DISS. ETH NO. 25694

REAL-TIME URBAN TRAFFIC CONTROL IN A CONNECTED AND AUTOMATED VEHICLE ENVIRONMENT

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

KAIDI YANG

Master of Science, Tsinghua University

born on 22.06.1990

citizen of
People's Republic of China

accepted on the recommendation of

Prof. Dr. Kay W. Axhausen
Prof. Dr. Monica Menendez
Prof. Dr. Hani S. Mahmassani
Prof. Dr. Yanfeng Ouyang
Prof. Dr. Xuegang (Jeff) Ban

2018

Foreword

Urban traffic control in a connected and automated vehicle environment is a very relevant topic these days in both academia and industry, as these technologies present enormous potential and significant challenges for the control and management of transportation systems. The work of Mr. Kaidi Yang focuses on the transition period of the technological development, an important period often overlooked by the previous research. Mr. Yang's work covers multiple aspects related to this topic, ranging from traffic estimation exploiting the information provided by connected vehicles, to traffic control considering multiple scales and multiple transportation modes. He uses an interdisciplinary approach that builds on traffic flow theory fundamentals and complements them with recent advances from machine learning, control theory, and operations research.

Mr. Yang's dissertation is rather interesting from both a scientific and a practical perspective. His work provides innovative methodologies and pragmatic guidelines on how these emerging technologies can be utilized to build a more efficient and sustainable urban transportation system. His work is also very timely, as many cities around the world are launching pilot projects utilizing connected and automated vehicles to improve traffic operations.

On behalf of the Traffic Engineering research group at the Swiss Federal Institute of Technology, Zurich, I thank Mr. Yang for his immense scientific creativity, attention to detail, and incredible ability to solve problems. Not only has his work contributed significantly to the advancement of traffic control in a connected and automated vehicle environment, but his vision has set the foundation for many other promising research directions.

Prof. Monica Menendez

Former Director of the Traffic Engineering research group at ETH Zurich

Currently Associate Professor of Civil Engineering at New York University Abu Dhabi (NYUAD) / Global Network Associate Professor of Civil and Urban Engineering at New York University (NYU)

Abstract

The promising development towards vehicle connectivity and autonomy will impose substantial societal and economic impact not only on the mobility systems but also on the entire cities. This dissertation will, from the perspective of the traffic operators, design effective traffic estimation and control strategies to maximize the benefits of these technologies for the urban traffic systems. We will propose both methodological frameworks and pragmatic guidelines to address this timely and relevant research topic. The main contributions of this dissertation are three-fold. First, this dissertation is among the first to handle the challenging issues in the transition period where vehicles with various technologies coexist in the traffic system (i.e. conventional, connected, and automated vehicles). We propose more accurate queue estimation methods that better exploit the limited information provided by connected vehicles, and develop efficient and robust control strategies to handle the uncertainties due to low penetration rates. Second, this dissertation develops a novel control framework for large-scale urban traffic networks at both the local and network levels. At the local level, we integrate traffic control at signalized intersections with trajectory design of automated vehicles. At the network level, we develop a multi-scale perimeter control strategy that fills two important research gaps: i) how the macroscopic control decisions can be translated into microscopic variables, and ii) how the control objectives at different levels can be synthesized. Third, this dissertation explicitly integrates priority schemes into the proposed control strategies, considering the interactions between different groups of vehicles (i.e. vehicles corresponding to different transportation modes, or with different occupancies, values of time, priority levels, etc.).

Part I addresses traffic estimation in a connected vehicle environment, with a particular focus on queue estimation at signalized intersections. The queue estimation results facilitate trajectory reconstruction and thus provide a holistic picture of the urban traffic system. These results also serve as essential inputs to traffic control and management strategies. In this part, we propose a computationally efficient methodology based on a convex optimization to exploit the information provided by connected vehicles. We fill the research gaps in two aspects. First, we relax the widely adopted assumption of uniform demand in a signal cycle. Second, we further reuse the information provided by upstream intersections in order to improve the performance of the algorithm. Simulation results show that the proposed strategy significantly improves the estimation accuracy with a reasonable solution time (0.8s), sufficient for most real-time applications. Results further show that the proposed algorithm is able to handle scenarios with penetration rates as low as 0.1 and is also robust to measurement noises.

Part II and Part III investigate advanced traffic control strategies exploiting the information and flexibility provided by connected and automated vehicles at both the local (Part II) and network (Part III) levels, and develop a multi-scale, multi-modal, and multi-technological control framework to maximize the potential of these technologies in urban traffic systems. In each part, we address the challenging issues in the transition period and study how the priority schemes can be integrated into the control strategies to differentiate different vehicle types.

In part II, we develop a bi-level optimization based strategy to integrate traffic signal timing and the trajectory planning of automated vehicles at local intersections. We further develop heuristics to switch between different signal control algorithms as the technology evolves. Simulation results show an evident decrease in the total number of stops and delay even when the penetration rate of the connected vehicles is lower than 50%. We further extend the proposed strategy to account for transit signal priority, considering bus stops and bus schedule. Simulation results show that such extension successfully improves passenger mobility over the original strategy. Results also demonstrate that it is valuable to consider bus stops, especially when they are near-side.

In part III, we develop a multi-scale perimeter control strategy based on a Model Predictive Control (MPC) algorithm where the network-level decision can be optimally distributed to local-level perimeter intersections to synthesize the competing objectives of both levels. Connected vehicles are assumed to be the only source of information. Simulation results show that the proposed strategy optimizes the performance at both the network and the perimeter intersections, providing much better outputs than the classical controllers. The multi-scale perimeter control strategy is further extended to a stochastic MPC that explicitly handles the uncertainties due to low penetration rates of connected and automated vehicles. It is shown that the total travel is significantly reduced by applying such stochastic MPC. This work is then integrated with priority lanes to prioritize certain groups of vehicles to maximize social welfare. Results show that by introducing the priority scheme, the social welfare can be improved.

The proposed estimation and control strategies in this dissertation provide insights on how the emerging technologies can be employed to build a more efficient and flexible urban multimodal urban transportation system. The findings serve as the cornerstone for some promising directions of future research, including the integration with special infrastructure, new mobility systems, heterogeneous data sources, multiple control strategies (e.g. routing), cyber-security, and interactions with other systems (e.g. logistics). This dissertation can be beneficial for traffic managers, local authorities, practitioners, and the automotive industry.

Zusammenfassung

Die vielversprechende Entwicklung in Richtung Fahrzeugkonnektivität und -autonomie wird nicht nur für die Mobilitätssysteme, sondern auch für die gesamten Städte erhebliche gesellschaftliche und wirtschaftliche Auswirkungen haben. Diese Dissertation wird aus Sicht der Verkehrsbetreiber wirksame Strategien zur Verkehrsschätzung und -steuerung entwickeln, um den Nutzen dieser Technologien für die städtischen Verkehrssysteme zu maximieren. Wir werden sowohl methodologische Rahmenbedingungen als auch pragmatische Richtlinien vorschlagen, um dieses aktuelle und relevante Forschungsthema zu behandeln. Diese Dissertation liefert drei Hauptbeiträge. Erstens ist diese Dissertation eine der ersten, die sich mit den anspruchsvollen Problemen in der Übergangszeit befasste, in der Fahrzeuge mit verschiedenen Technologien im Verkehrssystem nebeneinander existieren (d. h. konventionelle, verbundenen und automatisierte Fahrzeuge). Wir schlagen genauere Warteschlangenschätzungsmethoden vor, die die begrenzten Informationen, die von verbundenen Fahrzeugen zur Verfügung gestellt werden, besser nutzen und effiziente und robuste Steuerungsstrategien entwickeln, um die Unsicherheiten aufgrund niedriger Durchdringungsraten zu bewältigen. Zweitens entwickelt diese Dissertation ein neuartiges Steuerungsrahmenwerk für grosse städtische Verkehrsnetze sowohl auf lokaler als auch auf Netzebene. Auf lokaler Ebene integrieren wir die Verkehrskontrolle an signalisierten Kreuzungen mit der Trajektorienplanung automatisierter Fahrzeuge. Auf Netzwerkebene entwickeln wir eine Multiskalen-Perimeter-Steuerungsstrategie, die zwei wichtige Forschungslücken schliesst: i) wie die makroskopischen Steuerungsentscheidungen in mikroskopische Variablen übersetzt werden können und ii) wie die Kontrollziele auf verschiedenen Ebenen synthetisiert werden können. Drittens integriert diese Dissertation explizit Prioritätsschemata in die vorgeschlagenen Steuerungsstrategien, wobei die Wechselwirkungen zwischen verschiedenen Fahrzeuggruppen (d. H. Fahrzeugen, die verschiedenen Verkehrsträgern oder unterschiedlichen Belegungen, Zeitwerten, Prioritätsstufen usw. entsprechen) berücksichtigt werden.

Teil I befasst sich mit der Verkehrsabschätzung in einer Umgebung verbundener Fahrzeuge mit einem besonderen Fokus auf die Abschätzung von Warteschlangen an signalisierten Kreuzungen. Die Ergebnisse der Warteschlangenabschätzungen erleichtern die Rekonstruktion der Trajektorien und ergeben somit ein ganzheitliches Bild des städtischen Verkehrssystems. Diese Ergebnisse dienen auch als wesentlicher Input für die Verkehrssteuerungs- und -managementstrategien. In diesem Teil schlagen wir eine recheneffiziente Methodik vor, die auf einer konvexen Optimierung basiert, um die von verbundenen Fahrzeugen bereitgestellten Informationen zu nutzen. Wir schliessen die Forschungslücken in zwei Aspekten. Zunächst lockern wir die weit verbreitete Annahme einer konstanten Verkehrsnachfrage in einem Signalzyklus. Zweitens verwenden wir die Informationen, die von vorgelagerten Kreuzungen bereitgestellt werden, erneut, um die Performance des Algorithmus zu verbessern. Simulationsergebnisse zeigen, dass die vorgeschlagene Strategie die Schätzgenauigkeit mit einer angemessenen Rechenzeit (0,8 s) erheblich verbessert, was für die meisten Echtzeitanwendungen ausreichend ist. Die Ergebnisse zeigen weiter, dass der vorgeschlagene Algorithmus Szenarien mit Durchdringungsraten von 0,1 verarbeiten kann und ausserdem robust gegenüber Messrauschen ist.

Teil II und Teil III untersuchen fortgeschrittlich Strategien zur Verkehrssteuerung und nutzen dabei die Informationen und die Flexibilität, die verbundene und automatisierte Fahrzeuge sowohl im lokalen (Part II) als auch im Netzwerk (Teil III), und entwickeln einer multiskalare, multimodales und multitechnologischen Rahmen, um das Potenzial dieser Technologien in städtischen Verkehrssystemen zu maximieren. In jedem Teil behandeln wir die herausfordernden

Fragen der Übergangszeit und untersuchen, wie die Prioritätsschemata in die Kontrollstrategien integriert werden können, um verschiedene Fahrzeugtypen zu unterscheiden.

