significant advance over the current system capability of point presence and passage detection that is used in traffic control systems. Given enriched data from IntelliDrive\textsuperscript{SM}, the problem of traffic control can be solved in an innovative data-driven and mathematical way to produce robust and optimal outputs.

In this doctoral research, three different problems within a v2x environment—“enhanced pseudo-lane-level vehicle positioning”, “robust coordinated-actuated multiple priority control”, and “multimodal platoon-based arterial traffic signal control”, are addressed with statistical techniques and mathematical programming.

First, a pseudo-lane-level GPS positioning system is proposed based on an IntelliDrive\textsuperscript{SM} v2x environment. GPS errors can be categorized into common-mode errors and noncommon-mode errors, where common-mode errors can be mitigated by
differential GPS (DGPS) but noncommon-mode cannot. Common-mode GPS errors are cancelled using differential corrections broadcast from the road-side equipment (RSE). With v2i communication, a high fidelity roadway layout map (called MAP in the SAE J2735 standard) and satellite pseudo-range corrections are broadcast by the RSE. To enhance and correct lane level positioning of a vehicle, a statistical process control approach is used to detect significant vehicle driving events such as turning at an intersection or lane-changing. Whenever a turn event is detected, a mathematical program is solved to estimate and update the GPS noncommon-mode errors. Overall the GPS errors are reduced by corrections to both common-mode and noncommon-mode errors.

Second, an analytical mathematical model, a mixed-integer linear program (MILP), is developed to provide robust real-time multiple priority control, assuming penetration of IntelliDriveSM is limited to emergency vehicles and transit vehicles. This is believed to be the first mathematical formulation which accommodates advanced features of modern traffic controllers, such as green extension and vehicle actuations, to provide flexibility in implementation of optimal signal plans. Signal coordination between adjacent signals is addressed by virtual coordination requests which behave significantly different than the current coordination control in a coordinated-actuated controller. The proposed new coordination method can handle both priority and coordination together to reduce and balance delays for buses and automobiles with real-time optimized solutions.

The robust multiple priority control problem was simplified as a polynomial cut problem with some reasonable assumptions and applied on a real-world intersection at Southern Ave. & 67 Ave. in Phoenix, AZ on February 22, 2010 and March 10, 2010. The
roadside equipment (RSE) was installed in the traffic signal control cabinet and connected with a live traffic signal controller via Ethernet. With the support of Maricopa County’s Regional Emergency Action Coordinating (REACT) team, three REACT vehicles were equipped with onboard equipments (OBE). Different priority scenarios were tested including concurrent requests, conflicting requests, and mixed requests. The experiments showed that the traffic controller was able to perform desirably under each scenario.

Finally, a unified platoon-based mathematical formulation called PAMSCOD is presented to perform online arterial (network) traffic signal control while considering multiple travel modes in the IntelliDriveSM environment with high market penetration, including passenger vehicles. First, a hierarchical platoon recognition algorithm is proposed to identify platoons in real-time. This algorithm can output the number of platoons approaching each intersection. Second, a mixed-integer linear program (MILP) is solved to determine the future optimal signal plans based on the real-time platoon data (and the platoon request for service) and current traffic controller status. Deviating from the traditional common network cycle length, PAMSCOD aims to provide multi-modal dynamical progression (MDP) on the arterial based on the real-time platoon information. The integer feasible solution region is enhanced in order to reduce the solution times by assuming a first-come, first-serve discipline for the platoon requests on the same approach. Microscopic online simulation in VISSIM shows that PAMSCOD can easily handle two traffic modes including buses and automobiles jointly and significantly reduce delays for both modes, compared with SYNCHRO optimized plans.
Traffic signals play an important role in the transportation network of urban areas. With correct installation and control strategy, they can improve both traffic throughput and the safety of all road users. As indicated in the 2007 National Traffic Signal Report Card, an optimally operated traffic signal can reduce traffic delay by 15~40 percent, fuel consumption up to 10 percent, and harmful emission up to 22 percent (National Transportation Operations Coalition 2007).

Today, there are more than 300,000 traffic signals in the United States (National Transportation Operations Coalition 2007). The use of traffic signals at a busy intersection in a typical urban area might direct the movement of as many as 100,000 vehicles per day. It is estimated that many of these signals could be improved by updating equipment or by simply adjusting and updating the timing plans. Outdated or poor traffic signal timing accounts for a significant portion of traffic delay on urban arterials and traffic signal retiming is one of the most cost effective ways to improve traffic flow and is one of the most basic strategies to help mitigate congestion. The U.S. Department of Transportation’s (U.S. DOT) Intelligent Transportation Systems Joint Program Office maintains a benefit-cost database that documents many traffic signal studies from across the United States. These studies show that there is as much as a 1:40 cost-benefit from
signal retiming. That is, the benefits of investing in signal timing outweighs the costs by as much as 40:1 (Sunkari 2004).

Despite their important role in traffic management, traffic signals, once installed, are often not proactively managed. Maintenance activities are frequently delayed or canceled in reaction to shrinking budgets and staffs. More than half of the signals in North America are in need of repair, replacement, or upgrading (Federal Highway Administration 2008). In 2007, the National Traffic Signal Report Card concluded that the nation scored a ‘D’ in terms of the overall quality of traffic signal operation, if the nation supported its signals at an ‘A’ level.

Therefore, traffic signal timing plays a critical role within the overall transportation network. Signal timing offers the opportunity to improve the mobility and safety of the signalized street transportation system as well as to improve environmental conditions that result from vehicle emissions due to inefficient signal operations. According to the many improper implementations in United States (Federal Highway Administration 2008), there is a growing opportunity to improve the strategy of traffic signal control. This dissertation is intended to address opportunities to improve traffic signal control under new and emerging opportunities in advanced communications and an enriched data environment in a multi-modal transportation environment.

1.1 Current traffic control systems

The state-of-the-art in traffic control today is actuated traffic signal control where different conflicting movements of vehicles are controlled by phases that are called by
detectors when vehicles are present. Figure 1.1 depicts a typical intersection where presence detectors exist at each of the minor traffic movements - which include the main street left turns and side street movements. These presence detectors will call (e.g. request service) from the associated movement phase when a vehicle arrives at the detector. In addition, sometimes passage detectors are used to call a phase as well as to extend the green time as a vehicle approaches the intersection. Extension intervals are typically timed to provide sufficient time for a vehicle to clear the intersection stop bar after they cross the fixed detector position. The details of current traffic control systems will be introduced in Chapter 2.

Adjacent intersections, or collections of intersections on an arterial or a grid, can be coordinated to allow vehicles to progress along the arterial or desired direction of vehicle travel. Each intersection operates using the same actuated control principles as single, isolated intersection, but constrained by a timing plan that can provide coordination through a fixed cycle length, offset, and phase splits.
Figure 1.1. Typical intersection with detectors

The cycle length is typically chosen to provide sufficient time to serve all vehicles and pedestrians on all movements. Phases that are not called may be skipped and, if so, return their allocated green split time to the following phase or to the main-street coordinated phase. The offset is defined to be the amount of time between start of green at one intersection and the associated green at a downstream intersection, and is selected to provide progression in at least one direction (generally the direction with the largest volume of traffic) and possibly both directions. Generally, the offset is set to be the travel time between intersections plus some time for a standing queue to clear before a vehicle from the upstream signal would arrive. The splits are selected to provide sufficient time for each phase to serve the traffic movement demand. Minor phases are allowed to gap
out before the entire split is timed if no vehicles generate/request an extension from the

detectors.

A signal timing plan that contains a cycle length, offset, and phase splits may be
defined for each characteristic traffic pattern that might result from time-of-day demand -
such as morning and evening commutes, or special events such as sporting events,
school, shopping center activities, etc. Signal timing plans are sometimes defined for
special weather conditions such as snow, ice, and rain. Generally, signal timing plans are
selected on a time-of-day basis, but plans can be selected manually from a traffic control
center or a closed-loop master, or using a traffic responsive method that looks at data
from system detectors and selects a plan that has been defined as a good match based on
volume, occupancy, and/or speed data.

Modern traffic control systems have not changed significantly in the past 40-50 years.
Most major cities today have traffic control systems that include coordinated-actuated
controllers that can “adapt” to minor changes in traffic demand on a per-movement basis
as determined by simple point detection systems. Most of these systems have a higher
level control system, either a central traffic management system or a closed-loop master,
where different signal timing plans can be activated based on either a pre-determined
time-of-day schedule or, sometimes, based on system detectors that measure volume and
occupancy at single points on network links.

While these systems have performed adequately for long periods of time, there has
continually been a desire to develop traffic adaptive traffic signal control systems that can
make changes in how the traffic signals provide service so that they can improve
efficiency. Early traffic adaptive systems made minor adjustments to signal timing parameters based on observed patterns of traffic flow, but these systems were responding after the traffic flow had changed and were essentially reactive. In the past decade several attempts to develop proactive (or prediction based) adaptive control systems have occurred. There has been some moderate progress, but typically these systems have depended on elaborate communications, computation, and detection system that are difficult to maintain and require highly specialized knowledge and understanding to operate.

1.2 IntelliDrive<sup>SM</sup> - advanced vehicle communications and capabilities

With the advent of IntelliDrive<sup>SM</sup> for Mobility (RITA 2010), soon it will be possible to obtain additional network and vehicle operation information. IntelliDrive<sup>SM</sup>, formerly known as vehicle infrastructure integration (VII) (AASHTO 2009), is a suite of technologies and applications that use wireless communications to provide connectivity that can deliver transformational safety, mobility, and environmental improvements in surface transportation. IntelliDrive<sup>SM</sup> applications provide connectivity:

- with and among vehicles (V2V)
- between vehicles and the roadway infrastructure (V2I)
- among vehicles, infrastructure, and wireless devices (consumer electronics, such as cell phones and PDAs) that are carried by drivers, pedestrians, and bicyclists (v2x)
Like the Internet, which provides information connectivity, IntelliDrive\textsuperscript{SM} provides a starting point for transportation connectivity that ultimately will enable countless applications and spawn new industries. Today only the tip of the iceberg has been seen.

The two dominant communication channels will be 5.9 GHz Dedicated Short Range Communications (DSRC) with communication distance 500~1000 meters (Y. Liu et al. 2005) and cell phone based data connectivity with bandwidths of several 100 Kbit/sec. This will enable the vehicle to send and receive messages to and from other vehicles and the infrastructure to enhance safety and to provide probe vehicle data. Equipping vehicles with DSRC will also necessitate the installation of Global Positioning System (GPS) so that positioning capability will be available on all vehicles that communicate. The vehicles will send out dynamic data (e.g. vehicle position, speed, heading, acceleration, yaw rate, steering wheel angle, etc.), vehicle status data (e.g. electronic stability system data, wheel slip, anti-lock brake status, turn signals, windshield wiper status, rain sensor data, etc.) and possibly data from other autonomous safety systems on the vehicle (e.g. vision systems, forward collision radars and lidars). The entire IntelliDrive\textsuperscript{SM} system structure is shown as Figure 1.2.
Figure 1.2 IntelliDrive$^\text{SM}$ (VII) architecture data flow (Faradyne 2005)

The On-Board Equipment (OBE) is the vehicle side of the IntelliDrive$^\text{SM}$ system, as depicted in Figure 1.2. OBE’s are used to describe the functions performed within the vehicle in addition to the radio transmission element. An OBE is logically composed of a 5.9 GHz DSRC transceiver (OBE), a GPS system, an applications processor and interfaces to vehicle systems and the vehicle’s human machine interface (HMI) (Faradyne 2005). OBEs provide the communications both between the vehicles and the road-side units (RSU) and between the vehicle and other nearby vehicles. The OBEs may regularly transmit status messages to other OBEs to support safety applications. The OBEs may also gather data to support public applications. The OBEs will accommodate storage of many snapshots of data, depending upon its memory and communications capacity. After some period of time, the oldest data may be overwritten. The OBEs also
assembles vehicle data together with GPS data as a series of snapshots for transmission to the Roadside Equipments (RSEs).

RSEs may be mounted at interchanges, intersections, and other locations providing the interface to vehicles within their range. An RSE is composed of a DSRC transceiver, an application processor, and interface to the IntelliDriveSM back-haul communications network. A RSE may have a GPS unit attached. The RSE is connected to the IntelliDriveSM back-haul communications network that provides communications services between application servers and vehicles. Using its interface to the IntelliDriveSM back-haul communications network, it forwards probe data to the IntelliDriveSM message switches and can send private data to and from the OEMs (Original Equipment Manufacturer).

The RSE may also manage the prioritization of messages to and from the vehicle. Although the OBE has priorities set within its applications, prioritization must also be set within the RSE to ensure that available bandwidth is not exceeded. Local and vehicle-to-vehicle safety applications have the highest priority; messages associated with various public and private network applications have lower priority. Entertainment messages will likely have the lowest priority.

Although it is anticipated that infrastructure instrumentation of IntelliDriveSM may take many years, including time for instrumented vehicle penetration into the market, it is desirable to develop new traffic signal control algorithms and traffic management strategies that respond to changing traffic and environmental conditions to mitigate congestion.
1.3 Research objectives

In this doctoral research, three objectives are aimed under a v2x environment:

- Lane-level GPS positioning with sole GPS and v2x communications for enhanced safety control.
- Real-time robust multiple priority control with current coordinated-actuated traffic signal control system, assuming the penetration of IntelliDrive℠ is up to privileged vehicles.
- Online multi-modal traffic control with high penetration of IntelliDrive℠ in all vehicles.

1.3.1 Pseudo-lane-level GPS positioning

A key capability necessary for successful and wide-scale deployment of v2x applications is the ability to provide lane-level estimation of vehicle position. First, lane level position data enhances roadway safety by supporting collision avoidance system, such as Cooperative Intersection Collision Avoidance Systems (CICAS) in United States (Amanna 2009). Second, lane control with different advisory speed and lane restriction are feasible when lane level positioning is available. Third, driving behavior, such as lane changing, can be studied intensively based on lane level positioning data. Forth, lane level positioning can also benefit traffic operations and control; for example, lane level queue length could be obtained. This is a significantly challenging technical problem that
must engage public sector infrastructure as well as advanced vehicle technologies for positioning including the global positioning system (GPS), inertial navigation (INS), and other technologies such as vision based, radar and lidar sensors, and magnetic roadway markers. No single technology is currently capable of providing the required fidelity and reliability of position estimates. A solution that best leverages both the infrastructure capabilities and advanced positioning technologies is needed.

To be successful for wide scale deployment this solution must also be cost effective. A solution that is too expensive is unlikely to be widely deployed and supported. Modern vehicle positioning technology is capable of providing high fidelity positioning using a combination between Differential GPS (DGPS) and inertial navigation systems. These systems are generally very expensive - in the order of $20K-$80K - and are hence too expensive for wide scale deployment. Technological solutions using a combination of lidar, video, GPS, and inertial navigation have the potential to achieve the high level of accuracy, but are likely to be expensive and susceptible to environmental conditions as well as GPS interruptions.

The standard deviation of a non-differential low cost GPS position estimates is on the order of 10-20 meters (J. Farrell & Barth 1999) and (J. Farrell et al. 2003). This level of accuracy is not sufficient to estimate the vehicle lane status, which could be used to track the turning proportion and analyze driving behavior. Traffic density and queue length measurement and/or estimation could also be affected by the GPS positioning inaccuracy. Therefore, the first goal of this research is to provide a method that will correct low cost GPS error and achieve pseudo-lane level GPS positioning, where lane change behavior
could be tracked by the OBEs. In this context, pseudo means that lane level accuracy is achieved only under the assumption that v2x is available and there is no GPS outage.

1.3.2 Robust multiple priority control within a v2x environment

A variety of challenges as well as opportunities arise when considering traffic control within a v2x environment. One of the opportunities is that different classes of vehicles, such as passenger cars, transit vehicles (buses, light rail, street cars), trucks, and emergency vehicles, can be identified and can request priority signal timing treatment to allow multi-modal timing considerations. This opportunity presents some significant challenges including how to implement priority operations that resolve multiple conflicting requests from different classes of vehicles - such as emergency vehicles and buses, as shown as Figure 1.3. With V2I communication in an IntelliDriveSM world, vehicle information will be able to be obtained up to 1000 meters away from the road-side equipment (RSE) near the intersection.
Figure 1.3 Intersection layout with multiple priority requests

Traditional priority control system in United States can be categorized into emergency vehicle preemption and transit signal priority (TSP). Emergency vehicle can request signal preemption treatment by using either optical, acoustic, special inductive loop, or Global Positioning System (GPS) technology (Nelson & D. Bullock 2000). Preemption generally involves a control strategy that immediately switches from current phase to a pre-selected phase for the first received request. Transit vehicles can be served by either passive priority or active priority systems. Passive priority timing is achieved when signal plan parameters (offsets, green splits, phase insertion or rotation) are tuned in favor of the movements of transit vehicles (Evans & Skiles 1970)(Yagar & Han 1994)(Balke et al.
Active priority systems involve adapting the signal timing by extending the green or providing early green.

In current emergency vehicle preemption systems, only one request can be served at a time. Therefore, multiple requests with conflicting phases could create unsafe conditions resulting in situations where emergency vehicle accidents may occur (The Transportation Safety Advancement Group 2010). With V2I communication systems, the road-side equipment (RSE) can receive requests from multiple vehicles, prioritize the requests based on class and time, then work with the traffic signal controller to generate an optimal signal timing plan that simultaneously accommodates multiple requests in a safe and efficient manner. The RSE can also send request list information back to each of the requesting OBEs to enhanced intersection safety by providing feedback so that each vehicle is aware of the other vehicles on conflicting approaches.

Requests from multiple transit vehicles are pretty common in high population density urban areas. In other words, it is likely that more than one transit vehicle would approach an intersection at any time. The transit network is composed of different level (class) of bus lines (e.g. express and local) and bus frequency is generally high in large metropolitan areas of cities like New York City, Los Angeles and other major cities. One or more buses may arrive on one or more approaches of an intersection during any cycle. Every bus request has its own characteristics (e.g. class, lateness, occupancy) and the efficiency of signal priority for them would be different depending on the situation. Bus occupancy and adherence to the schedule could be considered for real-time active priority
control. Providing priority for a bus with high occupancy or late of schedule is much more efficient for an empty bus or a bus that is ahead of schedule.

The research issues to be addressed in priority control are as follows:

1. Show that first-come first-serve is not efficient (Head et al. 2006a) and formulate a mathematical program that simultaneously considers multiple priority requests in optimizing signal timing.

2. Develop a robust and reliable signal timing solution that accounts for the uncertainty of the traffic state (e.g. queue length) that may effect the actual arrival times of priority requesting vehicles.

3. Integrate priority control with the state-of-practice in traffic signal control that is coordinated-actuated control. First, coordination is considered within the multiple priority control formulation by adding virtual coordination requests. Second, vehicle actuations are considered when the optimal solutions are implemented by introducing the concept of green extension group (GEG). Therefore, the method developed in this dissertation addresses multiple priority requests, coordination, as well as passenger vehicle real-time actuations.

4. Implement a real-time algorithm that does not utilize a commercial solver (e.g. CPLEX) on an embedded platform and demonstrate priority control at a real field intersection.

1.3.3 Multi-modal traffic control within a v2x environment
The changes within a v2x environment include more than just changing how vehicles are detected and making small adjustments in signal timing parameters, but include actual consideration for multi-modal vehicle operation that includes passenger vehicles, transit, commercial vehicles, emergency vehicles, cycles, and pedestrians. It can include priority for transit and emergency vehicles. It will be possible to provide lane-by-lane and vehicle-by-vehicle controls to support highly cooperative and integrated behavior to utilize the network capacity in the most efficient and safe manner possible.

The advanced vehicle information and communications opens the opportunity for significant improvement in traffic signal control. The most obvious improvement is that vehicles can call (request) a phase as they approach a signalized intersection from any location on the roadway as opposed to only where the detectors have been installed. More correctly, they can continuously notify the traffic signal controller that they are on the approach and request service. In addition, the vehicle can communicate information about speed that can be used to determine when they would arrive at and cross the stop bar.

In the middle or final stage of deployment of v2x applications, the fraction of OBE-equipped vehicles will be relatively significant. New kinds of enriched traffic data from OBEs will easily overwhelm traditional traffic control logic. Not only will real-time positions and speeds be available, but also multi-modal traffic composition data with requested traffic control phases and arrival times throughout the network.

Given information about the current mix of traffic modes and the requested phases and arrival times the traffic signal control problem can be transformed to be a multi-modal
multi-priority request problem. New traffic control objectives with this multi-modal concept of traffic control include:

1. A clustering algorithm to quickly locate platoons by grouping nearby requests thus lowering the computation complexity of the area-wide optimization algorithm. The vehicles in a platoon can be treated as one single low priority request with a defined time interval for service.

2. A platoon-based multi-modal arterial traffic control formulation is addressed as a new concept of area-wide traffic control. Dynamical coordination, which is different than traditionally fixed-cycle, fixed-offset coordination is achieved by servicing the real-time platoon data. Platoons can be served in one cycle or split into two cycles depending on the total delay assessed in current intersection and downstream intersections.

1.4 Summary of the dissertation

Chapter 2 presents current literature review of previous work on IntelliDrive\textsuperscript{SM}, traffic signal control algorithms, traffic control systems, and probe vehicle technology. Relevant pioneering work on traffic control with advanced communications is also summarized.

Chapter 3 presents a framework for obtaining pseudo lane-level positioning using low-cost GPS, vehicle-to-infrastructure (v2i) communication, and driving event detection. In this context, pseudo means that lane level accuracy is achieved only under the assumption that v2i is available and there is no GPS outage. GPS errors can be categorized into common-mode errors and noncommon-mode errors, where common-mode errors can be
mitigated by differential GPS (DGPS) but noncommon-mode cannot. First, common-mode GPS error is cancelled from differential corrections broadcast from the road-side equipment (RSE). With v2i communication, a high fidelity roadway layout map and satellite pseudo-range corrections are broadcast by the RSE. The on-board equipment (OBE) corrects for the GPS common-mode errors based on the received pseudo-range corrections from the RSE, the current lane estimate, and the segment status determined by a general map matching algorithms. To enhance and correct the lane level positioning, a statistical process control approach is used to detect significant vehicle driving events such as turning at an intersection or lane-changing. Whenever a turn event is detected, a mathematical program is solved to estimate and update the GPS noncommon-mode errors. This chapter does not consider vehicle sensor data which could be used to improve position estimates (but requires an interface to the vehicle electronic system) and it is assumed that there is no GPS outage. Next Generation Simulation (NGSIM) data is used to validate driving behavior for turn movements and to calibrate the lane-changing detection model. A field experiment is conducted to validate the positioning models.

Chapter 4 examines the multiple priority problems in traffic signal control under the condition that OBEs are only installed on high priority class vehicles, such as emergency vehicle or transit vehicle. A priority request is sent to RSE if the emergency vehicle is approaching the intersection or the bus falls behind schedule. Given the current multiple priority request information, a mixed integer linear program (MILP) is solved in the RSE to obtain the optimal signal plan. First, a deterministic MILP is proposed only for multiple priority control of emergency vehicles. Second, a robust MILP is developed for
transit vehicles (e.g. buses) to accommodate the uncertainty in the traffic state (e.g. queues). Third, actuated control is integrated into the robust MILP formulation to mitigate the delay for passenger cars caused by priority control. Last but not least, the signal coordination constraints are added in the MILP formulation to achieve better performance on an arterial. Both experiments on isolated intersections and coordinated intersections are conducted to prove the efficiency of proposed strategy to achieve real-time control.

Chapter 5 presents an approximation algorithm to the mathematical program from chapter 4 for field implementation. Currently our OBEs and RSEs are running on embedded Linux systems, which is not a compatible operating system for sophisticated solvers such as CPLEX. Therefore, it is necessary to develop a solution algorithm for the multiple priority control problem that can perform in a reasonable fashion on an embedded computer. First the problem is transformed to a polynomial solvable cut problem according to some reasonable assumptions. Second, a phase-time diagram is developed to evaluate the feasible solutions and search for the sub optimal solutions. Finally, a real-world experiment is conducted in a live intersection of Southern Ave. and 67th Ave in Maricopa County, AZ. OBEs are installed on three “REACT” (Regional Emergency Action Coordinating Team) vehicles from Maricopa County Department of Transportation (MCDOT). One RSE is connected with and Econolite ASC/3 traffic controller in the cabinet. Different scenarios of multiple priority requests are tested and the results showed that the algorithm proposed could serve the multiple priority requests in real-time.
Chapter 6 outlines a methodology called PAMSCOD (Platoon-based Arterial Multi-modal Signal Control with Online Data) for multi-modal traffic control when market penetration of IntelliDrive$^\text{SM}$ is relatively high in passenger cars. Here multi-modes include emergency vehicles, buses and passengers cars. Due to the large number of passenger cars in the network, clustering methods are developed to group the nearby service requests into traffic platoons. Then a uniform request-based formulation is developed to optimize traffic signal control for concurrent different motorized travel modes, e.g. buses and emergency vehicles, given the assumption of v2x environment.
CHAPTER 2
LITERATURE REVIEW

Recent advances in communications standards and technologies provide the basis for significant improvements in traffic signal control capabilities. In the United States, the IntelliDrive™ program (originally called Vehicle Infrastructure Integration - VII) has identified 5.9GHz Digital Short Range Communications (DSRC) as the primary communications mode for vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) safety based applications. The IntelliDrive™ architecture (Faradyne 2005) also includes the use of other communications channels such as IEEE 802.11 (wi-fi), digital cellular, Bluetooth, etc. for non-safety critical applications. Regardless of the communications channel used, the ability for vehicles and the infrastructure to communicate information is a significant advance over the current system capability of point presence and passage detection that is used in traffic control systems.

This chapter reviews the literature related to IntelliDrive™ as well as traffic signal control strategies and probe vehicle techniques.

2.1 U.S. IntelliDrive™ (VII)

IntelliDrive™ (VII) has been demonstrated and evaluated in several states over the past several years. Michigan and California are both leading field operational tests (FOT). Large test beds were established in these two states. New York is in the early stages of
developing an IntelliDrive\textsuperscript{SM} program focused on commercial vehicles. Virginia conducts some early research on ramp metering control and signal dynamical gap with IntelliDrive\textsuperscript{SM}. Arizona focuses on researching and developing technology to assist emergency-responder vehicles.

2.1.1 California

In the VII California Program, Caltrans and Metropolitan Transportation Commission (MTC) have created a VII test bed in the Bay Area (Misener 2008a). The large-scale test bed extends over approximately 60 miles of roadway (freeways and arterials). The VII California test bed is continuing to expand. Currently there are 12 DSRC radios deployed with plans to grow this number up to 40. Several applications are being testing including (Misener 2008b): 1). Traveler information; 2. Ramp control; 3). Electronic payment (tolling); 4). Intersection safety, including a project called Cooperative Intersection Collision Avoidance Systems – Violation (CICAS-V) (J. Chang et al. 2007); 5). Curve overspeed warning; 6). OEM specific applications (K. Li et al. 2007).

2.1.2 Michigan

Michigan has embarked on an early IntelliDrive\textsuperscript{SM} (VII) deployment. Formally, Michigan has been designated the national IntelliDrive\textsuperscript{SM} (VII) proof of concept (POC). Preliminary testing has focused on proving that data can be shared between the infrastructure and vehicles in a timely and accurate manner to support IntelliDrive\textsuperscript{SM}
applications (Piotrowicz 2008). Michigan Department of Transportation (MDOT) has installed 60 RSEs and is cooperating with Chrysler to equip 15 vehicles. The MDOT Data Use Analysis and Processing (DUAP) project and Cooperative Intersection Collision Avoidance System (CICAS) are two significant projects conducted using the POC testbed.

The DUAP project (Mixon/Hil of Michigan, Inc. 2007) is a research program to determine how new VII data impacts safety, traffic operations and management, asset management, winter operations, and transportation planning. The program is focused on demonstration and assessment of data transformation and management, and DUAP system development is a means to that end. From a systems engineering functional viewpoint, the DUAP system has four high-level capabilities:

- Collecting data
- Consolidating the collected information
- Converting data into information needed by transportation agencies
- Communicating the unified information to various agencies and the public

The CICAS projects (McHale 2008) can be categorized by CICAS “Gap” projects and CICAS “Violations” projects. In CICAS “Gap” projects, there two sub projects CICAS-Stop Sign Assist project and CICAS-Signalized Left Turn Assist. CICAS-Stop Sign Assist project enhances driver’s decision at stop sign through information and warnings by dynamic sign, gaps assessed with infrastructure sensors and v2i warnings. In CICAS-Signalized Left Turn Assist project, drivers will be warned when it is unsafe to make a left turn because of oncoming vehicles, presence of pedestrians and other road
users. CICAS- Violations (CICAS-V) is a 4 year project to develop and evaluate a prototype system intended to assist drivers in reducing the frequency of crashes between vehicles due to violations of traffic signals and crashes between vehicles due to violations of stop signs.

2.1.3 New York

The New York IntelliDrive℠ program is designed especially for commercial vehicles. The commercial vehicle infrastructure integration (CVII) program (NYSDOT 2008) includes 13-mile test site on the New York State (NYS) Thruway Authority’s Spring Valley Corridor. The goal of the CVII Program is to develop, test and demonstrate commercial vehicle based data communication with the roadside equipment (RSE).

The Department and its partners desire to leverage the existing light vehicle based VII technology to enhance commercial vehicle safety, security, and mobility by partnering to develop, test and demonstrate a prototype system that utilizes the VII architecture and system requirements as well as the SAE J1708 vehicle data bus and the standard message sets defined in the SAE standards (SAE J1587, SAE J1939 and SAE 2735). Commercial vehicle in-vehicle hardware and software have been developed, tested and demonstrated to allow data message sets (DMS) to be wirelessly transmitted via DSRC.

2.1.4 Virginia
The Virginia Department of Transportation (VDOT) has been an active participant in the national IntelliDrive\textsuperscript{SM} development effort. They assessed national development activities and quantitatively evaluated two potential system operation applications: traffic monitoring and signal control (B. L. Smith et al. 2007).

The IntelliDrive\textsuperscript{SM} benefits to traffic signal control was analyzed to determine how traffic signal control could be improved. A dynamic gap-out feature was developed that takes advantage of higher resolution vehicle location data available in IntelliDrive\textsuperscript{SM}. In the dynamic gap-out system, vehicle headways were analyzed at a distance of 300 ft upstream of the stop bar. Based on this headway, the controller predicts a vehicle arriving at the stop bar before or after the gap timer expiring. If the headway gap at 300 feet upstream is larger than the gap time, the signal is allowed to gap out immediately, hence effectively transferring the additional gap time to the other phases (movements).

2.1.5 Arizona

The Arizona Department of Transportation (ADOT) developed a system called the Emergency IntelliDrive\textsuperscript{SM} (E IntelliDrive\textsuperscript{SM}) system that is focused on researching and developing technology to assist emergency-responder vehicles (ADOT 2008). This dissertation is partially funded by Arizona E- IntelliDrive\textsuperscript{SM} project, which has been conducted by the ATLAS (Advanced Traffic and Logistics Algorithms and Systems) research center in the College of Engineering at the University of Arizona.
The Arizona E- IntelliDrive℠ initiative has a very unique focus on incident management and emergency response, which is not currently being addressed by other state or national IntelliDrive℠ efforts.

The Arizona E- IntelliDrive℠ effort focused on four applications, including:

2. Preemption Operations at Ramp Meters.
3. Ad hoc Incident Warning Broadcast.
4. Lane, Road Closure, and Incident Information Communication to Traffic operation center (TOC).

These applications were developed and demonstrated in the Maricopa County DOT parking lot in 2008, at the AASHTO Annual Meeting 2009 in Palm Desert, CA, and the traffic signal priority application was tested and demonstrated at a live intersection in Maricopa County in 2010.

2.2 Traffic Signal Control

Traffic signal lights were invented nearly 150 years ago when the first traffic lights were installed outside the British Houses of Parliament in London by the railway engineer J. P. Knight. Traffic signal control can be categorized into three different control schemes: fixed-time traffic signal control, actuated traffic signal control, and adaptive traffic signal control.
2.2.1 Fixed-time traffic signal control

Many of state-of-the-practice pre-timed systems are operated in a time-of-day mode in which a day is segmented into a number of time intervals, and a signal timing plan is predetermined for each time interval. Typically 3–5 plans are run in a given day. The basic premise is that the traffic pattern within each interval is relatively consistent and the predetermined. Fixed time control has low cost installation and timing plan is best suited for the condition of this particular time of day. But it is not robust or adaptive to current traffic conditions, since real-world travel demands are intrinsically fluctuating, and traffic flows at intersections may vary significantly even for the same time of day and day of week (Yin 2008).

Usually fixed time control plan is produced from off-line signal optimization considering time-of-day constant flows. Some well-known traffic signal off-line optimization algorithms are listed as below,


MAXBAND is developed based on the fact that vehicles leaving from an upstream intersection are grouped into a “platoon” by the green light. It is desirable to set the signals at the downstream intersections in such way that the platoon of traffic is able to go without stop when traveling through the network. In other words, the control objective is to maximize the bandwidth. It provides progression along arterial, but the algorithm doesn’t work well for oversaturated intersections, since residual queue can easily disrupt the progression.
• MULTIBAND (Gartner, Assman et al. 1991) and (Stamatiadis & Gartner 1996)

MULTIBAND is another well-known off-line progression based optimization model developed by Gartner. MULTIBAND incorporates a systematic traffic-dependent criterion, which guarantees the suitable progression scheme for different traffic flow patterns. The method generates a variable bandwidth progression in which each directional road section can obtain an individually weighted bandwidth (hence, the term multi-band). Mixed-integer linear programming is used for the optimization.

• TRANSYT (Robertson 1969)

TRANSYT was first developed by Robertson but was substantially extended and enhanced later. TRANSYT has been widely recognized as one of the most useful tools in studying the optimization of area traffic control. It is the most known and most frequently applied signal control strategy, and it is often used as a reference method to test improvements enabled by real-time strategies.