Im Teil II entwickeln wir eine auf zwei Ebenen optimierte Strategie, um die Verkehrssignalen und die Trajektorienplanung von automatisierten Fahrzeugen an lokalen Kreuzungen zu integrieren. Wir entwickeln Heuristiken weiter, um mit der Weiterentwicklung der Technologie zwischen verschiedenen Signalsteuerungsalgorithmen zu wechseln. Simulationsergebnisse zeigen eine deutliche Abnahme der Gesamtzahl der Stopps und der Verzögerung, selbst wenn die Durchdringungsrate der angeschlossenen Fahrzeuge unter 50% liegt. Wir erweitern die vorgeschlagene Strategie weiter, um die Priorität des Transitsignals zu berücksichtigen, wobei Bushaltestellen und Busfahrpläne berücksichtigt werden. Simulationsergebnisse zeigen, dass eine solche Erweiterung die Mobilität der Passagiere gegenüber der ursprünglichen Strategie erfolgreich verbessert. Die Ergebnisse zeigen auch, dass es wichtig ist, Bushaltestellen zu berücksichtigen.

Im Teil III entwickeln wir eine multiskalare Perimeter-Steuerungsstrategie, die auf einem Model Predictive Control (MPC) -Algorithmus basiert, mit dem die Entscheidung auf Netzwerkebene optimal auf Perimeter-Kreuzungen auf lokaler Ebene verteilt werden kann, um die konkurrierenden Ziele der beiden Ebenen zu synthetisieren. Es wird angenommen, dass verbunden Fahrzeuge die einzige Informationsquelle sind. Simulationsergebnisse zeigen, dass die vorgeschlagene Strategie die Leistung sowohl an den Netzwerk- als auch an den Perimeter-Kreuzungen optimiert und deutlich bessere Ergebnisse als die klassischen Controller bietet. Die multiskalare Perimeter-Steuerungsstrategie wird auf einen stochastischen MPC erweitert, der die Unsicherheiten aufgrund niedriger Durchdringungsraten von verbundenen und automatisierten Fahrzeugen explizit behandelt. Es wird gezeigt, dass der Gesamtweg durch die Anwendung solcher stochastischer MPC deutlich reduziert wird. Diese Arbeit wird dann in Prioritätsspuren integriert, um bestimmte Fahrzeuggruppen zu priorisieren, um die soziale Wohlfahrt zu maximieren. Die Ergebnisse zeigen, dass durch die Einführung des Prioritätssystems die soziale Wohlfahrt verbessert werden kann.

Die in dieser Dissertation vorgeschlagenen Schätz- und Kontrollstrategien geben Aufschluss darüber, wie aufstrebende Technologien zum Aufbau eines effizienteren und flexibleren städtischen multimodalen städtischen Verkehrssystems eingesetzt werden können. Die Ergebnisse dienen als Eckpfeiler für einige vielversprechende Richtungen der zukünftigen Forschung, einschliesslich der Integration in spezielle Infrastruktur, neue Mobilitätssysteme, heterogene Datenquellen, mehrere Steuerungsstrategien (z. B. Routing), Cybersicherheit und Interaktionen mit anderen Systemen (z. B. Logistik). Diese Dissertation kann für Verkehrsmanager, lokale Behörden, Praktiker und die Automobilindustrie einen Mehrwert darstellen.

Acknowledgments

My deepest thanks go to my supervisor Prof. Monica Menendez. Monica has been an outstanding researcher, supportive mentor, inspiring leader, and reliable friend. She has provided me with insightful guidance and tremendous freedom to develop my research, helped me improve the presentation of ideas and findings, and given me important advice for professional and personal development. Her devotion to research, openness to new questions, and willingness to embrace challenges will impose a profound impact on my future career.

I am also very grateful to my official supervisor Prof. Kay W. Axhausen for his valuable advice and feedback. Being a world-renowned scholar, he truly inspired me with his vision, creativity, and rigorous attitude towards research, which would undoubtedly benefit my future research.

Furthermore, I would also like to express my sincere gratitude to my external examiners, Prof. Hani Mahmassani, Prof. Yanfeng Ouyang, and Prof. Xuegang (Jeff) Ban, for serving on my doctoral examination committee, reviewing my thesis, and attending my defense. Their insightful comments and constructive suggestions have significantly improved this dissertation.

Moreover, I am very fortunate to work at the Traffic Engineering (SVT) Group where I met incredible people with diverse cultural and disciplinary background: Dr. Ilgin S. Guler (currently Assistant Professor at the Pennsylvania State University), Dr. Javier Ortigosa, Dr. Qiao Ge, Dr. Jin Cao, Dr. Mireia Roca Riu, Dr. Nan Zheng (currently Senior Lecturer at Monash University), Dr. Haitao He, Dr. Mahnam Saeednia, Marco Rothenfluh, Lukas Ambühl, and Igor Dakic. They have always been there supporting me in both research and life, from settling down in Zurich, to practicing for my doctoral defense. I have gained many new perspectives from the discussions and interactions with them. Special thanks go to Ilgin, Mireia, and Nan, with whom I have established fruitful collaborations and published many academic papers together.

I would also like to thank Dr. Anastasios Kouvelas, Prof. Francesco Corman, and all the members of the Institute for Transport Planning and Systems (IVT) at ETH Zurich for their kindly support and thought-provoking discussions. I would like to give special credits to Beda Büchel for helping me translate the English abstract into German. I am also very lucky to give lectures or supervise many excellent bachelor and master students at ETH Zurich. Working with them makes me understand better the role of being a teacher and supervisor. I am also thankful to the researchers at EPF Lausanne, TU Delft, and IFSTTAR, from whom I often get insightful feedbacks at various transportation conferences. I am also grateful to the professors at Tsinghua University for their encouragement and insights, in particular to Prof. Li Li, Prof. Xin Pei, and Prof. Yi Zhang.

My life as a Ph.D. would not have been so colorful if it were not for the friends I met during social activities and sports in Switzerland. It is my pleasure to be part of the Tsinghua Alumni Association in Switzerland and participate in the organization of various memorable events, including the Belt and Road Davos Forum. I would also like to thank my flatmates, tennis partners, and hiking friends for making my life in Switzerland so enjoyable. I would also like to thank my long-time friends around the world for their time-to-time greetings and continuous support.

Last but most important, I give my most sincere gratefulness to my family, in particular to my mother, to whom I owe everything that I have achieved.

Contents

Foreword i

Abstract iii

Zusammenfassung v

Acknowledgments vii

1 Introduction 1
- 1.1 Research Motivation 1
- 1.2 Goal and Scope 2
  - 1.2.1 Research goals 2
  - 1.2.2 Research scope 3
- 1.3 Research Contributions 3
- 1.4 Organization 4

2 Background and Related Literature 7
- 2.1 Background Information, Assumptions, and Terminology 7
  - 2.1.1 Intersection control 7
  - 2.1.2 Vehicular technologies 7
- 2.2 Related Literature 9
  - 2.2.1 Traffic estimation in a connected and automated vehicle environment 9
  - 2.2.2 Traffic control of local intersections in a connected and automated vehicle environment 10
  - 2.2.3 Aggregated traffic control of large-scale urban traffic networks in a connected vehicle environment 11
- 2.3 Research Gaps 13

I Traffic Estimation in a Connected Vehicle Environment 15

3 Queue Estimation in a Connected Vehicle Environment 19
- 3.1 Objectives and Contributions 19
- 3.2 General Methodology 20
  - 3.2.1 Data labelling and preprocessing 21
  - 3.2.2 Identification of the critical points 22
  - 3.2.3 Estimation of the FoQ curve 23
  - 3.2.4 Estimation of the BoQ curve 24
  - 3.2.5 Calculation of the queue length 25
- 3.3 Reuse of Upstream Departure Information 25
  - 3.3.1 Estimation of the arrival flow 25
  - 3.3.2 Estimation of the queue length 26
- 3.4 Inclusion of the Intermediate States for Cases With Limited Data 27

3.5 Simulation Settings 28
3.6 Case Study and Results 30
  3.6.1 Results for an isolated intersection 30
  3.6.2 Value of considering piecewise linear BoQ 31
  3.6.3 Value of considering acceleration and deceleration 33
  3.6.4 Value of integrating flow information 34
3.7 Sensitivity Analysis of the Proposed Methodology 35
  3.7.1 Sensitivity analysis to model parameters 35
  3.7.2 Robustness to measurement errors 37
3.8 Implementation Details 38
  3.8.1 Online implementation 38
  3.8.2 Comparison between the two implementation methods 39
3.9 Conclusions and Future Work 41

II Local-level Traffic Control in a Connected and Automated Vehicle Environment 43

4 Intersection Control in a Connected and Automated Vehicle Environment 47
4.1 Objectives and Contributions 47
4.2 General Methodology 48
  4.2.1 Model inputs 49
  4.2.2 Upper level model: optimization of departure sequence 49
  4.2.3 Lower level model: trajectory design for automated vehicles 51
  4.2.4 A branch and bound solution algorithm 52
4.3 Simulation Settings 55
4.4 Case Study and Results 56
4.5 Sensitivity Analysis of the Proposed Strategy 58
  4.5.1 Robustness to location errors 58
  4.5.2 Robustness to arrival patterns 59
  4.5.3 Efficiency of the branch and bound algorithm 60
4.6 A Demand Responsive Control Scheme 60
4.7 Conclusions and Future Work 62

5 Intersection Control with TSP in a Connected Vehicle Environment 63
5.1 Objectives and Contributions 63
5.2 General Methodology 64
  5.2.1 Algorithm without considering bus stops 65
  5.2.2 Algorithms considering bus stops and signal delay 66
  5.2.3 Algorithm considering bus stops and schedule delay 68
5.3 Simulation Framework 68
5.4 Simulation Results and Algorithm Evaluation 70
  5.4.1 Value of connected vehicles and value of TSP 70
  5.4.2 Value of TSP considering bus stops 74
  5.4.3 Value of the algorithm considering schedule delay 74
5.5 Robustness of the Algorithm 76
  5.5.1 Sensitivity to location of bus stops 76
  5.5.2 Sensitivity to assumed bus occupancy 77
  5.5.3 Sensitivity to bus dwell time estimation error 77
  5.5.4 Trade-off between bus priority and the conflicting approach delay . . . 78
5.6 Conclusions and Future Work 79

III Network-level Traffic Control in a Connected Vehicle Environment 81

6 Multi-scale Perimeter Control in a Connected Vehicle Environment 87
6.1 Objectives and Contributions 87
6.2 General Methodology 88
6.2.1 Problem presentation 88
6.2.2 An MPC approach for multi-scale control 89
6.2.3 An approximation framework 92
6.3 Simulation Settings 94
6.4 Case Study on the Multi-scale Controller 94
6.4.1 Performance of the multi-scale controller 95
6.4.2 System performance under moderate noises 98
6.4.3 Performance deterioration with strong noises in the system 100
6.5 Stochastic Controller 102
6.5.1 Design of the stochastic MPC 102
6.5.2 Performance of the stochastic MPC 103
6.5.3 Value of connected vehicles 105
6.6 Conclusions and Future Work 106

7 Perimeter Control with Priority Lanes in a Connected Vehicle Environment 107
7.1 Objectives and Contributions 107
7.2 General Methodology 108
7.2.1 Problem presentation 108
7.2.2 MPC approach 110
7.2.3 Approximation framework 112
7.2.4 Online recalibration for the VOT distribution 113
7.3 Simulation Settings 114
7.4 Case Study and Results 116
7.4.1 Overall performance 116
7.4.2 Performance of individual groups of cars 117
7.4.3 Value of the recalibration algorithm 118
7.5 Sensitivity Analysis 119
7.5.1 Sensitivity to the assumed VOT 119
7.5.2 Sensitivity to storage capacity at the perimeter intersections 120
7.5.3 Sensitivity to scaling parameter in the logistic function 121
7.6 Conclusion and Future Work 122

8 Conclusions and Outlook 123
8.1 Summary 123
8.2 Limitations 124
8.3 Outlook 125

Bibliography 127

A Detailed Calculations 139
A.1 Inclusion of the Intermediate States for Cases With Limited Data 139
A.2 Calculation of the Optimal and Entering Speed 140
A.3 Kalman Filter for Location and Speed Estimation With Measurement Noises 141

Curriculum Vitae 145

List of Figures

1.1 Organization of the dissertation. 5

3.1 Flowchart of the proposed general methodology. 21
3.2 Illustration of the methodology. 22
3.3 Study area and vehicle trajectories of the simulation. 29
3.4 Performance of the proposed methodology for an isolated intersection. 30
3.5 Illustration of queue lengths. 32
3.6 Value of considering piecewise linear BoQ. 34
3.7 Value of considering acceleration and deceleration. 35
3.8 Value of flow information. 36
3.9 Sensitivity analysis to model parameters. 36
3.10 Sensitivity to the measurement errors 37
3.11 Comparison between two online implementation methods. 40

4.1 Flowchart of the proposed strategy. 48
4.2 Illustration of penalty. 51
4.3 Illustration of the speed calculation. 51
4.4 Illustration of the branch and bound algorithm 53
4.5 Simulation results for average number of stops per vehicle. 56
4.6 Simulation results for average delay per vehicle. 57
4.7 Robustness to measurement errors. 58
4.8 Average coefficient of variation. 59
4.9 Demand responsive control scheme based on connected vehicle technology. . . 61

5.1 Intersection layout. 64
5.2 Evaluation framework. 70
5.3 Average delay savings per passenger (both cars and buses) for algorithm with general TSP compared to the fixed-time strategy with TSP and the connected vehicle algorithm without TSP (balanced demand). 72
5.4 Average delay savings per passenger (both cars and buses) for algorithm with general TSP compared to the fixed-time strategy with TSP and the connected vehicle algorithm without TSP (unbalanced demand). 73
5.5 Average delay savings per passenger for the connected algorithm with TSP considering stops compared to algorithm with general TSP. 75
5.6 Average delay savings per passenger. 75
5.7 Sensitivity to location of stops. 76
5.8 Sensitivity to assumed occupancy. 77
5.9 Sensitivity to bus dwell time estimation error. 78
5.10 Average vehicle delay of buses and cars in the conflicting approach. 79

6.1 Control diagram for MPC. 90
6.2 Simulations settings 95
6.3 Accumulation comparison between three controllers. 96
6.4 Queue length comparison between three controllers. 97

6.5 Total queue length for all streams and intersections without noises. 97
6.6 System performance with moderate noises. 99
6.7 System performance with strong noises. 101
6.8 System performance of the stochastic MPC with strong noises. 104
6.9 Performance of the stochastic MPC in different penetration rates of connected vehicles. 105

7.1 Illustration of one possible layout at priority intersection i. 109
7.2 Layout of the intersections. 115
7.3 Distribution of car VOTs 115
7.4 Performance of the algorithm. 116
7.5 Performance of individual group of cars in the priority movement 117
7.6 Delay savings of the recalibration algorithm at the perimeter intersections. Penetration rate of connected vehicles is 100%. 118
7.7 Value of information provided by connected vehicles, $\zeta^0 = [4, 0.4]$. 119
7.8 Sensitivity to the assumed VOT $\bar{\sigma}$. 120
7.9 Sensitivity to storage capacity at the perimeter intersections $X^i_{m,\max}$ 121
7.10 Sensitivity to scaling parameter ω in the logistic function. 121

A.1 Effect of the Kalman filter. 143

List of Tables

2.1 Levels of Automation. (NHTSA, 2018) 8

3.1 Parameters for simulation. 29
3.2 Cycle-by-cycle maximum queue lengths. 33

4.1 Attributes of a node v. 53
4.2 Efficiency of the branch and bound algorithm (nodes visited). 60

5.1 Summary of results. 71

6.1 Comparison between the two MPC, the PID controller and the Bang-bang controller. 100

Chapter 1

Introduction

1.1 Research Motivation

Cities have a substantial impact on the economic growth and social progress of a nation. With the technological boom, the concept of smart cities has been attracting increasing attention from the academia, industry, and government to address the pressing challenges due to rapid urbanization (e.g. population explosion, pollution, traffic congestion, etc.) (Batty et al., 2012). Exploiting advanced technologies and rich operational data, smart cities connect people, goods, information, and city elements to provide sustainable and efficient city services, innovative business, and recuperate life quality (Harrison et al., 2010; Bakıcı et al., 2013). Many major cities have established smart city initiatives or projects, such as Amsterdam, Zurich, Berlin, Singapore, New York, Beijing, etc. It is anticipated that the global smart cities market size will reach USD 2.57 trillion by 2025 (Grand View Research, Inc., 2018).

Smart mobility is an essential paradigm within the smart city concept, setting the foundation of various urban activities. The goal of smart mobility is to promote the shift to more efficient and flexible multimodal transportation systems. Currently, with the surging population in urban areas, traffic congestion is soaring all around the world, giving rise to the loss of working hours, an increase in traffic accidents, and pollution. It is estimated that car drivers in Europe lose more than 33 hours per year in congestion (63 billion EUR in total monetary cost). Similarly, traffic congestion has cost 305 billion USD in 2017 in the U.S according to INRIX. Therefore, building a smart mobility system is a timely and relevant issue, which requires joint efforts from the academia, industry, and government.

The promising development towards vehicle connectivity and autonomy is one of the most exciting breakthroughs in smart mobility over the last decade. It is predicted that the number of connected vehicles sold globally will grow more than sixfold to 152 million by 2020 (McCarthy, 2015). On the other hand, commercial companies, including big technology firms (e.g. Google), start-ups (e.g. nuTonomy), and automotive manufacturers (e.g. BMW), are racing in an effort to become mainstream while building fully automated vehicles. Many high-end vehicles have already been equipped with lower level automation functions (e.g. adaptive cruise control and lane-keeping assistance). It is expected that the first fully automated vehicles will hit the market by 2020 (Center for Sustainable Systems, 2016, Alexander-Kearns et al., 2016). Many of the prototypes of the fully automated vehicles have already been tested on public roads.

It is widely envisioned that the technologies of connected and automated vehicles will impose revolutionary societal and economic impact not only on the smart mobility systems, but also on the smart cities in general. For traffic operators, these technologies provide remarkable flexibility for traffic estimation, modeling, control, and management, and thus have significant potential to improve the operations of traffic systems. For example, connected vehicles generate rich real-time information beneficial for the estimation of traffic systems, and automated vehicles enable large-scale algorithmic coordination. For passengers, these technologies can help improve safety, comfort, and convenience. Equipped with various sensors and advanced communication systems, connected and automated vehicles are able to identify and handle potential danger more timely. Fully automated vehicles are able to enhance mobility for those who are not able or willing to drive (e.g. young, elderly, or people with disabilities). Promising mobility concepts, such as Autonomous Mobility-on-demand and ride-sharing services, help provide a ubiquitous mobility service with a lower cost, which could, in turn, reduce car ownership. Moreover, these technologies can be beneficial for the entire smart cities, e.g. providing better access to city services, enabling more anticipated and effective logistic systems, etc.

Despite the potential benefits, there are still challenges in the utilization of these technologies, specifically summarized as follows.

1) The penetration rates of these technologies can only increase gradually, resulting in a transition period with an uncertain environment and limited information. However, for the promotion of these technologies, it is beneficial to demonstrate certain advantages even from the early stages of the deployment. This requires more accurate estimation methods and more robust control strategies for a period during which multiple technologies will coexist.
2) Urban transportation systems are typically large-scale and complex systems. It is essential to design effective yet computationally tractable control strategies at both the local (intersection) and network levels in a connected and automated vehicle environment. Moreover, although many researchers propose efficient hierarchical control strategies, it is not clear how the network level control decisions can be translated to local level variables and how the control objectives at both levels can be balanced.
3) Urban transportation systems are also multimodal systems where different transportation modes compete for limited road infrastructure and collaborate to serve the increasing transport demand. In such multimodal systems, it is desirable to provide priority to certain groups of vehicles, such as emergency vehicles and the vehicles with the ability to carry more passengers (e.g. public transport vehicles, ride-sharing vehicles, etc.). This requires the integration of priority schemes into the control framework, which adds to the complexity of the problem.

The solution to these challenges still remains as open research questions. This dissertation will address such research questions from the perspective of the traffic operators. We will design effective traffic estimation and control strategies to maximize the benefits of these technologies on urban traffic systems.

1.2 Goal and Scope

1.2.1 Research goals

The primary objective of this dissertation is to develop effective yet efficient traffic estimation and control strategies in a connected and automated vehicle environment. Specific objectives are listed below.

1) Develop traffic estimation and control strategies using the information provided by connected vehicles, which are able to exploit the benefits of connected and automated vehicles especially during the transition period.
2) Propose traffic signal control strategies at both local and network levels and investigate how the control at both levels can be bridged.
3) Integrate priority schemes into the proposed signal control schemes to provide priority to certain groups of vehicles (i.e. vehicles corresponding to different transportation modes, or with different occupancies, values of time, priority levels, etc.).

The outcome of this dissertation will provide scientific and pragmatic support for the deployment of urban traffic estimation and control strategies in the presence of connected and automated vehicles.