• TRANSYT-7F (Wallace et al. 1998) and (M. Li & Gan 1999)

TRANSYT-7F (TRAffic Network StudY Tool, version 7F) is a version of TRANSYT for United States. TRANSYT-7F has been used by practitioners for traffic network signal timing design and analysis. The latest version of TRANSYT-7F release 11 features genetic algorithm optimization of cycle length, phasing sequence, splits, and offsets (McTrans 2010). It combines an optimization process (including genetic algorithm, multi-period, and direct CORSIM
optimization) with a state-of-the-art macroscopic simulation model (including platoon dispersion, queue spillback, and actuated control simulation)

- PASSER II (Chaudhary & Chu 2003)

PASSER II was originally developed with Texas Department of Transportation (TxDOT) more than 30 years ago. The optimization technology used in PASSER II is simple but efficient. In addition, it has proven to produce high-quality timings for signalized arterials. Furthermore, bandwidth-based timings are easily recognized and appreciated by motorists in Texas and many other parts of the United States.

- Synchro (Trafficware 2009)

Synchro is a macroscopic traffic signal optimization model and software package produced by Trafficware. They have approximately 1600 users throughout North America and are used by most state DOT’s. Synchro uses SimTraffic as its microscopic simulation model to fully simulate signalized or unsignalized intersections.

One disadvantage of above fixed time control algorithms is that the traffic control problem is not addressed very well under oversaturated traffic condition, since traffic flow model is oversimplified in their algorithms. Recently a number of papers have developed dynamic traffic signal control formulations based on the cell transmission model (CTM) in (Daganzo 1994) and (Daganzo 1995). The significant benefit of embedding CTM in signal control is to capture traffic dynamics. CTM-based signal
control formulation is to address both unsaturated and oversaturated conditions considering shockwaves and physical queues. Lo formulated the network signal optimization problem as a mixed-integer linear programming problem using CTM in (Lo 1999) and (Lo 2001), assuming that the cycle lengths are fixed. Lin and Wang formulated a more computationally efficient version of the mixed integer linear program for the signal optimization problem with CTM in (Lin & C. Wang 2004). But only two-phase signal was considered in their work. (Beard & Ziliaskopoulos 2006) proposed a CTM-based system optimal signal optimization formulation combined with system optimal traffic assignment which provides several improvements over existing mixed-integer linear program formulations including turning movements for exclusive turn lanes. Most recently, (L. Zhang et al. 2010) examined the design of robust traffic signal control with the CTM. A scenario-based stochastic programming model was proposed to optimize the timing of pre-timed signals along arterials under day-to-day demand variations.

The disadvantage of CTM-based formulations is the complexity of the mixed-integer linear program (MILP). The problem size grows very quickly with the size of the network, the number of phases and the time horizon. The “curse of dimensionality” makes it impossible to solve these formulations directly using commercially available packages such as CPLEX, LINDO, etc. Therefore, Genetic Algorithm (GA) is widely accepted to solve the complicated analytical traffic signal control models (Abu-Lebdeh & Benekohal 2000)(B. Park et al. 1999)(Girianna & Benekohal 2002)(Lo 2001) and (L. Zhang et al. 2010), though it is likely to converge to local optimum.
2.2.2 Actuated traffic signal control

Nowadays, traffic actuated control is already implemented within controllers. It makes use of real-time measurements, provided by inductive loop detectors that are usually located some 40 m upstream of the stop line, to execute some more or less sophisticated vehicle-actuation logic (Papageorgiou et al. 2003).

The basic principles of timing the green interval in a traffic actuated controller is as follows:

- There is a minimal green time for each timing phase, so that vehicles have enough time to start and pass through the intersection and pedestrians can walk through the intersection before yellow signal. Minimal green that is too long may result in wasted time at the intersection; one that is too short may violate driver expectation or (in some cases) pedestrian safety.

- Each following vehicle generates a call (or actuation) to the traffic controller to ask for additional green time. This is called green extension or gap.

- There is also a maximal green time (or split in coordinated-actuated mode) for each timing phase, which is the limit of total green time when there is a conflicting phase vehicle call.

Figure 2.1 illustrates the timing diagram of actuated traffic controller.

The actuated controller can be configured to operate in one of two modes: fully actuated and semi-actuated (Federal Highway Administration 2007). In fully-actuated
mode, detection is provided on all approaches to the intersection, and the controller operates without a common background cycle (i.e., operating “free”).

Figure 2.1 Actuated phase timing diagram (Federal Highway Administration 2008)

In semi-actuated (or called coordinated-actuated) mode, detection is provided only on the side-street approaches (and perhaps main-street, left-turn movements). The main street signals remain green until a call for service is placed by the side-street detectors. Semi-actuated operation is used to provide progressive vehicle flow through a series of controlled intersections. In this mode, each controller in the coordinated system operates within a common background cycle length. The coordinator in the controller guarantees
that the coordinated phases (generally phase 2 in ring 1 and phase 6 in ring 2) will display green at a specific time within the cycle, relative to a system reference point established by the specified cycle length and system synch reference time. An offset time, relative to the system reference point, is specified for each controller in the series to maintain the smooth progression of vehicles through the intersections. The coordinator also controls when and for how long non-coordinated phases can indicate green so that the controller will return to the coordinated phases at the proper time.

2.2.2.1 Advanced features in coordinated-actuated traffic signal control

Each coordinated system has the set of parameters to be determined to achieve signal coordination. These settings are necessary inputs for coordination, which are listed as follows:

- **Cycle length**: Cycle length defines the time required for a complete sequence of indications. Usually the traffic engineer utilizes the greatest cycle length among all the intersections along arterial to accommodate the traffic and then design the rest of the progression scheme around that intersection.

- **Splits**: splits are the portion of time allocated to each phase at an intersection (including yellow and all red clearance time). For implementation in a signal controller, the sum of the phase splits must be equal to (or less than) the cycle length.
• Offsets: The offset is usually defined as the time differential between the initiation of green indications of the coordinated movements relative to the master intersection (i.e., the intersection dictates the signal timing requirements of the other intersections). The offset value is derived based upon the distance between the master intersection and the desired travel speed of traffic on the arterial. Figure 2.2 shows a time-space diagram illustrating offsets and bandwidth of a coordinated traffic signal system (Sunkari et al. 2004).

![Figure 2.2 Coordination on a time-space diagram (Sunkari et al. 2004)](image)

• Yield point: Yield point is only for coordinated phases. It is a point where the controller starts to make decisions to terminate the coordinated phase, as shown in Figure 2.3 (a).
Force-off point = 100s

Yield point = 25s

Force-off point = 75s

Yield point = 25s

Force-off point = 50s

Yield point = 25s

\( \phi_1 \) split = 25s
\( \phi_2 \) split = 25s
\( \phi_3 \) split = 25s
\( \phi_4 \) split = 25s

Demand = 25s
Demand = 15s
Demand = 40s
Demand = 25s

(a)

(b)
Force-offs: The force-offs are points where non-coordinated phases must end even if there is continued demand. The use of force-offs overlays a constraint on all non-coordinated phases to ensure that the coordinated phase will receive a minimum amount of time for each cycle, depicted in Figure 2.3 (a). There are two types of force-offs: fixed and floating (Federal Highway Administration 2008).

- Fixed force-offs: The fixed force-off maintains the phase’s force-off point within the cycle. If a previous non-coordinated cycle ends its phase early, any following phase may use the extra time up to that phase’s force-off.

\[
\begin{align*}
\phi_1 \text{ demand} &= 25\text{s} \\
\phi_2 \text{ demand} &= 15\text{s} \\
\phi_3 \text{ demand} &= 40\text{s} \\
\phi_4 \text{ demand} &= 25\text{s}
\end{align*}
\]

Figure 2.3 (a) Pre-defined splits; (b) Fixed force-offs implementation; (c) Floating force-offs implementation.
o Floating force-offs: Floating force-offs are limited to the duration of the splits that were programmed into the controller. The force-off maintains the non-coordinated maximum times for each non-coordinated phase in isolation of one another. Floating force-offs are more restrictive for the non-coordinated phases. If a phase does not use all of the allocated time, then all extra time is always given to the coordinated phase.

An example is presented in Figure 2.3 (a), (b) and (c) to distinguish fixed and floating force-offs. Suppose phase 2 is coordinated phase, cycle length is 100 seconds and each phase split is equal to 25 seconds, shown as Figure 2.3 (a). If the demand of phase 2 and phase 1 equal to the splits (25 seconds), phase 3 has lower demand (15 seconds) and phase 4 has larger demand (40 seconds) than splits, fixed force-offs and floating force-offs have totally different split implementation. In fixed force-offs, the force-off point of phase 4 is fixed at 50 s. The unused green time in phase 3 can be re-allocated to phase 4. In addition to the predefined split 25 seconds, phase 4 has total 35 seconds actual time. In floating force-offs, the maximal split of phase 4 is 25 seconds. So phase 4 needs to be forced off no matter that there are unused green times in previous phase 3. So extra green time are all assigned to coordinated phase (phase 2) in floating force-off. Therefore, fixed force-offs give beneficial to side streets if there are fluctuations in traffic demand and a phase needs more green time. And floating force-offs give more green on coordinated phase, which may result in early return to disrupt coordination, but may also clear the
queue on arterial in congested traffic condition. There are both trade-offs for fixed and floating force-offs.

2.2.3 Adaptive traffic signal control

Since 1970’s, first generation of adaptive traffic signal control systems was developed in UK and Australia, such as SCOOT and SCAT.

SCOOT was first developed by Robertson’s team (Hunt et al. 1982) and has been extended later in several respects. It is considered to be the traffic-responsive version of TRANSYT and has been applied to over 150 cities in the United Kingdom and elsewhere. SCOOT utilizes traffic volume and occupancy (similar to traffic density) measurements from the upstream end of the network links. It runs in a central control computer and employs a philosophy similar to TRANSYT. More precisely, SCOOT includes a network model that is fed with real measurements (instead of historical values) and is run repeatedly in real time to investigate the effect of incremental changes of splits, offsets, and cycle time at individual intersections (functionally decentralized operation).

SCOOT also has some vices (P. Martin 2001). Only up to 15% detector failure is accommodated. The performance of SCOOT degrades back to a fixed time plan if faults not rectified. And it is unable to accommodate oversaturation.

SCAT was installed in Sydney in 1970s, offers a substantial improvement to movement on arterial roads at low cost thereby enabling usage of the arterial road network to be optimized (Sim & Dobinson 1980). Now it is implemented in 50 cities worldwide, including Oakland County, Minneapolis and Atlanta in the USA. The main
objective of the system is to minimize overall stops and delay when traffic demand is less than system capacity. When demand approaches system capacity, SCATS maximizes throughput and controls queue formation (Lowrie 1982) and (Luk 1984).

UTOPIA (Urban Traffic Optimization by Integrated Automation)/SPOT (System for Priority and Optimisation of Traffic) was developed by FIAT Research Centre, Italy (Mauro & Taranto 1989). This system is a hierarchical control model in which UTOPIA is applied in area level control and SPOT is for local intersection control. It also contains three layer controls like other systems.

PRODYN is another real-time traffic control system developed by CERT/ONERA in France and implemented in three French cities (Henry et al. 1983). It includes the prediction model of arrival vehicles and estimates queues at each intersection for 16 time intervals of 5 seconds. Local optimization is made for the time horizon by a controller implementing the estimated control strategy for each successive period. The system transmits the predicted states to controllers downstream to improve their predictions.

In United States, during the past decades, the Federal Highway Administration (FHWA) has focused on the development and deployment of Real Time Traffic Adaptive Control System (RT-TRACS) in the USA. Several traffic adaptive signal control systems were developed since 1980’s.

The Optimized Policies for Adaptive Control (OPAC) was developed by Gartner (Gartner 1983)(Gartner, Tarnoff et al. 1991) and became a part of RT-TRACS of FHWA. It was developed earlier for isolated intersection control, which could be expanded to control a subnetwork with a group of intersections. It is based on the Dynamic
Programming to minimize the total intersection delay and stops over a user-specified rolling horizon interval. During optimization, it progressively selects from among a number of possible signal patterns at each intersection. The patterns are recalculated based on updated traffic data over a shorter time interval and used for computing the globally optimized solution. Sensors are placed upstream of stop line to predict arrival flow pattern.

RHODES is another part of RT-TRACS of FHWA, developed by ATLAS center, University of Arizona (Head & P. Mirchandani 1992) (Sen & Head 1997)(P. Mirchandani & Head 2001). University of Arizona’s prototype is composed of a main controller (called RHODES), APRES-NET, which simulates platoons, REALBAND (a section optimizer), PREDICT, which simulates individual vehicles, and COP (a local optimizer). This prototype, which is a hierarchical control system, has three levels of optimization, namely intersection control, network control and network loading. For local intersection control, the signal phase durations are optimized by a Dynamic Programming approach. The decision is re-evaluated every 7 to 15 seconds using a decision horizon of 90 seconds. At the second level, the optimization of network flow control is performed based REALBAND. This model attempts to form progression bands based on actual observed platoons in the network. All the possible resolution of a conflict among the predicted platoon movements are listed as the decision tree. Then the best one based on the performance index is chosen as the optimal setting. At the highest level, network loading predicts the general demand over longer periods of time, typically one hour.
Both OPAC and RHODES were developed in the U.S. and were implemented in the 1990s and early 2000s. These systems were both offspring from FHWA’s RT-TRACS development effort (Selinger & Schmidt 2009).

Most recently, in 2001, FHWA initiated the ACS-Lite (adaptive control systems) program to assess, and then pursue, the best, most cost-effective solution for applying ACS technology to current, state-of-the-practice closed loop traffic signal control systems. This effort is intended to make ACS technology accessible to many jurisdictions without the upgrade and maintenance costs required to implement ACS systems that provide optimized signal timings on a second-by-second basis (Luyanda et al. 2003).

Due to past wireless communication limitation, none of the current adaptive signal control makes use of probe vehicle data to gain better performance.

2.3 Probe vehicle technology

The state of the art in traffic monitoring is to utilize wireless location technology as a means to track “probe” vehicles as they traverse the transportation network. The probe vehicle “track” provides information on vehicle locations over time, which can be used to derive travel times and speeds on particular roadway links. The wireless location technology most commonly used is cellular phone GPS locations and/or cell handoff information.

2.3.1 Traffic state and travel time estimation with probe data
2.3.1.1 Transit as probes

Some researchers proposed the use of mass transit buses as probes since they can be equipped with AVL (automated vehicle location) technologies. Buses can be easily tracked since they have fixed routes and schedule. It will provide relative stable data set than passenger cars. However, the percentage of bus in entire traffic is relative low to estimate traffic state in real time.

(Hall & Vyas 2000) found that when automobiles have long delays, buses traveling nearby on the same route are also likely to be delayed. The reverse situation, however, is not always true, because buses frequently wait for extended periods when they run ahead of schedule. Any useful bus probe algorithm needs to distinguish between actual congestion and a stopping delay.

(Cathey & Dailey 2002) developed a mass transit tracking system based on AVL data and a Kalman filter to estimate vehicle position and speed were described, as were a system of virtual probe sensors that measure transit vehicle speeds by using the track data.

(Bertini & Tantiyanugulchai 2004) showed that actual arterial traffic conditions may be explained by using transit vehicle AVL information. The set of transit data, bus movements generated from the maximum instantaneous speed achieved between each stop pair was found to most reliably depict the traffic movement of non-transit vehicles.

(Cathey & Dailey 2003) presented a corridor approach to travel-time estimates by using transit vehicles as probes. This work provided speed estimates that track the
significant changes identified in inductance-loop data but appears to provide a conservative estimate of the speed.

2.3.1.2 GPS positioning as probes

With the development of GPS positioning technology in last decades, GPS error (10–20 meters) (J. Farrell & Barth 1999) can be much lower compared with cellular positioning (~100 meters) (Ygnace et al. 2000). Many researchers started to utilize GPS to estimate traffic state and travel time. However, the market penetration of GPS equipment is not as high as cell phone. So some researchers use smart cell phone with GPS device to collection probe data, which seems much more realistic for widely implementation of estimation algorithm.

It is showed that the number of probe vehicles required increases nonlinearly as the reliability criterion is made more stringent. Probe vehicles appear to be an attractive source of real-time traffic information in heavily traveled, high-speed corridors such as freeways and major arterials during peak periods, but they are not recommended for coverage of minor arterials or local and collector streets or during off-peak hours.

(Quiroga & D. Bullock 1998) collected tremendous GPS historical data for considering three analysis: segment lengths, sampling rates, and central tendency. The sampling rate analysis addresses the effect of collecting GPS data at different sampling periods and shows that for a segment to have GPS data associated with it, the GPS sampling period should be smaller than half the shortest travel time associated with the segment. The analysis also shows a tradeoff between sampling rates and segment speed.
reliability, and emphasizes the need for even shorter GPS sampling periods (1–2 s) in order to minimize errors in the computation of segment speeds. The central tendency analysis compares harmonic mean speeds and median speeds and shows that median speeds are more robust estimators of central tendency than harmonic mean speeds.

(Y.B. Yim & Cayford 2001) used a vehicle equipped with differential GPS (DGPS), and managed to match its route for 93% of the distance it traveled.

Recently, some researches combine GPS device and cellular phone for travel time and traffic state estimation. (Young 2007) encompassed two primary methods: GPS data obtained from fleet management services and geo-location schemes that leverage cellular phone infrastructure. (Yoon et al. 2007) identified traffic conditions on surface streets given location traces collected from on-road vehicles—this requires only GPS location data, plus infrequent low-bandwidth cellular updates. (Herrera & Bayen 2009) developed a real experiment using GPS with cell phones to estimate traffic state, called Mobile Millennium (see http://traffic.berkeley.edu/). In this experiment, cell phones equipped with a Global Positioning System (GPS) provide new opportunities for location based services and traffic estimation. When traveling on board vehicles, these phones are able to accurately provide position and velocity of the vehicle, and can be used as probe traffic sensors. This article presents a new technique to incorporate mobile probe measurements into highway traffic flow models, and compares it to a Kalman filtering approach.

2.3.1.3 Cellular phone as probes
Although Cellular phone positioning technique has less accuracy, cellular phone also has much high market penetration in real world. Due to its positioning error, cellular phone positioning is not good for intra-city (arterial) traffic estimation. However, it could be efficiently implemented for sparse network, like highway networks.

Some researchers in California PATH (Partners for Advanced Transit and Highways) are the pioneers in this area. (Sanwal & Walrand 1995) utilized vehicles as sensors instead of the conventional stationary sensors (such as the inductive loops used in many places) for highway travel time estimation. (Westerman et al. 1996) proposed four possible methods for estimating real time travel times and performing automatic incident detection for ATMIS based on induction loop or probe vehicle data alone. It concludes that the fourth approach statistical techniques is the best, which focuses on the macroscopic level of traffic and to analyze how information about these macroscopic traffic characteristics can be extracted from received probe vehicle data. (Ygnace et al. 2000) revealed that at least 5% of freeways travelers are equipped with a cell phone; one can predict a 95% accuracy in freeway link travel time estimates. (Y.B. Yim 2003) did some surveys for cell phone penetration and claimed that Cellular probe technology one of the potentially promising technologies for data collection of accurate travel time.

Recently, (Bar-Gera 2007) found that there is a good match between the two measurement methods, indicating that the cellular phone-based system can be useful for various practical applications such as advanced traveler information systems and evaluating system performance for modeling and planning. Equipping floating vehicles with GPS can improve the accuracy of the measurements. (Valerio et al. 2009) outlined a
unified framework that encompasses UMTS and GPRS data collection in addition to GSM, and prospectively combines passive and active monitoring techniques.

2.3.2 Traffic control with probe data

Currently there are very few literatures about traffic control with probe data. (Comert 2008) proposed a probabilistic method to estimate queue length given the last probe vehicle position in the queue. Based on the queue information, max green parameter was adjusted to achieve better performance. However, no detailed signal control scheme or analytical model was proposed in this work.

(H. Park 2008) utilized VII enhanced data and developed three VH-enabled ramp metering algorithms (the variable speed limit, the lane changing advisory, and the GAP). The results showed that VII-enabled ramp metering algorithms improved the network performance by providing 4.3% more vehicle miles traveled while reducing vehicle hours traveled by 4.6%, which resulted in 9.3% higher average speeds.

(J. Y. Park 2009) developed a network wide signal control system based on Persistent Traffic Cookies (PTC), which is similar as IntelliDriveSM. A decentralized control embedded with indirect signal coordination scheme was presented. Signal optimization is accomplished at each local intersection by a dynamic programming approach with the predicted arrival patterns resulting from PTC data. However, probe data uncertainty was not considered in this work.
2.4 Summary

Advanced traffic management is a cost-effective option to reduce total delay, fuel consumption and air pollution in urban networks. Nevertheless, Adaptive signal control, the most advanced scheme for real-time traffic responsive operations, is still not widely used due to inadequate sensor systems and the deficiencies in the control algorithms.

With the advent of advanced communication systems nowadays, the traffic data are dramatically enriched. In order to implement advanced traffic control systems in the field, the adaptive signal control systems need to be re-developed in more simple and direct way, given that full size of real-time traffic data is available.

This dissertation presents a innovative pseudo-lane-level GPS positioning system, a robust mixed integer linear program (MILP) for multiple priority signal control, and a platoon-based multi-modal arterial traffic control approach, all within a v2x environment.
CHAPTER 3
PSEUDO-LANE-LEVEL, LOW-COST GPS POSITIONING WITH VEHICLE-TO-INFRASTRUCTURE COMMUNICATION AND DRIVING EVENT DETECTION

3.1 Introduction

Recently, the concept of cooperative systems have gained increased attention by both infrastructure owner-operators and vehicle manufacturers because of the potential of wireless communications between vehicles and the roadside to provide a safer and more efficient operating environment. Vehicle-to-vehicle (v2v) or vehicle-to-infrastructure (v2i) - generally referred to as v2x - has the potential to transform travel as we know it today. v2x applications combine leading edge technologies such as advanced wireless communications, on-board computer processing, advanced vehicle-sensors, GPS navigation, smart infrastructure, and others—to provide the capability for vehicles to identify potential collision and hazards on the roadway and communicate relevant information to give driver alerts, warnings, and critical traffic control information (RITA 2010). A key capability necessary for successful and wide-scale deployment of v2x applications is the ability to provide accurate lane level estimation of vehicle position. First, lane level position data enhances roadway safety by supporting collision avoid system, such as Cooperative Intersection Collision Avoidance Systems (CICAS) in United States (Amanna 2009). Second, lane control with different advisory speed and lane restriction are available with lane level positioning. Third, driving behavior -such as
lane changing - can be studied intensively based on lane level positioning data. Forth, lane level positioning can also benefit traffic operations and control; for example, lane level queue length could be obtained. This is a challenging technical problem that must engage the infrastructure as well as advanced vehicle technologies.

To be successful for wide scale deployment any solution must also be cost effective. A solution that is too expensive is unlikely to be widely deployed and supported. Although the use of low cost GPS receivers for navigation has recently become very popular as a variety of units from Garmin, TomTom, and others have flooded the market, the accuracy requirements of navigation and v2x are significantly different. The standard deviation of a non-differential GPS position estimates is on the order of 10-20 meters (J. Farrell & Barth 1999) (J. Farrell et al. 2003). Increased accuracy in few meters or even centimeters can be achieved through different kinds of Differential GPS (J. Farrell & Barth 1999) (H. Blomenhofer et al. n.d.) and (Tan et al. 2003). DGPS is an excellent positioning tool, but GPS receivers on most of vehicles are not capable of receiving differential corrections and differential receivers are more expensive and many times require subscriptions to correction services (OminiSTAR 2010) that are costly. In order to make GPS positioning systems popular, many researchers focus on how to correct the error from low-cost, non-differential GPS (Clanton et al. 2009) and (Toledo-Moreo & Zamora-Izquierdo 2009).

This chapter presents a potential solution to the low-cost positioning problem and includes four low-cost elements. The first low-cost element is the use of GPS (not necessarily differential GPS) that is available on many vehicles, hand held devices, and on v2x radio units, e.g. Dedicated Short Range Communication (DSRC) radio units
(Savari 2010). The second element is the high fidelity maps of key infrastructure elements that provide information about intersection and roadway geometry, called MAPs (defined in SAE DSRC-J2735), which contain very accurate GPS waypoints in the center of each roadway lane. These maps are to be provided as part of the infrastructure-to-vehicle communications (Le et al. 2009). The third element is low cost vehicle sensors that can be used to enhance position information, when GPS signals are erroneous or undetectable in places such as urban canyons. The fourth element is the cooperation between equipped vehicles by sharing information about current GPS position error. These four key elements can be combined to provide highly accurate and reliable vehicle position estimates that will enable new safety and efficiency applications.

This chapter explores a low cost positioning framework based on solely GPS and detailed maps (called the MAP) of the roadway system. GPS positioning with other vehicle sensors will be considered in future work. This chapter is organized as follows. The system structure is proposed in section 2. Section 3 presents a statistical process approach to detect lane-changing and turn movements. Section 4 develops a lane alignment optimization model to estimate GPS noncommon-mode errors. Section 5 reports the findings of a field test of the proposed positioning system. Conclusions and remarks are in Section 6.
3.2 The V2I positioning environment

In the environment of v2x, each equipped vehicle has on-board equipment (OBE), which communicates with road-side equipment (RSE) or other vehicles equipped with OBEs by some reliable wireless communication technology such as DSRC. The RSE broadcasts a high fidelity “map” (MAP). The vehicle will receive the MAP and using the received GPS position will estimate its current position, shown as Figure 3.1.

MAPs are an integral part of the infrastructure of a v2x system. MAPs are small ASCII text file that describes the roadway geometry in terms of segments, lanes,
forms the MAP. It is assumed that the MAP should be based on accurate GPS measurements, which can be obtained using survey grade RTK-GPS equipment and on a frequency of waypoints that captures the roadway geometrics including curvature, intersection geometry, and lane drop geometry. The requirement for highly accurate waypoints in the MAP is important due to the additive nature of the error that includes both the MAP accuracy and the real-time measurements.

In addition to the MAP, GPS corrections can also be broadcast from RSE since the position of RSE is fixed and surveyed. Given that the range of DSRC radio is less than 1km (Y. Liu et al. 2005), a small range local-area DGPS system can be established in this v2i environment either by position domain corrections and pseudorange domain corrections (Kaplan & Hegarty 2006). In position domain corrections, the coordinate differences between the surveyed RSE position and the position estimated from GPS measurements are communicated from the RSE to the OBEs. The latitude, longitude and height differences are directly broadcasts from RSE to nearby OBEs. Although the position domain corrections are the simple to implement, it requires that both receivers use the same set of satellites and the same position solution techniques on all receivers, which is very hard to be ensured because of the variety of GPS receiver providers in the low cost market. In pseudorange domain corrections, the reference station determines and disseminates pseudorange corrections for each visible satellite. Since it is a local-area DGPS system, the common-mode noises sources are cancelled to achieve 1m accuracy. Detailed discussion of DGPS algorithms can be found in (J. Farrell & Barth 1999) and (Kaplan & Hegarty 2006).
In this chapter, the authors suggest using pseudorange domain corrections. However, the details of how to implement pseudorange domain DGPS is not the scope of this chapter. It is assumed that the RSE-based local-area DGPS accuracy is achieved by low cost GPS under good visibility conditions. In order to test if this assumption is valid, a simple test was conducted in the intersection of Mountain and Speedway, Tucson, AZ when it was sunny and clear. A stationary RSE with low cost GPS was installed on the top of a traffic controller cabinet for 11 hours. The GPS position errors are shown in Figure 3.2.

![GPS Error Chart]

Figure 3.2 The GPS error in the test site in Tucson, AZ

Figure 3.2. The average GPS error is 1.29m with standard deviation 0.748m, which nearly matches the accuracy of code-based DGPS.
Although the GPS noncommon-mode error is unknown and difficult to track, it can be estimated when some specific driving events occur, e.g. vehicle right hand turn or left hand turn. Given a current MAP and measured vehicle trajectory, it is very simple to identify a vehicle turn movement occurs at an intersection. First, the actual inbound and outbound lanes could be estimated by the MAP network and map matching algorithms surveyed in (Quddus et al. 2007), given the 1m DGPS accuracy. The measured vehicle trajectory can also be divided into inbound and outbound trajectories after the vehicle turn movement is completed and detected. The offset between the actual and measured vehicle inbound and outbound trajectory can be regarded as the current GPS noncommon-mode error. A turn event-driven lane alignment optimization model is solved to capture the GPS noncommon-mode error to provide an offset that can be used for correction. The occurrence of a turn event or lane change event is monitored by using an exponentially weighted moving average (EWMA) statistical process control (SPC) chart based on the vehicle heading in relation to the roadway heading. The vehicle lateral deviation is tracked in order to detect the number of lanes changed. Lane changing events can also be used to determine the vehicle lane status, as well as to correct previous lane status estimated by the map matching algorithm.

Figure 3.3 shows an illustration of the actual and measured position of a vehicle after it makes a right hand turn at an intersection. In this situation, it is not known if the vehicle is in the right most lane or the left lane, but a combination of this driving event and the previous error estimate can be used to provide an accurate and reliable estimate of the position error.
The entire system structure is shown as Figure 3.4. First, an event-separated Extended Kalman Filter (EKF) is chosen to estimate the vehicle state from the raw GPS data and the estimated GPS errors (including common-mode errors from RSE and noncommon-mode errors from an optimization model). Due to page limitation, the EKF discussion is omitted in this chapter. Interested readers can find a detailed introduction to EKF in (Zhao et al. 2003)(Welch & Bishop 1995). The map matching algorithm is used to estimate the vehicle’s initial lane and segment status as well as to update the status based on the EKF estimate. Second, given the vehicle heading from the EKF states and the lane heading from MAP, the heading error is monitored by an exponentially weighted moving average (EWMA) SPC control chart. The EWMA control charts track both lane change and turn events. Once the EWMA data exceeds the defined control limits of lane change
events, a lane change is detected. The number of lanes changed can be estimated by the vehicle lateral deviation. Similarly, turning events are detected based on defined control limits. Finally, the vehicle lane status is updated or corrected by the vehicle status management module. When a turn event is detected, the lane alignment optimization module is triggered to update the estimated GPS noncommon-mode error. This information is provided to the v2x applications.

Two major contributions in this system are the EWMA control chart to monitor events and the event-driven lane alignment optimization to estimate the GPS noncommon-mode error.
3.3 Extended Kalman filter

The Kalman filter provides an efficient computational (recursive) means to estimate the state of a process. Kalman filters are very powerful in several aspects including that they support the estimation of past, present, and even future system states and they can do so even when the precise nature of the modeled system is unknown (Zhao et al. 2003). The two main features of the Kalman filter formulation and problem solution are vector modeling of the dynamic process under consideration and recursive processing of the noisy measurement data (Misener & Shladover 2006).

In the vehicle states tracking application of a Kalman filter the nonlinear dynamical system model must be linearized. A Kalman filter that linearizes about the current mean and covariance is referred to as Extended Kalman Filter (EKF). Although the linearization and Gaussian distributed noise assumption in the EKF may seriously affect the accuracy of the obtained solution, or can sometimes lead to divergence of the system (Welch & Bishop 1995), the EKF can handle approximate nonlinear filtering in real time without the curse of dimensionality, which greatly reduces the computational complexity of the system.

Since the low cost GPS receiver is the only sensor/information source used in this chapter, the longitude, latitude, heading and velocity are the only measurements available (assuming that altitude remains constant within the localized plane centered by the MAP). We assume that the speed of the vehicle is constant during each GPS measurement interval (1 second). Since the GPS data is in the WGS-84 coordinate
system, we convert each GPS measurement from onto a local planar coordinate system (El-Rabbany 2006).

The following states are selected,

\[ X(\tau) = [x(\tau) \quad y(\tau) \quad \sin \phi_v(\tau) \quad \cos \phi_v(\tau) \quad v_v(\tau)]^T \]

Where

\[ x(\tau) = \text{vehicle location in the east direction at time step } \tau, \text{ measured in meters.} \]
\[ y(\tau) = \text{vehicle location in the north direction at time step } \tau, \text{ measured in meters.} \]
\[ \sin \phi_v(\tau) = \text{sine of the vehicle heading at time step } \tau, \text{ in radians, with north being zero heading and clockwise being positive.} \]
\[ \cos \phi_v(\tau) = \text{cosine of the vehicle heading at time step } \tau, \text{ in radians, with north being zero heading and clockwise being positive.} \]
\[ v_v(\tau) = \text{velocity of the vehicle at time step } \tau, \text{ in m/s.} \]

The process to be estimated is now governed by the non-linear stochastic difference equations,

\[ x(\tau) = x(\tau - 1) + v(\tau - 1) \sin \phi_v(\tau - 1) + w_x(\tau - 1) \quad (3.1) \]
\[ y(\tau) = y(\tau - 1) + v(\tau - 1) \cos \phi_v(\tau - 1) + w_y(\tau - 1) \quad (3.2) \]
\[ \sin \phi_v(\tau) = \sin \phi_v(\tau - 1) + w_1(\tau - 1) \quad (3.3) \]
\[ \cos \phi_v(\tau) = \cos \phi_v(\tau - 1) + w_2(\tau - 1) \quad (3.4) \]
\[ v_v(\tau) = v_v(\tau - 1) + w_3(\tau - 1) \quad (3.5) \]
The choice of \( \sin \phi_v(\tau) \) and \( \cos \phi_v(\tau) \) as state variables instead of \( \phi_v(\tau) \) simplifies the state equations since \( \phi_v(\tau) \) is in radians and is cyclic, meaning that \(-\pi\) and \(\pi\) are the same state, and this discontinuity or “state jump” is not easily accounted for in the EKF.