1.2.2 Research scope

In this dissertation, we constrain the scope to traffic estimation and control in an urban scenario, with a particular focus on signalized intersections in a connected and automated vehicle environment.

1) We only study traffic estimation and control. Although the developed strategies will be based on several traffic models, such as car following models, kinematic wave theory, and Macroscopic Fundamental Diagram (MFD), we do not investigate how these models are derived. In other words, these models are assumed as exogenous inputs to the proposed estimation and control strategies. Detailed discussions of these models (e.g. the car following and mechanical dynamics of automated vehicles or the existence and partitioning for the MFD) are beyond the scope of this dissertation.
2) We only propose signal control strategies for signalized intersections in an urban scenario. This is because traffic signals are essential components in urban traffic systems. Properly designed signal control strategies can effectively reduce travel time, fuel consumption and accidents. We do not consider highway scenarios or urban traffic systems that do not include signalized intersections (e.g. parking). Moreover, we do not study traffic control strategies other than signal control (e.g. routing).
3) We only perform centralized control where the system only has one controller (i.e. the central controller). Through centralized control, we aim to find the global optimal control decisions to maximize the potential of the emerging technologies. Decentralized control, although can be beneficial in some cases, is beyond the scope of this dissertation.
4) We assume that connected and automated vehicles are the only information source to clearly evaluate the benefits of these technologies. We do not study the fusion with other data sources, such as loop detectors and video cameras. Data fusion is beyond the scope of this dissertation, but it is expected to improve the results obtained here.
5) We assume that the information provided by connected and automated vehicles is accurate, with the only exception of measurement errors. We do not consider the potential issues in communication systems, e.g. packet loss or communication interference. We further assume that the connected and automated vehicles are faithful in providing the information. In other words, we do not consider the scenarios where these vehicles are hacked or the owners intentionally forge the information for their own benefits. Cyber-secure control against malicious attacks is beyond the scope of this dissertation, but recommended as future work.
6) We only focus on the supply side. In other words, we assume that the traffic demand is given as an exogenous input. We do not study how the demand can be modelled and how the proposed strategies will influence such demand. This is because the proposed strategies are real-time, and thus will have marginal impact on the demand in the short term.

1.3 Research Contributions

The major contributions of this dissertation are three-fold.

1) This dissertation is among the first to handle the challenging issues during the transition period where vehicles with various technologies coexist in the traffic systems (i.e. conventional, connected, and automated vehicles) . We study the real-time queue estimation (trajectory reconstruction) to better utilize the limited connected vehicle data (Chapter 3) and propose various control strategies to account for the uncertainties due to low penetration rates during the transition period (Chapter 4 and Chapter 6). The proposed strategies in Chapter 3–7 are tested in scenarios with various penetration rates of the technologies.
2) This dissertation develops a novel control framework for large-scale urban traffic networks

at both the local and network levels. At the local level, we jointly optimize the signal timings of local intersections and the trajectory planning of automated vehicles (Chapter 4). At the network level, we pay particular attention to i) how the macroscopic control decisions can be translated into microscopic variables; and ii) how the control objectives at different levels can be synthesized. In particular, we make the first methodological effort to integrate the control of perimeter intersections and perimeter control of large-scale urban networks (Chapter 6).

3) This dissertation explicitly integrates priority schemes into the proposed control strategies, considering the interactions between different groups of vehicles (i.e. vehicles corresponding to different transportation modes, or with different occupancies, values of time, priority levels, etc.). Specifically, we provide priority to public transport vehicles at local intersections (Chapter 5), and to vehicles with higher value of time to enter the network (Chapter 7). We further analyze how the control strategies can be adapted to different infrastructure settings. This research aims to promote a shift to more sustainable transportation modes.

As part of these contributions, we propose not only methodological frameworks, but also pragmatic guidelines. More detailed contributions will be described within each chapter.

1.4 Organization

The overall methodologies adopted in this research and the corresponding findings are presented in the reminder of this dissertation. Chapter 2 provides a comprehensive review of the most important research works on traffic estimation and control in a connected and automated vehicle environment. Chapter 3 – Chapter 7 contain the contributions of this dissertation, and are organized into three parts. Chapter 8 concludes the dissertation and proposes future open questions. The overall structure of this dissertation is illustrated in Figure 1.1. An overview of each part is given below. Notice that we maintain the same notation within each part and provide the list of the most important variables in the abstract of the part.

Part I: Traffic estimation in a Connected Vehicle Environment

Part I addresses traffic estimation in a connected vehicle environment, in particular queue estimation.

Chapter 3 proposes a convex optimization based methodology to estimate queue profile in a connected vehicle environment, which can be further utilized for trajectory reconstruction, flow estimation, etc. We exploit the information of connected vehicles in two aspects. First, we relax the widely adopted assumption of uniform demand in a signal cycle. Second, we further reuse the information provided by upstream intersections in order to improve the performance of the algorithm.

Part II: Local-level Traffic Control in a Connected and Automated Vehicle Environment

Based on the estimation results, Part II develops a control strategy at the local level which handles scenarios with various priority requirements and different technology development stages.

Chapter 4 establishes a joint optimization strategy that integrates the signal timing and trajectory planning of automated vehicles in a scenario where conventional, connected but non-automated, and automated vehicles co-exist in the traffic systems.

Chapter 5 extends the strategy developed in Chapter 4 to provide priority to public transport vehicles. Traffic signals are explicitly coordinated with bus stops and bus schedule to minimize total passenger delay.

Part III: Network-level Traffic Control in a Connected Vehicle Environment

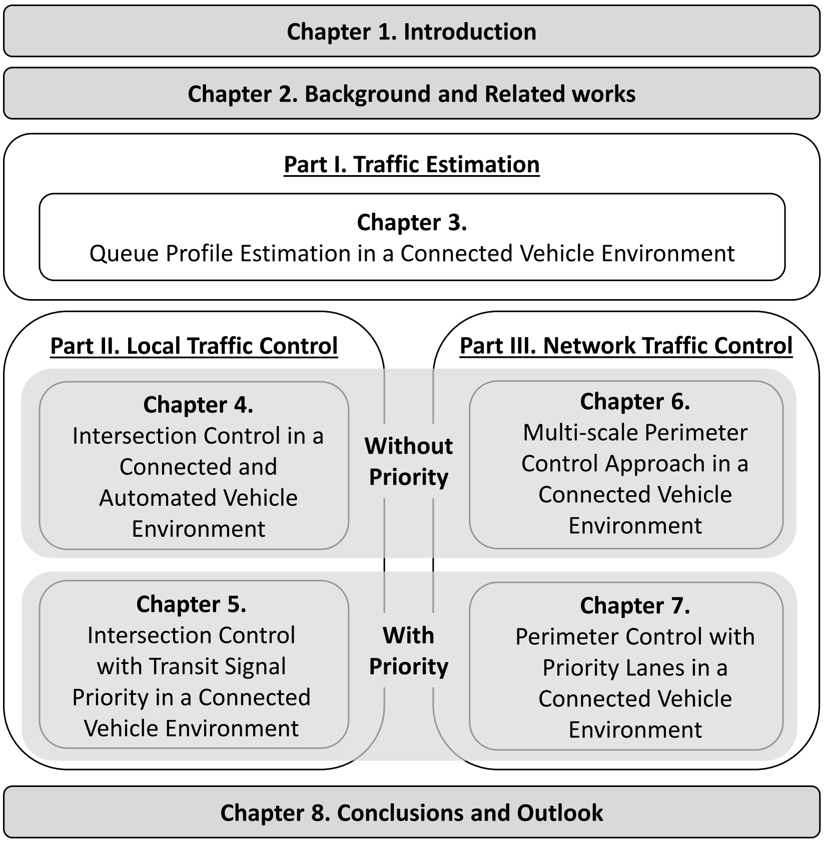

Chapter 1. Introduction
Chapter 2. Background and Related works
Part I. Traffic Estimation
Chapter 3.
Queue Profile Estimation in a Connected Vehicle Environment
Part II. Local Traffic Control
Part III. Network Traffic Control
Chapter 4.
Intersection Control in a Connected and Automated Vehicle Environment
Without Priority
Chapter 6.
Multi-scale Perimeter Control Approach in a Connected Vehicle Environment
Chapter 5.
Intersection Control with Transit Signal Priority in a Connected Vehicle Environment
With Priority
Chapter 7.
Perimeter Control with Priority Lanes in a Connected Vehicle Environment
Chapter 8. Conclusions and Outlook

Figure 1.1: Organization of the dissertation.

Part III develops a control strategy at the network level and extends it to provide priority to certain groups of vehicles during the transition period of technology development.

Chapter 6 develops a multi-scale perimeter control strategy based on Model Predictive Control (MPC). It integrate the optimal control of perimeter intersections into the perimeter control scheme to jointly optimize the traffic performance at both levels, only based on the information of connected vehicles. The proposed strategy is extended to a stochastic MPC to handle the uncertainties due to the low penetration rate of connected vehicles.

Chapter 7 extends Chapter 6 by providing priority to allow certain group of vehicles to enter the protected region with less delay.

Chapter 3 – Chapter 7 are the updated versions of papers published, presented in, or submitted to peer reviewed journals and conferences. These are all original work and first authored by the doctoral candidate. A list of the relevant papers is provided below.

Papers in Peer Reviewed Journals

- Yang, K. and M. Menendez (in press) Queue estimation in a connected vehicle environment: A convex approach, *IEEE Transactions on Intelligent Transportation Systems*. DOI: `10.1109/TITS.2018.2866936`.

- Yang, K., S. I. Guler and M. Menendez (2016a) Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles, *Transportation Research Part C: Emerging Technologies*, **72**, 109–129.

- Yang, K., M. Menendez and S. I. Guler (2018b) Implementing transit signal priority in a connected vehicle environment with and without bus stops, *Transportmetrica B: Transport*

Dynamics, 1–23.

- Yang, K., N. Zheng and M. Menendez (2018d) Multi-scale perimeter control approach in a connected-vehicle environment, *Transportation Research Part C: Emerging Technologies*, **94**, 32 – 49.

Papers in Refereed Conference Proceedings

- Yang, K. and M. Menendez (2017) A convex model for queue length estimation in a connected vehicle environment, paper presented at the *Transportation Research Board 96th Annual Meeting*.

- Yang, K., S. I. Guler and M. Menendez (2015) A transit signal priority algorithm under connected vehicle environment, paper presented at the *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 66–70.

- Yang, K., M. Menendez and S. I. Guler (2016b) Using connected vehicle technology to optimize transit signal priority, paper presented at the *Transportation Research Board 95th Annual Meeting*.

- Yang, K., N. Zheng and M. Menendez (2017b) Multi-scale perimeter control approach in a connected-vehicle environment, paper presented at the *22nd International Symposium on Transportation and Traffic Theory (ISTTT)*, vol. 23, 101–120.

- Yang, K., N. Zheng and M. Menendez (2018e) A perimeter control approach integrating dedicated express toll lanes, paper presented at the *Transportation Research Board 97th Annual Meeting*.

- Yang, K., N. Zheng and M. Menendez (2017a) Integrating perimeter control with dedicated express toll lanes, paper presented at the *Traffic and Granular Flow Conference (TGF 2017)*.

Chapter 2

Background and Related Literature

In this chapter, we provide background information on connected and automated vehicles, present a short literature review, and identify the research gaps. Section 2.1 presents the background information, assumptions and terminology that will be used throughout the paper. Section 2.2 reviews the related literature on traffic estimation and control. Section 2.3 summarizes the research gaps.

2.1 Background Information, Assumptions, and Terminology

2.1.1 Intersection control

In this dissertation, we assume that there is a central controller at the local intersections and the urban network. It is also assumed that the controllers between different levels or between different intersections are able to communicate with each other via communication systems. The roles of these controllers are two-fold: 1) to perform traffic estimation using the collected information, and 2) to calculate and execute the control decisions based on the estimation results. We further assume that the central controller at the network level is able to influence or overwrite the control decisions at the local intersections.

The control decisions are either the departure sequence in which individual vehicles leave the intersection (Part II) or the green time for each signal phase (Part III). The major goal is to minimize delay for all the vehicles (or passengers) at the intersections or inside an urban network. For vehicles arriving at the intersections, we distinguish the following terminology.

Approaches are defined as links used by traffic approaching the intersection (e.g. an intersection with four approaches refers to an intersection with four links carrying incoming traffic).

Movements are defined as the traffic moving in the same direction (e.g. through, left turning, etc.) in the same approach.

Streams are defined as the traffic in the same movement and with the same right of way, which are combined in order to simplify the calculations for design of the signal control strategy (e.g. left turning maneuvers that receive the green signals at the same time). Notice that if two lanes in the same movement receive different green times (e.g. with priority lanes, see Chapter 7), they are seen as different streams.

In the simple intersections considered in Part II, approaches, movements, and streams are equal. However, in Part III, we consider complex and complete intersections where these concepts can be different.

2.1.2 Vehicular technologies

It is also assumed that the vehicle-to-infrastructure (V2I) communication systems are installed along with the central controller. V2I communication systems have been widely employed in traffic signal control strategies (He et al., 2014; Li et al., 2014b; Hu et al., 2015). There are several platforms that enable V2I communications, such as Dedicated Short Range Communications (DSRC), 3G/4G, Bluetooth, etc. Compared to the other platforms, DSRC has several advantages (Guo and Balon, 2006; Andrews and Cops, 2009): 1) it is more robust to radio interference and extreme weather conditions; 2) it works with high vehicle speeds; 3) it has small latency (0.002s); 4) it helps to protect privacy; and 5) the transmission rage of DSRC is 100-1000 meters. Therefore, in this dissertation, we assume DSRC communication technology for V2I communication.

We categorize vehicles based on their connectivity and autonomy as follows.

Conventional Vehicles are defined as the vehicles that are unable or unwilling to communicate to the central controller or other vehicles by any means, nor to perform the commands advised by the central controller.

Connected vehicles are the vehicles that can communicate to the central controller through V2I communication systems. The information that connected vehicles report includes their position, speed, direction, etc.

Automated vehicles are the connected vehicles that can drive themselves and strictly follow the trajectory planning commands informed by the infrastructure.

Notice that in this dissertation, there is in general an overlap between the definitions of connected and automated vehicles. An additional distinction between connected and automated vehicles is made only in Chapter 4.

It is worth noting that in 2014, SAE international, an automotive standardization body, published a classification system for self-driving vehicles based on different levels of automation, ranging from fully manual to fully automated (SAE, 2014). The six levels (0–5) of automation, classified based on human involvement, are shown in Table 2.1.

Table 2.1: Levels of Automation. (NHTSA, 2018)

Levels of automation	Who does what, when
Level 0	The human driver does all the driving.
Level 1	An advanced driver assistance system on the vehicle can sometimes assist the human driver with either steering or braking/accelerating, but not both simultaneously.
Level 2	An advanced driver assistance system on the vehicle can itself actually control both steering and braking/accelerating simultaneously under some circumstances. The human driver must continue to pay full attention (“monitor the driving environment”) at all times and perform the rest of the driving task.
Level 3	An Automated Driving System on the vehicle can itself perform all aspects of the driving task under some circumstances. In those circumstances, the human driver must be ready to take back control at any time when the ADS requests the human driver to do so. In all other circumstances, the human driver performs the driving task.
Level 4	An Automated Driving System on the vehicle can itself perform all driving tasks and monitor the driving environment - essentially, do all the driving - in certain circumstances. The human need not pay attention in those circumstances.
Level 5	An Automated Driving System on the vehicle can do all the driving in all circumstances. The human occupants are just passengers and need never be involved in driving.

In this dissertation, we assume that the level of automation of the automated vehicles is in general above Level 4. Nevertheless, the automated vehicles we consider may also include certain vehicles in Level 2-3 if they are capable of executing the commands of the infrastructure and are sufficiently safe to operate in a mixed traffic scenario.

2.2 Related Literature

Over the last decade, connected and automated vehicles have attracted enormous research interest, ranging from traffic modeling (Talebpour et al., 2015), traffic safety (Rahman and Abdel-Aty, 2018; Xie et al., 2018), traffic operations and control (Li et al., 2014a; Florin and Olariu, 2015) for both urban and highway systems, energy consumption (Vahidi and Sciarretta, 2018), novel mobility concepts such as Autonomous Mobility-on-Demand systems (Hyland and Mahmassani, 2018; Pavone et al., 2012), to their implications on policies (Bösch et al., 2018). In this section, we focus on the traffic operation side and present related works on the traffic estimation and control strategies in a connected and automated vehicle environment. Notice that the literature has evolved since the publication of some of the chapters on academic journals. For the integrity of the dissertation, we do not include the literature that supersedes the journal articles on which the dissertation is based. Interested readers can refer to recent survey papers (e.g. Guo et al. (2019)).

2.2.1 Traffic estimation in a connected and automated vehicle environment

The promising development of connected and automated vehicle technology provides unique opportunity for more accurate traffic estimation. Compared to the traditional roadside detectors (e.g. loop detectors), connected and automated vehicles can provide real-time and detailed information with a better spatial coverage, and without the extra cost for installation and maintenance. Queue estimation is an important aspect of traffic estimation, as it can be used for trajectory reconstruction, delay evaluation, traffic flow estimation, etc. In other words, it provides valuable and holistic information about the traffic dynamics. Moreover, in urban traffic networks, queue lengths are an essential input to adaptive signal control strategies (Webster, 1958; Allsop, 1971; Chang and Lin, 2000; Guler et al., 2014; Li et al., 2013; Yang et al., 2016a, 2017b, 2018b) and performance measurements at signalized intersections or arterials (Balke et al., 2005). Therefore, in this section, we will focus on the literature on queue estimation. For literature review of traffic estimation in a more general context, interested readers can refer to Mori et al. (2015); Darwish and Bakar (2015).

One challenge on queue estimation using the connected and automated vehicle technology is the penetration rate, which is expected to remain low in the near future. The penetration rate highly influences the estimation accuracy, as only connected vehicles can report information. Another challenge is the sampling rate, i.e. the frequency at which a vehicle reports information. This can also be calculated as the inverse of the time interval between two consecutive reports. Some of the existing literature requires that the connected vehicles report information every second (Sun and Ban, 2013; Gómez et al., 2015; Dakic and Menendez, 2018). However, the sampling rate could be lower in reality due to the transmission and storage capacity. Moreover, the obtained information can also be noisy, as there might be data corruption or GPS noises. Therefore, the proposed methods should be robust enough to handle these challenges.

The existing literature on queue estimation can be mostly classified into two categories. The first category uses queuing theory, usually with stochastic arrivals. For example, Comert and Cetin (2009); Comert (2013b,a) proposed a probabilistic model to estimate the expected queue length, assuming a Poisson arrival process of vehicles and a Bernoulli distribution for whether a vehicle is connected or not. Amini et al. (2016) used a stochastic gradient descent algorithm and a queuing diagram to estimate the queue lengths. However, these works cannot capture the spatial traffic dynamics (e.g. the propagation of traffic waves) due to the point queue assumption.

The second category is based on Lighthill-Whitham-Richards (LWR) kinematic wave theory (Lighthill and Whitham, 1955b,a; Richards, 1956). Some works use the sampled travel time obtained from the trajectory data to estimate the queue length (e.g. Ban et al., 2009, 2011; Hao et al., 2012, 2014; Hao and Ban, 2015). Ban et al. (2009, 2011) fitted the queue curves based on the observation of a decreasing delay pattern. A support vector machine (SVM) based methodology was proposed in Hao et al. (2012) to identify the critical points for the new cycle based on delay and travel time. Hao et al. (2014) and Hao and Ban (2015) considered the acceleration and deceleration process to calculate the queue location in the discharging process and to reconstruct the long queue when spillbacks happen, respectively. Other works directly use the trajectory data to estimate the piecewise linear back of queue (BoQ) that consists of a series of shockwaves. Through the BoQ curve, not only queue length and platoons, but also traffic flow, density, and even trajectories can be recovered. The estimation of the BoQ curve can be formulated into a regression and/or classification problem. The general process for estimating the BoQ curve follows two steps. In the first step, critical points where the traffic states change are identified on the time-space diagram. Each critical point is a two dimensional vector: time t and location x. There are a few different definitions of critical points. Sun and Ban (2013) estimated the critical points as the first trajectory points with speed lower than a threshold; Izadpanah et al. (2009) and Hiribarren and Herrera (2014) fitted the vehicle trajectory into a piece-wise linear function, and identified the critical points as the intersection between each two pieces; Cheng et al. (2012) determined the critical points using both speed and acceleration information; Ramezani et al. (2015) defined the critical points as the first calculated x-t point with zero speed. In the second step, the existing literature has obtained the BoQ curve from the critical points using variational theory (Sun and Ban, 2013; Mehran et al., 2012), fundamental diagram (Hiribarren and Herrera, 2014), linear regression (Izadpanah et al., 2009; Cheng et al., 2012), or piecewise linear fitting (Ramezani et al., 2015).