For the measurement equations, the observation variables are chosen to be the same as the state variables.

\[
Z(\tau) = [x_m(\tau) \ y_m(\tau) \ \sin \phi_{vm}(\tau) \ \cos \phi_{vm}(\tau) \ v_{vm}(\tau)]^T
\]

Where
\( x_m(\tau) \) = the corrected measurement of the vehicle easting position at time step \( \tau \), in meters.
\( y_m(\tau) \) = the corrected measurement of the vehicle northing position at time step \( \tau \), in meters.
\( \sin \phi_{vm}(\tau) \) = the sine of the vehicle heading measurement at time step \( \tau \), in radians, with north being zero heading and clockwise being positive.
\( \cos \phi_{vm}(\tau) \) = the cosine of the vehicle heading measurement at time step \( \tau \), in radians, with north being zero heading and clockwise being positive.
\( v_{vm}(\tau) \) = the measurement of velocity of the vehicle at time step \( \tau \), in m/s.

Note that \( x_m(\tau) \) and \( y_m(\tau) \) are not equal to the raw position \( x'(\tau) \) and \( y'(\tau) \) from the GPS receiver, but are the corrected measurements that include the last event driven estimate of the GPS error \( \Delta_x'(M) \) and \( \Delta_y'(M) \), which occurred before time step \( \tau \).

\[
x_m(\tau) = x'(\tau) + \Delta_x'(M) \quad (3.6)
\]
\[ y_m(\tau) = y'(\tau) + \Delta_y'(M) \] (3.7)

Where, \( \Delta_y'(M) \) and \( \Delta_y'(M) \) are the GPS error estimates (offsets) for the east and north direction, respectively, from the \( M^{th} \) event.

And the measurement equations are defined as,
\[
x_m(\tau) = x(\tau) + v_1(\tau) \tag{3.8}
\]
\[
y_m(\tau) = y(\tau) + v_2(\tau) \tag{3.9}
\]
\[
\sin \phi_m(\tau) = \sin \phi(\tau) + v_3(\tau) \tag{3.10}
\]
\[
\cos \phi_m(\tau) = \cos \phi(\tau) + v_4(\tau) \tag{3.11}
\]
\[
v_m(\tau) = v_5(\tau) \tag{3.12}
\]

The nonlinear state equations and measurement equations above can be written in matrix form,
\[
X_\tau = f(X_{\tau-1}) + w_{\tau-1} \tag{3.13}
\]
\[
Z_\tau = h(X_\tau) + v_\tau \tag{3.14}
\]

Where \( w_\tau = [w_1(\tau) \ w_2(\tau) \ w_3(\tau) \ w_4(\tau) \ w_5(\tau)]^T \) and \( v_\tau = [v_1(\tau) \ v_2(\tau) \ v_3(\tau) \ v_4(\tau) \ v_5(\tau)]^T \) represents the vector of process noise and the measurement noise, respectively. The function \( f \) can be used to compute the predicted state from the previous estimate and similarly the function \( h \) can be used to compute the predicted measurement from the predicted state.
$f$ and $h$ are nonlinear and cannot be applied directly. Hence, they are linearized about the previous and current states, respectively, and are written as follows (Welch & Bishop 1995)

$$X_{\tau} \approx \tilde{X}_{\tau} + A(X_{\tau-1} - \tilde{X}_{\tau-1}) + w_{\tau-1}$$ (3.15)

$$Z_{\tau} \approx \tilde{Z}_{\tau} + H(X_{\tau} - \tilde{X}_{\tau}) + v_{\tau}$$ (3.16)

Where

$X_{\tau}$ and $Z_{\tau}$ are the actual state and measurement vectors,

$\tilde{X}_{\tau}$ and $\tilde{Z}_{\tau}$ are the approximate state and measurement vectors defined as follows

$$\tilde{X}_{\tau} = f(\hat{X}_{\tau-1})$$

$$\tilde{Z}_{\tau} = h(\tilde{X}_{\tau})$$

$\hat{X}_{\tau}$ is the a posteriori estimate of the state at step $\tau$, and $A$ is the Jacobian matrix of partial derivatives of $f$ with respect to $x$, that is

$$A_{[i,j]} = \frac{\partial f_i}{\partial X_j}(\hat{X}_{\tau-1}) = \begin{bmatrix}
1 & 0 & v_x(\tau - 1) & 0 & \sin \phi_x(\tau - 1) \\
0 & 1 & 0 & v_y(\tau - 1) & \cos \phi_x(\tau - 1) \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}$$ (3.17)

$H$ is the Jacobian matrix of partial derivatives of $h$ with respect to $X$, that is an identity matrix.
At each time step the Jacobian is evaluated using the current predicted states. This process essentially linearizes the non-linear function around the current estimate. The complete set of EKF equations is shown below

\[
\dot{X}_\tau^- = f(\hat{X}_{\tau-1})
\]  
(3.19)

\[
P^-_{\tau} = A_{\tau} P_{\tau-1} A_{\tau}^T + Q
\]  
(3.20)

\[
K_{\tau} = P^-_{\tau} H_{\tau}^T (H_{\tau} P^-_{\tau} H_{\tau}^T + R)
\]  
(3.21)

\[
\hat{X}_{\tau} = \hat{X}_{\tau}^- + K_{\tau} (Z_{\tau} - h(\hat{X}_{\tau}^-))
\]  
(3.22)

\[
P_{\tau} = (I - K_{\tau} H_{\tau}) P_{\tau}^-
\]  
(3.23)

The filter is started with \( P_0 = I \), and the covariance matrix \( Q \) and \( R \) are both defined as fixed diagonal matrices,

\[
Q = \begin{bmatrix}
0.01 & 0 & 0 & 0 \\
0.01 & 0 & 0 & 0 \\
0 & 0.0001 & 0 & 0 \\
0 & 0 & 0.0001 & 0.01
\end{bmatrix}
\]
3.4 Lane status monitor – EWMA SPC control chart

Statistical process control (SPC) consists of a diverse set of tools for quality monitoring and process improvement. The most common method in the SPC “tool-set” is the control chart. A control chart is used to track changes in the mean and variance of a dependent-variable time-series (Shewhart 1931). The chart contains a center line that represents the average value of the quality characteristic (dependent variable or control data) being monitored and corresponds to the in-control state. Two additional horizontal lines, called the upper control limit (UCL) and the lower control limit (LCL) are also shown on the chart. These control limits represent the statistical decision value that is used to determine the in-control and out-of-control state of the process. As long as the points (control data) remain within the control limits, the process is assumed to be in-control, and no action is necessary. However, a point that exceeds the control limits is interpreted as evidence that the process is out-of-control. The traditional Shewhart control chart (Shewhart 1931) uses only the information about the process contained in the most recent sample observation and ignores any information given by the entire time series of points. The exponentially weighted moving average (EWMA) control chart utilizes a

\[
R = \begin{bmatrix}
0.08 & 0 & 0 \\
0.08 & 0.001 & 0 \\
0 & 0.001 & 0.1
\end{bmatrix}
\]
weighted average of all past and current data to detect small process shifts (Montgomery 2008).

The control data used in EWMA is defined as

\[ W(t) = \lambda Y(t) + (1 - \lambda)W(t - 1) \]  
\[ W(0) = \bar{Y} \]  

The UCL and LCL of the EWMA control charts are

\[ UCL = \bar{Y} + L\sigma \sqrt{\frac{\lambda}{2 - \lambda}}[1 - (1 - \lambda)^{2t}] \]  
\[ LCL = \bar{Y} - L\sigma \sqrt{\frac{\lambda}{2 - \lambda}}[1 - (1 - \lambda)^{2t}] \]

Where \( Y(t) \) is the observation at time \( t \), and \( W(t) \) is the EWMA data at time \( t \). \( \sigma \) is the standard deviation of control variable \( Y \). The starting value \( W(0) \) is equal to the average of preliminary data, \( \bar{Y} \). \( \lambda \in (0,1] \) is a constant which assigns weight between new data and past data. \( L \) is a factor which defines sensitivity of detection and false alarms and can be interpreted as a multiplier of the standard deviation for control limits.

To implement EWMA SPC control on lane changing detection, the observation data is defined as,

\[ Y(\tau) = \varepsilon_h(\tau) = 180 \ast \frac{\phi_v(\tau) - \phi_l(\tau)}{\pi} \]  

Where \( \varepsilon_h(\tau) \) is the heading error (degrees) between the vehicle heading \( \phi_v(\tau) \) (radians) and the lane heading \( \phi_l(\tau) \) (radians) at time \( \tau \).
In this chapter, both the detection of lane change events and turn (right or left) events are important. Lane change events are used to update the vehicle lane status. For the detection of lane change events, one assumption is made:

Assumption: If the curvature of roadway is not sufficiently captured by discrete waypoints, then splines would be required from the RSE MAP (this is not addressed in this implementation).

Figure 3.5 shows the components of the lane change model. Assumption 1 ensures that the curvature of the roadway can be captured at any time. To simplify calibration, average heading error $\overline{\varepsilon_h}$ is calculated from $\varepsilon_h(\tau)$.

The UCL and LCL of lane change events are defined as,

$$UCL_{lane} = \overline{Y} + L_{lane}\sigma_{lane}\sqrt{\frac{\lambda}{2-\lambda}} [1 - (1 - \lambda)^{2\tau}]$$  \hspace{1cm} (3.29)

$$LCL_{lane} = \overline{Y} - L_{lane}\sigma_{lane}\sqrt{\frac{\lambda}{2-\lambda}} [1 - (1 - \lambda)^{2\tau}]$$  \hspace{1cm} (3.30)

and the UCL and LCL of turn events could be defined as,

$$UCL_{turn} = \overline{Y} + L_{turn}\sigma_{turn}\sqrt{\frac{\lambda}{2-\lambda}} [1 - (1 - \lambda)^{2\tau}]$$  \hspace{1cm} (3.31)
Since the common-mode GPS errors are corrected by differential corrections, the average heading error $\bar{Y}$ should be zero. $L_{\text{lane}}$ and $L_{\text{turn}}$ are usually equal to 3, as typical 3-sigma control limits. The most important parameter is the standard deviation of heading error $\sigma_{\text{lane}}$, which affects the UCL and LCL of lane change events.

In order to calibrate $\sigma_{\text{lane}}$, the lane changing process is modeled as a deterministic process given assumption 2, as shown in Figure 3.5. $T$, $l_n$, $v_k$, $v_c$ and $|\varepsilon_h|$ denote the duration of lane changing, the lane width, lane longitudinal speed, lateral speed and the absolute value of average heading error in the lane change, respectively. The process of lane changing could be described as follows: Suppose lane changing starts from time $\tau = 1$, EWMA SPC control data $W(\tau)$ will increase until $\tau = T$, when the lane changing

$$LCL_{\text{turn}} = \bar{Y} - L_{\text{turn}} \sigma_{\text{turn}} \sqrt{\frac{\lambda}{2 - \lambda} \left[1 - (1 - \lambda)^{2\tau}\right]}$$

(3.32)
process completes, shown as equation (3.33)-(3.36).

\[ W(1) = \bar{\lambda} \bar{\varepsilon}_h \]  

(3.33)

\[ W(2) = (1 - \lambda) \bar{\lambda} \bar{\varepsilon}_h + \bar{\lambda} \bar{\varepsilon}_h \]  

(3.34)

\[ W(T) = \left( (1 - \lambda)^{T-1} + (1 - \lambda)^{T-2} + \ldots + 1 \right) \bar{\lambda} \bar{\varepsilon}_h \]  

(3.35)

\[ W(T) = \left( 1 - (1 - \lambda)^T \right) \bar{\varepsilon}_h \]  

(3.36)

A lane changing event is claimed to be detected, if \( W(T) \) is greater than UCL or less than LCL, depicted in equation (3.37).

\[ |W(T)| \geq 3 \sigma_{\text{lane}} \sqrt{\frac{\lambda}{2 - \lambda}} \]  

(3.37)

If a driver changes behavior and does not commit to the lane change, the data may indicate the start of the change, but will include the return to the original lane. The latest time to “regret” is \( \tau = \frac{T}{2} \). Therefore \( W(T/2) \) should stay in the control limits in order to avoid false alarm, shown as equation (3.38).

\[ |W(T/2)| \leq 3 \sigma_{\text{lane}} \sqrt{\frac{\lambda}{2 - \lambda}} \]  

(3.38)

A bound on \( \sigma_{\text{lane}} \) can be defined as (3.39) by combining (3.36)-(3.38) as

\[ \bar{\varepsilon}_h \left[ \frac{1 - (1 - \lambda)^{T/2}}{3 \sqrt{\frac{\lambda}{2 - \lambda}}} \right] \leq \sigma_{\text{lane}} \leq \bar{\varepsilon}_h \left[ \frac{1 - ((1 - \lambda)^T)}{3 \sqrt{\frac{\lambda}{2 - \lambda}}} \right] \]  

(3.39)

\(|\bar{\varepsilon}_h|\) in equation (3.39) can be substituted by a inverse trigonometric function
arctan(·), derived from Figure 3.5. (Suppose the output of arctan(·) is in degrees)\\

$$[\mathcal{E}_k] = \arctan \left( \frac{l_n}{v_k T} \right)$$ \tag{3.40}\\

Therefore, the bound of $\sigma_{\text{lane}}$ is determined by lane width $l_n$, vehicle speed $v_k$ and lane changing duration $T$ given a fixed $\lambda$, as shown in (3.41).

$$\arctan \left( \frac{l_n}{v_k T} \right) \left( 1 - \left( 1 - \lambda \right)^{\frac{T}{2}} \right) \leq \sigma_{\text{lane}} \leq \arctan \left( \frac{l_n}{v_k T} \right) \left( 1 - \left( 1 - \lambda \right)^{\frac{T}{2}} \right)$$ \tag{3.41}\\

The false alarm (false positive) and miss detection (false negative) errors are both undesirable for lane changing detection. The smaller the value of $\sigma_{\text{lane}}$, the higher the probability of a false alarms. The larger the value of $\sigma_{\text{lane}}$, the higher the probability of miss detection. Therefore, the median value is selected to be the value of $\sigma_{\text{lane}}$, shown in equation (3.42).

$$\sigma_{\text{lane}} = \frac{\arctan \left( \frac{l_n}{v_k T} \right)}{6 \sqrt{\lambda}} \left( \left( 1 - \left( 1 - \lambda \right)^{\frac{T}{2}} \right) + \left( 1 - \left( 1 - \lambda \right)^{\frac{T}{2}} \right) \right)$$ \tag{3.42}\\

In (3.42), $\lambda$, $l_n$ and $v_k$ are assumed known. The only random variable is the duration of the lane changing event, $T$. The duration of the lane change event is modeled by Toledo and Zohar in (Toledo & Zohar 2007). They found that the range of lane change duration varies from 1 second to 13 seconds with mean 4.6 seconds and standard deviation 2.3 seconds. Thiemann et al. (Thiemann et al. 2008) examined the Next Generation Simulation data (NGSIM) (Federal Highway Administration 2009) from
Federal Highway Administration (FHWA) and showed that the mean duration of lane changing is 4.01 seconds with standard deviation 2.31 seconds, which comply with the findings of Toledo and Zohar.

To implement the EWMA SPC control chart, the initial value of $T$ for real-time application can be determined as the mean of lane changing duration in previous studies, approximately 4~5 seconds. Since different people have different driving behaviors, it is likely to have a different $T$ for each driver. Given the assumption that drivers behave somewhat consistently when changing lanes, $T$ could be estimated from driver’s historical lane changing data and the position of surrounding vehicles by vehicle-to-vehicle communication, which could be considered as future research. The relationship

![Relationship between $\sigma_{lane}$ and $T$](image)

**Figure 3.6** The relationship between $\sigma_{lane}$ and $T$, given $\lambda = 0.4$, $l_n = 3.2m$ and $v_k = 13.33m/s$
between $\sigma_{lane}$ and $T$ can be described as shown in Figure 3.6 as a monotonically decreasing curve, given fixed $\lambda$, $l_n$ and $v_k$. 
Figure 3.7 (a) An example of vehicle events detection; (b) Events detection with EWMA control chart
When a turn even occurs, the EWMA control data, $W(\tau)$, exceeds the control limits established for a turn event. Both the lane and vehicle headings will change when the segment status is updated to match the MAP map, hence the EWMA control data, $W(\tau)$, will drop back into the turn detection control limits. However, the EWMA control data will not converge back into the control limits of lane change event immediately even if the new heading observation is within the lane control limits due to the “inertial effect” of EWMA control chart (Montgomery 2008). It usually takes several seconds for the EWMA data to drop into the control limits of lane change event after a turn event is detected. As a result, any new lane change event occurring within the duration of “inertial effect” would not be detected. This inertial lag can be addressed by restarting a new control chart when a new turn event occurs.

An example of the EWMA control chart applied to driving event detection is shown in Figure 3.7(a). A vehicle heads eastbound, then merges into the right lane and executes a right turn. The on-board GPS receiver outputs data every 1 second. The process of vehicle movement is monitored by the EWMA control chart as shown in Figure 3.7(b). The heading error $\varepsilon_h(t)$ is treated as raw data. Every second the EWMA data $W(t)$ is calculated by using (3.24). The parameter values are set to: $\bar{Y} = 0$, $\lambda = 0.4$, $L_{lane} = 3$, $L_{turn} = 3$, and $\sigma_{turn} = 18$. According to field data $l_n = 3.3m$, $v_k = 11.5m/s$ and $T = 5s$, $\sigma_{lane}$ is equal to 1.799 by equation (3.42). The UCL and LCL for lane change event detection and turn event detection are: $UCL_{lane} = 2.7$, $LCL_{lane} = -2.7$, $UCL_{turn} = 27$, $LCL_{turn} = -27$. To address the latency of lane changing detection, the model proposed in equation 3.38
shows that the lane changing can be detected after half time of lane time, which can be validated in the example in Figure 3.7(b). The lane changing time takes about 5~6 seconds by counting from third heading error points (stars in Figure 3.7 (b)) deviated from zero line. The EWMA data exceeds UCL at third seconds, when the lane changing event is detected. So our proposed algorithm can detect lane change once the vehicle passes the lane marker, which is assumed to be the middle of lane changing time.

3.5 Turn event-driven lane alignment optimization

The EKF and SPC control chart provides state updates and detects turning events. However, the vehicle position is still uncertain due to the potentially noncommon-mode GPS errors (0.1-4m). In order to estimate the noncommon-mode error, the vehicle turning inbound and outbound trajectories are combined with the MAP to measure the vehicle offset from the estimated actual lane inbound and outbound trajectories, shown in Figure 3.8 (a) & (b). A turn event-driven lane alignment optimization problem is solved to estimate the average noncommon-mode error in the process of turn movements.

To better understand driver’s behavior for turn movement, more than 2000 turns are observed in NGSIM raw video data in Lankershim Boulevard in Los Angeles, CA and Peachtree Street in Atlanta, GA. Table 3.1 shows that the probability of not drifting lanes is pretty low, about 0.6 for left turns and 0.4 for right turns. Therefore, it is hard to precisely predict which lane the driver selects after the turn. However, the lane number could be set to an initial estimate using the map-matching algorithm. If the initial lane number is correct, the subsequent lane changing events will be reasonable. Otherwise, if
the initial lane estimate is incorrect, the subsequent lane changing may violate the geometry of roadway, for example, a detected right lane change violates the previous status that the vehicle was in the most right lane. The optimization problem can be re-solved to re-estimate noncommon-mode error after the vehicle’s previous lane number is determined.
Figure 3.8 (a) GID map with inbound-outbound trajectory of two type turns; (b) Line approximations for inbound-outbound trajectories of right turn
There are two characteristics of this problem that provide an opportunity to estimate the error: First, the GPS inbound-outbound trajectory across the intersection contains information about the direction of the turn that the driver makes. Second, given a MAP and a turn type, the “true” inbound-outbound trajectories can be compared to the measured trajectory and the GPS error \((\Delta'_x, \Delta'_y)\) can be estimated from the lane and turn alignments.

Two scenario sets \(S_x\) and \(S_y\) are created by sampling some of the recorded GPS points. \(S_x\) contains some \(x_s, s \in S_x\) which are used to test the vertical distance between two horizontal lines, while \(S_y\) contains some \(y_s, s \in S_y\) which are used to estimate the horizontal distance between two vertical lines as shown as Figure 3.8(b).

The lane alignment optimization problem can be stated as:

Objective function: \(\min Z = \bar{\Delta}_x + \bar{\Delta}_y\)
Subject to the constraints:

\[ \Delta_x = \frac{1}{|S_y|} \sum_{x \in S_y} (x_m(y_s) - x(y_s))^2 \]

\[ \Delta_y = \frac{1}{|S_x|} \sum_{y \in S_y} (y_m(x_s) - y(x_s))^2 \]

\[ x_m(y_s) = cy_s - c\Delta_y + d + \Delta_x, \quad s \in S_y \]

\[ x(y_s) = c'y_s + d' \quad s \in S_y \]

\[ y_m(x_s) = ax_s - a\Delta_x + b + \Delta_y, \quad s \in S_x \]

\[ y(x_s) = a'x_s + b' \quad s \in S_x \]

This optimization problem can be solved as a weighted least square problem. \( Z \) is minimized when its gradient with respect to each variable is equal to zero,

\[
\begin{cases}
\frac{\partial Z}{\partial \Delta'_x} = 0 \\
\frac{\partial Z}{\partial \Delta'_y} = 0
\end{cases}
\] (3.43)

The optimal solution is derived from (3.43) as follows,

\[
\begin{aligned}
\Delta'_x &= \frac{(c^2 + 1)K_1 + (a + c)K_2}{2(ac-1)^2} \\
\Delta'_y &= \frac{(a^2 + 1)K_2 + (a + c)K_1}{2(ac-1)^2}
\end{aligned}
\] (3.44)

Where \( K_1 = \frac{2a}{|S_x|} \sum_{x \in S_x} X_x - \frac{2}{|S_y|} \sum_{y \in S_y} Y_y \) and \( K_2 = \frac{2c}{|S_y|} \sum_{y \in S_y} Y_y - \frac{2}{|S_x|} \sum_{x \in S_x} X_x \)

Therefore, the GPS noncommon-mode error can be roughly captured from turn events.
3.6 Field data results

To evaluate the effectiveness of the pseudo-lane level position estimation system, an experiment was conducted around the Mountain and Speedway intersection in Tucson, AZ as shown in Figure 3.9. The OBE installed on the test vehicle features a 500Mhz processor, 256MB of memory, 4GB of compact flash disk space, multiple radios (WiFi, DSRC) and an integrated USB GlobalSat BU-353 GPS receiver and antenna. The DSRC communication range of v2i has been measured to be about 600~700meters in this test site.

Figure 3.9 The test intersection in Tucson, AZ

Once a turn is detected and finished, two linear equations are estimated by the trajectory lines. The optimal solution is calculated by (3.44), which is considered as the
GPS noncommon-mode error. Then the states of the EKF are applied with this GPS drift error. The GPS error is corrected just after the turn event as shown in Figure 3.10. A lane change detection example with 1-lane left turn, 1-lane right turn and 2-lane left turn is depicted in Figure 3.11.

Figure 3.10 GPS noncommon-mode error fixed by lane alignment optimization for the left turn event
Three routes were driven with each having different lane change durations. The results are summarized in Table 3.2. The Number of false positives (a detected event which is false) and the number of false negatives (a true event without detection) are both equal to 2, out of 61 total lane-change events. The total lane changing detection rate is about 93%. All of the turns were detected because heading errors are significant and easy to detect. There was no observed error for turn detection. Even though a few lane change detection failures occurred due to the inaccuracy of the low cost GPS, the vehicle lane status was reset each time that a turn event was detected.
3.7 Summary

A pseudo lane-level positioning system was developed using only low cost GPS in a v2i environment. The system consists of three major components: v2i communication of GPS common-mode corrections; an Exponentially Weighted Moving Average (EWMA) control chart monitor for lane changing and turn detection; and, a lane alignment optimization model to estimate the noncommon-mode GPS errors. Pseudo lane-level positioning is achieved under the assumptions that no GPS outage occurs, drivers change lanes at a constant speed, and the roadway geometry is well captured in the MAP.

Future research will focus on solving some potential problems existing in this initial approach. First, the problem of GPS blockage can be addressed by other vehicle sensors,

<table>
<thead>
<tr>
<th></th>
<th>1st experiment</th>
<th>2nd experiment</th>
<th>3rd experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>lane change duration (sec)</td>
<td>2~3</td>
<td>4~6</td>
<td>7+</td>
</tr>
<tr>
<td># of lane changing</td>
<td>20</td>
<td>22</td>
<td>19</td>
</tr>
<tr>
<td># of false positives</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td># of false negatives</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
such as gyro, odometer and vehicle wheel encoders that can also address the GPS drift issue at low speeds and provide important information about $\sigma_{\text{lane}}$. Second, GPS noncommon-mode error estimation on a straight road is another challenge. It is likely that some vehicles will not execute turning maneuvers and the time between GPS offset updates may be long enough that the GPS drift will significantly affect positioning. This might be addressable by cooperative sharing of information using vehicle-to-vehicle (v2v) communications where all of the equipped vehicles on a street share GPS offset estimates and updates based on the population information.
CHAPTER 4

ROBUST ACTUATED PRIORITY TRAFFIC SIGNAL CONTROL WITH VEHICLE-TO-INFRASTRUCTURE COMMUNICATIONS

This chapter examines priority based traffic signal control using vehicle-to-infrastructure communication to send priority requests from a vehicle to an intersection. Priority control allows certain classes of vehicles, such as emergency vehicles or buses, to receive preferential treatment at a traffic signal. At any time it is possible that more than one qualified vehicle is approaching an intersection and each vehicle may send a priority request. The traffic control algorithm must consider multiple requests as well as be robust to uncertainty in the desired service time. Given the current information from multiple priority requests, a mixed integer linear mathematical program (MILP) is solved for each intersection to obtain an optimal signal timing plan. This chapter first presents a deterministic MILP that is applicable for control of multiple emergency vehicles. Then a robust MILP is developed for transit vehicles (e.g. buses) where the desired service time might be uncertain due to operational factors such queuing and passenger boarding/alighting. The solution to the robust control problem is integrated with actuated traffic signal control to be responsive to real-time non-priority vehicle demand. Finally, coordination between adjacent signals is achieved by adding coordination requests along with other priority requests. Due to the limited number of binary variables in the formulation, the robust MILP can be implemented in real-time signal control. The
proposed approach is compared with state-of-practice coordinated-actuated traffic signal control with transit signal priority (TSP) under several operating scenarios using microscopic traffic simulation. The simulation experiments show that the priority based traffic signal control is able to reduce transit delay by 18% and all vehicle delay by 3%.

4.1 Introduction

With the advent of IntelliDrive\textsuperscript{SM} for Mobility in United States (RITA 2010)(AASHTO 2009), it may soon be possible to obtain additional information about the network state and vehicle operations. IntelliDrive\textsuperscript{SM}, formerly known as Vehicle Infrastructure Integration (VII) (Faradyne 2005), is a suite of technologies and applications that use wireless communications to provide connectivity which includes vehicle-to-vehicle (v2v) communication and vehicle-to-infrastructure (v2i) communication, called v2x in general.

A variety of challenges as well as problems arise when considering traffic control within a v2x environment. One of the challenges is how to implement priority operations that resolve multiple conflicting requests from a variety of different classes of vehicles, including emergency vehicles and buses. With v2i communication in an IntelliDrive\textsuperscript{SM} world, vehicle information will be able to be obtained up to 1000 meters away from the road-side equipment (RSE) at an intersection.

Traditional priority control system in United States can be categorized into Emergency vehicle preemption and transit signal priority (TSP). An emergency vehicle can request signal preemption treatment by using either optical, acoustic, special inductive loop
technology, or based on Global Positioning System (GPS) positions (Nelson & D. Bullock 2000). Preemption generally involves a control strategy that immediately switches from current phase to a pre-selected phase for the first received request. Transit priority can use the same technology, but can be served by minor modifications to traffic signal plan parameters (offset adjustment, green split reallocation, phase insertion or phase rotation) to favor the movements of transit vehicles (Evans & Skiles 1970) (Yagar & Han 1994) (Balke et al. 2000) (Furth & Muller 2000) (Skabardonis 2000)(Baker et al. 2002) (Head 2002) (H. Liu et al. 2003) and (H. Smith et al. 2005).

In current emergency vehicle preemption systems, only one request is served at a time. Therefore, if multiple vehicles are approaching an intersection at one time and they request conflicting phases the first request received would be served even if a safer and more efficient solution could be achieved by considering all active request simultaneously. While emergency vehicle operators are trained to be observant and vigilant, there have been cases where two emergency vehicles have collided in an intersection (ABC13 2009). Roadway safety has been noted as a significant emergency responder issue (The Transportation Safety Advancement Group 2010). With vehicle-to-infrastructure communication systems, the signal controller can generate an optimal signal plan for multiple requests and send feedback back to the vehicles so that they are aware of potential conflicting requests and planned signal controls.

Transit signal priority is a popular tool for improving transit performance and reliability (H. Smith et al. 2005). Typically the priority strategy include extending a phase to allow a transit vehicle to pass or terminating conflicting phases allowing early service
to reduce delay. However, it is possible, and maybe likely, that more than one bus may arrive on conflicting approaches at an intersection during a cycle. In this case there is a need to simultaneously consider the multiple requests for priority is a way that is not disruptive, or inefficient, to other traffic. Other factors, such as occupancy and schedule adherence are important considerations that can be used to manage priority requests, but it is still likely that multiple transit vehicles will desire priority.

Advanced communication technologies have been applied on transit signal priority (TSP) control projects in the past (G. Chang et al. 1996) (Liao & Davis 2007) and (Ekeila et al. 2009). However, very few references can be found that address the multi-priority request issue. Head et al. (Head et al. 2006a) proposed a mixed integer nonlinear programming (MINP) formulation which could accommodate multiple priority requests and minimize the total priority delay. However, there are several shortcomings in this formulation. First, the MINP formulation takes relatively long time to generate optimal solutions due to its nonlinearity. Second, bus arrival times are assumed to be deterministic, which is reasonable for emergency vehicles but not realistic for buses. Third, vehicle actuation is not incorporated into the formulation which limits the control to behave as fixed-time and does not allow for vehicle detection to be used to take advantage of gaps in traffic flow.

The goal of this chapter is to address the multiple priority request issue within coordinated traffic signal operations through a robust mathematical optimization approach and an implementation that allows vehicle actuation. Robustness of proposed approach is represented in two-folds. The first robustness is from the definition of robust
optimization. Robust optimization is defined as a modeling methodology, combined with computational tools, to process optimization problems in which the data are uncertain and is only known to belong to some uncertainty set (Ben-Tal and Nemirovski 2002). Estimated time arrivals of priority request are considered as an uncertainty interval with unknown distribution, rather than a point of time arrival. The other robustness is including actuated control when it comes to implementation of our proposed methods, since future vehicle actuations are unknown. Therefore, priority control and vehicle actuations are both considered to improve priority delay as well as traffic delay.

In this paper, three traffic modes are considered: emergency vehicles, buses, and passenger vehicles, within a decision framework that can accommodate pedestrians and bicycles. One assumption is made:

*The sequence of phases in a ring is fixed and phase skipping is not allowed.*

This is a reasonable assumption since phase rotation and skipping can cause confusion to the motorist, loss of coordination, and long delay to the traffic stream (Skabardonis 2000). It is understood that phase rotation, such as lead-lag and lag-lead, can produce useful behavior is some circumstances.

This chapter is organized as follows. Table 4.1 contains a summary of the model notation. Section 4.2 introduces a new signal plan modeling tool called phase-time diagram, which enhances the precedence graph representation of a dual-ring controller developed in (Head et al. 2006a) and provides a useful visualization of the priority timing problem. Section 4.3 presents a deterministic mixed integer linear program (MILP) formulation to linearize the MINP formulation in (Head et al. 2006a). This deterministic
MILP formulation is designed for emergency vehicles, which are assumed to travel at a constant speed. Section 4.4 addresses the uncertainty of bus arrival and proposes a robust MILP formulation. In Section 4.5, actuated control is integrated into the robust formulation to improve the efficiency of the control strategy by reducing the delay for passenger cars. In Section 4.6, additional constraints for signal coordination are added into the robust MILP formulation to simultaneously consider both a progression, or “green wave”, for passenger vehicles and priority for requesting vehicles. Section 4.7 presents two numerical examples for isolated intersections and coordinated intersections, respectively, comparing the different strategies. Concluding remarks along with future extensions are reported in the Section 4.8.