Although the methods developed in the aforementioned works perform well, there is still room for improvement. For example, most of the aforementioned works (e.g. Ban et al., 2009, 2011; Hao et al., 2012, 2014; Hao and Ban, 2015; Izadpanah et al., 2009; Cheng et al., 2012) rely on the assumption of constant arrival rate (thus a linear BoQ curve) in each signal cycle. However, such assumption cannot capture the variation in demand. In an urban network, the arrival rate to the downstream intersection may be affected by the signal timings of the upstream intersections, and thus is varying. In such cases, relaxing the assumption of a linear BoQ curve may yield better results. (Ramezani et al., 2015; Sun and Ban, 2013; Mehran et al., 2012) aimed to estimate a non-linear BoQ and proposed complex non-convex models, which can be time consuming to find the global optimum. Hiribarren and Herrera (2014) directly obtained the shockwave for each trajectory point using the fundamental diagram, which works for very low penetration rates, but might be sensitive to GPS noises. Furthermore, none of these works explore the queue estimation considering the upstream intersections. The discharging process of the upstream intersections provides additional flow information that can be utilized to improve the accuracy of the queue estimation. Finally, most of these works focus on the queue estimation of undersaturated scenarios, whereas for the purpose of traffic control, oversaturated scenarios are of more significance.

2.2.2 Traffic control of local intersections in a connected and automated vehicle environment

This section presents a short literature review on signal control strategies based on connected and automated vehicle technology. The interested readers can refer to Florin and Olariu (2015) for a comprehensive survey of traffic signal control strategies using connected and automated vehicles from a technology perspective, and Li et al. (2014a), which summarized the general traffic control strategies and highlighted the transition from feedback control to feed-forward control thanks to the connected vehicle technology.

The existing research on intersection control using connected and/or automated vehicles can be classified into two categories. The first category utilizes vehicle-to-infrastructure communication, assuming that these vehicles only serve as information providers. Some studies optimize either signal phases (Priemer and Friedrich, 2009; Hu et al., 2015) or the departure sequence in which individual cars leave the intersection (Wu et al., 2007; Pandit et al., 2013). Most of the early works assume all or a majority of the vehicles are connected. Only a few recent works relax this assumption by taking into consideration incomplete information. The arrival information of conventional vehicles is estimated using either traffic models (He et al., 2012; Guler et al., 2014; He et al., 2014; Feng et al., 2015), statistical methods (Lee et al., 2013), or simulations (Goodall et al., 2013). However, the benefit of the connected and automated vehicles is not fully exploited in this category, as these works assume only uni-directional communication. The second category takes advantage of automated driving and integrates trajectory design into the signal control scheme. Vehicle trajectories can be designed to minimize evacuation time (Li and Wang, 2006), or to provide cooperative control (Lee and Park, 2012). A reservation based strategy that reserves space at the intersection for each car in advance was proposed for automated vehicles only (Dresner and Stone, 2004) and for connected vehicles with human drivers mixed with automated vehicles (Dresner and Stone, 2006). Au and Stone (2010) presented a trajectory planning algorithm for automated vehicles to reduce the number of stops. Li et al. (2014b) presented an online algorithm to optimize vehicle trajectory and traffic signal simultaneously for an intersection of two one-way streets. A rolling horizon optimization model is adopted to identify each control stage. In each stage, trajectory is designed and signal timing plans are enumerated to minimize total delay. However, this work is based on enumeration, which cannot be extended to larger scale problems. Kamal et al. (2015) used model predictive control to coordinate automated vehicles at an unsignalized intersection to maximize the intersection capacity and avoid collision. One limitation in the previous works with this category is that they assume all vehicles to be automated or connected.

Some studies consider a multimodal environment to provide buses with transit signal priority without over-sacrificing the performance of cars in the conflicting approaches. He et al. (2012) developed an algorithm to identify platoons and existing queues, and formulated a mixed integer linear program (MILP) model to optimize signal timings using the queueing information, priority requests of public transport vehicles, and platooning information. Results show a decrease in delay with penetration rates greater than 40%. He et al. (2014) later extended the above model to address the cases with lower penetration rate (only priority eligible vehicles are connected). Hu et al. (2015) proposed another control strategy based on MILP that minimizes total passenger delay for all traffic users at signalized intersections or along a corridor, which is only activated when the bus is behind schedule and no additional delay per passenger is imparted by using this strategy. This strategy was extended to accommodating conflict request in Hu et al. (2016), and tested in a the Smart Road testbed at Blacksburg, Virginia (Lee et al., 2017). However, none of these works have explicitly considered the coordination between the traffic signals with bus stops and the bus schedule.

2.2.3 Aggregated traffic control of large-scale urban traffic networks in a connected vehicle environment

Real-time traffic control strategies at the aggregated level have been receiving significant research attention, in particular the perimeter control (also known as gating). The basic concept is to restrict the incoming flow through traffic signals at the boundaries of a pre-defined region to prevent congestion inside. The controllers of this type are typically designed based on the

MFD, also known in the literature as the Network Fundamental Diagram (NFD), which was initially proposed in Godfrey (1969) and followed in Mahmassani et al. (1987) and Daganzo (2007). Since the demonstration of its existence (Geroliminis and Daganzo, 2008), the MFD has received enormous attention for its elegance, which facilitates traffic control. The most important contribution of the MFD model lies in the fact that the model features the so-called critical accumulation of the network, which serves as the control goal and facilitates the design of various control schemes for traffic signals in real time. It is shown in the literature that MFD-based perimeter control strategies are effective in regulating the global traffic performance in both single/two regions (Haddad and Geroliminis, 2012; Keyvan-Ekbatani et al., 2012; Geroliminis et al., 2013; Haddad et al., 2013; Aboudolas and Geroliminis, 2013; Keyvan-Ekbatani et al., 2015; Haddad, 2015) and multiple regions (Aboudolas and Geroliminis, 2013; Ramezani et al., 2015; Kouvelas et al., 2016). Perimeter control can be viewed as a high-level regional control scheme and may be combined with other strategies (e.g. local or distributed controllers) in a hierarchical control framework (Kouvelas et al., 2016; Daganzo, 2007; Keyvan-Ekbatani et al., 2012; Geroliminis et al., 2013; Aboudolas and Geroliminis, 2013; Ramezani et al., 2015; Kouvelas et al., 2016).

Regardless of what type of control strategies are applied, the optimal flow allowed to enter the network has to be distributed to the local intersections. In other words, as long as the controllers determine the total flow, the flow should be allocated to each individual perimeter intersection through the control of the traffic signals. Nearly all the existing studies emphasize that delay may be caused at local intersections when applying perimeter control (Keyvan-Ekbatani et al., 2012; Geroliminis et al., 2013; Haddad et al., 2013; Aboudolas and Geroliminis, 2013; Hajiahmadi et al., 2015; Haddad, 2017). However, no work quantitatively and systematically treats the delay at these intersections when designing the controllers for network-wide applications. Considering detailed performance of the local intersections may complicate the dynamics of the system, and consequently the optimization problem. Thus, given its challenging complexity, controllers treating specifically the intersections are rarely reported in the literature. Although some initial efforts were made, e.g. combining adaptive traffic signal settings inside the network (Kouvelas et al., 2016) and considering the queue length at the perimeter (Keyvan-Ekbatani et al., 2016; Haddad, 2017), the interaction between the network level perimeter control and the local level intersection control has not been fully considered.

In addition, to the best of our knowledge, there is no work on perimeter control in a connected vehicle environment. Connected vehicles are becoming an important source of information, as the application of connected vehicle technology is expected to keep growing rapidly in most places around the world. Admittedly, traditional data sources such as loop detectors and video cameras would be helpful to measure traffic accumulations (e.g. see Ambühl and Menendez, 2016) for control purposes. However, connected vehicles can provide more anticipated and more diverse information with a wider coverage, particularly suitable for developing multi-scale control solutions. The challenge, nevertheless, lies in the uncertainties due to its limited penetration rates, especially in the near future. Although some works propose robust perimeter control strategies (Geroliminis et al., 2013; Haddad, 2015; Ampountolas et al., 2014), these works assume perfect information on the accumulation of vehicles, thus may not be applicable to the connected vehicle environment, as a biased state estimation would have significant impact on control.

Moreover, none of the aforementioned control strategies provide a scheme to differentiate the incoming vehicles to the network. Vehicles are treated the same in the existing perimeter control schemes, either blocked simultaneously or allowed to enter the network. However, the overall system might benefit from providing certain priority to some vehicles. Although some studies extend the network-wide control to multimodal networks (Ampountolas et al., 2014; Chiabaut, 2015; Ampountolas et al., 2017), they only account for the interactions with public transport, rather than providing priority to public transport vehicles. Furthermore, most developed strategies do not consider the value of time (VOT) of each vehicle. Even in the single mode case, some vehicles have higher occupancies or high-valued riders. Providing priority to such vehicles can increase the social welfare (see, for example, high occupancy vehicle (HOV) lanes(Menendez and Daganzo, 2007)). Therefore, it is essential to develop a general framework to incorporate a priority scheme to perimeter control strategies.

2.3 Research Gaps

Based on the literature review in Section 2.2, we focus on the following research gaps.

For traffic estimation, the information provided by connected vehicles can be further exploited to handle the uncertainties in the transition period. Specifically, to the best of our knowledge:

E1. There is not a computationally efficient method to capture the variation of demand within a signal cycle.

E2. None of the existing works explore the queue estimation reusing the departure information from the upstream intersections.

For traffic control at the local level, to the best of our knowledge:

L1. Few works have integrated the signal timing with trajectory planning of automated vehicles, especially during the transition period where various levels of technologies coexist in the traffic systems.

L2. No work has studied the transit signal priority coordinated with bus stops and bus schedule in a connected vehicle environment.

For traffic control at the network level, to the best of our knowledge:

N1. When designing the perimeter control for network-wide applications, no work has quantitatively and systematically discussed how the macroscopic control decisions can be optimally distributed to the local levels, nor treated the delay at these intersections .

N2. There is no work on perimeter control in a connected vehicle environment. Additionally, the existing robust perimeter control strategies are not suitable to handle the uncertainties due to low penetration rates of connected vehicles.

N3. In a perimeter control scheme, no work has differentiated the incoming vehicles to the network or provided priorities to certain groups of vehicles.

PART I

Traffic Estimation in a Connected Vehicle Environment

Abstract

Traffic estimation aims to obtain the current traffic condition accurately and efficiently based on various data sources, serving as an essential pre-requisite for effective traffic control and management. Recently, connected and automated vehicle technologies has emerged as an alternative to traditional traffic detectors. Equipped with Global Positioning System (GPS) devices and wireless communication systems, connected and automated vehicles are capable of reporting real-time information (e.g. position, speed, acceleration rate and direction) to each other (vehicle-to-vehicle, V2V) or to the roadside infrastructure (V2I), providing a better spatial coverage of information (Herrera et al., 2010; Ambühl and Menendez, 2016).