Table 4.1 Symbol definition of decision variables and data

<table>
<thead>
<tr>
<th>Type</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets</td>
<td>$p \in P$</td>
<td>The set of phases</td>
</tr>
<tr>
<td></td>
<td>$p_c \in P_c$</td>
<td>The set of coordinated phases $P_c \subseteq P$</td>
</tr>
<tr>
<td></td>
<td>$k \in K$</td>
<td>The set of cycles</td>
</tr>
<tr>
<td></td>
<td>$(j, p) \in J \times P$</td>
<td>The set of priority requests $(j\text{th request that is active for phase } p)$, $J \subseteq \mathbb{Z}$ ($J$ is a subset of the integers)</td>
</tr>
<tr>
<td></td>
<td>$(p_c, k) \in P_c \times K$</td>
<td>The set of coordination requests (for phase $p_c$ in cycle $k$)</td>
</tr>
<tr>
<td>Decision</td>
<td>$a_{pk}$</td>
<td>Maximal available green extension time for actuated control at phase $p$ during cycle $k$</td>
</tr>
<tr>
<td>variables</td>
<td>$d_{jpk}$</td>
<td>Priority delay in cycle $k$ for priority request $(j, p)$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>$d'_{jpk}$</td>
<td>Maximal priority delay in cycle $k$ for priority request $(j,p)$</td>
<td></td>
</tr>
<tr>
<td>$d'_{pck}$</td>
<td>Coordination delay in cycle $k$ for coordinated phase $p_c$</td>
<td></td>
</tr>
<tr>
<td>$g_{pk}$</td>
<td>Green time for phase $p$ during cycle $k$</td>
<td></td>
</tr>
<tr>
<td>$g'_{pk}$</td>
<td>Necessary green time (the maximal one of minimal green and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>pre-allocated green time) for phase $p$ during cycle $k$ starting from</td>
<td></td>
</tr>
<tr>
<td>$t'_{pk}$</td>
<td>$t'_{pk}$ Starting time of phase $p$ during cycle $k$</td>
<td></td>
</tr>
<tr>
<td>$t'_{pk}$</td>
<td>Latest starting time of phase $p$ during cycle $k$</td>
<td></td>
</tr>
<tr>
<td>$v_{pk}$</td>
<td>Phase duration time of phase $p$ during cycle $k$, including clearance</td>
<td></td>
</tr>
<tr>
<td>$v'_{pk}$</td>
<td>Maximal phase duration time of phase $p$ during cycle $k$, including</td>
<td></td>
</tr>
<tr>
<td></td>
<td>clearance time</td>
<td></td>
</tr>
<tr>
<td>$\theta_{jpk}$</td>
<td>$\theta_{jpk} = 1$, the priority request $(j,p)$ is served in cycle $k$; else, not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>served in cycle $k$</td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>Common cycle length for coordination</td>
<td></td>
</tr>
<tr>
<td>$g_{min}^{pk}$</td>
<td>Minimal green time for phase $p$ during cycle $k$</td>
<td></td>
</tr>
<tr>
<td>$g_{max}^{pk}$</td>
<td>Maximal green time for phase $p$ during cycle $k$</td>
<td></td>
</tr>
<tr>
<td>$M$</td>
<td>A large number</td>
<td></td>
</tr>
<tr>
<td>$O$</td>
<td>Offset for the current intersection</td>
<td></td>
</tr>
<tr>
<td>$r_p$</td>
<td>Red clearance time for phase $p$</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>$Q_{jp}$</td>
<td>Necessary green time to clear the queue for priority request $(j,p)$</td>
<td></td>
</tr>
<tr>
<td>$Q_{p,k}^c$</td>
<td>Necessary green split for coordinated request $(p_c,k)$</td>
<td></td>
</tr>
<tr>
<td>$R_{jp}$</td>
<td>Estimated time of arrival for priority request $(j,p)$</td>
<td></td>
</tr>
<tr>
<td>$R_{p,k}^c$</td>
<td>Estimated time arrival for coordination request $(p_c,k)$</td>
<td></td>
</tr>
<tr>
<td>$\overline{R}_{jp}$</td>
<td>Lower bound of estimation of time arrival for priority request $(j,p)$</td>
<td></td>
</tr>
<tr>
<td>$\overline{R}_{p,k}$</td>
<td>Upper bound of estimation of time arrival for priority request $(j,p)$</td>
<td></td>
</tr>
<tr>
<td>$R_{p,k}^c$</td>
<td>Lower bound of time interval for coordination request $(p_c,k)$</td>
<td></td>
</tr>
<tr>
<td>$\overline{R}_{p,k}^c$</td>
<td>Upper bound of time interval of time arrival for coordination request $R_{p,k}^c$</td>
<td></td>
</tr>
<tr>
<td>$w_{jp}$</td>
<td>Weight for priority request $(j,p)$</td>
<td></td>
</tr>
<tr>
<td>$w_{p,k}$</td>
<td>Weight for coordination request $(p_c,k)$</td>
<td></td>
</tr>
<tr>
<td>$y_p$</td>
<td>Yellow clearance time for phase $p$</td>
<td></td>
</tr>
</tbody>
</table>

4.2 Phase-time diagram: a new tool to model traffic signal controller logic

The signal controller model considered in this research is based on the standard North American NEMA dual-ring, eight-phase controller. A four-legged intersection with eight movements is shown in Figure 4.1(a). Typically, each ring in the controller contains 4 phases, depicted in Figure 4.1(b). A barrier exists that crosses both rings between groups
of conflicting movements so that all phases in one group have to terminate before any phase in the other group starts.

![Dual-ring controller diagram](image)

**Figure 4.1 Dual-ring, eight-phase controller**

The dual-ring controller can be modeled by a traditional precedence graph as depicted in Figure 4.2 (Head et al. 2006b). Arcs in the precedence graph represent the duration of phases, while nodes represent the phase transitions. Phase intervals can be easily visualized in the precedence graph by decomposing each arc into its respective interval precedence graph. However, the precedence graph is only a one dimensional graph which has some drawbacks: 1) the slope of arcs and vertical position of nodes are not quantified; 2) the feasible timing region defined by minimal and maximal green is not visualized; 3) The priority requests are shown only as events in time with an association to an arc, not as possibly being served in multiple cycles.
To better model the signal controller logic with priority requests, a new tool called a phase-time diagram is proposed. Given the previous assumption of a fixed phase sequence, a phase-time coordinate system is constructed with one horizontal time axis and two vertical phase axes, as shown in Figure 4.3. The origin denotes the current time and current phase. Phases in ring 1 are evenly distributed on the left vertical axis in a

![Phase-time diagram representation of a dual-ring controller](image)

Figure 4.2 Precedence graph representation of a dual-ring controller

![Phase-time diagram representation of a dual-ring controller](image)

Figure 4.3 Phase-time diagram representation of a dual-ring controller

To better model the signal controller logic with priority requests, a new tool called a phase-time diagram is proposed. Given the previous assumption of a fixed phase sequence, a phase-time coordinate system is constructed with one horizontal time axis and two vertical phase axes, as shown in Figure 4.3. The origin denotes the current time and current phase. Phases in ring 1 are evenly distributed on the left vertical axis in a
sequence starting from the current phase, while the phases in ring 2 are shown on the right vertical axis. Based on the initial settings, the properties of phase-time diagram are listed as below:

1. (Physical meaning of arc slope) Nodes represent phase transition events, and arcs represent phase duration time. The slope of the arcs depends on the phase duration. If a phase times the minimum time, the arc will be short with a high slope \( \frac{1}{g_{pk}^{\text{min}}} \). If a phase times for the maximal time, the arc will be long with a low valued slope \( \frac{1}{g_{pk}^{\text{max}}} \).

2. (Feasible region) Any piecewise line starting from the origin stands for a signal plan in the phase-time diagram. However, the feasible region of the signal plan is bounded in a fan-shaped area by the shortest path (fastest timing as determined by each phase’s minimal green times) and the longest path (slowest time as determined by each phase’s maximal green times). Any piecewise linear path through each phase between the shortest and longest path is feasible.

3. (Request representation) A priority (or service) request is associated with a desired service time and service phase (P) is denoted as \( R_j \), which represents the arrival time of \( j \)th request for phase \( p \). Any request \( R_{jp} \) will be served in one of the future cycles during phase \( p \) depending on the realization of the signal plan. \( R_{jp} \) can be depicted as cyclic serving bars (CSB) on the phase-time diagram.

4. (Delay visualization) There are two cases that occur when a request \( R_{jp} \) is served by a signal plan: without delay or with delay. If the piecewise line of a signal plan intersects a CSB at any point, \( R_{jp} \) is served at that point in time without delay. If the
piecewise line of a signal plan does not intersect any of the CSBs, $R_{jp}$ gets served at the moment when the piecewise line cross right hand side of a CSB for the first time. The corresponding delay for $R_{jp}$ is horizontal distance from the piecewise line to the served CSB, depicted as dashed lines in Figure 4.3.

Two priority requests $R_{18}$ and $R_{13}$ are shown as CSBs in Figure 4.3. $R_{18}$ is served in cycle 1 by phase 8 in ring 2 as shown by the piecewise line. $R_{13}$ is served in cycle 2 by phase 3 in ring 1. However, both of requests are delayed in this illustration for the sample signal plan. Thus the delay for $R_{18}$ occurs during the first cycle and the delay for $R_{13}$ during the second cycle. The delay is calculated as $d_{181} = t_{81} - R_{18} = 21$ seconds; $d_{132} = t_{32} - R_{13} = 9$ seconds.

The diagram presented here is completely extensible to controllers with more or fewer rings and phases and can accommodate any ring, barrier, and phase configuration. In fact, the phase-time diagram can be considered as a projection from precedence graph with determined vertical positions for each node.

The phase-time diagram also provides an intuitive visualization for the problem of determining the best signal timing plan given a set of priority requests. Any piecewise linear path in the feasible region can be selected as the timing plan. The path that minimizes the total delay would be the optimal plan. Similarly, the phase-time diagram will allow consideration of other controller behaviors – such as coordination and phase actuation in subsequent research.
4.3 Deterministic mixed integer linear program (MILP) formulation

Head et al. (Head et al. 2006a) proposed a mixed integer nonlinear programming (MINP) formulation which could accommodate multiple priority requests and minimize the total priority delay. However, the solution times increase dramatically due to nonlinearity of the formulation. For practical purposes, a MILP formulation is proposed in this section to linearize the MINP formulation in (Head et al. 2006a). Also the total number of integer variables is reduced by 50% in MILP formulation. The MILP model is:

Minimize \[ \sum_{j,p,k} w_{jp} d_{jpk} \]  

\[ (4.1) \]

Subject to

\[ t_{1,1} = 0, t_{5,1} = 0 \]  

\[ (4.2) \]

\[ t_{2,k} = t_{1,k} + v_{1,k} \quad \forall k \]  

\[ (4.3) \]

\[ t_{6,k} = t_{5,k} + v_{5,k} \quad \forall k \]  

\[ (4.4) \]

\[ t_{3,k} = t_{2,k} + v_{2,k}, \quad t_{3,k} = t_{6,k} + v_{6,k} \quad \forall k \]  

\[ (4.5) \]

\[ t_{7,k} = t_{2,k} + v_{2,k}, \quad t_{7,k} = t_{6,k} + v_{6,k} \quad \forall k \]  

\[ (4.6) \]

\[ t_{4,k} = t_{3,k} + v_{3,k} \quad \forall k \]  

\[ (4.7) \]

\[ t_{8,k} = t_{7,k} + v_{7,k} \quad \forall k \]  

\[ (4.8) \]

\[ t_{1,k+1} = t_{4,k} + v_{4,k}, \quad t_{4,k+1} = t_{8,k} + v_{8,k} \quad \forall k \]  

\[ (4.9) \]

\[ t_{5,k+1} = t_{4,k} + v_{4,k}, \quad t_{5,k+1} = t_{8,k} + v_{8,k} \quad \forall k \]  

\[ (4.10) \]

\[ v_{pk} = g_{pk} + y_{p} + r_{p} \quad \forall p,k \]  

\[ (4.11) \]
The objective of the mathematical model is to minimize the total weighted delay for all active priority requests. The weights $w_jp$ can be considered as function of real-time bus occupancy and adherence of schedule or as mission priority for emergency vehicles. Constraints (4.2)-(4.10) represent the same precedence constraints depicted in Figure 4.2 to model a set of precedence relationships starting at $t = 0$ and considering $|K|$ total cycles.

Different initial phases will result in different structure of constraints from (4.2)-(4.10). There are 8 different starting phase combinations given the standard NEMA 2-ring 8-phase configuration. The precedence constraints can be modified according to alternative ring configurations. Constraints (4.11)-(4.12) are phase interval constraints which define the feasible phase duration based on minimal and maximal green times as well as yellow and all red clearance intervals. Constraints (4.13)-(4.16) are enhanced phase service selection constraints. As depicted in Figure 4.3, request $(j,p)$ (at time $R_{jp}$)
could be served in several cycles by phase $p$ on CSBs. Binary variables $\theta_{jp}$ are introduced to address the combinatorial optimization problem to select the cycle to serve request $(j,p)$ . When $\theta_{jp} = 1$, $R_{jp}$ is bounded by constraints (4.13) and (4.14), which are modified to be

$$
t_{p,k-1} + g_{p,k-1} \leq R_{jp} \leq t_{pk} + g_{pk} \quad \forall j, p, k
$$

which means request $(j,p)$ is served in cycle $k$. When $\theta_{jp} = 0$, constraints (4.13) and (4.14) are relaxed meaning request $(j,p)$ is not served in cycle $k$.

Constraint (4.17) is used to evaluate the delay when the request $(j,p)$ is served in cycle $k$. Combining the nonnegativity constraint (4.18) and minimal objective function (4.1), priority delay can be assessed by the equivalent equation (4.20) as below when $\theta_{jp} = 1$:

$$
d_{jp} = \max \left\{ 0, t_{pk} - R_{jp} \right\}
$$

Although MILP address the same problems as MINP, the solution times in the new MILP formulation are decreased by a factor of 5 to 10 times even for small problems.

4.4 Robust mixed integer linear program (MILP) formulation

Traffic dynamics include significant randomness due to the driver behavior based flow processes. Due to the uncertainty of the traffic state (e.g. queue length), the actual arrival times of priority requesting vehicles may drift away from the predicted or most desired time. The optimal solution from deterministic MILP could actually be sub-optimal or even result in worse performance due to this uncertainty. Therefore, it is necessary to
develop a formulation which can accommodate the stochastic factors and produce a signal plan that is robust in providing good performance when uncertainty is a factor.

When considering that priority request times $R_{jp}$ are random variables, it is very difficult for a decision maker to explicitly model the underlying probability distribution that would be required for a traditional stochastic optimization approach. In the case of priority control with deterministic requests, the request is usually planned to be served at the end of the requested phase to minimize the delay of other conflicting requests. The worst case occurs when the vehicles cannot be served in the planned phase and the phase is unnecessarily extended.

A robust optimization approach, which performs well across all scenarios and hedges against the worst of possible scenarios, is another way to deal with randomness (Kouvelis & Yu 1997). Inspired by the merits of robust optimization, randomness with robustness can be addressed in three ways:

1. Replace point arrival time predictions with request arrival time intervals to increase the chance that the vehicle can be served when the signal plan provides the service opportunity.

2. The objective function is modified to minimize the total possible maximal delay for each request interval. This objective is considered to be robust in the determination of the traffic signal controls.

3. Queued vehicles are treated as aggregated requests and are addressed in the green time constraints to ensure that a queue can be cleared during the service phase to allow the requesting vehicle to successful be served.
In the robust MILP formulation, it is assumed that $R_{jp}$ varies in an interval $R_{jp} \in [\underline{R}_{jp}, \overline{R}_{jp}]$ with unknown probability distribution, defined by a upper bound value $\overline{R}_{jp}$ and a lower bound value $\underline{R}_{jp}$. With interval arrivals, cyclic serving bars (CSB) on phase-time diagram become cyclic serving rectangles (CSR), shown in Figure 4.4(a). There are four possible cases of signal plan to cross CST in the first cycle, shown in Figure 4.4(b):

- **Case 1**: $t_{pk} + g_{pk} < \underline{R}_{jp}$. The request cannot be served.
- **Case 2**: $\overline{R}_{jp} \leq t_{pk} + g_{pk} < \underline{R}_{jp}$. There is some likelihood that the request will not be served.
- **Case 3**: $\overline{R}_{jp} \leq t_{pk} + g_{pk}$ and $t_{pk} \leq \underline{R}_{jp}$. The request can be served without any delay.
- **Case 4**: $\overline{R}_{jp} \leq t_{pk} + g_{pk}$ and $\underline{R}_{jp} \leq t_{pk}$. The request can be served, but it is likely there will be some delay. The maximal possible delay equals $\max\{0, t_{pk} - \underline{R}_{jp}\}$. 
According to the analysis of the different cases, Case 1 and Case 2 have the worst outcome - there is a good chance that the request may not be served. Case 3 and Case 4 are acceptable since the request will certainly be served. The sufficient but not necessary condition to ensure the request can be served in phase $p$ and cycle $k$ is:

\[
(4.21)
\]

Therefore, Constraints (4.13) and (4.14) are rewritten in the robust MILP as:

\[
(4.22)
\]

\[
(4.23)
\]

The objective function of robust MILP formulation is modified to minimize the maximum possible delay, $d'_{jp,k}$, according to the interval request $R_{jp} \in [R_{jp}, \overline{R}_{jp}]$, shown as equation (4.24).
Minimize $\sum_{p,k,j} w_{p,k} d'_{p,k}$ \hspace{1cm} (4.24)

By analysis in Case 4, the maximum possible delay is always achieved by the difference between starting time of phase $p$, $t_{p,k}$ and the earliest arrival time, $R_{j,p}$. The constraint (4.17) could be rewritten as

$$d'_{j,p,k} \geq t_{p,k} - R_{j,p} - (1 - \theta_{j,p,k})M \hspace{0.5cm} \forall j, p, k \hspace{1cm} (4.25)$$

Congested traffic conditions or an unexpected arrival of many buses requesting the same phase could result in some of the buses not being served as desired. Explicit consideration of the queue length in front of the buses can be utilized to calculate the green time required to clear the queue. Therefore, the necessary green time, $Q_{j,p}$, to clear the queue for request $R_{j,p}$, can be added into the robust formulation to accommodate congestion. If $Q_{j,p} > g_{j,p,k}^{\text{max}}$, define $Q_{j,p} = g_{j,p,k}^{\text{max}}$ as the best possible solution given the maximum green time available. A new constraint is added to the formulation as

$$g_{j,p,k} \geq Q_{j,p} - (1 - \theta_{j,p,k})M \hspace{0.5cm} \forall j, p, k \hspace{1cm} (4.26)$$

4.5 Integration of robust MILP formulation and actuated control

4.5.1 Green extension representation in MILP

The state-of-the-practice in traffic control today for a single intersection is actuated traffic signal. Actuated control can be programmed to adapt to vehicle demand by serving phases when there are vehicles present, changing phases lengths as vehicles arrive, forcing-off a phase, and many other operations with the purpose of shifting capacity
where and when it is needed. For example, the green time of a phase is extended by
detector calls as vehicle approach an intersection.

Actuated signal control is complicated due to the fact that cycle times and splits are
determined based on actual real-time vehicle demand, which is random by nature.
Therefore, it is impossible to derive an exact signal plan for actuated control in advance.
However, a flexible signal plan including flexible phase duration times could be
generated combining consideration for actuation events with priority constraints. A
flexible phase duration time $v'_{pk}$ is defined in three components: necessary green time
$g'_{pk}$, which denotes the larger one of minimal green time and pre-allocated green time,
additional maximal possible green extension time $a_{pk}$ and clearance time $y_p + r_p$,
illustrated in Figure 4.5. The necessary green duration is enforced, while green extension
is optional. For example, phase $p$ must start timing no later than time $t'_{pk}$ and stop timing
no earlier than time $(t'_{pk} + g'_{pk})$. And the duration of phase $p$ could be extended as long as
$(g'_{pk} + a_{pk})$. There are two stages to implement the flexible phase. In the first stage, the
controller holds phase $p$ green for $g'_{pk}$. After $g'_{pk}$ times out, the controller implements
actuated control logic with real-time demandactuating the phase for each detected vehicle
by adding the programmed extension time up to the maximal green time when the phase is forced-off immediately. Given this new flexible behavior, \( v_{pk} \) and \( t_{pk} \) in precedence constraints (4.2)-(4.10) are replaced by \( v'_{pk} \) and \( t'_{pk} \), which denotes the latest starting time of phase \( p \) at cycle \( k \). In other words, phase \( p \) in cycle \( k \) could start timing before \( t'_{pk} \) but not start after \( t'_{pk} \).

The phase interval constraints (4.11) and (4.12) are modified as below:

\[
v'_{pk} = g'_{pk} + a_{pk} + y_{p} + r_{p}, \quad \forall p, k
\]  
(4.27)

\[
g_{pk}^{\text{min}} \leq g'_{pk}, \quad \forall p, k
\]  
(4.28)

\[
g'_{pk} + a_{pk} \leq g_{pk}^{\text{max}}, \quad \forall p, k
\]  
(4.29)

The objective function is modified to not only to minimize the priority delay, but also maximize the total maximal possible green extension time to each phase for actuated control without adding any additional priority delay.

\[
\text{Minimize } \alpha \sum_{j,p,k} w_{jp} d'_{jp} - \beta \sum_{p,k} a_{pk}
\]  
(4.30)

Where \( \alpha \) and \( \beta \) are the coefficients to balance the weight of two summation items. In order to assign more weighs on total priority delay, \( \beta \) is set to a much smaller value than \( \alpha \). As for other constraints in robust MILP with actuated control, \( g_{pk} \) is replaced with \( g'_{pk} \) in constraints (4.22), (4.23) and (4.26), and \( t_{pk} \) is replaced with \( t'_{pk} \) in constraints (4.25) as follows.
\[ \overline{R}_{jp} \leq t'_{pk} + g'_{pk} + (1 - \theta_{jpk})M \quad \forall j, p, k \]  
(4.31)

\[ \overline{R}_{jp} \geq t'_{p,k-1} + g'_{p,k-1} - (1 - \theta_{jpk})M \quad \forall j, p, k \]  
(4.32)

\[ d'_{jp} \geq t'_{pk} - \overline{R}_{ip} - (1 - \theta_{jpk})M \quad \forall j, p, k \]  
(4.33)

\[ g'_{pk} \geq Q_{jp} - (1 - \theta_{jpk})M \quad \forall j, p, k \]  
(4.34)

The optimal solutions of the previous robust MILP formulation is not unique in most cases, since there are many timing options to achieve minimal delay given limited fixed priority requests. So the overall goal of this enhanced robust MILP formulation with actuated control is to select a timing plan among those minima-priority-delay plans to maximize the flexibility to reduce the delay for passenger vehicles in real-time actuation demand.

Integration of the robust MILP formulation and actuated control will result in the phase timing gapping out if no vehicles are detected, extending the phase if a priority request can be served, or forcing-off the phase when the maximal green extension is reached. Therefore, the efficiency of green time usage is improved for passenger cars.

4.5.2 Green extension group (GEG) and \( a_{pk} \) reassignment

When the MILP formulation is solved, the real-time traffic demand for each phase is unknown. Without real-time actuation (detection) information, the green extension \( a_{pk} \) cannot be accurately assigned in the green extension formulation presented in Section
4.5.1. In order to understand how to assign the phase green extension $a_{pk}$ without any additional priority delay, the concept of a green extension group (GEG) is developed. Green extensions $a_{pk}$ from the MILP optimal solutions are clustered into GEGs along the phases in a controller ring. Within each GEG, $a_{pk}$ contributes to a total amount of group green extension time with each other. When the signal timing plan is implemented, $a_{pk}$ is reassigned within the GEG according to real-time actuations using the actuated signal control logic.

GEG is accounted for on each controller since each ring operates independently, except for the barrier crossing that must be coordinated between rings. The barrier is not considered in the algorithm presented here since the actuated controller logic enforces the barrier constraint. The generation of GEG takes two steps as defined in the algorithm in Figure 4.6. First, given a defined set of fixed priority request intervals, the robust MILP is solved to determine the best $\theta_{j pk}$, which assigns each priority request to a phase and a cycle. Once $\theta_{j pk}$ is determined, $\sum_j \theta_{j pk}$ indicates the number of requests served by phase $p$ during cycle $k$. Second, phases sequenced along a ring and across a few cycles can be separated by positive $\sum_j \theta_{j pk}$. Each separation is treated as one GEG. The $i$th GEG set is defined as $s_{GEG[i]}$ contains all the phases $(p, k)$, which shares the same amount of green extension time, which is denoted $t_{GEG[i]}$. The total green extension time $t_{GEG[i]}$ in $i$th GEG is determined by:

$$t_{GEG[i]} = \sum_{p,k} a_{pk} \quad \forall (p,k) \in s_{GEG[i]} \quad (35)$$
The green extension implementation in the robust MILP utilizes the same control logic as actuated signal control. There are two differences: 1) the minimal green time in actuated control is substituted by necessary green time $g'_{pk}$ as determined by the solution to the MILP problem; 2). The green extension time is not only constrained by the the maximal green (or split) but also by the remaining unassigned green extension time in the current GEG.

**Procedure GEG_in_Ring(r)**
1. Obtain $\theta_{jp_k}$ and $a_{pk}$ by solving the robust MILP formulation.
2. Calculate $w_{pk} = \sum_j \theta_{jp_k}$ for phase $p = 1..|P|$ and cycle $k = 1..|K|$.
3. Let $k = 1, p = \text{first_phase_of_ring}(r), i = 1, s\_GEG[1] = \{\}$ and the total green extension time $t\_GEG[1] = 0$, 
4. While $k < |K|$ do
5. While $p < \text{num_phases_in_ring}(r)$ do
6. If $w_{pk} \leq 0$ then
7. $s\_GEG[i] = s\_GEG[i] \cup \{(p,k)\}$
8. $t\_GEG[i] = t\_GEG[i] + a_{pk}$
9. Else
10. $i = i + 1$
11. $s\_GEG[i] = s\_GEG[i] \cup \{(p,k)\}$
12. $t\_GEG[i] = t\_GEG[i] + a_{pk}$
13. End if
14. $p = \text{next_phase_in_ring}(r)$
15. End while
16. $k = k + 1$ //next cycle
17. End while
18. End procedure

Figure 4.6 The procedure for generating GEG in a ring
Figure 4.7 (a) An example of green extension group and reassignment; (b) Green extension reassignment in GEG2
A detailed example is illustrated in Figure 4.7(a). Phases sequenced along 1st ring within 2 cycles can be represented as a sequence of phases denoted $\phi_1, \phi_2, \phi_3, \phi_4, \phi_1, \phi_2, \phi_3, \phi_4$. Given three priority request intervals, $R_{13}$, $R_{12}$ and $R_{22}$, the optimal signal plan and cycle service selection variables $\sum_j \theta_{jpk}$ are obtained from the robust MILP solution. The corresponding sequence of $\sum_j \theta_{jpk}$ is represented as 00100200, where the first priority request is served in phase 3 of cycle 1, and the other two priority requests are served in phase 2 of cycle 2. The sequence of phases is separated into three GEGs by the positive $\sum_j \theta_{jpk}$ values. The starting time of each GEG[i] is determined as well as the duration of the first phase in GEG[i] (except GEG[1]) to ensure that the priority request can be served during the allocated time interval, which is exactly same as requested $Q_{jp}$. However, the duration of each phase in the GEG is flexible and determined by the phase minimum time and the vehicle actuations. All phases in a GEG must share the total green extension time.

Implementation of GEG[2] is depicted in Figure 4.7(b) using a cycle style diagram. GEG[2] contains three phases in sequence: $\phi_3$ and $\phi_4$ in cycle 1, and $\phi_1$ in cycle 2. First, $\phi_3$ must start timing at 45 seconds and is held for 15s. Then $\phi_3$ can be extended by real-time vehicle actuations until either it gaps out or it reaches the maximum time allowed by the maximum green or the total green time extension available $t_{\text{GEG}[2]}$.

After $\phi_3$ terminates, $\phi_4$ starts timing. The process is similar, except the necessary green time, $g_{41}$, is not constrained by the duration of the priority request interval. $\phi_4$ times up to either its maximum green time or until the remaining green extension time is
used, $t_{GEG[2]} = 0$. Note that this method of green time allocation is identical to the floating force-off modes in actuated-coordinated traffic control. Similarly, this could be executed using fixed force-offs as well. The details of coordination force-off modes can be found in the Traffic Signal Timing Manual (Federal Highway Administration 2008). Floating force-offs flavor to assign slack, or remaining green extension time, to coordinated phases or priority requested phases in this research, where fixed force-offs allow more unused green extension time to be used by non-coordinated or non-priority requested phases. They both have different trade-offs. Force-off modes are an important consideration in coordinated traffic control. Therefore, the defined robust MILP formulation can also be applied for coordination with priority requests. The final goal of this Chapter is to show how the MILP formulation can address the multiple priority request issue on coordinated traffic control utilizing the mathematic optimization approach as well as actuated control implementation which systematically plans for priority requests, while accommodating uncertain vehicle flow as measured through vehicle detection and phase actuation.

4.6 Robust signal coordination with priority

The state-of-the-practice in traffic control today is coordinated-actuated traffic signal control. Coordination aims to provide smooth progression of vehicle platoons through the determination of traffic plans that contain appropriate offsets, splits, and cycle times at each intersection. The benefits that can be obtained from coordination drive the need to
develop an analytical framework that simultaneously considers signal coordination and priority.

Traditionally, signal coordination with priority has been achieved using either passive priority or active priority. In passive priority control, the signals along an arterial are coordinated in a way to create progression for buses. Thus buses would be able to travel through intersections with minimal stops and delay (Estrada et al. 2009) (Furth et al. 2010). However, passive priority may result in increased delay for passenger vehicles due to speed differences between buses and passenger vehicles. In active priority control, the state-of-practice is signal plan transition from coordination plan to transit signal priority (TSP) plan when a delayed bus checks in. The transition methods usually include green extension, red truncation and phase omits. After the bus checks out, the plan is transited back from TSP plan to coordination by add, subtract, short way, dwell, max dwell and so on (S.G. Shelby et al. 2006). This method of dealing active priority performs well under low priority frequency. However, it cannot handle conflicting priority requests or high frequency priority request, since the logic is limited to a serving a single request at a time and typically limited to only one request every two cycles. There is no known literature that considers the combined coordination and priority problem in single mathematical formulation within an actuated control strategy.

There are three major factors considered in a coordinated signal timing plan: offset, phase sequence and splits, and a common cycle length. These three factors can be easily represented as priority request intervals. Suppose the coordinated phase is \( p_c \) in one ring. Requests \( R_{p,k}^c \in \left[ R_{p,k}^c, R_{p,k}^c \right] \) are called the coordination request to distinguish them from
priority requests. The offsets from the master clock corresponds to the lower bound \( R_{p,k}^c \).

The phase split of the coordinated phase \( p_c \) can be considered as \( Q_{p,k}^c \), which ensures the desired green time for the coordinated phase as

\[
g'_{p,k} \geq Q_{p,k}^c \quad \forall p_c, k
\] (4.36)

Hence, the upper bound \( \bar{R}_{p,k}^c \) equals \( R_{p,k}^c + Q_{p,k}^c \). The background common cycle length can be ensured by scheduling the lower bound \( \underline{R}_{p,k}^c \) for each coordinated phase \( p_c \) in each cycle \( k \), such that

\[
\bar{R}_{p,k}^c = \underline{R}_{p,k}^c + C \quad \forall p_c, k
\] (4.37)

Within the MILP formulation, the introduction of coordination requests does not increase the number of binary decision variables since coordination request \( R_{p,k}^c \) should be always served in phase \( p_c \) of cycle \( k \). Hence, the phase selection constraints are not needed for coordination requests. Therefore, constraints (4.31)-(4.33) can be rewritten as follows.

\[
\bar{R}_{p,k}^c \leq t'_{p,k} + g'_{p,k} \quad \forall p_c, k
\] (4.38)

\[
\underline{R}_{p,k}^c \geq t'_{p, k-1} + g'_{p, k-1} \quad \forall p_c, k
\] (4.39)

\[
d_{p,k}^c \geq t'_{p,k} - \bar{R}_{p,k}^c \quad \forall p_c, k
\] (4.40)
The objective of the robust MILP formulation with signal coordination is not only to minimize the priority delay and maximize the total green extension, but also to minimize the coordination delay. A weight $w_{p,k}$ is introduced for each coordination request delay $d^c_{p,k}$. The objective in equation (3.30) is modified as follows.

$$\text{Minimize } \alpha \sum_{j,p,k} w_{jp} d'_{jp k} + \gamma \sum_{p,k} w_{p,k} d^c_{p,k} - \beta \sum_{p,k} a_{pk} \quad (4.41)$$

Where $\gamma$ is the coefficient for adjusting the total coordination request delay.

Therefore, fixed-cycle, fixed-offset coordination problems are modeled as priority control problems that can simultaneously address coordination and priority control in the same formulation.

An example of priority control with signal coordination is illustrated in Figure 4.8. Consider ring 1 over two cycles, where there is a single priority request at $t=R_{14}$ for phase $\phi 4$ and two coordination requests $R^c_{21}$ and $R^c_{22}$ for coordinated $\phi 2$. Suppose the coordination parameters are provided by some signal optimization software, such as TRANSYT (Wallace et al. 1998), SYNCHRO (Trafficware 2009) or PASSER (Chaudhary & Chu 2003). In this example, the common cycle length is 80 seconds, and the offset is 10 seconds. The distance between interval $R^c_{21}$ and $R^c_{22}$ is one cycle length. The interval length of $R^c_{21}$ and $R^c_{22}$ equals to the designed $\phi 2$ split minus the clearance time.

The optimal solution is given by solving the robust MILP formulation with the coordination constraints. The formulation can produce the best plan over the next few cycles in real-time without the need for traditional plan transition. To implement the
optimal signal plan, the concept of GEG is executed exactly as described in Section 4.5.2. Initially, coordinated phase 2 ($\phi_2$) in cycle 1 and cycle 2 create GEG[1] and GEG[2]. Next, priority request $R_{14}$ served in phase 4 ($\phi_4$) of cycle 2 generates GEG[3]. If there were no priority request in the formulation, the GEG would equal the total available time across all the phases in each cycle. The specific implementation of each GEG depends on different force-off modes as discussed in Section 4.5.2.

It should be noted that a different trade-off value of the signal plan can be generated by adjusting the weights in equation (4.41). The general idea is that the priority weight could be higher than the coordination weight in low volume traffic, and the opposite in high volume traffic.

![Figure 4.8 Priority requests with coordination](image)

4.7 Numerical experiments
The numerical experiments were conducted using VISSIM, a microscopic simulation tool. To better simulate the real traffic signal controller logic, the ASC/3 SIL (software in the loop) controller was installed with VISSIM. The ASC/3 SIL feature allows a VISSIM user to utilize the same logic as a physical ASC/3 controller during the simulation. This includes the transit signal priority (TSP) provided as an advanced feature of the controller firmware (Econolite 2009).

The entire evaluation platform contains VISSIM with COM (Component Object Model), ASC virtual controller as the simulation environment and GAMS as optimization solver, depicted in Figure 4.9. Simulation in VISSIM can be easily controlled in COM, which can be created with a variety of programming languages, including C++. First, COM runs a VISSIM model and continuously reads vehicle data as VISSIM simulates the movement of vehicles. When there is no priority request the actuated control logic is performed by the ASC/3 SIL. When a vehicle that is designated to generate a priority
request is detected, a MILP program is formulated by the COM component and sent to GAMS for solution. After retrieving optimal signal plan from GAMS, the COM component implements the signal timing schedule by sending phase control commands (hold and force-off) to the ASC/3 SIL.

The number of integer variables in proposed MILP formulations is $N \times K$, where $N$ is total number of requests and $K$ is the number of optimized cycles. $N$ depends on the maximal number of co-existing buses at the intersection, which is generally not a very large number. Given small $N$ and $K$, proposed MILP formulation can be solved less than 0.1 seconds, which is suitable for real-time TSP control.

![Figure 4.10 Layouts of a two-intersection arterial](image-url)

Figure 4.10 Layouts of a two-intersection arterial
Numerical experiments on a simple two-intersection arterial model that was based on a short section of Speedway Blvd. in Tucson, AZ, bounded by from Campbell Ave. to Cherry Ave. Four conflicting bus routes were added in the model shown as Figure 4.10. There are six bus stops on the bus routes. All of bus stops are far-side stops. It is assumed that each bus is behind schedule. So every bus sends a priority request when it approaches intersection.

In this experiment, eight methods were tested and compared with two bus frequency and three different traffic volume scenarios as shown in Table 4.2. Four of the control methods are non-coordinated control methods and the other four are coordinated control methods, as summarized in Table 4.3.

### Table 4.2 Traffic volume for speedway

<table>
<thead>
<tr>
<th>Volume (veh./h)</th>
<th>Cross street</th>
<th>Eastbound</th>
<th>Westbound</th>
<th>Northbound</th>
<th>Southbound</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L T R L T R L T R L T R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Cherry</td>
<td>9 462 28</td>
<td>18 417 14</td>
<td>35 19 42</td>
<td>57 18 20</td>
</tr>
<tr>
<td></td>
<td>Campbell</td>
<td>50 397 116</td>
<td>87 335 39</td>
<td>91 216 36</td>
<td>54 243 24</td>
</tr>
<tr>
<td>2</td>
<td>Cherry</td>
<td>22 1110 68</td>
<td>44 1002 32</td>
<td>93 46 102</td>
<td>138 44 48</td>
</tr>
<tr>
<td></td>
<td>Campbell</td>
<td>119 952 278</td>
<td>130 583 58</td>
<td>218 519 86</td>
<td>54 243 24</td>
</tr>
<tr>
<td>3</td>
<td>Cherry</td>
<td>37 1849 114</td>
<td>74 1670 54</td>
<td>156 76 170</td>
<td>230 73 81</td>
</tr>
<tr>
<td></td>
<td>Campbell</td>
<td>199 1586 463</td>
<td>349 1339 155</td>
<td>363 864 144</td>
<td>216 972 96</td>
</tr>
</tbody>
</table>

### Table 4.3 Description of compared different methods

<table>
<thead>
<tr>
<th>Modes</th>
<th>Methods</th>
<th>Actuated</th>
<th>Bus priority</th>
<th>Signal coordination</th>
<th>Solving MILP formulation</th>
<th>Robust requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free</td>
<td>ASC free</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Determ. free</td>
<td>✓ ✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
In free control methods, “ASC free” utilizes the ASC/3 SIL actuated control at each isolated intersection without priority control. “Determ. free” is based on the solution to the deterministic formulation proposed in Section 4.3 and implements its optimal plan with actuated control only for cases with no bus approaching. “ASC-TSP free” utilizes the ASC/3 SIL actuated control with TSP logic triggered by bus check-in and check-out detectors. The basic strategy of TSP in ASC/3 SIL is early green (red truncation) and green extension. Transit Priority operation is on a first-come first-serve basis. Only one transit vehicle can modify timing during any given cycle. The detailed settings of TSP in ASC/3 SIL can be found in Zlatkovic and the Econolite Controller Programming Manual (Zlatkovic et al. 2010) to (Econolite 2009). “Robust free” is based on the solution to the robust MILP formulation without coordination requests as defined in Section 4.4.

### Table 4.4 Basic optimal coordination timing plan from SYNCHRO with 90s common cycle

<table>
<thead>
<tr>
<th>Volume</th>
<th>Cross street</th>
<th>Offset (s)</th>
<th>Split (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\phi_1$</td>
</tr>
<tr>
<td>1</td>
<td>Cherry</td>
<td>12</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Campbell</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>Cherry</td>
<td>80</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Campbell</td>
<td>0</td>
<td>16</td>
</tr>
</tbody>
</table>
In the coordinated control methods, the basic coordination signal plan is based on optimal signal timing parameters obtained from SYNCHRO 7.0, shown as Table 4.4. The only difference between the coordinated control methods is how they realize the basic coordination plan and whether priority requests are considered. “ASC coord.” and “Robust coord.” consider coordination without priority control. “ASC coord.” implements the signal plan in Table 4.4 using the ASC/3 SIL coordinator. The “Robust coord.” method solves the MILP by adding coordination requests in rolling horizon fashion.

“ASC-TSP coord.” and “Robust priority coord.” address the combined coordination and priority problems in different ways. “ASC-TSP coord.” performs similarly to the “ASC-TSP free”, except for coordination, while “Robust priority coord.” solves the MILP by considering the current active coordination requests and priority requests simultaneously.

The experiments involve two bus frequencies (determined by headway) under three different traffic demand volumes. A total of six scenarios are compared using the eight different control methods. In each scenario, the simulation is replicated using ten different random seeds in VISSIM.

The measured average delay under each scenario is presented in Table 4.5. Six different measurements are provided: 1). “All” is the average delay for all vehicles in the

*: -- means data not applicable
network; 2). “Main” is the average delay for vehicles traveling on the arterial; 3). “Mbus” is the average bus delay on arterial; 4). “Side” is the average vehicle delay on side streets; 5). “Sbus” is the average bus delay on the side streets; 6). “Left-turn” is the average all left-turn vehicle delay.

Several observations from the experimental results in Table 4.5 can be made:

1. Coordinated methods have much lower overall vehicle delay as well as bus delay than free methods under medium and high volume. However, the free methods outperform coordinated methods under low volume. This is expected since the study network has two intersections that are relatively closely spaced so that coordination would be expected to improve overall performance.

2. The deterministic MILP performs fairly well under low volume, but the delay increases significantly in medium and high volume.

3. The robust methods have much less bus delay as well as less overall vehicle delay compared with state-of-practice ASC-TSP methods, shown in Figure 4.11. In addition, the robust methods provide relatively stable performance for buses from side street and left-turn vehicles, illustrated in Figure 4.12.
Table 4.5 Measured average delay under each scenario with eight different methods

<table>
<thead>
<tr>
<th>Bus frequency</th>
<th>Volume</th>
<th>Measurement</th>
<th>Free-actuated</th>
<th>Coordinated-actuated</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>ASC free</td>
<td>Determin. free</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ASC-TSP free</td>
<td>ASC-TSP coord.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Robust free</td>
<td>Robust coord.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ASC coord.</td>
<td>ASC-TSP coord.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Robust priority coord</td>
<td></td>
</tr>
<tr>
<td>5 min/bus</td>
<td>1 (low volume)</td>
<td>All</td>
<td>13.08</td>
<td>13.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main</td>
<td>11.21</td>
<td>11.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBus</td>
<td>11.72</td>
<td>3.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Side</td>
<td>18.84</td>
<td>19.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sbus</td>
<td>19.38</td>
<td>10.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left-turn</td>
<td>21.72</td>
<td>20.13</td>
</tr>
<tr>
<td></td>
<td>2 (medium volume)</td>
<td>All</td>
<td>23.81</td>
<td>21.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main</td>
<td>20.09</td>
<td>20.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBus</td>
<td>21.59</td>
<td>18.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Side</td>
<td>35.48</td>
<td>26.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sbus</td>
<td>40.04</td>
<td>18.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left-turn</td>
<td>39.31</td>
<td>29.45</td>
</tr>
<tr>
<td></td>
<td>3 (high volume)</td>
<td>All</td>
<td>42.42</td>
<td>81.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main</td>
<td>38.59</td>
<td>89.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBus</td>
<td>43.80</td>
<td>88.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Side</td>
<td>54.66</td>
<td>58.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sbus</td>
<td>45.28</td>
<td>64.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left-turn</td>
<td>147.95</td>
<td>140.2</td>
</tr>
<tr>
<td>10 min/bus</td>
<td>1 (low volume)</td>
<td>All</td>
<td>12.76</td>
<td>12.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main</td>
<td>11.14</td>
<td>10.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBus</td>
<td>11.24</td>
<td>4.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Side</td>
<td>17.92</td>
<td>19.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sbus</td>
<td>18.85</td>
<td>17.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Left-turn</td>
<td>20.85</td>
<td>21.56</td>
</tr>
<tr>
<td></td>
<td>2 (medium volume)</td>
<td>All</td>
<td>23.43</td>
<td>21.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Main</td>
<td>19.34</td>
<td>18.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MBus</td>
<td>20.17</td>
<td>13.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Side</td>
<td>36.41</td>
<td>31.92</td>
</tr>
</tbody>
</table>

Table dimensions: 612.0x792.0
<table>
<thead>
<tr>
<th></th>
<th>Sbus</th>
<th>25.24</th>
<th>16.39</th>
<th>10.11</th>
<th>21.87</th>
<th>22.01</th>
<th>14.83</th>
<th>23.67</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left-turn</td>
<td>36.95</td>
<td>32.70</td>
<td>41.95</td>
<td>36.53</td>
<td>38.66</td>
<td>38.46</td>
<td>69.01</td>
<td>42.23</td>
</tr>
<tr>
<td>All</td>
<td>39.85</td>
<td>69.85</td>
<td>36.57</td>
<td>46.17</td>
<td>21.93</td>
<td>21.85</td>
<td>26.68</td>
<td>24.11</td>
</tr>
<tr>
<td>Main</td>
<td>35.02</td>
<td>69.82</td>
<td>29.47</td>
<td>39.23</td>
<td>18.39</td>
<td>18.17</td>
<td>21.76</td>
<td>18.15</td>
</tr>
<tr>
<td>Mbus</td>
<td>38.24</td>
<td>67.00</td>
<td>19.37</td>
<td>20.36</td>
<td>17.63</td>
<td>17.83</td>
<td>16.52</td>
<td>11.48</td>
</tr>
<tr>
<td>Side</td>
<td>55.26</td>
<td>70.95</td>
<td>58.81</td>
<td>68.76</td>
<td>33.37</td>
<td>33.69</td>
<td>42.15</td>
<td>43.36</td>
</tr>
<tr>
<td>Sbus</td>
<td>48.60</td>
<td>61.50</td>
<td>45.51</td>
<td>19.37</td>
<td>22.64</td>
<td>24.74</td>
<td>28.25</td>
<td>31.96</td>
</tr>
<tr>
<td>Left-turn</td>
<td>144.1</td>
<td>113.4</td>
<td>151.4</td>
<td>115.8</td>
<td>158.9</td>
<td>145.6</td>
<td>238.8</td>
<td>164.4</td>
</tr>
</tbody>
</table>
Figure 4.11 Overall vehicle delay and bus delay under “ASC-TSP coord.” and “Robust priority coord.”

Figure 4.12 Side bus delay and left-turn delay under “ASC-TSP coord.” and “Robust priority coord.”
4. Pure robust coordination results in higher bus delay compared with robust coordination with priority control. However, when the traffic demand increases, pure robust coordination performs better than robust coordination with priority as shown in Figure 4.13. That means priority control results in larger delay either for passenger vehicles and buses by frequently disrupting the coordination when the traffic demand is high.

Closer examinations of the results based on observation 3 are shown in Table 4.6. In the high volume cases (Volume 3), the robust MILP outperforms ASC-TSP under all bus frequency scenarios and measurements. Considering overall vehicle delay, the ASC-TSP has better results than the robust MILP under low and medium volume. However, ASC-TSP has very large average bus delay under almost all cases, except for low bus frequency in medium volume. It is also observed that the ASC-TSP increases average left-turn delay dramatically under every scenario. Considering average delay reduction,
robust MILP decreases overall vehicle delay, bus delay and left-turn delay by 3%, 18% and 37%, respectively.

Table 4.6 Delay changes in both percentage (%) and seconds (s) from ASC-TSP to robust priority

<table>
<thead>
<tr>
<th></th>
<th>High bus freq.</th>
<th>low bus freq.</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Volume 1</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>13.13%</td>
<td>7.05%</td>
<td>10.09%</td>
</tr>
<tr>
<td>Bus</td>
<td>-41.26%</td>
<td>-3.43%</td>
<td>-22.35%</td>
</tr>
<tr>
<td>Left-turn</td>
<td>-50.73%</td>
<td>-24.83%</td>
<td>-37.78%</td>
</tr>
<tr>
<td><strong>Volume 2</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>0.27%</td>
<td>4.70%</td>
<td>2.49%</td>
</tr>
<tr>
<td>Bus</td>
<td>-54.71%</td>
<td>49.90%</td>
<td>-2.41%</td>
</tr>
<tr>
<td>Left-turn</td>
<td>-48.02%</td>
<td>-38.80%</td>
<td>-43.41%</td>
</tr>
<tr>
<td><strong>Volume 3</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>-37.6%</td>
<td>-9.65%</td>
<td>-23.63%</td>
</tr>
<tr>
<td>Bus</td>
<td>-50.86%</td>
<td>-10.39%</td>
<td>-30.63%</td>
</tr>
<tr>
<td>Left-turn</td>
<td>-33.75%</td>
<td>-31.17%</td>
<td>-32.46%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>-8.07%</td>
<td>0.70%</td>
<td>-3.68%</td>
</tr>
<tr>
<td>Bus</td>
<td>-48.94%</td>
<td>12.03%</td>
<td>-18.46%</td>
</tr>
<tr>
<td>Left-turn</td>
<td>-44.17%</td>
<td>-31.60%</td>
<td>-37.89%</td>
</tr>
</tbody>
</table>

4.8 Summary

The multiple priority control problem is examined in this Chapter under the condition that the vehicle-to-infrastructure communication is available for each emergency vehicle or transit vehicle. Given the current multiple priority request information, a mixed integer linear program (MILP) is solved to obtain the optimal signal plan. First, a deterministic
MILP formulation is proposed only for multiple priority control of emergency vehicles due to their deterministic point time of arrival. Next, a robust MILP formulation is developed for transit vehicles (e.g. buses) to address the randomness of time arrivals by introducing the concept of a request interval and necessary green time. Third, actuated control is integrated into the robust MILP formulation to mitigate the delay for passenger cars based on real-time demand using vehicle detectors. Finally, signal coordination is combined with priority control in the formulation by adding coordination requests. The proposed approach is compared with state-of-practice coordinated-actuated traffic signal control with Transit signal priority (TSP) over several scenarios. The numerical experiments show that the robust MILP approach is able to reduce priority delay as well as all vehicle delay and achieve real-time robust control.

In this Chapter, it is assumed that the market penetration of IntelliDriveSM, or other vehicle to infrastructure communications, is significant for emergency vehicles and transit vehicles. Our ongoing research includes addressing multi-modal traffic signal control with penetration of IntelliDriveSM for different priority classes.
CHAPTER 5: A HEURISTIC ALGORITHM TO IMPLEMENT MULTIPLE PRIORITY CONTROL FOR EMBEDDED CONTROL AT A SINGLE INTERSECTION

5.1 Introduction

Multiple priority control aims to solve the problem that multiple vehicles, such as emergency vehicles or transit buses, may approach an intersection and request priority at the same time. This research is motivated and supported by the Arizona E-IntelliDrive\textsuperscript{SM} (called E-VII before 2009) project conducted by a partnership among the Arizona DOT, Maricopa County DOT, FHWA, University of Arizona, Arizona State University and the private sector. The goal of the Arizona E-IntelliDrive\textsuperscript{SM} initiative is to develop and test advanced technologies and integrate roadway systems with emergency responder vehicles to improve response efficiency as well as enhance responder safety (ADOT 2008). The details of Arizona E-IntelliDrive\textsuperscript{SM} field implementation will be presented in section 5.3.

Arizona E-IntelliDrive\textsuperscript{SM} provided the hardware and technologies of v2x environment necessary to conduct a field implementation of multiple priority control. Given the v2x environment for emergency vehicles, the control strategy developed in Chapter 4 are able to be implemented at a real intersection. However, CPLEX is not compatible with small kernel embedded Linux systems, such as those used and installed on both RSE and OBE. It is necessary to develop a heuristics algorithm to find near-optimal signal plan for multiple priority requests without the use of solvers for mathematical programming problems.
In this Chapter, the multiple priority control problem is simplified to a polynomial solvable cut problem by adding few assumptions. Each cut set corresponds to a unique serving sequence of multiple priority requests. First, the cut problem is proved to be a polynomial solvable problem. Second, an exhaustive search algorithm is proposed to search for the good solutions within a specified tolerance range. The total priority delay of each cut set is assessed and visualized by a phase-time diagram as developed in Chapter 4. Finally, the performance of proposed heuristic algorithm is compared with the robust MILP and other methods in microscopic simulation experiments.

5.2 Simplification of multiple priority control problems

5.2.1 Simplification to a polynomial solvable problem

In the robust MILP developed in Chapter 4, the $\theta_{jkp}$ variables are the only integer decision. These integer decision variables denote the assignment of request $(j,p)$ to be solved in cycle $k$. So $\theta_{jkp}$ are the only “hard” variables in the formulation. It is necessary to find a heuristic method to seek near-optimal $\theta_{jkp}$ to solve the entire multiple priority control problem. The problem of seeking near-optimal $\theta_{jkp}$ can be simplified as an cut problem, since the realizations of $\theta_{jkp}$ essential assign or cut request $(j,p)$ to cycle $k$. In order to trim undesired solutions and enhance the feasible region, two reasonable assumptions are made as follows.

Assumption 1: The sequence of phases in a ring is fixed and phase skipping is not allowed.
Assumption 2: A First-come, first-serve rule holds for all requests for the same phase.

Assumption 1 is a reasonable assumption since phase rotation and skipping can cause confusion to the motorist, loss of coordination, and long delay to the traffic stream (Skabardonis 2000). It is understood that phase rotation, such as lead-lag and lag-lead, can produce useful behavior in some circumstances. But the solution space for signal plans is significantly reduced by the fixed phase sequence assumption. Assumption 2 follows a widely accepted rule - first-come, first-serve in queuing theory (Gross et al. 2008).

Note that the assumption 2 of first-come first-serve is only applicable for the requests for the same phase, meaning that the preceding priority request (j-1,p) should always be served before the succeeding priority request (j,p). Assumption 2 establishes a precedence relationship between priority vehicles that are adjacent and requesting the same phase. If the succeeding priority request (j,p) is assigned in cycle k, the preceding priority request (j-1,p) cannot be assigned to cycle k+1, k+2,…. Suppose the number of request is $|J_p|$ for phase p, and the number of considered cycles is $|K|$. Given assumption 1 and 2, the total number of possible cut combinations for priority request (j,p) within cycle $|K|$ can be calculated by Remark 4.1.

Remark 4.1 Given assumption 1 and 2, the size of feasible region to cut linked $|J_p|$ requests into $|K|$ cycles is equal to $\prod_{p=1}^{[p]} \frac{(|J_p| + |K| - 1)!}{(|K| - 1)! |J_p|!}$. 
Proof: Assumption 1 allows considering the cut problem separately for each phase \( p \). Therefore, the total number of possible cut combinations is equal to the product of the number of cut combinations for each phase \( p \). Considering the request list in each phase \( p \), the problem is to assign \( |J_p| \) requests into \( |K| \) cycles. According to Assumption 2, the cut problem is equivalent to dividing the \( |J_p| \) sequenced linked nodes into \( |K| \) sets. An empty set is allowable. So \( |K|-1 \) cuts can separate \( |J_p| \) linked nodes into \( |K| \) set, as shown as Figure 5.1. The total number of cut combinations is equal to 
\[
C(|J_p| + |K| - 1, |K| - 1)
\]
where \( C(n,r) \) denotes the number of \( r \)-combinations from a given set of \( n \) elements. Figure 5.2 illustrates the case when \( |K|=2 \). The cut can be located before the first request or after the last request. So the total number of cut combinations is \( |J_p|+1 \). Therefore, combining with assumption 1 and 2, the size of feasible region to assign \( |J_p| \) requests into \( |K| \) cycles is equal to 
\[
\prod_{p=1}^{P} C(|J_p| + |K| - 1, |K| - 1) = \prod_{p=1}^{P} \frac{(|J_p| + |K| - 1)!}{(|K| - 1)! |J_p|!}
\]

The number of cycles, \( |K| \), considered in Remark 4.1 affects the complexity of solving the cut problem. The problem can be reduced to a very simple cut problem if only two cycles are enough long to serve all the priority requests. Thus assumption 3 is added to further reduce the feasible region.
Figure 5.1 Illustration of cut $|J_p|$ requests into $|K|$ cycles

Phase $p$ linked request list: cut $c_p = j$

<table>
<thead>
<tr>
<th>$c_p$</th>
<th>$\phi(p-1)$</th>
<th>$\phi(p)$</th>
<th>$\phi(p+1)$</th>
<th>$\phi(p-1)$</th>
<th>$\phi(p)$</th>
<th>$\phi(p+1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1,2,...,$</td>
<td>J_p</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>1</td>
<td>...</td>
<td>2,...,$</td>
<td>J_p</td>
<td>$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>J_p</td>
<td>$</td>
<td>...</td>
<td>1,2,...,$</td>
<td>J_p</td>
<td>-1$</td>
</tr>
<tr>
<td>$</td>
<td>J_p</td>
<td>+1$</td>
<td>1,2,...,$</td>
<td>J_p</td>
<td>$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2 Priority request cut table for phase $p$ in two cycles
Assumption 3: All the requests can be served in 2 cycles.

Assumption 3 matches the communication range limit of DSRC. DSRC has a communication range of about 500~1000 meters (Y. Liu et al. 2005), which indicates that the latest arrival of priority vehicle is approximately 37.5~75 seconds from the current time, assuming the vehicle speed is 48km/h. Typically, the cycle length is greater than 50 seconds, hence, two cycles are long enough to serve all priority requests.

Given Assumption 1, 2 and 3, the size of feasible region is equal to \( \prod_{p=1}^{\lvert P \rvert} (\lvert J_p \rvert + 1) \).

Therefore, the NP-hard cut problem is reduced as a polynomial solvable cut problem, since the number of phases in a ring \( \lvert P \rvert \) is fixed. \( \prod_{p=1}^{\lvert P \rvert} (\lvert J_p \rvert + 1) \) can be written as \( O(\lvert J \rvert^{\lvert P \rvert}) \) in big O notation, where \( \lvert J \rvert \) denotes the maximal number of requests for one phase.

Remark 4.2 concludes the simplification as follows:

**Remark 4.2:** Given assumption 1, 2 and 3, the complexity of assigning \( \lvert J_p \rvert \) requests into 2 cycles is equal to \( O(\lvert J \rvert^{\lvert P \rvert}) \), which is polynomial.

5.2.2 Revised exhaustive search algorithm and delay evaluation

It is desired to find near-optimal solutions via a fast algorithm rather than enumerate all possible solutions in the feasible region; even though Section 5.2.1 shows that the size of the feasible region of the cut problem is polynomial.

Based on Assumption 1, 2 and 3, only one cut needs to be selected in each request list \( J_p \), which is the set with all request for phase \( p \), ordered by the lower bound of arrival
time $R_{jp}$. Whether to serve a request in the first cycle or the second cycle depends on the length of time (interval) between two adjunct requests. Define interval $\tau_{jp}$ as follows:

$$
\tau_{jp} = \begin{cases} 
\frac{R_{jp}}{R_{jp}} & \text{if } j = 1 \\
\frac{R_{jp}}{R_{j-1,p}} & \text{if } 2 \leq j \leq |J_p| 
\end{cases} 
$$

(5.1)

The larger the value of $\tau_{jp}$, the higher probability that the delay will be smaller if a cut is made between request $j$ and request $j-1$. Denote the cuts as $c_p = j \ (j > 0)$ if a cut is made before request $j$ in request list $J_p$, illustrated in Figure 5.2. Note that the number of possible cuts is equal to $|J_p|+1$. The best cut $c_p = j^*$ when $\tau_{j^*,p}$ is largest in all $\tau_{jp}$ and also larger than a defined threshold $\gamma$, which decides if cut in $|J_p|+1$. If every interval $\tau_{jp}$ is less than the threshold $\gamma$, $c_p = |J_p|+1$, meaning that the entire request list can be cut into cycle 1. After the cuts are made to each request list $J_p$, the entire cut combination corresponds to a request serving sequence, which can be evaluated easily using the phase-time diagram proposed in Chapter 4. The heuristic algorithm aims to find a near-optimal assignment with acceptable priority delay through enumerating different cuts from high $\tau_{jp}$ to low $\tau_{jp}$, including threshold $\gamma$. 

1. Calculate $\tau_{jp}$ by equation (5.1)

2. Define vector $V_p = (\tau_{1p}, \tau_{2p}, ..., \tau_{J_p}, \gamma)$ \hspace{1cm} $\forall p, |J_p| > 0$

3. Let $i_p = 1$ \hspace{1cm} $\forall p$, $c_p = 0$ \hspace{1cm} $\forall p$, minimal delay $d^* = \infty$, best cut group $C^* = \{0\}$

4. While $i_p \leq |J_p| + 1$

   If $|J_p| > 0$; $c_p = \text{find\_cut\_order}(i_p, V_p)$; Else $c_p = 0$.

   ......

   While $i_p \leq |J_p| + 1$

   If $|J_p| > 0$; $c_p = \text{find\_cut\_order}(i_p, V_p)$; Else $c_p = 0$.

   ......

   While $i_{|p|} \leq |J_{|p|}| + 1$

   If $|J_{|p|}| > 0$; $c_{|p|} = \text{find\_cut\_order}(i_{|p|}, V_{|p|})$; Else $c_{|p|} = 0$.

   Evaluate delay $d_k$ for selected cut group $C_k = \{c_1, c_2, ..., c_{|p|}, c_{|p|}\}$ in all phases, $d_k = \text{phase\_time\_diagram}(c_1, c_2, ..., c_{|p|}, c_{|p|})$, where $k \in J_1 \times J_2 \times ... \times J_{|p|}$,

   If $d_k < d^*$, $d^* = d_k$ and $C^* = C_k$.

   If $d^* \leq \xi$, stop and return $d^*$.

   $i_{|p|} = i_{|p|} + 1$

   End while $i_{|p|}$

   ......

   $i_p = i_p + 1$

   End while $i_p$

   ......

   $i_{l} = i_{l} + 1$

   End while $i_{l}$

Return $d^*$

Figure 5.3 Revised exhaustive algorithm to find the near-optimal solutions
The detailed algorithm is presented in Figure 5.3. Sub procedure \textit{find\_cut\_order}(i_p, V_p) returns the \textit{i}th “best” cut position for phase \textit{p} given in an interval vector \( V_p = (\tau_{\text{p1}}, \tau_{\text{p2}}, \ldots, \tau_{|J_p|}, \gamma) \) \( \forall i_p, |J_p| > 0 \). So the \textit{i}th “best” cut position is the position of the request \((1, 2, \ldots, |J_p|+1)\) in vector \( V_p \) with the \textit{i}th largest value. The sub procedure \textit{phase\_time\_diagram}( \{c_1, c_2, \ldots, c_p, \ldots, c_{|J_p|}\} ) returns the delay of input cut group \( \{c_1, c_2, \ldots, c_p, \ldots, c_{|J_p|}\} \) by plotting the phase time diagram proposed in Chapter 4.

A small example of delay evaluation is illustrated in Figure 5.4. There two priority request lists \( J_2 = \{1, 2\} \) for phase 2 and \( J_4 = \{1\} \) in phase 4. All the cyclic serving rectangles (CSR) are plotted on Figure 5.4 (a). Suppose a cut group \( \{c_1 = 0, c_2 = 2, c_3 = 0, c_4 = 2\} \) is evaluated using the phase-time diagram, meaning that the cut before both request 2 in \( J_2 \) and \( J_4 \). Since \( 2 = |J_4|+1 \), this cut assigns all the requests in \( J_4 \) to the first cycle. Figure 5.4 (b) plots the exact serving rectangles (SR) after cuts. Bounded by the slopes of minimal green and maximal green, a signal plan is generated to travel through each SR with minimal delay starting from the origin – which represents the current time. In Figure 5.4 (b), the cut is evaluated by a feasible signal plan with no delay, meaning that the feasible plan is also an optimal plan. Therefore, the phase time diagram also provides the optimal signal timing plan when evaluating the cut group \( \{c_1 = 0, c_2 = 2, c_3 = 0, c_4 = 2\} \).

The near-optimal plan can be implemented with real vehicle actuations using the same concept of GEG developed in Chapter 4 section 4.5.2.
5.2.3 Simulation results

To evaluate the heuristic, same numerical experiments from Chapter 4 (Figure 4.10) were conducted. A simple two-intersection arterial was modeled in VISSIM on a short section of Speedway Blvd. in Tucson, AZ, bounded by from Campbell Ave. to Cherry Ave. Four conflicting bus routes were added in the model as shown in Figure 4.10. The experiments are configured as follows:

- 3 traffic volumes: low, medium, large.

Figure 5.4 (a) CSRs on the phase-time diagram (b) Phase-time diagram evaluation of an assignment of serving $R_{22}$ in the second cycle

[Diagram showing phase-time diagrams with CSRs and cycle timing]

5.2.3 Simulation results

To evaluate the heuristic, same numerical experiments from Chapter 4 (Figure 4.10) were conducted. A simple two-intersection arterial was modeled in VISSIM on a short section of Speedway Blvd. in Tucson, AZ, bounded by from Campbell Ave. to Cherry Ave. Four conflicting bus routes were added in the model as shown in Figure 4.10. The experiments are configured as follows:

- 3 traffic volumes: low, medium, large.
- Performance comparisons of 4 non-coordinated control methods: “ASC free”, “Determ free”, “Robust free”, “ASC-TSP free” and “Heuristic”, where “ASC free” is full actuated control; “Determ free” and “Robust free” represents the deterministic formulation and robust formulation for non-coordinated priority control, respectively; “ASC-TSP free” is fully actuated control with TSP enabled. “Heuristic” denotes the proposed algorithm in this chapter.

- 10 runs per volume per method with different random seeds.

- Each run lasts 1 hour. A total of 150 simulation runs were conducted.
Average total vehicle delay and average bus are measured at three volume levels. The performances of five methods are shown in Figure 5.5. Some observations are as follows:

1. Every method performs very well under low volume. It is expected that “ASC free” would have very large bus delay, since bus priority is not considered in “ASC free” method.

2. “Determ” method has large variations. It can generate a large amount of delay especially for large volume, since a deterministic request that is not served due to random effects can result in very poor performance.
3. “ASC-TSP free” doesn’t work very well under large volume traffic with multiple bus priority.

3. “Robust free” method outperforms other methods with very stable solutions under all traffic volumes.

4. “Heuristic” method shows close results compared with “Robust” method with the additional benefit that it can be implemented for real-time control.

<table>
<thead>
<tr>
<th></th>
<th>Volume 1</th>
<th>Volume 2</th>
<th>Volume 3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total vehicle delay</td>
<td>-2.59%</td>
<td>-2.53%</td>
<td>2.31%</td>
<td>-0.94%</td>
</tr>
<tr>
<td>Bus delay</td>
<td>-7.24%</td>
<td>15.87%</td>
<td>5.31%</td>
<td>4.65%</td>
</tr>
</tbody>
</table>

Furthermore, delay increment percentages of “Heuristic” are shown in Table 5.1, compared with “Robust free”. The average total vehicle delay is the same between these two methods. There is only about 5% delay increment in bus delay for using “Heuristic” algorithm.

The results of the microscopic simulation study shows that proposed heuristic algorithm can successfully estimate the near-optimal sequence of serving requests in the existing request table. Therefore, the proposed algorithm was coded in C++ and implemented in an embedded Linux system on an RSE as a solver to produce near-optimal signal plans for a table of request as received from OBEs as priority requesting vehicles approach an intersection.
5.3 Field implementation

To implement the proposed algorithm in the field for evaluation and demonstration, it is necessary to have a basic v2x environment with OBEs, RSEs and intersections. The multiple priority control algorithm presented in this Chapter was implemented based the v2x environment provided by Arizona E-IntelliDrive\textsuperscript{SM} (called E-VII before 2009) project. The goal of the Arizona E-IntelliDrive\textsuperscript{SM} initiative is to develop and test advanced technologies and integrate roadway systems with emergency responder vehicles to improve emergency response efficiency as well as enhance responder safety (ADOT 2008).

5.3.1 System structure

The entire system of multiple priority control was implemented at a real intersection at Southern Ave. & 67 Ave and tested on February 22, 2010 and March 10, 2010. The intersection layout is shown in Figure 5.6. The intersection is a typical signalized four-legged two-way intersection. The traffic signal controller for this intersection is an Econolite ASC/3 (Econolite 2010). The location of RSE and MAP waypoints are shown in Figure 5.6.
Figure 5.6 Test intersection layouts at Southern Ave. & 67 Ave. Phoenix, AZ (Google Earth)

The system structure of implementation is illustrated in Figure 5.7. The RSE was installed in the Cabinet, connected with the Econolite ASC/3 traffic controller via Ethernet. The RSE reads traffic status using NTCIP objects from the signal controller and broadcasts the signal status, intersection MAP (Society of Automotive Engineers 2006) and current request table to the DSRC network. When receiving new priority request or updated priority request, the RSE updates the request table and runs the heuristic algorithm to produce a new near-optimal signal plan based on current signal status and phase-ring configurations. The new signal plan is then implemented on traffic signal controller by NTCIP phase control objects including phase hold and phase force-offs commands.
When approaching intersection, a vehicle equipped with an OBE will receive the intersection MAP, signal status and current request table through the DSRC network as it enters the communication range (about 1km). Combining the intersection MAP, current GPS position, vehicle heading, and speed (obtained from a GPS receiver), a simple algorithm running on the OBE is able to calculate the desired service phase and the relative arrival time at the stop bar. Using this information, a priority request is sent from the OBE to the RSE including the requesting phase, upper and lower bounds on when the vehicle will arrive, and the time required to clear the front queue if the vehicle joins the queue (detected as stopping some distance from the stopbar).