Various methods have been proposed to estimate traffic flow, travel time, and queue length. Among them, queue estimation is an indispensable component for urban intelligent transportation systems. The queue estimation results can be used for trajectory reconstruction, providing a holistic picture of the urban traffic system. Moreover, these results are an essential input and performance measure to adaptive signal control strategies.

Chapter 3 will investigate the research question of queue estimation in a connected vehicle environment and address two important research gaps established in Section 2.3 (E1 and E2). We will pay particular attention to how the information provided by connected vehicles can be better exploited to improve the accuracy of the queue estimation, especially in the transition period with limited penetration rates. First, unlike most of the previous works that assumed constant arrival demand within a signal cycle, i.e. a linear back of queue (BOQ) curve, this work considers a piecewise linear BoQ curve to account for more practical scenarios with varying demand. Second, the proposed methodology is extended to reuse the estimated departure information from the upstream intersections to further improve the estimation accuracy. Third, this work explicitly handles cases with low penetration rates and low sampling rates, as well as measurement noises. Simulation results show that this treatment significantly improves the estimation accuracy of the queue estimation. Moreover, the estimation problem is formulated into a convex optimization model, ensuring efficient computation. Two online implementation approaches are presented to perform real-time queue estimation. Simulation results show that the proposed framework can be solved within a reasonable time (0.8s), which is sufficient for most real-time applications.

For the reader's convenience, a list of the most important variables is given below.

The variables below are given sets (inputs)

N	set of all equipped vehicles, indexed by n
J	set of all trajectory points, indexed by j
J_n	set of trajectory points associated to vehicle n
P	set of signal cycles, indexed by p
V	set of all trajectory points classified as in the free flow state
V_p	set of trajectory points classified as in the free flow state for cycle p
S	set of all trajectory points classified as in the stopped state
S_p	set of trajectory points classified as in the stopped state for cycle p
I	set of all trajectory points classified as in the intermediate state
I_p	set of trajectory points classified as in the intermediate state for cycle p
A_p	set of trajectory points classified as in the accelerating state for cycle p
D_p	set of trajectory points classified as in the decelerating state for cycle p
B_p	set of BoQ critical points for cycle p
F_p	set of FoQ critical points for cycle p

The variables below are given scalars (inputs)

x_j	location coordinate of trajectory point j
t_j	time coordinate of trajectory point j
v_j	speed coordinate of trajectory point j
r_p	start of the red signal in cycle p
g_p	start of the green signal in cycle p
$\underline{u}$	low speed threshold to detect stopped trajectory points
$\bar{u}$	high speed threshold to detect free-flow trajectory points
k_{jam}	jam density on the link
u_f	free flow speed on the link
w	absolute value of the backward wave speed on the link
T_{step}	time step for the back of queue estimation
π_i	time of the ith time step for the BoQ curve estimation

The variables below are associated with decision variables

α_i	slope of the back of queue curve in the ith time interval
β_i	intercept of the back of queue curve in the ith time interval
$BoQ_p(t)$	Back of queue curve plane due to the red signal in cycle p on the x-t
$FoQ_p(t)$	Front of queue curve due to the red signal in cycle p on the x-t plane

Chapter 3

Queue Estimation in a Connected Vehicle Environment

This chapter is partially based on the following papers.

- Yang, K. and M. Menendez (in press) Queue estimation in a connected vehicle environment: A convex approach, *IEEE Transactions on Intelligent Transportation Systems*. DOI: `10.1109/TITS.2018.2866936`.
- Yang, K. and M. Menendez (2017) A convex model for queue length estimation in a connected vehicle environment, paper presented at the *Transportation Research Board 96th Annual Meeting*.

3.1 Objectives and Contributions

Queue estimation is a crucial aspect of traffic estimation. Existing literature focuses on queuing theory (Comert and Cetin, 2009; Comert, 2013b,a) or LWR kinematic wave theory (e.g. Ban et al., 2009, 2011; Hao et al., 2012, 2014; Hao and Ban, 2015; Ramezani et al., 2015). LWR kinematic wave theory has the advantage to account for the spatial traffic dynamics, and therefore attracts enormous attention. However, as described in Chapter 2.3 (research gaps E1 and E2), only a few works are able to capture the variation in demand (Hao and Ban, 2015; Ramezani and Geroliminis, 2015). One the other hand, to the best of our knowledge, there are no works that explore queue estimation considering the spatial relation between multiple intersections.

To address these research gaps, in this chapter, we propose a convex optimization based methodology to estimate the queue profiles at urban intersections. The obtained queue length will serve as an essential input to the adaptive signal control strategies proposed in Part II and Part III at the local and network levels, respectively. The queue profiles can also be used for trajectory reconstruction, providing holistic information about the traffic system. In particular, we will make the following contributions. Contributions 1)-3) are to better exploit the information provided by connected vehicles, and contribution 4) is to solve pragmatic issues.

1) We relax the widely adopted assumption of a constant arrival rate by estimating a piecewise linear BoQ curve with a convex optimization model. The convexity of the formulated model guarantees a low computational cost in the BoQ estimation.
2) The convex optimization model is further extended to reuse the estimated discharging flow from the upstream intersections to improve the estimation accuracy at the downstream intersection, especially if the penetration rate is low. The effects of platoon dispersion are considered. The proposed methodology does not require the tracking of individual vehicles through multiple intersections, which helps protect privacy.
3) We explicitly handle scenarios with low data quality. On the one hand, an alternative method is proposed to deal with the cases with low sampling rates and low penetration rates by using the acceleration and deceleration information. On the other hand, the determination of the critical points and the BoQ/FoQ (front of queue) curves is based on the whole dataset, which makes the methodology more robust to measurement noises.

4) The proposed methodology is of pragmatic significance. First, it is able to work in both undersaturated and oversaturated scenarios. Second, we propose an online implementation framework, which is both accurate and computationally efficient. Third, the methodology proposed in this chapter is independent of the demand, hence it can handle any demand distribution (known or not).

The chapter is organized as follows. The convex formulation for the queue estimation for a single signalized intersection is proposed in Section 3.2. Section 3.3 utilizes the information provided by the upstream intersections to further improve the accuracy of the queue estimation. Section 3.4 extends the proposed methodology to handle the cases with limited data. Section 3.5 describes the simulations settings, and Section 3.6 shows the queue estimation results on the Next Generation Simulation (NGSIM) dataset (Alexiadis et al., 2004) and the simulated dataset of an arterial of Wehntalerstrasse, Zurich, Switzerland. A sensitivity analysis is performed in Section 3.7. Section 3.8 studies the online implementation of the methodology. Section 3.9 concludes this chapter.

3.2 General Methodology

This section proposes a general methodology for queue estimation at a signalized intersection using connected vehicle technology.

The signal timing plan can be either fixed, actuated or adaptive. In this chapter, it is assumed that the signal timings are available, i.e. the start of the red signal, r_p, and the start of the green signal, g_p, are available for each cycle p. This assumption can be relaxed, however, with classification algorithms (e.g. Hao et al., 2012) or clustering algorithms (e.g. Ramezani et al., 2015).

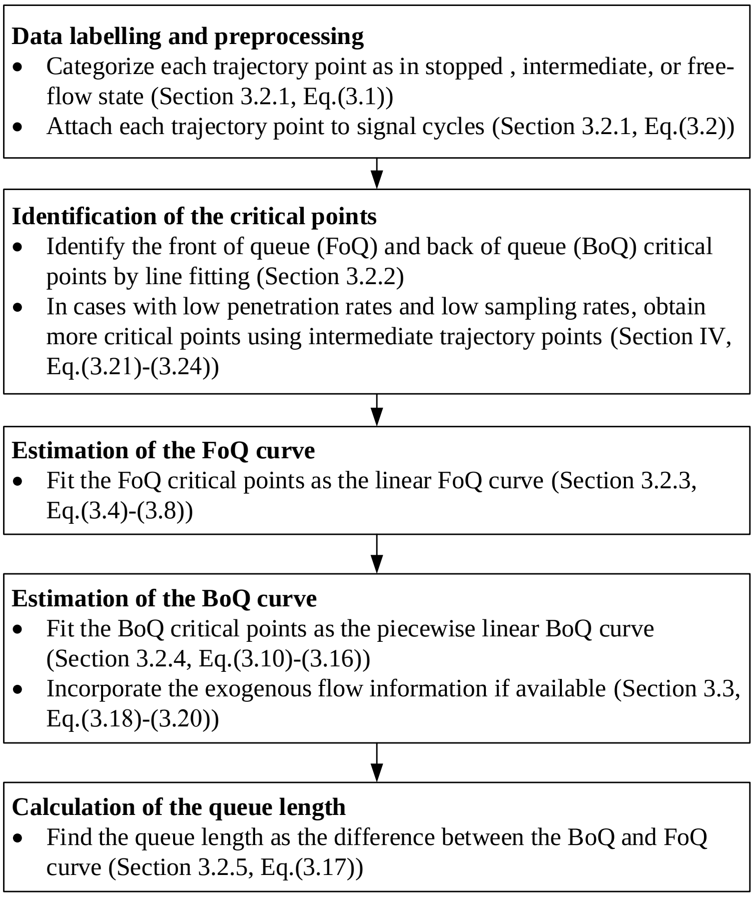
The fundamental diagram of the link is assumed to be triangular with known parameters (free flow speed, jam density, backward wave speed). In practice, the shape and parameters of the fundamental diagram can be calibrated using trajectory data (see Chiabaut et al., 2009).

A certain percentage of vehicles are assumed to be connected vehicles, which may include automated vehicles. It is assumed that these vehicles are able to communicate with the intersection controller if they are within a certain range of it. This range is upper bounded by the physical limit of the DSRC communication range and the length of the links (as otherwise route choice of the vehicles will be needed). On the other hand, for privacy concern, it is desirable if the trajectories of connected vehicles are not tracked through multiple intersections. Therefore, this range is assumed to be the minimum of the city block size and the DSRC communication range. Connected vehicles send trajectory information (time, location and speed) to the controller at some sampling rates, which do not have to be constant for each trajectory, and do not need to be synchronized in time.

In this section, it is assumed that there is at least one stopped trajectory point and one moving trajectory point for each connected vehicle. This assumption is made for presentation simplicity here, and will be relaxed in Section 3.4.

The set of trajectory points is defined as J. The trajectory information is defined as a vector (t_j, x_j, v_j), where the three coordinates represent current time, location, and speed, respectively. In this chapter, the location of a vehicle j, x_j, is assumed to be 0 at the intersection, and a negative value if the vehicle is on the upstream link of the intersection.

Figure 3.1 shows the flowchart of this methodology. Given a set of trajectory points, the methodology estimates the queue length in five steps: 1) data labelling and preprocessing; 2) identification of the critical points; 3) estimation of the FoQ curve; 4) estimation of the BoQ curve; and 5) calculation of the queue length. The rest of this section presents the main parts of the five steps in detail. The other parts, i.e. incorporating the flow information for BoQ estimation and including the intermediate states, will be presented in Section 3.3 and Section 3.4, respectively. An illustration of the process is given in Figure 3.2.