Figure 5.7 System structure of field implementation
5.3.2 RSE and OBE introductions

The OBE and RSE hardware are provided by Savari Networks (Savari 2010). They are both embedded computer systems with a 500Mhz processor, 256MB of memory, 4GB of compact flash disk space, multiple networks (WiFi, DSRC, bluetooth and Ethernet) and include an integrated USB GlobalSat BU-353 GPS receiver.

5.3.2.1 RSE configurations

An RSE is usually mounted on a light or signal pole beside the traffic signal control cabinet or put inside the cabinet with an external antenna, as shown as Figure 5.8. There are three programs running on RSE:

- **RSE_BROADCASTER**: reads controller status, broadcasts intersection MAP, signal status and the request table on DSRC network every half seconds.
- **RSE_LISTENER**: listens on a specific port for priority requests, manages the request table.
- **RSE_SOLVER**: solve the multiple priority problems if there are any changes to the request table and implements the optimized signal plan on the controller via NTCIP.
5.3.2.2 OBE configurations

The OBE is mounted on the dash of each vehicle, as shown as Figure 5.9. There are three programs running on OBE:

- **OBE_LISTENER**: listens on a specific port for the intersection MAP, signal status and request table.
- **OBE_REQUESTER**: sends a priority requests when it is either in range of an intersection stop bar or stopped in the queue.
- **OBE_WEBDESIGNER**: dynamically generate web pages for the webserver on OBE. This provides an interface between the driver and OBE. The service of each OBE includes four states: 1) OBE approaching an intersection; 2) receiving MAP...
and request table, 3) sending a priority request and 4) traveling through the intersection. The web pages corresponding to four steps are shown in Figure 5.10 (a)-(d).
Figure 5.9 OBE installation (a)
Figure 5.10 Web page displays (a) OBE approaches a intersection; (b) OBE receives MAP and request table; (c) OBE sends a priority request; (d) OBE travels through the intersection. (GID is a former name of MAP in SAE J2735)

5.3.3 NTCIP implementation

The National Transportation Communications for Intelligent Transportation System Protocol (NTCIP) is a family of standards maintained by NEMA, AASHTO and ITE. The NTCIP standards provide the rules and vocabulary for electronic traffic equipment from different manufacturers to communicate and operate with each other. NTCIP compliant devices must follow this standard. The NTCIP is the first protocol for the transportation industry that provides a communications interface between disparate hardware and software products. The NTCIP effort not only maximizes the existing
infrastructure, but it also allows for flexible expansion in the future, without reliance on specific equipment vendors or customized software (Institute of Transportation Engineers 2010).

It is necessary to understand tree structure of NTCIP objects. In order to read or write an object in the ASC controller, the object identification number (OID) should be located and specified in the NTCIP tree. An example of reds group 1 OID is “1.3.6.1.4.1.1206.4.2.1.1.4.1.2.1”. The object in the left of each dot is the parent of the object in the right. This could be depicted in a tree structure, shown as Figure 5.11.

The NTCIP set of standards defines the Point to Multi Point Protocol (PMPP) for communication with traffic devices. PMPP is a specialization of the HDLC (High-Level Data Link Control) protocol which can use SNMP (Simple Network Management Protocol) for the information field. The SNMP is already a widely accepted protocol on the internet for remote devices.

In the field test, Net-SNMP is adapted to perform communications between the RSE and the traffic controllers. Net-SNMP is a suite of applications used to implement different version of SNMP through IPv4 or IPv6 (Net-SNMP 2010).
5.3.4 DSRC field experiences

Dedicated Short-Range Communications (DSRC) is 75 MHz of spectrum at 5.9 GHz allocated by the Federal Communications Commission (FCC) to “increase traveler safety, reduce fuel consumption and pollution, and continue to advance the nation’s economy”
The DSRC communication range was measured under both an urban intersection and a suburban intersection, shown in Figure 5.12 (a) and (b) respectively. Both tests are configured with a data rate 3Mb/s in the control channel. The urban intersection is located at Mountain Ave. and Speedway Blvd. Tucson, AZ. This intersection is on the campus of University of Arizona which is surrounded by large buildings and trees. The communication range is affected by the blockage. It varies from 300 meters to 600 meters. The suburban intersection is at Southern Ave. and 67 Ave. in Phoenix, AZ. There is no any obvious blockage around this intersection. The DSRC communication distance falls in the range of 700 meters to 1100 meters.
5.3.5 Field tests

With the support of Maricopa County’s Regional Emergency Action Coordinating (REACT) team, Three REACT vehicles were equipped with OBEs, shown as Figure 5.13 (a) and (b). Different scenarios were tested with single vehicle, two vehicles and three vehicles. Multiple test vehicles were positioned and coordinated to arrive at the intersection according to the test plan. The test plan included the following tests:

1. Single vehicle approaching intersection
   a. Crossing the desired service phase
   b. During the desired service phase
2. Two vehicles approaching intersection
   a. Vehicles on conflicting movements
   b. Vehicle on concurrent movements

3. Three vehicles approaching intersection: two vehicles on concurrent movements and one vehicle on conflicting movement.
   a. The timing phase is the same as two-vehicle requested phase.
   b. The timing phase is the same as one-vehicle requested phase.

The detailed results of each scenario are described in Table 5.1.

<table>
<thead>
<tr>
<th>Number of test vehicles</th>
<th>scenarios</th>
<th>Test descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>When the vehicle is less than 30 seconds away from the intersection, the OBE sends a priority request to the RSE. A “Force Off” was observed on the status display of the traffic controller.</td>
</tr>
<tr>
<td>b</td>
<td></td>
<td>When the vehicle is less than 30 seconds away from the intersection, the OBE sends a priority request to the RSE. A “Green Hold” was observed on the status display of traffic controller.</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>When both vehicles were sending requests to the intersection, the traffic controller was able to show a “Green Hold” status for the phase that was timing when the requests were received, essentially holding the phase to allow the vehicle to be served. After the serving vehicle in the controller displayed a “Force off” status to terminate the current phase immediately and then changed to serve the remaining vehicle on the conflicting phase.</td>
</tr>
<tr>
<td>b</td>
<td></td>
<td>Two cases were tested. In the first case, two vehicles approached during the green phase. When the requests were received, a “Green Hold” was observed in the status display of traffic controller. The signal remained green until both vehicles passed through the signal. In the second case, two vehicles approached intersection during the red phase. A “Force Off” was observed in the status display of traffic controller and then “Green Hold” was observed after the controller switched to the desired service phase.</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>A “Green Hold” was observed in the controller status display until two vehicles traveled through the intersection. After both vehicles were cleared, a “Force Off” status was immediately displayed until</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>the phase changed. The third vehicle was then served and allowed to cross the stop bar. There was some delay for the third vehicle.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A “Green Hold” was observed on the controller status display until the single vehicle traveled through the intersection. After the single vehicle was cleared, a “Force Off” was immediately observed in the controller status display. The first vehicle of the two-vehicle group was able to pass through stop bar with some delay, and the second vehicle arrived just as the signal turned green.</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.13 Field tests (a) REACT vehicles ready to test; (b) A priority request is being served.
One of field experiments is shown in Table 5.3 and Figure 5.14. Two conflicting emergency vehicles were approaching the intersection southbound (phase 2) and westbound (phase 4), respectively. Table 5.3 presents the recorded data and output control command in RSE from 10:23:39 am to 10:24:26. The notations of Table 5.3 head are illustrated as below:

- **T**: relative time, starting from 0
- **Time stamp**: actual time stamp
- **ϕ1–8**: the real-time signal status of phase 1-8: red is 0, yellow is 2 and green is 1
- **CMD**: the commands sent from RSE to traffic controller, including three types: “NONE” is no command, “Hold” is phase hold and “FORCEOFF” is phase force-off
- **ID1(2)**: the OBE ID of first (second) request entry in the request table
- **Type1(2)**: the request type of first (second) request entry in the request table, including three types: “New”
- **Cls1(2)**: the priority class level of first (second) request entry in the request table
- **ETA1(2)**: the estimated time of arrival of first (second) request entry in the request table
- **P1(2)**: the requested phase of first (second) request entry in the request table
- **Q_t1(2)**: the needed time to clear front queue of first (second) request entry in the request table. If Q_t>0, the vehicle stops in a queue. Otherwise the vehicle is moving.
Table 5.3 Field test data with two conflicting priority requests on March 10, 2010

<table>
<thead>
<tr>
<th>T</th>
<th>Time stamp</th>
<th>φ1</th>
<th>φ2</th>
<th>φ3</th>
<th>φ4</th>
<th>φ5</th>
<th>φ6</th>
<th>φ7</th>
<th>φ8</th>
<th>CMD</th>
<th>ID1</th>
<th>Type1</th>
<th>Cls1</th>
<th>ETA1</th>
<th>P1</th>
<th>Q_t1</th>
<th>ID2</th>
<th>Type2</th>
<th>Cls2</th>
<th>ETA2</th>
<th>P2</th>
<th>Q_t2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10:23:39</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>NONE</td>
<td>OBE002</td>
<td>New</td>
<td>1</td>
<td>27</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>10:23:40</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>26</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>NEW</td>
<td>1</td>
<td>28</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10:23:41</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>24</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>29</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>10:23:42</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>24</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>25</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>10:23:43</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>22</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>25</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>10:23:44</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>21</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>23</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>10:23:45</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>22</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>10:23:46</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>19</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>21</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>10:23:47</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>20</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>10:23:48</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>18</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>19</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10:23:49</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>18</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>10:23:50</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>15</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>17</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>10:23:51</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>10:23:52</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>13</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>14</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>10:23:53</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>12</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>14</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>10:23:54</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>13</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>10:23:55</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>13</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>10:23:56</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>HOLD</td>
<td>OBE002</td>
<td>Update</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>OBE003</td>
<td>UPDATE</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Time</td>
<td>Current State</td>
<td>Device</td>
<td>Time</td>
<td>Current State</td>
<td>Device</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>--------</td>
<td>------------</td>
<td>---------------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:23:57</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:23:58</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:23:59</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:00</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:01</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:02</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:03</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:04</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:05</td>
<td>OBE002 UPDATE</td>
<td>1</td>
<td>01:00:00</td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:06</td>
<td>OBE002 CLEAR</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:07</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:08</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:09</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:10</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:11</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:12</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:13</td>
<td>FORCEOFF</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:14</td>
<td>HOLD</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:15</td>
<td>HOLD</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:24:16</td>
<td>HOLD</td>
<td></td>
<td></td>
<td>OBE003 UPDATE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|   | Time       | Action | Value | Value | Value | Value | Value | Value | Hold
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>10:24:17</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>10:24:18</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>10:24:19</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>10:24:20</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>10:24:21</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>10:24:22</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>10:24:23</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>10:24:24</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>10:24:25</td>
<td>HOLD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>10:24:26</td>
<td>NONE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 15

OBE003 UPDATE 1 0 4 0

OBE003 UPDATE 1 0 4 0
At 10:23:39 on March 10th, 2010, the first request was received as a new request from OBE002, requesting phase 2 with estimated time of arrival 27 seconds and time to clear the queue 0 second. One second later at 10:23:40, the second request comes from OBE003, requesting phase 4 with estimated time of arrival 28 seconds, while the estimated time of arrival of OBE002 reduced to 26 seconds. At 10:23:40, the proposed heuristic algorithm was called to solve the conflicted multiple priority requests given that current timing phases are 2 and 6. Due to almost the same arrival of conflicting requests, at least one vehicle needs to stop before the stop bar. The phase-time diagram of these two requests is shown on Figure 5.14 (a), considering two cycles and 4 seconds robust interval. Since no protected left-turn phase is used in this intersection, only two phases exist in a ring. A best s-t cut combination was found \{c_1 = 0, c_2 = 2, c_3 = 0, c_4 = 2\} to
serve both requests in cycle 1. Since the current timing phase at time 10:23:40 is phase 2 and 6, the serving sequence is to hold phase 2 and 6 to serve OBE002 first and then force-off to OBE003. The entire plan was implemented by phase hold and phase force-offs through NTCIP protocol from 10:23:41 to 10:24:26. Actually the heuristic algorithm was solved every one second to correspond to the real-time traffic and request changes.

These demonstrations showed that the multiple priority control system using the DSRC communications was successful at controlling a traffic signal. The field test shows the proposed heuristic algorithm is ready to be applied on a real isolated intersection to serve multiple priority requests.

5.4 Summary

A “solver free” heuristic algorithm was presented in this Chapter for field application. The multiple priority control problem was simplified to a polynomial solvable cut problem by adding few assumptions. Each cut combination corresponds to a unique serving sequence of multiple priority requests. First, the cut problem is proved to be polynomial solvable problem. Second, a revised exhaustive search algorithm is proposed to search for the best solutions within a defined tolerance range. The total priority delay of each cut combination is assessed using the phase-time diagram developed in Chapter 4. Finally both microscopic simulations with VISSIM and a field test on a real intersection confirm the effectiveness of the proposed algorithm.

In future research, heuristic algorithms can be revised to consider more than two cycles. Currently only one single, isolated intersection is considered. Heuristic algorithms
should be studied for arterial priority control. Further field implementations are required to test the algorithm’s effectiveness at consecutive intersections along an arterial.
6.1 Introduction

The previous two chapters presented algorithms for multi-modal priority control for emergency and transit vehicles. These algorithms were intersection based algorithms for either isolated operation or within a traditional cycle, offset, and split based coordination plan. Systems of traffic signals, such as along an arterial, need to be coordinated based on the actual traffic composition and the flow of vehicles in platoons. With the advent of vehicle based communications, it is possible to know where most of the vehicles are on the network and plan traffic signal control to best serve all vehicles in the network.

This Chapter presents a unified platoon-based formulation (PAMSCOD: Platoon-based Arterial Multi-modal Signal Control with Online Data) to concurrently optimize network traffic signal control for different travel modes, given the assumption that advanced communication systems are available between travelers and traffic controllers. In this Chapter, four modes of traffic composition are considered: emergency vehicles, buses, trucks and passenger vehicles in a decision framework that can accommodate pedestrians and bicycles. First, when approaching the intersection, travelers are able to send “green light” request to the traffic controller. The “green light” request including travel mode, position, speed and requested traffic signal phase. Single requests are
categorized and clustered into platoons by priority level and phase. Finally, a mixed-integer linear program (MILP) is solved online for future optimal signal plans based on the real-time arterial request platoon data and traffic controller status.

One feature of PAMSCOD is network level dynamical signal coordination without the constraint of common cycle length and fixed offsets. State-of-practice traffic actuated coordinated control requires fixed common cycle length and offsets, which works only for automobiles under a certain range of traffic flow. In addition to consideration of multi-modes, the large variance of traffic flow would degrade the performance of offline optimized time-of-day plans (Yin 2008). In order to achieve robust and online network traffic signal control, the concept of multi-modal dynamical progression (MDP) is proposed in this paper. Regardless of cycle length and offsets, multi-modal dynamical progression along an arterial is realized in the MILP formulation by adjusting phase durations to minimize the overall multi-modal delay both at the current intersection and downstream intersections.

Large MILP formulations for signal control problems are well known to be NP-hard and not easily solvable in real-time. To address the issue of computation complexities, three measures are taken into accounts. First, the automobile requests are aggregated into platoons requesting specific phases to reduce total number of integer variables since the number of integer variables contributes significantly to the complexity of MILP formulation. Second, the integer feasible solution region is enhanced by assuming a first-come, first-serve discipline for the requests on the same approach. For example, suppose two platoons are requesting the same phase. The first platoon should always travel
through intersection before the second platoon. Third, multi-modal dynamical progression is only considered between adjacent intersections due to the large variations of real-time traffic data.

It is well known that traffic signal control is very difficult during oversaturated traffic condition (Gazis 1964). Oversaturated traffic situations are caused by traffic demands that exceed the available capacity and can produce queues that grow over time. Residual queues can overflow the storage capacity of urban streets and physically block intersections to cause de facto red (Abu-Lebdeh & Benekohal 1997) when queue build-up lead to complete blockage of upstream signals, and no traffic can discharge when the traffic signal is green. With real-time queue length and queue size is available from advanced vehicle based communications, the link capacity between adjacent intersections is known. Therefore, PAMSCOD is able to control the discharge rate from upstream intersection to avoid the de facto red.

The follow chapter is organized as follows. Section 6.2 presents a fast algorithm to identify platoons in real-time. Section 6.3 proposes a MILP to deal with online arterial (or network) platoon-based multi-modal traffic signal control. Both dynamical progression and queue management are addressed in this section. Section 6.4 compares the performance of PAMSCOD within different methods under different traffic volume and scenarios. Section 6.5 presents a summary of the model and the performance comparison.

6.2 Platoon-based signal optimization
Platoon based traffic signal control is not new in previous research literatures. Platoon dispersion and secondary flows were considered via a simplified platoon-dispersion algorithm in the TRANSYT-7F model (Robertson 1969) and (Wallace et al. 1998). Dell'Olmo and Mirchandani (Dell'Olmo & P.B. Mirchandani 1995) proposed a real-time platoon-based network level control algorithm, called REALBAND, that attempts to resolve future platoon conflicts. REALBAND generates platoon information based on loop detector detector data (Gaur & P.B. Mirchandani 2001) and creates a decision tree based on the projected platoon flow to find the best root-to-leaf decision by minimizing the total platoon delay. The point detectors used in REALBAND only provide a snapshot of the traffic platoon data, which may vary significantly along the arterial according due to dispersion, departures, and vehicles joining the platoon. Dell'Olmo and Mirchandani didn’t present an effective way to prune the decision tree, which grows exponentially with an increase in the number of considered intersections and conflicted platoons. Recently, a platoon-based traffic signal control algorithm was developed by Jiang (Jiang et al. 2006) however this algorithm only suitable for isolated intersection with major-minor traffic volume.

In the model presented in this chapter, the platoon information is obtained directly by communicating between the traffic controller and the vehicles. Individual vehicles are continuously tracked using the advanced communication system and platoons are identified using a one-dimensional clustering algorithm.

6.2.1 Hierarchical platoon recognition
The first step in the development of the platoon based control strategy is the identification of platoons from individual vehicle data. Assume that each vehicle along arterial is represented as a single point at a specific time. Figure 6.1(a) illustrates the four characterizing variables of a platoon: time of leading vehicle arrival $T_a$, time of tail vehicle arrival $\bar{T}_a$, platoon average speed $V_p$, platoon size $N_p$ and time needed to clear the platoon $T_p$. Platoon 1.1 is actual a queue in front of a stop bar. In this platoon, $T_a = \bar{T}_a = 0$, and $T_p = 3600N_p / S_r + T_{st}$, where $N_p$ is number of vehicles in the platoon, $S_r$ is saturation flow, and $T_{st}$ is start-up time. Platoon 2 in Figure 6.1 is a moving platoon, where $0 \leq T_a < \bar{T}_a$ and $T_p = \bar{T}_a - T_a$.

Overall platoons can be easily recognized by car-car distance, headways, or link density. Traffic engineers can easily tell different platoons by the snapshot of individual car locations on the arterial. Any platoon identification algorithm in which the goal of the algorithm is to filter out individual cars and identify platoons in groups. In previous studies, Gaur and Mirchandani (Gaur & P.B. Mirchandani 2001) utilized link density to identify platoon. Jaing (Jiang et al. 2006) directly separated platoons by the vehicle headways. Under different geometry of street, the threshold level (e.g. headway) may vary to correctly identify platoons. However, none of above algorithm is able to generate multiple layer partitions according to different threshold levels. Single threshold may separate the vehicle improperly under different environment.

Figure 6.1 (a) gives a snapshot of on one link of a signalized intersection. Black dots represent the vehicle positions. The proposed platoon recognition algorithm can be
described in three steps. First, all the vehicle are mapped to a one-dimensional axis. Second, the proposed idea aims to cluster vehicles with different levels of headways. In this example, 2 seconds and 1 second, two levels of headways are adopted, shown as Figure 6.1 (b). In higher level (2 seconds) clustering, it outputs two platoons: platoon 1.1 and platoon 1.2. In lower level clustering, platoon 2.1-2.4, four platoons are generated. Third, given the desired number output of platoon is 3, the algorithm combines the platoon 2.3 and platoon 2.4 to be a new platoon 2.3’ and output three platoons 2.1, 2.2 and 2.3’.

Figure 6.1 (a) Original point vehicles on a link; (b) Platoon one-dimensional mapping and 2-level clustering with 2s and 1s headways respectively

\[
\begin{align*}
\text{Stop bar} & \quad T_a = T_a = 0 \\
T_p &= \frac{3600 N_p}{S_r + T_{st}} \\
0 \leq T_a &< T_a \\
T_p &= T_a - T_a
\end{align*}
\]
The problem of platoon recognition can be addressed as a clustering problem. A simple one-dimensional hierarchical clustering method with a specified threshold is proposed to achieve multiple level platoon recognition. The algorithm is composed of the following steps:

1. Map all the vehicle points onto a spatial axis, ordered by distance to the stop bar, as shown in Figure 6.1 (b). Suppose the set of vehicles on one link is \( N \). For each vehicle \( i \) in \( N \), the speed and location of vehicle is denoted as \( v[i] \) and \( l[i] \), respectively. Calculate headway array \( h[i] \) and spacing array \( s_{p}[i] \) as follows:

\[
s_{p}[i] = \begin{cases} 0 & i = 1 \\ \frac{l[i] - l[i-1]}{v[i]} & \text{otherwise} \end{cases} \quad (6.1)
\]

\[
h[i] = \begin{cases} 0 & \forall v[i] = 0 \\ s_{p}[i]/v[i] & \text{otherwise} \end{cases} \quad (6.2)
\]

2. Let \( c \in M = \{1, 2, 3, \ldots\} \) be the set of clustering levels. The threshold headway in each level \( c \) in \( M \) is denoted as \( \varepsilon[c] \). For each level \( c \) in \( M \) and each vehicle \( i \) in \( N \), set partition flag array \( f[c][i] \) as follows:

\[
f[c][i] = \begin{cases} 0 & \forall h[i] < \varepsilon[c] \\ 1 & \text{otherwise} \end{cases} \quad (6.3)
\]

3. The number of platoons in each level \( c \) is denoted as \( p_{n}[c] = \sum_{i \in N} f[c][i] + 1 \). Given \( p_{n}^{*} \), the satisfactory number of output platoon, which locates between outputs of \( c_{1} \) and \( c_{2} \) : 

\[ p_{n}[c_{1}] \leq p_{n}^{*} < p_{n}[c_{2}] \]
4. Generate the platoon list $P_l[j]$ for $j$th platoon such that the number of output platoons equals to $p_n^*$:

Let $j = 1$ and $n = 0$, $n$ stores the number of platoon generated by level $c_2$, but not by level $c_1$.

For $i$ in $N$

If $f[c_1][i] = 0$ and $f[c_2][i] = 0$

Add $i$ in platoon list $P_l[j]$

Else if $f[c_1][i] = 0$ and $f[c_2][i] = 1$ and $n < p_n^* - p_n[c_1]$

$j = j + 1$

$n = n + 1$

Add $i$ in platoon list $P_l[j]$

Else if $f[c_1][i] = 1$ and $n \geq p_n^* - p_n[c_1]$

$j = j + 1$

Add $i$ in platoon list $P_l[j]$

End if

End for

The headway threshold should be different with traffic condition and geometry of roadway. (Dell'Olmo & P.B. Mirchandani 1995) suggested 2 seconds for low volume traffic and 1 second for high volume traffic. (Jiang et al. 2006) claimed that critical headway is 2.5 seconds, which located in the middle of commonly assumed saturated headway (2.0 s) and the desired allowable gap (3.0 s).
With hierarchical platoon recognition, the output number of platoons could be specified as user’s input $p_n^*$. Take Figure 6.1 (b) for example. Suppose two-level platoon recognition is considered with $\varepsilon[1] = 2\sec$ and $\varepsilon[1] = 1\sec$. The largest headways are $h[12] = 2s$, $h[16] = 1s$ and $h[20] = 1s$. The first level platoon recognition ends up with 2 platoons, separated by $h[12]$, while the second level platoon recognition ends up with 4 platoons, separated by $h[12]$, $h[16]$ and $h[20]$.

Suppose $p_n^* = 3$ is the desired number of platoon in Figure 6.1(b). Neither the first level nor the second level outputs 3 platoons. However, 3 platoons can be retrieved by combing the results of the first level clustering and the partial of the second clustering. The output of step 4 will combine platoon 2.3 and platoon 2.4 as a single platoon, denoted as 2.3&2.4 in the second level. So the output 3 platoons will be platoon 1.1, platoon 2.2, and platoon 2.3&2.4.

It is obvious that the complexity of proposed algorithm is $O(|M| \cdot |N|)$. Therefore, the algorithm is suitable for real-time platoon recognition. In the environment of this work, each vehicle is traceable according to their unique on-board equipment (OBE) id. The platoon recognition algorithm seems very easy under the assumption that 100% penetration of communications is available. Future research can focus on how to dynamically identify platoon by partial penetration and point detector data.

6.3 Mixed integer linear program (MILP) in PAMSCOD
Several MILP formulations have been proposed to solve the network traffic signal control problem over the past few decades. The original formulation was MAXBAND (Little 1966) and (Little et al. 1981). MULTIBAND, another well-known traffic signal control MILP, was developed later by Gartner in (Gartner, Assman et al. 1991) (Stamatiadis & Gartner 1996). Both MAXBAND and MULTIBAND aimed to maximize the “green wave” bandwidth by optimizing cycle length, splits and offsets. Neither of them is able to deal with oversaturated condition due to the assumptions in the traffic model.

Recently a number of papers have developed dynamic traffic signal control formulations based on the cell transmission model (CTM) (Daganzo 1994) and (Daganzo 1995). The CTM-based signal control formulation can address both unsaturated and oversaturated conditions by considering shockwaves and physical queues. Lo (1999, 2001) formulated the network signal optimization problem as a mixed-integer linear programming problem using CTM (Lo 1999) and (Lo 2001) and assuming that the cycle length is fixed. Lin and Wang (2004) formulated a more computationally efficient version of the MILP for the signal optimization problem based on the CTM (Lin & C. Wang 2004). But only two-phase signals were considered in their model. For the CTM-based formulations with the MILP approach, the problem size can grow very quickly with the size of the network and the time horizon. The “curse of dimensionality” makes it impossible to solve these formulations online directly even using commercially available packages such as CPLEX, LINDO, etc.
In this chapter, a platoon-based MILP is presented that reduces the number of integer variables required and that accommodates both unsaturated and oversaturated conditions by considering shockwaves and physical queues. The model notation is summarized in Table 6.1.

Table 6.1 Notation definition of decision variables (lower case) and data (upper case)

<table>
<thead>
<tr>
<th>Type</th>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets</td>
<td>$m \in M$</td>
<td>The set of travel modes</td>
</tr>
<tr>
<td></td>
<td>$i, n \in I$</td>
<td>The set of intersections</td>
</tr>
<tr>
<td></td>
<td>$i_d \in I_d(i, p)$</td>
<td>The set of downstream intersections for current phase $p$ at intersection $i$ $(I_d(i, p) \subseteq I)$</td>
</tr>
<tr>
<td></td>
<td>$p \in P$</td>
<td>The set of phases</td>
</tr>
<tr>
<td></td>
<td>$p \in P_t$</td>
<td>The set of through traffic phases $(P_t \subseteq P)$</td>
</tr>
<tr>
<td></td>
<td>$j, j_1 \in J$</td>
<td>The set of platoons</td>
</tr>
<tr>
<td></td>
<td>$k, c \in K$</td>
<td>The set of cycles</td>
</tr>
<tr>
<td></td>
<td>$p \in \Delta_{s1}(i)$</td>
<td>The starting phase in ring 1 at intersection $i$ $(\Delta_{s1}(i) \subseteq P)$</td>
</tr>
<tr>
<td></td>
<td>$p \in \Delta_{s2}(i)$</td>
<td>The starting phase in ring 2 at intersection $i$ $(\Delta_{s2}(i) \subseteq P)$</td>
</tr>
<tr>
<td></td>
<td>$p \in \Delta_{non}(i)$</td>
<td>The set of nonexistent phases (compared with a standard NEMA 8-phase controller) at</td>
</tr>
</tbody>
</table>
The set of past phases in the first cycle at intersection $i$ ($\Delta_{\text{non}}(i) \subseteq P$)

$p \in \Delta_p(i)$

The set of $j$ th valid platoons in model $m$, intersection $i$, and phase $p$

$(m,i,p,j) \in \Gamma = \{(m,i,p,j) \mid Np(m,i,p,j) > 0\}$

The set of platoons on arterial Eastbound. ($\Gamma_2 \subseteq \Gamma$)

$(m,i,p,j) \in \Gamma_2 = \{(m,i,p,j) \mid i \leq |l| - 1, p = 2, Np(m,i,p,j) > 0\}$

The set of platoons on arterial Westbound. ($\Gamma_6 \subseteq \Gamma$)

$(m,i,p,j) \in \Gamma_6 = \{(m,i,p,j) \mid i \geq 2, p = 6, Np(m,i,p,j) > 0\}$

$c_i(m,i,p,j)$

The cut ratio of serving platoon $(m,i,p,j) \in \Gamma$

$d_{\text{pen}}(m,i,p,j)$

Delay penalty to cut a platoon to serve in two cycles for platoon $(m,i,p,j) \in \Gamma$

$d_q(m,i,p,j)$

Queue delay at the current intersection for platoon $(m,i,p,j) \in \Gamma$

$\text{Decision variables}$

$d_{ql}(m,i,p,j)$

Leading vehicle queue delay in current intersection for platoon $(m,i,p,j) \in \Gamma$

$d_s(m,i,p,j)$

Signal delay in current intersection for platoon $(m,i,p,j) \in \Gamma$

$d_{sl}(m,i,p,j)$

Leading vehicle signal delay in current intersection for platoon $(m,i,p,j) \in \Gamma$
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d'_q(m,i,p,j)$</td>
<td>Queue delay in downstream intersection for platoon $(m,i,p,j) \in \Gamma_2 \cup \Gamma_6$</td>
</tr>
<tr>
<td>$d'_s(m,i,p,j)$</td>
<td>Signal delay in downstream intersection for platoon $(m,i,p,j) \in \Gamma_2 \cup \Gamma_6$</td>
</tr>
<tr>
<td>$e(i,p)$</td>
<td>Elapsed green time for phase $p$ at intersection $i$</td>
</tr>
<tr>
<td>$g(i,p,k)$</td>
<td>Green time for phase $p$ during cycle $k$ at intersection $i$</td>
</tr>
<tr>
<td>$s(i,p,k)$</td>
<td>Slack variables for phase $p$ during cycle $k$ at intersection $i$</td>
</tr>
<tr>
<td>$s_1(i)$</td>
<td>Slack variable for ring 1 at intersection $i$</td>
</tr>
<tr>
<td>$s_2(i)$</td>
<td>Slack variable for ring 2 at intersection $i$</td>
</tr>
<tr>
<td>$t(i,p,k)$</td>
<td>Starting time of phase $p$ during cycle $k$ at intersection $i$</td>
</tr>
<tr>
<td>$v(i,p,k)$</td>
<td>Phase duration time of phase $p$ during cycle $k$ at intersection $i$</td>
</tr>
<tr>
<td>$\theta(m,i,p,j,k)$</td>
<td>0-1 binary variables. Whether to serve platoon $(m,i,p,j)$ in cycle $k$ at current intersection (if $\theta(m,i,p,j,k) = 1$, the platoon $(m,i,p,j)$ is served in cycle $k$ at current intersection; else, not served in cycle $k$)</td>
</tr>
<tr>
<td>$\theta'(m,i,p,j,c)$</td>
<td>0-1 binary variables. Whether to serve platoon $(m,i,p,j)$ in cycle $c$ at downstream</td>
</tr>
</tbody>
</table>
intersection (if $\theta'(m, i, p, j, c) = 1$, the platoon $(m, i, p, j)$ is served in cycle $c$ at downstream intersection; else, not served in cycle $c$)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Weighting factor to the sum of slack variables</td>
</tr>
<tr>
<td>$C_i(i, p)$</td>
<td>Link remaining storage capacity before solving the MILP</td>
</tr>
<tr>
<td>$C_r$</td>
<td>Reference cycle length (s)</td>
</tr>
<tr>
<td>$E(i, p)$</td>
<td>Elapsed green times for starting phase $p$ at intersection $i$ ($p \in \Delta_1(i) \cup \Delta_2(i)$) (s)</td>
</tr>
<tr>
<td>$G_{\min}(i, p)$</td>
<td>Minimal green time for phase $p$ at intersection $i$ (s)</td>
</tr>
<tr>
<td>$G_{\max}(i, p)$</td>
<td>Maximal green time for phase $p$ during at intersection $i$ (s)</td>
</tr>
<tr>
<td>$F_d(i, n)$</td>
<td>Platoon dispersion factor on link $(i, n)$</td>
</tr>
<tr>
<td>$H_p(m, i, p, j)$</td>
<td>Average headway of a platoon $(m, i, p, j)$ (s)</td>
</tr>
<tr>
<td>$H_s(i, p)$</td>
<td>Saturation flow headway for phase $p$ at intersection $i$</td>
</tr>
<tr>
<td>$H_{s1}$</td>
<td>Saturation flow headway for a single lane (s)</td>
</tr>
<tr>
<td>$L(i, n)$</td>
<td>Link length between intersection $i$ and $n$ (m)</td>
</tr>
<tr>
<td>$L_s$</td>
<td>Average vehicle spacing in queue (m)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$M$</td>
<td>A large number</td>
</tr>
<tr>
<td>$N_i(i, p)$</td>
<td>Number of lanes at intersection $i$ served by phase $p$</td>
</tr>
<tr>
<td>$N_p(m, i, p, j)$</td>
<td>Number of vehicles in platoon $(m, i, p, j)$</td>
</tr>
<tr>
<td>$O_1(i)$</td>
<td>Initial time for ring 1 at intersection $i$ (s)</td>
</tr>
<tr>
<td>$O_2(i)$</td>
<td>Initial time for ring 2 at intersection $i$ (s)</td>
</tr>
<tr>
<td>$P_r(m, i, n)$</td>
<td>Platoon remaining factor for platoon traveling from intersection $i$ to $n$ ($P_r(m, i, i) \in [0,1]$)</td>
</tr>
<tr>
<td>$R(i, p)$</td>
<td>Red clearance time for phase $p$ at intersection $i$ (s)</td>
</tr>
<tr>
<td>$S_r(i, p)$</td>
<td>Saturation rate for phase $p$ at intersection $i$ (veh/h)</td>
</tr>
<tr>
<td>$T_{\bar{a}}(m, i, p, j)$</td>
<td>Time arrival for leading vehicle in platoon $(m, i, p, j)$ (s)</td>
</tr>
<tr>
<td>$T_{\bar{a}}(m, i, p, j)$</td>
<td>Time arrival for tail vehicle in platoon $(m, i, p, j)$ (s)</td>
</tr>
<tr>
<td>$T_p(m, i, p, j)$</td>
<td>Green time needed to clear the platoon $(m, i, p, j)$ (s)</td>
</tr>
<tr>
<td>$T_{st}$</td>
<td>Start-up lost time (s)</td>
</tr>
<tr>
<td>$T_{i}(m, i, n)$</td>
<td>Travel time between intersection $i$ and $n$ for mode $m$ (s)</td>
</tr>
<tr>
<td>$V_f(m, i, p)$</td>
<td>Free flow speed for model $m$ in phase $p$ at</td>
</tr>
</tbody>
</table>
6.3.1 Unified precedence constraints

The signal controller model examined is based on the standard North American NEMA dual-ring eight-phase controller. A four-legged intersection with eight movements is shown in Figure 6.2(a). Typically, each ring in the controller contains 4 phases, depicted in Figure 6.2(b). A barrier exists across both rings between groups of conflicting movements so that all phases in one group have to terminate before any phase in the other group starts. The dual-ring controller can be modeled by a traditional precedence graph as depicted in Figure 6.3 (Head et al. 2006b). Arcs in the precedence graph represents the duration of phases, while nodes represents the phase transitions. Phase intervals can be easily visualized in the precedence graph.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>Intersection $i$ (m/s)</td>
</tr>
<tr>
<td>$V_s(m,i,p,j)$</td>
<td>Average speed for platoon $(m,i,p,j)$ (m/s)</td>
</tr>
<tr>
<td>$V_s$</td>
<td>Shock wave speed (m/s)</td>
</tr>
<tr>
<td>$W(m,i,p,j)$</td>
<td>Weight for platoon $(m,i,p,j)$</td>
</tr>
<tr>
<td>$Y(i,p)$</td>
<td>Yellow clearance time for phase $p$ at intersection $i$ (s)</td>
</tr>
<tr>
<td>$Z$</td>
<td>Allowable additional green rest time (s)</td>
</tr>
</tbody>
</table>
Figure 6.2 (a) Intersection layout; (b) A dual-ring, eight-phase controller
Phase sequence and phase initialization are very important considerations for online optimization. In formulation of the MILP, identification of the current phase to be the starting phase in the MILP and accounting for the time that has already elapsed in the cycle.
current phase define the phase time constraints (minimum green time) in the MILP formulation. In general, in an 8-phase controller, as shown in Figure 6.3, the starting phase pair can only be one of eight different cases: 1 and 5, 1 and 6, 2 and 5, 2 and 6, 3 and 7, 3 and 8, 4 and 7, or 4 and 8. The starting phases at each intersection can be different depending on the current status when the MILP problem is formulated. Similarly, different intersection may have different phase configurations, e.g. sequences, depending on the intersection geometry and operational considerations.

One important issue is how to handle different starting phases at different intersections in the unified precedence constraints formulation. To achieve this, it is assumed that two-ring structure is applicable on all the intersections and phase 1 and 5 are always considered as first phases in a phase configuration. Therefore, the entire precedence constraints are presented as follows:

\[
\begin{align*}
t(i,1,1) & = s_1(i) \quad \forall i \\
t(i,5,1) & = s_2(i) \quad \forall i \\
t(i, p, l) & = O_1(i) \quad \forall i, p \in \Delta_{i1}(i) \\
t(i, p, l) & = O_2(i) \quad \forall i, p \in \Delta_{i2}(i) \\
t(i,2,k) & = t(i,1,k) + v(i,1,k) + s(i,1,k) \quad \forall i, k \\
t(i,6,k) & = t(i,5,k) + v(i,5,k) + s(i,5,k) \quad \forall i, k \\
t(i,3,k) & = t(i,2,k) + v(i,2,k) + s(i,2,k) \quad \forall i, k
\end{align*}
\]
\[ t(i, 3, k) = t(i, 6, k) + v(i, 6, k) + s(i, 6, k) \quad \forall i, k \] (6.11)

\[ t(i, 7, k) = t(i, 2, k) + v(i, 2, k) + s(i, 2, k) \quad \forall i, k \] (6.12)

\[ t(i, 7, k) = t(i, 6, k) + v(i, 6, k) + s(i, 6, k) \quad \forall i, k \] (6.13)

\[ t(i, 4, k) = t(i, 3, k) + v(i, 3, k) + s(i, 3, k) \quad \forall i, k \] (6.14)

\[ t(i, 8, k) = t(i, 7, k) + v(i, 7, k) + s(i, 7, k) \quad \forall i, k \] (6.15)

\[ t(i, 1, k + 1) = t(i, 4, k) + v(i, 4, k) \quad \forall i, k \] (6.16)

\[ t(i, 1, k + 1) = t(i, 8, k) + v(i, 8, k) \quad \forall i, k \] (6.17)

\[ t(i, 5, k + 1) = t(i, 4, k) + v(i, 4, k) \quad \forall i, k \] (6.18)

\[ t(i, 5, k + 1) = t(i, 8, k) + v(i, 8, k) \quad \forall i, k \] (6.19)

\[ v(i, p, k) = g(i, p, k) + Y(i, p) + R(i, p) \quad \forall i, p \notin \Delta_{non}(i) \cup \Delta_p(i), k \] (6.20)

\[ v(i, p, k) = 0 \quad \forall i, p \in \Delta_{non}(i) \cup \Delta_p(i), k \] (6.21)

\[ g(i, p, k) = 0 \quad \forall i, p \in \Delta_{non}(i) \cup \Delta_p(i), k \] (6.22)

\[ s(i, p, k) = 0 \quad \forall i, p, k \geq 2 \text{ or } \forall i, p \notin \Delta_{s1}(i) \cup \Delta_{s2}(i) \cup \Delta_p(i), k = 1 \] (6.23)

\[ \sum_p \sum_k s(i, p, k) \leq Z \quad \forall i, p, k \] (6.24)

\[ G_{\min}(i, p) \leq g(i, p, k) \leq G_{\max}(i, p) \quad \forall i, p \notin \Delta_{s1}(i) \cup \Delta_{s2}(i) \cup \Delta_p(i) \cup \Delta_{non}(i), k \] (6.25)
\[ g(i, p, k) \geq G_{\min}(i, p) - E(i, p) \quad \forall i, p \in \Delta_{s1}(i) \cup \Delta_{s2}(i), k \quad (6.26) \]

A static set \( \Delta_{non} \) and three dynamic sets \( \Delta_{s1}, \Delta_{s2} \) and \( \Delta_p \) are introduced in this chapter to deal with different phase configuration and starting phases. \( \Delta_{non} \) represents those phases which are not used at a particular intersection. \( \Delta_{s1} \) and \( \Delta_{s2} \) denote the starting, or currently active, phases for ring 1 and ring 2 respectively. \( \Delta_p \) is the set of past phases, or phases that have already completed service at the current time, in the first cycle, assuming that the cycle always starts from phase 1 and 5. \( \Delta_{non} \) is fixed over the time horizon of consideration, while \( \Delta_{s1}, \Delta_{s2} \) and \( \Delta_p \) are variant depending on the starting phases and the elapsed time in the starting phases at the current time as the MILP is formed. Figure 6.4 illustrates these sets in a standard NEMA 8-phase configuration an example at one intersection \( i \). The current timing phases are phase 4 and 7 with some elapsed time \( E(i, 4, 1) \) and \( E(i, 7, 1) \), respectively. So the minimal green time for phase 4 and 7 in the first cycle should be reduced by the elapsed times. Past phases in cycle 1 are phase 1,2,3,5 and 6. So the duration of phases in 1,2,3,5 and 6 is 0 in cycle 1. Slack variables account for the green dwell time and maintain barrier constraints.

The unified precedence constraints have the following features:

1. The set of starting phases always include phases 1 and 5. The actual starting phases depend on the state of the controller at the time the MILP is formed and are denoted sets \( \Delta_{s1} \) and \( \Delta_{s2} \); constraint (6.4) and (6.5) in the formulation above.
2. If one or more of the starting phases is timing the yellow or red clearance, the issue is addressed in constraint (6.6) and (6.7) by introducing initial times $O_1(i)$ and $O_2(i)$, which are calculated by subtracting the elapsed yellow or all red times from the total clearance time $Y(i, p) + R(i, p)$.

3. The dual ring structure is represented by the unified constraints (6.6)-(6.19).

4. Different starting phases are modeled by set $\Delta_p(i)$. If phase $p \in \Delta_p(i)$, then the phase duration of phase $p$ should be 0, depicted in constraint (6.21) and (6.22).

5. Slack variables are introduced only for green rest to fulfill the barrier constraints in the first cycle. That means the maximal green constraints could be violated if the first barrier constraints need to be satisfied. Constraint (6.23) and (6.24) make sure the slack variables are equal to zero in all the phases after starting phases. Green rest means when one ring reaches barrier, as well as maximal green, but the other ring doesn’t reach the barrier. So the ring will rest on the barrier to satisfy the barrier constraint, regardless of the maximal green time constraint.

6. The minimal green times of the starting phases can be less than the pre-defined minimum green time, $G_{\min}(i, p)$, since the starting phases have already been timing for $e(i, p)$.

6.3.2 Platoon delay categorization and evaluation
Delay is perhaps the most commonly reported performance index in the literature of traffic signal control. Within the PAMSCOD formulation, before delay can be assessed, it is necessary to determine which cycle is selected to serve platoon \((m, i, p, j)\) thereby determining the correct phase starting time, \(t(i, p, k)\), to calculated delay of platoon \((m, i, p, j)\).

6.3.2.1 Platoon serving cycle selection

Figure 6.5 illustrates a platoon moving towards an intersection. This platoon can travel through the intersection either during cycle 1 or during cycle 2, assuming only two cycles are considered. If the platoon is served in the first cycle, the constraint (6.27) would be satisfied as follows:

\[
\frac{T_a(m, i, p, j)}{T_a(m, i, p, j)} + T_p(m, i, p, j) \leq t(i, p, 1) + g(i, p, 1) \quad \forall (m, i, p, j) \in \Gamma
\]  

(6.27)

where \(T_a(m, i, p, j)\) is the arrival time of leading vehicle in the platoon, and \(T_p(m, i, p, j)\) is the green time needed to clear the platoon. Constraint (6.27) requires the starting time of phase \(p\) in cycle 1 to occur such that platoon \((m, i, p, j)\) will be served during the green signal of the first cycle. If the platoon is served in the second cycle, two constraints should be fulfilled as follows:

\[
\frac{T_a(m, i, p, j)}{T_a(m, i, p, j)} + T_p(m, i, p, j) \leq t(i, p, 2) + g(i, p, 2) \quad \forall (m, i, p, j) \in \Gamma
\]  

(6.28)

\[
\frac{T_a(m, i, p, j)}{T_a(m, i, p, j)} + T_p(m, i, p, j) \geq t(i, p, l) + g(i, p, l) \quad \forall (m, i, p, j) \in \Gamma
\]  

(6.29)
Constraint (6.28) is the same as constraint (6.27). Constraint (6.29) ensures that the platoons arrive after the green time of the first cycle has completed, hence the platoons must be served in the second cycle. Binary variables \( \theta(m,i,p,j,k) \) are introduced to write constraint (6.27)-(6.29) in the unified formulation to handle the cycle service selection problem as constraint (6.30) and (6.31).

\[
\begin{align*}
T_g(m,i,p,j) + T_p(m,i,p,j) & \leq t(i,p,k) + g(i,p,k) \\
+ M(1 - \theta(m,i,p,j,k)) & \quad \forall (m,i,p,j) \in \Gamma, k \\
T_g(m,i,p,j) + T_p(m,i,p,j) & \geq t(i,p,k-1) + g(i,p,k-1) \\
- M(1 - \theta(m,i,p,j,k)) & \quad \forall (m,i,p,j) \in \Gamma, k
\end{align*}
\] (6.30)  
(6.31)

\[
\sum_k \theta(m,i,p,j,k) = 1 \quad \forall (m,i,p,j) \in \Gamma
\] (6.32)

For each valid platoon \( (m,i,p,j) \in \Gamma \), the corresponding \( \theta(m,i,p,j,k) = 1 \) makes constraint (6.28) and (6.29) go into effect, so the platoon \( (m,i,p,j) \) will be served in cycle...
Otherwise, platoon \((m,i,p,j)\) will not be served in cycle \(k\). Constraint (6.32) ensures there is only one cycle selected to serve platoon \((m,i,p,j)\). An example considering two cycles is illustrated in Figure 6.5. The approaching platoon \((m,i,p,j)\) can either be served in the first cycle or the second cycle depending on the duration of each phase.

6.3.2.2 Platoon split and link capacity

Constraint (6.28) and (6.29) only allow a platoon to be served completely during a cycle. However, there are some cases when platoons need to be split and served within two cycles or more. These cases include:

1. Platoon length may grow quickly during heavy demand or oversaturated traffic condition. In this case, \(T_p(m,i,p,j)\) could be greater than the maximal phase green times, \(G_{\text{max}}(i,p,j)\). In this case the demand exceeds the service capacity.

2. Downstream dynamic link storage capacity \(C_s(i_d,p)(i_d \in I_d(i,p))\) may be less than the size of the platoon \(N_p(m,i,p,j)\). Hence, the storage capacity is not sufficient to store the whole platoon.

3. Two conflicting platoons are approaching the intersection on the main street and side street simultaneously. Suppose the size of the main street platoon is much greater than the side street platoon. Assuming the the side street currently being served, e.g. the side street phase is in the green state, the signal may decide to switch to serve main street before completely serving the side street platoon, in
order to lower the total delay on the arterial. In this case, the side street platoons may be split to yield to main street platoons.

Platoon splitting in the first two cases is due to capacity constraints which cannot be violated. Platoon splitting in the third case is due to the performance benefit of platoon splitting, which increases arterial progression. The details of arterial progression will be discussed in the next section.

A positive variable \( c_i(m, i, p, j) \in [0, 1] \) is introduced to “cut” the platoons into two cycles, assuming that platoon can be served in no more than two cycles. Therefore, constraint (6.30) and (6.31) are rewritten as:

\[
\begin{align*}
T_s(m, i, p, j) + T_p(m, i, p, j) * c_i(m, i, p, j) & \leq t(i, p, k) + g(i, p, k) \\
+ M(1 - \theta(m, i, p, j, k)) & \quad \forall (m, i, p, j) \in \Gamma, k \quad (6.33)
\end{align*}
\]

\[
\begin{align*}
T_s(m, i, p, j) + T_p(m, i, p, j) * c_i(m, i, p, j) & \geq t(i, p, k - 1) + g(i, p, k - 1) \\
- M(1 - \theta(m, i, p, j, k)) & \quad \forall (m, i, p, j) \in \Gamma, k \quad (6.34)
\end{align*}
\]

The first part of platoon \((m, i, p, j)\) is served in cycle \(k\), while the remaining is assigned to be served in cycle \(k+1\). The remaining \(N_p(m, i, p, j) * (1 - c_i(m, i, p, j))\) vehicles are delayed at least one cycle. Assuming a platoon can be served in at most two cycles, a delay penalty, \(d_{pen}(m, i, p, j)\), is added into formulation to account for the delay, as follows:

\[
\begin{align*}
d_{pen}(m, i, p, j) = N_p(m, i, p, j) * (1 - c_i(m, i, p, j)) * C_r & \quad \forall (m, i, p, j) \in \Gamma \quad (6.35)
\end{align*}
\]
where $C_r$ is the reference cycle length; which represents an empirical estimation of cycle length such as average over the past several cycles. In this chapter, $C_r$ is selected to be about $60\sim150$ seconds depending on the network saturation level.

If one platoon is cut in the past cycle, the remaining vehicles in the platoon become the first queue at the stop bar. That means the remaining vehicles in the platoon will be served in the next cycle without any queue delay. If the remaining platoon has to be further cut in the next cycle, the new delay penalty will be considered in the formulation by solving another MILP based on a rolling horizon optimization approach. Therefore constraint (6.35) can roughly estimate how much delay will be generated for platoon splits.

Green times are subject to the required time to clear the platoon in selected cycle. So additional lower bounds of $g(i, p, k)$ are added in the formulation as constraint (6.36):

$$g(i, p, k) \geq T_p(m, i, p, j) \times c_i(m, i, p, j) - M(1 - \theta(m, i, p, j)) \quad \forall (m, i, p, j) \in \Gamma \quad (6.36)$$

Queue spill back sometimes occurs during oversaturated condition which can cause de facto red (Abu-Lebdeh & Benekohal 1997). De facto red is an extreme case of queue build-up due to complete blockage of upstream signals where no traffic can discharge during a green phase. To avoid de facto red and assign green times efficiently, the link storage capacity is utilized to set an upper limit on the split ratio in constraint (6.37):

$$c_i(m, i, p, j) \leq C_i(i_d, p) / N_p(m, i, p, j) \quad \forall (m, i, p, j) \in \Gamma \quad (6.37)$$

The later experiments with microscopic simulation prove the significance of constraint (6.37) under oversaturated condition. The network throughput is increased by $10\sim20\%$
and overall delay is decreased by 5~10% by including (6.36) in the MILP formulation for the oversaturated arterial.

6.3.2.3 Delay evaluation

To evaluate the total delay of a platoon, delay is divided into two categories: signal delay and queue delay. These two kinds of delay can co-exist or exist separately for each platoon (Head 1995). There are four combinations that must be considered. Platoons which travel through the intersection with free flow speed have no signal delay or queue delay. Platoons in the queue stopped at the stop bar by a red traffic light have signal delay, but no queue delay. Platoons that arrive at the back of the queue but eventually travel through the intersection during the green phase have queue delay but no signal delay, depicted in Figure 6.6 (a). Platoons stopped by a queue delayed by the traffic signal will have both queue delay and signal delay, as shown in Figure 6.6(b). Figure 6.6 (a) and (b) presents a case of two platoons requesting the same phase p. Platoon \((m,i,p,1)\) is a queue waiting at the stop bar, while platoon \((m,i,p,2)\) is a moving platoon approaching the intersection.

In Figure 6.6 (a), \(t_1\) is the time point when the leading vehicle of platoon 2 joins the platoon 1. And \(t_2\) is the time point when the leading vehicle of platoon 2 starts to discharge. \((t_2 - t_1)\) represents the leading vehicle queue delay \(d_{qi}(m,i,p,j)\) in the platoon \((m,i,p,2)\). To simplify the calculation, the average queue delay can be approximated by the leading vehicle queue delay, shown as follows:
\[d_q(m, i, p, j)/N_p(m, i, p, j) \geq t(i, p, k) + \sum_{j < j} N_p(m, i, p, j) * L_s / N_i(i, p) / V_s \]
\[- (T_a(m, i, p, j) - \sum_{j < j} N_p(m, i, p, j) * L_s / N_i(i, p) / V_p(m, i, p, j)) \]
\[- M(1 - \theta(m, i, p, 1)) \quad \forall (m, i, p, j) \in \Gamma, k \]

where \( \sum_{j < j} N_p(m, i, p, j) \) represents the total number of vehicles in front of platoon \((m, i, p, j)\). So \( \sum_{j < j} N_p(m, i, p, j) * L_s / N_i(i, p) \) is the total queue length in front of platoon \((m, i, p, j)\). \( t_2 \) can be found to be \( t(i, p, k) + \sum_{j < j} N_p(m, i, p, j) * L_s / N_i(i, p) / V_s \) and \( t_1 \) is \( T_a(m, i, p, j) - \sum_{j < j} N_p(m, i, p, j) * L_s / N_i(i, p) / V_p(m, i, p, j) \). Also the start of the phase \( p \) green time, \( t(i, p, k) \), is a decision variable determined by the first queuing platoon decision variable \( \theta(m, i, p, 1, k) \).
Figure 6.6 (a) Queue delay when two platoons are served in the same cycle; (b) Queue delay and signal delay when two platoons are served in different cycles
The queue delay can affect the arrival of a platoon; hence constraints (6.33) and (6.34) hold only if the queue delay is equal to zero. Otherwise, queue delay should be included in constraints (6.33) and (6.34) to estimate the realized arrival time of the platoon at the stop bar, as shown as (6.39) and (6.40):

\[
\frac{T_w(m,i,p,j) + d_p(m,i,p,j)}{N_p(m,i,p,j) + T_p(m,i,p,j)c_i(m,i,p,j)}
\leq t(i,p,k) + g(i,p,k) + M(1-\theta(m,i,p,j,k)) \quad \forall (m,i,p,j) \in \Gamma, k
\]  

(6.39)

\[
\frac{T_w(m,i,p,j) + d_p(m,i,p,j)}{N_p(m,i,p,j) + T_p(m,i,p,j)c_i(m,i,p,j)}
\geq t(i,p,k-1) + g(i,p,k-1) - M(1-\theta(m,i,p,j,k)) \quad \forall (m,i,p,j) \in \Gamma, k
\]  

(6.40)

Signal delay is generated by stopping the moving platoon due to a red phase rather than the leading platoon. An example of signal delay is illustrated in Figure 6.6 (b). The platoon \((m,i,p,2)\) is stopped by the discharging queue and then arrives during the red phase. The leading vehicle signal delay can be determined based on the starting time of phase \(p\) in cycle \(k+1\), \(t(i,p,k+1)\).

Delay can be calculated as the total area between the cumulative arrival curve and departure curve (Gazis 1964). It is assumed that the headways between vehicles are the same in a platoon, so the platoon flow rate can be treated as constant arrival rates, shown as Figure 6.7.
In Figure 6.7, the average arrival headway for a platoon is

\[ H_p(m, i, p, j) = \frac{\overline{T_a}(m, i, p, j) - \overline{T_a}(m, i, p, j)}{N_p(m, i, p, j)} \quad \forall (m, i, p, j) \in \Gamma \]  

(6.41)

where \( \overline{T_a}(m, i, p, j) - \overline{T_a}(m, i, p, j) \) represents the arrival time difference between the leading vehicle and trailing vehicle in the platoon. If it is a queuing platoon, equation (6.41) holds with,

\[ \overline{T_a}(m, i, p, j) = T_a(m, i, p, j) = 0 \]  

(6.42)

which makes \( H_p = 0 \) for queuing platoons.

The headway for saturation departure can be calculated as
where $H_s$ is the average headway of all the departure vehicles across all the output lanes; $H_{s1}$ is saturation flow headway in a single lane (usually 2 seconds), and $N_i(i, p)$ is number of lanes being served by phase $p$ at intersection $i$. The signal delay of the leading platoon is the summation of the vehicle delays measured between the arrival curve and departure curve in illustrated the Figure 6.7 and depicted in constraint (6.44),

$$d_s(m, i, p, j) \geq \sum_{n=1}^{N_p(m, i, p, j)} \{t(i, p, k) + (n-1)H_s(m, i, p, j) - [T_\alpha(m, i, p, j)] + (n-1)H_p(i, p)]\} - M(1 - \theta(m, i, p, j, k)) \quad \forall j = 1, (m, i, p, j) \in \Gamma, k$$

Note that the service time of the non-leading platoon $(m, i, p, j)$ is not always equal to $t(i, p, k)$. If there are any preceding platoons served in the same cycle as platoon $(m, i, p, j)$, the service time of platoon $(m, i, p, j)$ will be delayed. It is necessary to account for the the green time used to serve the preceding platoons, which is

$$\sum_{j=1}^{j-1} [\theta(m, i, p, j, k) * T_p(m, i, p, j_1)]$$

An additional constraint (6.45) is added into the MILP formulation.

$$d_s(m, i, p, j) \geq \sum_{n=1}^{N_p(m, i, p, j)} \{t(i, p, k) + \sum_{j=1}^{j=1} [\theta(m, i, p, j_1, k) * T_p(m, i, p, j_1)] + (n-1)H_s(m, i, p, j) - [T_\alpha(m, i, p, j)] + (n-1)H_p(i, p)]\} - M(1 - \theta(m, i, p, j, k)) \quad \forall j \geq 2, (m, i, p, j) \in \Gamma, k$$

6.3.3 Multi-modal dynamical progression
The state-of-practice of traffic signal control strategy is coordinated-actuated signal control. Coordinated-actuated signals can offer additional flexibility compared with fixed-time traffic signals because of their ability to respond to cycle-by-cycle variation in traffic demand while still being able to provide progression for the arterial movement. Traditionally, arterial coordination is managed through the determination of appropriate offsets, splits, and a common cycle length at each intersection. The “optimal” parameters of coordination signal plan can obtained by off-line signal optimization software, such as TRANSYT (Wallace et al. 1998), SYNCHRO (Trafficware 2009) and PASSER (Chaudhary & Chu 2003). However, the traffic flows optimized in signal optimization software are considered to be deterministic. Actually real-time traffic flow, even time-of-day traffic flow, can vary significantly (Yin 2008). Past on-line traffic-responsive (adaptive) signal control systems (Hunt et al. 1982) (Luk 1984) are able to adjust the coordination parameters (cycle length, offsets and splits) to fit the current detected traffic data, but the optimized plan assuming a constraint that every intersection must have the same cycle length may not be the best solution for every intersection on the arterial based on the real-time traffic data. It is not necessary to maintain the concept of a common cycle length as well as an offset that is constant during each cycle, since it is assumed that multi-modal traffic data is available in real-time under the assumption of vehicle-to-infrastructure communications.
In this chapter, a platoon-based dynamic coordination strategy is proposed as part of PAMSCOD to consider not only the current intersection but also the progression through downstream intersections, given the path information of each vehicle. The concept is to consider platoon queue delay and signal delay in the current intersection as well as its potential queue delay and signal delay at the next downstream intersection. Another binary variable \( \theta(m,i,p,j,c) \) is introduced into the PAMSCOD formulation to perform cycle selection at downstream intersection, \( i_d \in I_d(i,p) \), for platoon \( (m,i,p,j) \) approaching the current intersection \( i \) for phase \( p \). Again, it is assumed that only two

![Figure 6.8 Queue delay and signal delay at downstream intersection](image-url)
cycles will be considered to serve a platoon. The platoon service cycle selection constraints are shown as below,

\[
t(i, p, k) + \sum_{j=1}^{i-1} [\theta(m, i, p, j, k) * T_p(m, i, p, j, k)] + F_d * T_p(m, i, p, j) * c_i(m, i, p, j) \\
+ T(i, i_d) + T_u + d'_q(m, i, p, j) / N_p(m, i, p, j) \leq t(i_d, p, c) + g(i_d, p, c) \\
+ M(2 - \theta(m, i, p, j, k) - \theta'(m, i, p, j, c)) \quad \forall (m, i, p, j) \in \Gamma_2 \cup \Gamma_6, k, c
\]  

(6.46)

\[
t(i, p, k) + \sum_{j=1}^{i-1} [\theta(m, i, p, j, k) * T_p(m, i, p, j, k)] + F_d * T_p(m, i, p, j) * c_i(m, i, p, j) \\
+ T(i, i_d) + T_u + d'_q(m, i, p, j) / N_p(m, i, p, j) \geq t(i_d, p, c - 1) + g(i_d, p, c - 1) \\
- M(2 - \theta(m, i, p, j, k) - \theta'(m, i, p, j, c)) \quad \forall (m, i, p, j) \in \Gamma_2 \cup \Gamma_6, k, c
\]  

(6.47)

\[
\sum_{c} \theta'(m, i, p, j, c) = 1 \quad \forall (m, i, p, j) \in \Gamma_2 \cup \Gamma_6
\]  

(6.48)

Constraints (6.46) and (6.47) are quite similar to constraints (6.30) and (6.31). One difference is that platoon dispersion is considered in both constraints to ensure enough phase green time is allocated at the downstream intersection to accommodate a platoon that is arriving at the upstream signal. The other difference is that (6.46) and (6.47) go into effect when both \( \theta \) and \( \theta' \) are equal to 1, meaning that the departure and arrival occurs in cycle \( k \) and cycle \( c \), at intersection \( i \) and \( i_d \), respectively.

Queue delay and signal delay are evaluated base on the time-space diagram construction, as depicted in Figure 6.8. Another two constraints are included in the formulation to account for downstream queue delay and signal delay as follows:

\[
d'_q(m, i, p, j) / N_p(m, i, p, j) \geq t(i_d, p, c) + \sum_{j=1}^{i-1} N_p(m, i_d, p, j) * L_s / N_l(i_d, p) / V_s \\
- \{t(i, p, k) + [L(i, i_d) - \sum_{j=1}^{i-1} N_p(m, i_d, p, j) * L_s / N_l(i_d, p)] / V_p(m, i, p, j) \} \\
- M(2 - \theta(m, i_d, p, l, c) - \theta(m, i, p, j, k)) \quad \forall (m, i, p, j) \in \Gamma_2 \cup \Gamma_6, k, c
\]  

(6.49)
\[ d'_{x}(m, i, p, j) \geq P_r(m, i, i_d) \sum_{n=1}^{N_{d}(m, i, p, j)} [t(i_d, p, c) + \sum_{n=1}^{\infty} [\theta(m, i_d, p, j_1, c) \cdot T_{x}(m, i_d, p, j_1)]] + (n-1)H_{x}(m, i, p, j) - [t(i, p, k) + T_{x}(m, i, i_d) + d'_{x}(m, i, p, j)]/N_{p}(m, i, p, j) + T_{s} + (n-1)H_{p}(i, p) - M(2 - \theta(m, i, p, j, k) - \theta'(m, i, p, j, c)) \quad \forall (m, i, p, j) \in \Gamma_2 \bigcup \Gamma_6, k, c \]

In constraint (6.49), queue delay is assessed by determining the departure cycle \( k \) at intersection \( i \) and shockwave starting time from cycle \( c \) at downstream intersection \( i_d \).

The platoon departure cycle is controlled by binary variable \( \theta(m, i, p, j, k) \), while the shockwave starting time is controlled by binary variable \( \theta(m, i_d, p, j_1, c) \), denoting the first platoon to arrive at the downstream intersection is served in cycle \( c \).

In constrain (6.50), signal delay is addressed by choosing the departure cycle \( k \) at intersection \( i \) and then the departure cycle \( c \) at downstream intersection \( i_d \), controlled by binary variables \( \theta(m, i, p, j, k) \) and \( \theta'(m, i_d, p, j, c) \), respectively. \( P_r(m, i, i_d) \in [0, 1] \) represents the percentage of the remaining vehicles in the platoon after traveling through link \((i, i_d)\).

6.3.4 Formulation enhancement

So far the major part of formulation has been presented in the last few sections. The number of integer variables equal to \( 2|M||I||P||J| \), which increase proportionally with the number of modes (\( M \)) considered, the number of the intersections (\( I \)), the number of phases (\( P \)), and the average number of platoons (\( J \)) for each phase and intersection pair. Therefore, the feasible region of binary decision variables \( \theta \) and \( \theta' \) grows quickly with the network size and traffic condition (undersaturated, saturated). It is desired that this
formulation could solve online traffic signal control problems. Thus the solution time
plays an important role in the. To reduce the solution time, any reduction in the size of
the feasible region will help the solver to decrease the time searching undesired solutions.
In this Section, the formulation can be enhanced by adding a first-come first-serve (FCFS)
rule.

6.3.4.1 First-come first-serve (FCFS) rule

The first-come first-serve rule is a widely accepted rule in queuing theory (Gross et al.
2008). This rule may not hold for different travel modes due to different priority levels.
For example, all other vehicles need to pull over when an emergency vehicle is
approaching. However, the first-come first-serve rule is applicable in the same travel
mode for the same phase. So the preceding platoon \((m,i,p,j-1)\) should always be served
before platoon \((m,i,p,j)\). With the assumption of first-come first-serve, the precedence
relationship is established between adjacent platoons that are served by the same phase
and intersection. The feasible region of integer variables \(\theta\) and \(\theta'\) can be enhanced by
addressing first-come first-serve. Constraint (6.51) and (6.52) presents the first-come first
serve rule for the same intersection. If \(\theta(m,i,p,j,k) = 1\), the successive platoons can
only be served in cycle \(k\) or later than cycle \(k\).

\[
\theta(m,i,p,j,k) \leq \sum_{c=k}^{k} \theta(m,i,p,j+1) \quad \forall (m,i,p,j) \in \Gamma
\]  
(6.51)

\[
\theta'(m,i,p,j,k) \leq \sum_{c=k}^{k} \theta'(m,i,p,j+1) \quad \forall (m,i,p,j) \in \Gamma_2 \cup \Gamma_6
\]  
(6.52)
where $\sum_{c=k}^{[K]} \theta(m, i, p, j + 1)$ and $\sum_{c=k}^{[K]} \theta'(m, i, p, j + 1)$ enhance the feasible region to constraint (6.32) and (6.47), respectively. Constraint (6.53) presents the first-come first serve rule for the consecutive intersections.

Any platoon at a downstream intersection must be served before any platoon from the upstream intersection can be served.

$$\theta(m, i_d, p, j, k) \leq \sum_{c=k}^{[K]} \theta'(m, i, p, j, c) \quad \forall (m, i, p, j) \in \Gamma_2 \bigcup \Gamma_6, i_d \in I_d(i, p)$$  \hspace{1cm} (6.53)

These constraints are included in the PAMSCOD formulation below to improve the computational efficiency of the model.

### 6.3.5 Objective and summary of PAMSCOD

Usually the objective of traffic signal control algorithms is to minimize the disutility function, such as travel time, delay, number of stops, or maximize the utility function, such as network throughput. The proposed formulation aims to serve all the platoons in $|K|$ cycles using a rolling horizon approach. Hence, the throughput cannot be evaluated in the formulation. The number of stops is not computable due to the lack of an exact traffic flow model in the formulation. However, the total delay can be approximately assessed by constraints in derived in the previous sections.

The objective function will be to minimize the total weighted delay as well as the sum of slack variables, total green rest time in equation (6.53). The delay weight factor
$W(m,i,p,j)$ can be set to different values for each mode as well as each different platoon.

Weight factors can depend on the priority level of the mode and can be adjusted for individual vehicle according to other real-time information, such as occupancy. It is assumed that emergency vehicle will receive a very high level of priority and can be considered in a complementary and separate formulation.

The summary of formulation is depicted as follows.

Objective: Minimize

$$
\sum_{(m,i,p,j) \in t} W(m,i,p,j)[d_s(m,i,p,j) + d_q(m,i,p,j) + \alpha \sum_{(i,p,k)} s(i,p,k) (6.54)]
$$

Subject to

Precedence constraints: (6.4)-(6.26)

Selection constraints at current intersections: (6.32), (6.36), (6.39), (6.40)

Delay evaluation at current intersections: (6.35), (6.38), (6.44), (6.45)

Link capacity constraints: (6.37)

Selection constraints at downstream intersections: (6.46), (6.47), (6.48)

Delay evaluation at downstream intersections: (6.49), (6.50)

Formulation enhancement constraints: (6.51)-(6.53)

Variables $\theta$ and $\varphi$ are binary decision variables and all other variables are nonnegative.
The number of integer variables in PAMSCOD is equal to $2|M||I||P||J|$, where 2 means two integer variables $\theta$ and $\varphi$, $|M|$ is number of travel modes considered, $|I|$ is number of intersections controlled, $|P|$ is number of phases, and $|J|$ is the number of existing platoon. The solution times of this formulation vary depending on the traffic saturation condition. The formulation was tested with GAMS/CPLEX 10.1 on a PC with dual core 2.67G Hz with 3.5G memory. For uncongested traffic conditions with no residual queue, the solution times are less than 1 seconds in most of the time. However, for congested traffic condition with large residual queues, the solution times can extend from few seconds up to 1 minute to achieve an optimality gap less than 10%.

The solution times will have a significant impact on the implementation of an algorithm such as PAMSCOD in the field. In future research, either a simplified formulation or a heuristic algorithm needs to be developed for real-time applications.

6.4 Design of experiment

Two most importation travel modes are considered in the experiments: automobiles and buses. Due to some limitations of implement IntelliDriveSM for bicycles and pedestrians at starting stage, non-motorized travel modes are not included in the tests. Other motorized travel modes can be added in future effortlessly with the same proposed formulation.

The algorithm test configuration is the same as the one in Chapter 4. The numerical experiments were conducted in VISSIM, a microscopic simulation tool. To better simulate the real traffic signal controller, the ASC/3 SIL (software in the loop) was
installed with VISSIM, which is a version of the ASC/3 software that can become a virtual controller as part of VISSIM. The ASC/3 SIL feature allows a VISSIM user to circumvent the issue of configuring a physical ASC/3 controller to run during the simulation, including transit signal priority (TSP) (Econolite 2009).

The entire evaluation platform contains VISSIM with COM (Component Object Model) and ASC virtual controller as simulation environment and GAMS/CPLEX as solver, depicted in Figure 6.9. Simulation in VISSIM can be easily controlled in COM, which can be created with a variety of programming languages, like C++. First, COM runs VISSIM model and keeps reading vehicle data from VISSIM. When there is no priority request, actuated control is conducted by ASC/3 SIL. Once a priority request is detected, a MILP program will be formulated by COM and sent to GAMS. After retrieving optimal signal plan from GAMS, COM implements the plan by sending phase control commands to ASC/3 SIL, such as phase hold and force-off.

![Figure 6.9 Evaluation platform](image-url)
Numerical experiment with eight-intersection arterial was carried out on Speedway Blvd. in Tucson, AZ, starting from Euclid Ave. to Alveron Way. Since the last intersection is 1600 meters from seventh intersection, the coordination factor is very low between last two intersections. So the last intersection can be treated as an isolated intersection, which is excluded in PAMSCOD online optimization. Ten conflicting bus routes were added in the model shown as Figure 6.10, which are real bus routes operated by Suntran (Suntran 2010). Two out of ten bus routes travel on the arterial, while others travel on the side streets. All bus stops are far-ended from approaching intersections. Bus frequencies are set to 5–10 minutes. It is assumed that each bus is behind schedule. So every bus sends priority request when it approaches intersection.

6.4.1 Illustration of solutions from PAMSCOD

Signal-timing optimization software such as TRANSYT-7F and SYNCHRO are commonly used as benchmarks for good arterial signal timing. In this chapter, performances of PAMSCOD are compared with SYNCHRO under different level of flows. Seven different deterministic flows are designed as the basic flow levels of experiment. They corresponds to different levels of intersection saturation rate 0.3, 0.6, 0.8, 0.9, 1.0, 1.1 and 1.2, estimated by intersection capacity utilization in SYNCHRO (Husch & Albeck 2003). The “optimal” solutions obtained from SYNCHRO can be considered as near-optimal solutions for deterministic flows, since the input flows of VISSIM are the same as input flows of SYNCHRO. Four sets of stochastic volumes are generated by normal distribution, using deterministic flows as mean and 20~30% of
deterministic flow as variance, shown in Table 6.2. The random flow sets are selected to have the similar total volumes.

First the output cycle lengths of PAMSCOD are compared with SYNCHRO’s optimal cycle length. Average cycle lengths with seven volume levels are recorded at two most saturated intersections: Euclid & Speedway and Campbell & Speedway. Figure 6.11 (a) and (6) plots the curves of cycle length with PAMSCOD and SYNCHRO at these two intersection respectively. The results show that PAMSCOD’s cycle lengths are little lower than SYNCHRO’s optimized cycle length, but PAMSCOD’s cycle lengths follow the tendency that arterial cycle length in proportion to flow levels. And previous study shows that cycle lengths in a small range may result in similar bottom delay (Miller 1963). Therefore the output cycle lengths of PAMSCOD are reasonable to be understood.
Figure 6.10 Speedway arterial in Tucson, AZ.
<table>
<thead>
<tr>
<th>SR*</th>
<th>Stochastic volume set 1</th>
<th>Stochastic volume set 2</th>
<th>Stochastic volume set 3</th>
<th>Stochastic volume set 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spdwy_EB</td>
<td>542 1028</td>
<td>1154 1538</td>
<td>1623 2353</td>
<td>3028 4590</td>
</tr>
<tr>
<td>Euclid_NB</td>
<td>210 534</td>
<td>582 1121</td>
<td>1032 1685</td>
<td>1747 425</td>
</tr>
<tr>
<td>Euclid_SB</td>
<td>290 421</td>
<td>722 1036</td>
<td>1822 1881</td>
<td>1533 375</td>
</tr>
<tr>
<td>Park_NB</td>
<td>77 183</td>
<td>396 363</td>
<td>400 471</td>
<td>703 171</td>
</tr>
<tr>
<td>Park_SB</td>
<td>109 249</td>
<td>245 722</td>
<td>482 602</td>
<td>616 178</td>
</tr>
<tr>
<td>Moutn_NB</td>
<td>118 172</td>
<td>230 443</td>
<td>509 513</td>
<td>674 519</td>
</tr>
<tr>
<td>Moutn_SB</td>
<td>120 296</td>
<td>347 277</td>
<td>892 759</td>
<td>665 201</td>
</tr>
<tr>
<td>Cherry_NB</td>
<td>105 272</td>
<td>341 710</td>
<td>594 702</td>
<td>563 154</td>
</tr>
<tr>
<td>Cherry_SB</td>
<td>154 175</td>
<td>399 509</td>
<td>392 572</td>
<td>729 120</td>
</tr>
<tr>
<td>Camb’l_NB</td>
<td>201 847</td>
<td>1134 1222</td>
<td>1524 2247</td>
<td>3442 178</td>
</tr>
<tr>
<td>Camb’l_SB</td>
<td>519 592</td>
<td>1318 2182</td>
<td>3498 2149</td>
<td>2441 847</td>
</tr>
<tr>
<td>Tucson_NB</td>
<td>178 95</td>
<td>365 773</td>
<td>1107 913</td>
<td>756 1073</td>
</tr>
<tr>
<td>Tucson_SB</td>
<td>171 363</td>
<td>410 1043</td>
<td>732 961</td>
<td>754 201</td>
</tr>
<tr>
<td>Country_NB</td>
<td>329 473</td>
<td>1227 1165</td>
<td>873 1494</td>
<td>1693 459</td>
</tr>
<tr>
<td>Country_SB</td>
<td>283 615</td>
<td>687 1073</td>
<td>1310 1152</td>
<td>2198 842</td>
</tr>
<tr>
<td>Alverno_NB</td>
<td>385 685</td>
<td>572 833</td>
<td>1505 1807</td>
<td>1267 429</td>
</tr>
<tr>
<td>Alverno_SB</td>
<td>375 644</td>
<td>1095 1735</td>
<td>1748 2707</td>
<td>1523 429</td>
</tr>
<tr>
<td>Spdwy_WB</td>
<td>425 643</td>
<td>1006 1832</td>
<td>1905 2677</td>
<td>2338 429</td>
</tr>
<tr>
<td>Total volume</td>
<td>4590 8286</td>
<td>12231 17946</td>
<td>22258 25479</td>
<td>26877 5689750 14746 16787 20534 22700 27054</td>
</tr>
</tbody>
</table>

*: approximate saturation rate
Figure 6.11 Comparisons between optimized cycle length from SYNCHRO and average cycle length in PAMSCOD at two critical intersections. (a) Intersection at Euclid & Speedway; (b) Intersection at Campbell & Speedway.
Figure 6.12 Online optimized signal plan from PAMSCOD when SR=0.9 in set 3 (intersection 1 is in the bottom)

Table 6.3 Performance comparisons with four methods. (Th. = throughput; Dv, Db = average delay on each phase for all vehicles and buses respectively; Sv, Sb = average number of stops on each phase for all vehicles and buses respectively)
<table>
<thead>
<tr>
<th>set</th>
<th>SR</th>
<th>demand</th>
<th>ASC free</th>
<th>ASC coord</th>
<th>Arterial</th>
<th>TSP coord</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>9064</td>
<td>4600.50</td>
<td>17.50</td>
<td>14.82</td>
<td>21.48</td>
</tr>
<tr>
<td>1</td>
<td>0.9</td>
<td>18128</td>
<td>17997.00</td>
<td>22.19</td>
<td>21.90</td>
<td>21.90</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>16785.7</td>
<td>16724.50</td>
<td>37.17</td>
<td>31.70</td>
<td>24.80</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>20533.1</td>
<td>18713.00</td>
<td>60.75</td>
<td>53.30</td>
<td>53.30</td>
</tr>
<tr>
<td>3</td>
<td>1.1</td>
<td>22985.1</td>
<td>19840.50</td>
<td>80.94</td>
<td>51.78</td>
<td>47.60</td>
</tr>
<tr>
<td>1</td>
<td>1.2</td>
<td>25753.2</td>
<td>20876.50</td>
<td>98.51</td>
<td>75.15</td>
<td>61.66</td>
</tr>
<tr>
<td>1</td>
<td>0.3</td>
<td>4567.926</td>
<td>4613.50</td>
<td>10.05</td>
<td>15.97</td>
<td>21.50</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td>9749.1</td>
<td>9830.00</td>
<td>17.45</td>
<td>18.20</td>
<td>17.25</td>
</tr>
<tr>
<td>1</td>
<td>0.9</td>
<td>14745</td>
<td>14825.00</td>
<td>28.73</td>
<td>29.77</td>
<td>24.84</td>
</tr>
<tr>
<td>2</td>
<td>0.9</td>
<td>16857.5</td>
<td>16724.50</td>
<td>37.25</td>
<td>40.98</td>
<td>24.84</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>20533.1</td>
<td>18713.00</td>
<td>60.75</td>
<td>62.14</td>
<td>41.71</td>
</tr>
<tr>
<td>2</td>
<td>1.1</td>
<td>22985.1</td>
<td>19840.50</td>
<td>80.94</td>
<td>58.10</td>
<td>47.60</td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>27053.2</td>
<td>20876.50</td>
<td>98.51</td>
<td>75.15</td>
<td>61.66</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>4655.437</td>
<td>4690.50</td>
<td>11.29</td>
<td>16.20</td>
<td>22.00</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>8845.152</td>
<td>8990.50</td>
<td>16.23</td>
<td>17.01</td>
<td>16.58</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
<td>12214.4</td>
<td>12222.00</td>
<td>22.35</td>
<td>22.89</td>
<td>23.64</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>18835.93</td>
<td>18152.00</td>
<td>44.46</td>
<td>41.71</td>
<td>34.96</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>20533.1</td>
<td>18691.00</td>
<td>54.38</td>
<td>61.48</td>
<td>41.46</td>
</tr>
<tr>
<td>3</td>
<td>1.1</td>
<td>24491.45</td>
<td>19022.50</td>
<td>69.14</td>
<td>73.90</td>
<td>42.52</td>
</tr>
<tr>
<td>3</td>
<td>1.2</td>
<td>25057.84</td>
<td>20543.00</td>
<td>89.66</td>
<td>99.10</td>
<td>62.84</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>4717.146</td>
<td>4744.00</td>
<td>11.34</td>
<td>20.18</td>
<td>22.14</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>19660.2</td>
<td>18579.00</td>
<td>49.35</td>
<td>52.20</td>
<td>40.41</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>21833.41</td>
<td>18793.00</td>
<td>57.45</td>
<td>58.71</td>
<td>43.42</td>
</tr>
<tr>
<td>4</td>
<td>1.1</td>
<td>24755.01</td>
<td>20311.00</td>
<td>67.81</td>
<td>74.55</td>
<td>49.85</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>25853.04</td>
<td>20582.50</td>
<td>82.64</td>
<td>98.88</td>
<td>58.39</td>
</tr>
<tr>
<td>1</td>
<td>0.3</td>
<td>4532</td>
<td>4600.50</td>
<td>11.18</td>
<td>10.66</td>
<td>5.54</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td>9064</td>
<td>9112.50</td>
<td>16.02</td>
<td>16.29</td>
<td>5.90</td>
</tr>
<tr>
<td>1</td>
<td>0.9</td>
<td>18128</td>
<td>17997.00</td>
<td>42.63</td>
<td>44.73</td>
<td>7.77</td>
</tr>
<tr>
<td>1</td>
<td>1.2</td>
<td>18947.50</td>
<td>38.41</td>
<td>40.52</td>
<td>0.73</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>22257.3</td>
<td>19385.00</td>
<td>57.73</td>
<td>56.65</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>25478.1</td>
<td>20798.00</td>
<td>94.51</td>
<td>101.70</td>
<td>1.39</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.3</td>
<td>4567.926</td>
<td>4613.50</td>
<td>10.05</td>
<td>11.55</td>
<td>2.95</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>9749.1</td>
<td>9830.00</td>
<td>17.45</td>
<td>18.20</td>
<td>2.95</td>
</tr>
<tr>
<td>2</td>
<td>0.9</td>
<td>14745</td>
<td>14825.00</td>
<td>28.73</td>
<td>29.77</td>
<td>2.95</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>16724.50</td>
<td>37.25</td>
<td>40.98</td>
<td>2.95</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.1</td>
<td>18713.00</td>
<td>60.75</td>
<td>62.14</td>
<td>2.95</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.2</td>
<td>19840.50</td>
<td>80.94</td>
<td>83.10</td>
<td>2.95</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>4655.437</td>
<td>4690.50</td>
<td>11.29</td>
<td>11.55</td>
<td>2.95</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>8845.152</td>
<td>8990.50</td>
<td>16.23</td>
<td>17.01</td>
<td>2.95</td>
</tr>
<tr>
<td>3</td>
<td>0.8</td>
<td>12214.4</td>
<td>12222.00</td>
<td>22.35</td>
<td>22.89</td>
<td>2.95</td>
</tr>
<tr>
<td>3</td>
<td>0.9</td>
<td>18835.93</td>
<td>18152.00</td>
<td>44.46</td>
<td>41.71</td>
<td>2.95</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>20533.1</td>
<td>18691.00</td>
<td>54.38</td>
<td>61.48</td>
<td>2.95</td>
</tr>
<tr>
<td>3</td>
<td>1.1</td>
<td>24491.45</td>
<td>19022.50</td>
<td>69.14</td>
<td>73.90</td>
<td>2.95</td>
</tr>
<tr>
<td>3</td>
<td>1.2</td>
<td>25057.84</td>
<td>20543.00</td>
<td>89.66</td>
<td>99.10</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>0.3</td>
<td>4717.146</td>
<td>4744.00</td>
<td>11.34</td>
<td>11.55</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>9178.99</td>
<td>9196.50</td>
<td>16.17</td>
<td>16.74</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>0.8</td>
<td>12428</td>
<td>12508.00</td>
<td>20.44</td>
<td>21.04</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>19660.2</td>
<td>18579.00</td>
<td>49.35</td>
<td>52.20</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>21833.41</td>
<td>18793.00</td>
<td>57.45</td>
<td>58.71</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>1.1</td>
<td>24755.01</td>
<td>20311.00</td>
<td>67.81</td>
<td>74.55</td>
<td>2.95</td>
</tr>
<tr>
<td>4</td>
<td>1.2</td>
<td>25853.04</td>
<td>20582.50</td>
<td>82.64</td>
<td>98.88</td>
<td>2.95</td>
</tr>
</tbody>
</table>
Figure 6.13 (a) Network throughput; (b) Average all vehicle delay on each phase at each intersection. (c) Average bus delay on each phase at each intersection.
Second the output signal plans of PAMSCOD are visualized on the space-time diagram, depicted as Figure 6.12. From bottom to up, the number of intersection increases from 1 to 7. It is obvious to find a two-stage two-way progression on the diagram. One two-way progression is from intersection 1 (Euclid & Speedway) to intersection 5 (Campbell & Speedway). The other progression two-way progression is from intersection 5 (Campbell & Speedway) to intersection 7 (Country club & Speedway). The results are expected since intersection 5 (Campbell & Speedway) is near saturated (saturation rate is 0.9) and has very larger volume on northbound (2742 veh/hr) & southbound (2296 veh/hr) on Campbell than eastbound (1489 veh/hr) and westbound (1262 veh/hr) on Speedway. PAMSCOD outputs smaller eastbound & westbound green time than northbound & westbound, which seems very reasonable. Therefore, the volume pattern of intersection 5 (Campbell & Speedway) is detected by PAMSCOD correctly, and good splits in PAMSCOD have been achieved.

In addition to progression, a large network can be partitioned into sub-network according to the time-space diagram from PAMSCOD. Network partition can reduce the total number of concurrently considered intersections as well as the MILP solution times a lot. If the solutions in a sub-network show a good two-way progression under high volume experiments, this sub-network can be treated as an isolated network. The details algorithms of network partition are considered as our future topics.
6.4.2 Experiment results

Finally four different methods are compared within four volume sets. “ASC Free” treats each intersection separately and applies actuated control in each isolated intersection without bus priority. “ASC Coord” represents the coordinated-actuated non-priority signal control based on the optimal signal timing plans obtained from SYNCHRO 7.0. “TSP Coord” only adds transit signal priority (TSP) on “ASC coord”. The detailed settings of TSP in ASC/3 SIL please refer to (Zlatkovic et al. 2010) and (Econolite 2009).

Table 6.3 presents the detailed results for a 10 minute warm-up period and one-hour simulation in VISSIM. Performances among four different methods are compared in network throughput, average all vehicle delay and number of stops for each phase at each intersection, average bus delay and number of stops per phase intersection pair. Further, the polynomial regression lines are plot in Figure 6.13 (a)-(c) for four methods based on the data pairs of demand-throughput, demand-average vehicle delay and demand-average bus delay. Performance improvements of PAMSCOD are summarized in Table 6.4, compared with other methods. Several observations are listed as follows:

1. When the saturation rate is lower than 0.9, all the vehicles are served within four methods. However, vehicles are partially served when the saturation rate is near or higher than 0.9. The trend line in Figure 6.13 (a) as well as data in Table 6.4 shows that PAMSCOD has always highest through put among all the methods, especially when the traffic condition is oversaturated. When considering multi-modes (buss and automobiles)
in coordination, “TSP Coord” has the lowest throughput among all the methods for congested traffic conditions.

2. When the saturation rate is lower than 0.6, “ASC Free” always is the best method to minimized the total vehicle delay, which is acknowledged by most of researchers. However, “ASC Free” gets worse and “ASC Coord” gets better when the saturation rate increases higher than 0.6. Although “ASC Coord” can be treated as near optimal method, PAMSCOD outperforms “ASC Coord” in average all vehicle delay by 5%, shown in Table 6.4. Since “TSP Coord” put absolute priority for buses, the average all vehicle delay of PAMSCOD output as 38% less average all vehicle delay as “TSP Coord”.

3. With regard to average bus delay, the bus delay of “TSP Coord” can be treated as the lower bound, since it provides priority to any bus approaching intersection, regardless of other passenger vehicles’ delay or throughput. PAMSCOD yields only 2% more bus delay than “TSP coord”, but 7% more network throughput as well as 38% less all vehicle delay.

According to above observations, PAMSCOD outputs the lower bound of all vehicle delay, compared with “ASC Coord”, optimized by SYNCHRO with deterministic flows. And PAMSCOD outputs the near lower bound of bus delay, compared with “TSC Coord”, considering absolute bus priority on both directions.
Table 6.4 Performance improvements of PAMSCOD compared with other methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>0.3</th>
<th>0.6</th>
<th>0.8</th>
<th>0.9</th>
<th>1</th>
<th>1.1</th>
<th>1.2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Through put</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASC Free</td>
<td>0.10%</td>
<td>0.04%</td>
<td>0.17%</td>
<td>2.13%</td>
<td>3.86%</td>
<td>3.36%</td>
<td>0.32%</td>
<td>1.43%</td>
</tr>
<tr>
<td>ASC Coord</td>
<td>-0.01%</td>
<td>0.00%</td>
<td>1.59%</td>
<td>6.81%</td>
<td>6.79%</td>
<td>5.39%</td>
<td>1.06%</td>
<td>3.09%</td>
</tr>
<tr>
<td>TSP Coord</td>
<td>0.11%</td>
<td>0.21%</td>
<td>2.54%</td>
<td>13.86%</td>
<td>11.68%</td>
<td>13.14%</td>
<td>7.92%</td>
<td>7.07%</td>
</tr>
<tr>
<td>Avg. all delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASC Free</td>
<td>-28.85%</td>
<td>-6.63%</td>
<td>10.33%</td>
<td>32.61%</td>
<td>36.28%</td>
<td>38.13%</td>
<td>46.13%</td>
<td>18.28%</td>
</tr>
<tr>
<td>ASC Coord</td>
<td>11.95%</td>
<td>3.54%</td>
<td>6.88%</td>
<td>12.35%</td>
<td>1.79%</td>
<td>-0.56%</td>
<td>2.11%</td>
<td>5.44%</td>
</tr>
<tr>
<td>TSP Coord</td>
<td>37.36%</td>
<td>42.23%</td>
<td>60.12%</td>
<td>63.44%</td>
<td>38.50%</td>
<td>19.72%</td>
<td>8.32%</td>
<td>38.53%</td>
</tr>
<tr>
<td>Avg. bus delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASC Free</td>
<td>-21.95%</td>
<td>9.03%</td>
<td>21.16%</td>
<td>61.66%</td>
<td>91.63%</td>
<td>79.27%</td>
<td>93.62%</td>
<td>47.77%</td>
</tr>
<tr>
<td>ASC Coord</td>
<td>52.82%</td>
<td>38.99%</td>
<td>29.63%</td>
<td>37.41%</td>
<td>42.58%</td>
<td>31.64%</td>
<td>23.75%</td>
<td>36.69%</td>
</tr>
<tr>
<td>TSP Coord</td>
<td>-2.61%</td>
<td>-7.12%</td>
<td>-11.44%</td>
<td>12.34%</td>
<td>10.49%</td>
<td>-11.97%</td>
<td>-10.41%</td>
<td>-2.96%</td>
</tr>
</tbody>
</table>

6.5 Summary

A unified platoon-based formulation called PAMSCOD is presented in this chapter to optimize arterial (network) traffic signals for concurrent different travel modes, given the assumption that advanced communication systems are available between riders and traffic controllers.

First, a hierarchical platoon recognition algorithm is proposed to identify platoons in real-time. This algorithm can output desired number of platoons.

Second, a mixed-integer linear program (MILP) is solved online for future optimal signal plans by feeding the real-time arterial request platoon data and traffic controller status. Different with traditional common network cycle length, PAMSCOD aims to provide multi-modal dynamical progression on arterial based on the real-time platoon information. Furthermore, the integer feasible solution region is enhanced a lot by assuming first-come first-serve to the requests on the same approach, in order to reduce the solution times.
Microscopic online simulation shows that PAMSCOD can handle two traffic modes including buses and automobiles jointly and reduce a significant delay on both modes.

Future research can focus on how to reduce the solution times of PAMSCOD by considering network partition, reducing the complexity of the MILP or developing heuristic algorithms. Another issue is that high penetration of IntelliDrive\textsuperscript{SM} is assumed in this work. Further low penetration can affect the accuracy of platoon recognition. A platoon recognition algorithm needs to be provided to fuse data from traditional point detectors and V2X communications. However, PAMSCOD still works given the platoon information and traffic controller settings.
CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Summary of the Research

Modern traffic control systems have not changed significantly in the past 40-50 years. The most widely applied traffic signal control systems are still time-of-day coordinated-actuated system, since many existing advanced adaptive signal control systems are complicated and fathomless for most agencies. With the advent of IntelliDriveSM for Mobility, soon it will be possible to obtain additional network and vehicle real-time information. Given enriched data from IntelliDriveSM, the mechanism of traffic control could be transformed to produce a robust and optimal output with clear mathematical formulations. The major contributions were three-fold in this doctoral research:

1. A pseudo-lane-level GPS positioning system is developed based on IntelliDriveSM v2x structure. Statistical methods as well as mathematical optimization are applied to track driving events and reduce GPS errors.

2. An analytical mathematical model is developed for robust real-time multiple priority control, assuming penetration of IntelliDriveSM is limited to emergency vehicles and transit vehicles. This model accommodates advanced features of modern traffic controller, such as green extension and vehicle actuations. The main idea of proposed model is re-developed with a heuristic algorithm and applied on a real intersection at Southern Ave. & 67 Ave. Phoenix, AZ.
3. A unified platoon-based mathematical formulation called PAMSCOD is presented to optimize arterial traffic signals for concurrent multiple travel modes in the environment of IntelliDrive\textsuperscript{SM} with high penetration including passenger vehicles.

7.1.1 Pseudo lane-level GPS positioning in a v2x environment

A framework is presented in this research to obtain pseudo lane-level positioning using low-cost GPS, vehicle-to-infrastructure (v2i) communication, and driving event detection. In this context, pseudo means that lane level accuracy is achieved only under the assumption that v2i is available and there is no GPS outage.

GPS errors can be categorized into common-mode errors and noncommon-mode errors, where common-mode errors can be mitigated by differential GPS (DGPS) but noncommon-mode cannot. First, common-mode GPS error is cancelled using differential corrections broadcast from the road-side equipment (RSE). With v2i communication, a high fidelity roadway layout map and satellite pseudo-range corrections are broadcast by the RSE. The on-board equipment (OBE) corrects for the GPS common-mode errors based on the received pseudo-range corrections from the RSE, the current lane estimate, and the segment status determined by a general map matching algorithms.

To enhance and correct the lane level positioning, a statistical process control approach is used to detect significant vehicle driving events such as turning at an intersection or lane-changing. Whenever a turn event is detected, a mathematical program is solved to estimate and update the GPS noncommon-mode errors. This work does not
consider vehicle sensor data which could be used to improve position estimates (but requires an interface to the vehicle electronic system) and it is assumed that there is no GPS outage. Next Generation Simulation (NGSIM) data is used to validate driving behavior for turn movements and to calibrate the lane-changing detection model. A field experiment is conducted to validate the positioning models. The total lane changing detection rate is about 93%. All of the turns were detected because heading errors are significant and easy to detect. There was no observed error for turn detection.

7.1.2 Multiple priority control and field implementation in a v2x environment

This study examines priority based traffic signal control using vehicle-to-infrastructure communication to send priority requests from a vehicle to an intersection. Priority control allows certain classes of vehicles, such as emergency vehicles or buses, to receive preferential treatment at a traffic signal. At any time it is possible that more than one qualified vehicle is approaching an intersection and each vehicle may send a priority request. The traffic control algorithm must consider multiple requests as well as be robust to uncertainty in the desired service time.

Given the current information from multiple priority requests, a mixed integer linear mathematical program (MILP) is solved for each intersection to obtain an optimal signal timing plan. This research first presents a deterministic MILP that is applicable for control of multiple emergency vehicles. Then a robust MILP is developed for transit vehicles (e.g. buses) where the desired service time might be uncertain due to operational factors such queuing and passenger boarding/alighting. The solution to the robust control
problem is integrated with actuated traffic signal control to be responsive to real-time non-priority vehicle demand.

Finally, coordination between adjacent signals is achieved by adding coordination requests along with other priority requests. Due to the limited number of binary variables in the formulation, the robust MILP can be implemented in real-time signal control. The new approach is compared with state-of-practice coordinated-actuated traffic signal control with transit signal priority (TSP) under several operating scenarios using microscopic traffic simulation. The simulation experiments show that the priority based traffic signal control is able to reduce transit delay by 18% and all vehicle delay by 3%. For “solve free” field implementation, a heuristic algorithm is developed to run on an embedded Linux operating system. The multiple priority control problem is simplified to a polynomial solvable cut problem by adding few assumptions. Each cut combination corresponds to a unique serving sequence of multiple priority requests. First, the cut problem is proved as a polynomial solvable problem. Second, a revised exhaustive search algorithm is proposed to look for the good solutions in the tolerance range. The total priority delay of each cut combination is assessed by phase-time diagram. Finally both microscopic simulations with VISSIM and a field test on a real intersection at Phoenix, AZ confirm the effectiveness of proposed algorithm.

7.1.3 Multi-modal traffic control in a v2x environment
A unified platoon-based formulation called PAMSCOD is presented in this research to optimize arterial (network) traffic signals for concurrent different travel modes, given the assumption of high penetration of IntelliDrive\textsuperscript{SM} is available.

First, a hierarchical platoon recognition algorithm is proposed to identify platoons in real-time. This algorithm can output the number of platoons on the approach to an intersection. Second, a mixed-integer linear program (MILP) is solved for future optimal signal plans by feeding the arterial request platoon data and traffic controller status. Different that traditional common network cycle length, PAMSCOD aims to provide multi-modal dynamical progression on arterial based on the real-time platoon information. Furthermore, the integer feasible solution region is enhanced in order to reduce the solution times by assuming a first-come, first-serve discipline for the requests on the same approach.

Microscopic online simulation shows that PAMSCOD can easily handle two traffic modes including buses and automobiles and can significantly reduce the delay of both modes. Compared with coordinated-actuated traffic signal control optimized by SYNCHRO, the average overall vehicle delay and average bus delay is reduced by 5% and 47%, respectively. Compared with TSP configured in coordinated-actuated traffic signal control optimized by SYNCHRO, the average overall vehicle delay in PAMSCOD is reduced by 38%, and average bus delay is increased by only 3%. However, the throughput of PAMSCOD is increased by more than 10% for congested cases, compared with TSP coordinated-actuated traffic control optimized by SYNCHRO.
7.2 Future Research Topics

Since this doctoral research includes both positioning techniques and traffic control strategies. Two different directions are discussed as follows.

For GPS positioning, future research topics should include:

- The problem of GPS blockage can be addressed by other vehicle sensors, such as gyro, odometer and vehicle wheel encoders that can also address the GPS drift issue at low speeds and provide important information about $\sigma_{lane}$.

- GPS noncommon-mode error estimation on a straight road is another challenge. It is likely that some vehicles will not execute turning maneuvers and the time between GPS offset updates may be long enough that the GPS drift will significantly affect positioning. This might be addressable by cooperative sharing of information using vehicle-to-vehicle (v2v) communications where all of the equipped vehicles on a street share GPS offset estimates and updates based on the population information.

For traffic signal control under v2x environment, the future research topics are described as bellow:

- In this research, it is assumed that the market penetration of IntelliDrive$^\text{SM}$ is either complete for priority required vehicles, such as emergency vehicles and transit vehicles, or significantly high for automobiles. Potential research includes addressing multi-modal traffic signal control with different penetration ratess of IntelliDrive$^\text{SM}$. 
• Low penetration of IntelliDrive\textsuperscript{SM} can affect the accuracy of platoon recognition in PAMSCOD. A platoon recognition algorithm needs to be provided to fuse data from both traditional point detectors and v2x communications.

• The solution times of PAMSCOD increase significantly during congested and oversaturated traffic conditions. Future research can focus on how to reduce the solution times of PAMSCOD by considering network partition, reducing the complexity of the MILP or developing heuristic algorithms.

• Another interesting topic would be to determine what level of penetration of IntelliDrive\textsuperscript{SM} would be required to significantly improve the performance of traffic signal control. This research could be done by feeding PAMSCOD different levels of online traffic data.
REFERENCES


Head, K., Gettman, D. & Wei, Z., 2006a. A Decision Model for Priority Control of Traffic Signals. Transportation Research Record: Journal of the Transportation


May 7, 2010].


Shewhart, W., 1931. *Economic Control of Quality of Manufactured Product*,