Data labelling and preprocessing
- Categorize each trajectory point as in stopped , intermediate, or free-flow state (Section 3.2.1, Eq.(3.1))
- Attach each trajectory point to signal cycles (Section 3.2.1, Eq.(3.2))

Identification of the critical points
- Identify the front of queue (FoQ) and back of queue (BoQ) critical points by line fitting (Section 3.2.2)
- In cases with low penetration rates and low sampling rates, obtain more critical points using intermediate trajectory points (Section IV, Eq.(3.21)-(3.24))

Estimation of the FoQ curve
- Fit the FoQ critical points as the linear FoQ curve (Section 3.2.3, Eq.(3.4)-(3.8))

Estimation of the BoQ curve
- Fit the BoQ critical points as the piecewise linear BoQ curve (Section 3.2.4, Eq.(3.10)-(3.16))
- Incorporate the exogenous flow information if available (Section 3.3, Eq.(3.18)-(3.20))

Calculation of the queue length
- Find the queue length as the difference between the BoQ and FoQ curve (Section 3.2.5, Eq.(3.17))

Figure 3.1: Flowchart of the proposed general methodology.

3.2.1 Data labelling and preprocessing

In this step, the trajectory points will be labelled as in a stopped state, free-flow state, or intermediate state, and then attached to each signal cycle.

A trajectory point is identified as stopped if the speed is lower than a critical value $\underline{u}$, and free-flow if the speed is higher than a critical value $\bar{u} > \underline{u}$. Define set S as the set of all trajectory points in the stopped state, set V as the set of all trajectory points in the free-flow state. The other trajectory points are identified as in the intermediate state (i.e. accelerating/decelerating state), denoted as set I. This can be summarized as

$$j \in \begin{cases} S, & \text{if } v_j \le \underline{u} \\ I, & \text{if } \underline{u} < v_j \le \bar{u} \\ V, & \text{if } v_j > \bar{u} \end{cases} \tag{3.1}$$

In this section, intermediate trajectory points are not considered for simplicity. However, these trajectory points can be useful if there are no stopped or free-flow trajectory points for a connected vehicle trajectory, which will be discussed in Section 3.4.

Each trajectory point (stopped, free-flow, or intermediate) is attached to a signal cycle. This can be described in Eq.(3.2). Note that different trajectory points of a vehicle may be attached to different cycles, if this vehicle queues for multiple cycles. In other words, a vehicle can be

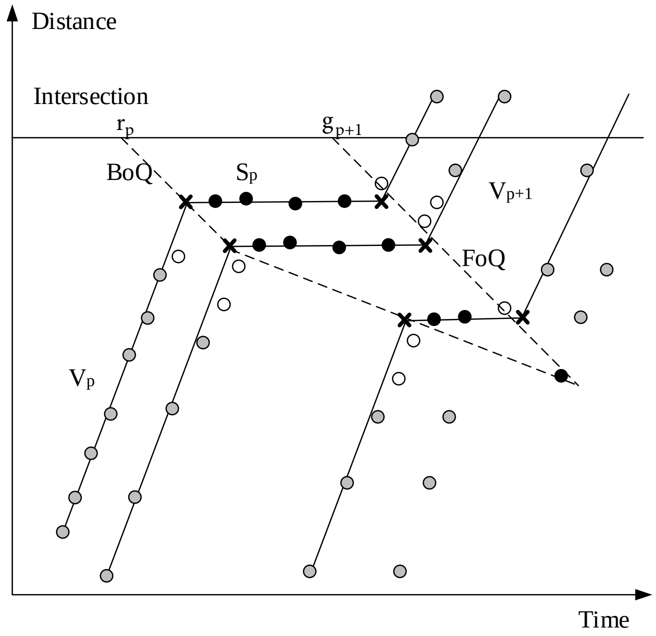


Figure 3.2: Illustration of the methodology. The gray dots are marked as free-flow, the white dots are in the intermediate state, the black dots are stopped. The points with cross marks represent the critical points. The solid lines represent the reconstructed theoretical trajectories; the dashed lines represent the front of queue (FoQ) curve and the back of queue (BoQ) curve.

included in the queue of multiple cycles.

$$j \in \begin{cases} V_p, \text{ if } g_p \leq x_j/w + t_j < g_{p+1} \text{ and } j \in V \\ I_p, \text{ if } g_p \leq x_j/w + t_j < g_{p+1} \text{ and } j \in I \\ S_p, \text{ if } g_p \leq x_j/w + t_j < g_{p+1} \text{ and } j \in S \end{cases} \tag{3.2}$$

where V_p, I_p and S_p represent the set of free-flow, intermediate, and stopped trajectory points in a signal cycle, respectively.

Notice that in scenarios with extremely sparse data (i.e. with low sampling rates or low penetration rates), there could be large error in identifying the signal cycle of the trajectory point. In such cases, it can be beneficial to incorporate probabilistic methods into the proposed methodology.

3.2.2 Identification of the critical points

Critical points represent the transition between two traffic states. As is shown in Figure 3.2, the FoQ critical points separate the stopped state in cycle p and the free-flow state in cycle $p+1$, and represents the time and location where a vehicle starts to be discharged. The BoQ critical points separate the free-flow state in cycle p to the stopped state in cycle p, and represents the process where a vehicle comes to a stop.

Let J_n be the trajectory points reported by the connected vehicle n. Then for cycle p, the set $J_n \cap V_p$ represents the free-flow trajectory points of connected vehicle n in this cycle; the set $J_n \cap S_p$ represents the stopped trajectory points of connected vehicle n in this cycle.

The FoQ critical point for trajectory n in signal cycle p can be calculated as the intersect between the lines formed by trajectory points in $J_n \cap S_p$ (the stopped points for trajectory n in signal cycle p) with a slope of 0 and $J_n \cap V_{p+1}$ (the moving points for trajectory n in signal cycle

$p + 1$) with a slope of u_f. Both lines can be derived using linear regression, and the critical points can be identified by solving the linear equations. Denote the set of FoQ critical points in cycle p as F_p.

For trajectory n and signal cycle p, each BoQ critical point is determined as the intersect between two lines: 1) the line fitted on set $J_n \cap V_p$ (the moving points of vehicle n for trajectory in signal cycle p) with a slope of u_f; and 2) the line fitted on set $J_n \cap S_p$ (the stopped points of vehicle n for trajectory in signal cycle p) with a slope of 0. Denote the set of BoQ critical points in cycle p as B_p.

This definition of the critical points approximates each vehicle trajectory as a piecewise linear curve of stopped segments and free-flow segments. This is consistent with the kinematic wave theory with the assumption of a triangular fundamental diagram. This approximation is also invariant with respect to vehicle delay.

3.2.3 Estimation of the FoQ curve

As is shown in Figure 3.2, the FoQ curve between cycles p and $p + 1$ is the discharging line in cycle p. It is a straight line that i) separates the stopped trajectory points in cycle p, S_p, and the free-flow trajectory points in cycle $p + 1$, V_{p+1}; ii) crosses the critical points in set F_p between S_p and V_{p+1}; iii) has a slope of $-w$, where w is the absolute value of the backward wave speed.

Denote the FoQ curve as a line on the $x - t$ plane represented by

$$x + wt - h = 0 \tag{3.3}$$

where h is the decision variable, representing the intercept on the distance axis, x. Then estimating the FoQ curve is equivalent to determining the intercept h that satisfies the criteria listed above.

Hence, the problem of estimating FoQ can be formulated into the following optimization model Eq.(3.4)-Eq.(3.8).

$$\min \sum_{j \in F_p} \epsilon_j^2 + \lambda_1 \sum_{j \in S_p} e_j + \lambda_2 \sum_{j \in V_{p+1}} e_j \tag{3.4}$$

$$\text{s.t. } x_j + wt_j - h = \epsilon_j, \ j \in F_p \tag{3.5}$$

$$x_j + wt_j - h \le e_j, \ j \in S_p \tag{3.6}$$

$$x_j + wt_j - h \ge -e_j, \ j \in V_{p+1} \tag{3.7}$$

$$e_j \ge 0, \ j \in V_{p+1} \cup S_p \tag{3.8}$$

where the decision variable is the intercept h. Constraint Eq.(3.5) defines the error of fitting the critical points with Eq.(3.3); Eq.(3.6) and Eq.(3.7) guarantee that the discharging line separates the stopped and moving states. The slack decision variables e_j are introduced to handle the case where the stopped state and the free flow state are not strictly separable by straight lines. This could happen in scenarios with multiple lanes, or with large measurement errors. The objective function Eq.(3.4) minimizes the summed square fitting errors and classification errors. λ_1 and λ_2 are regularization factors that control the weight of the the classification errors. If λ_1 and λ_2 are small, the model gives more emphasis to fitting than to classification. Besides, if $\lambda_1 > \lambda_2$, the model focuses more on the stopped state.

The optimization model Eq.(3.4)-Eq.(3.8) is convex, which can be solved in polynomial time with the interior-point method or Newton's method. Solving this model gives the discharging line (i.e. FoQ curve) $x + wt - h = 0$.

Note that the formulation Eq.(3.4)-Eq.(3.8) relies on the underlying assumption that there is at least one FoQ critical point. In other words, it is assumed in this section that there exists at least one vehicle n such that both $V_{p+1} \cap J_n$ and $S_p \cap J_n$ are not empty. This assumption will be relaxed in Section 3.4.

3.2.4 Estimation of the BoQ curve

According to LWR theory, the BoQ curve is modelled as a piecewise linear curve $x = x(t)$ that crosses the critical points in set B_p and separates the moving state V_p and the stopped state S_p (see Figure 3.2). All the trajectory points in V_p should be below the BoQ curve, and all the trajectory points in S_p should be above the BoQ curve.

The estimation of the BoQ curve can be formulated as a classification problem with a piecewise linear boundary. In this chapter, the piecewise linear boundary is determined using the algorithm proposed in Huang et al. (2013). The key idea of this model is to divide the region of interest into some subregions and find a classification line in each region through a global optimization model. For the BoQ curve estimation problem, the time region of interest Π is chosen as the time period between the start time of a cycle to the time when the queue is cleared. The BoQ curve begins from the start time of the red signal in cycle p, r_p. The time when the BoQ curve ends (the queue is cleared) is generally unknown. It can be chosen as a time sufficiently large. For real-time queue estimation, the end time of the region can be the current time. Then the region of interest is divided into time intervals $\Pi_i = \{j|\pi_i \le t_j < \pi_{i+1}\}, i = 1, \cdots, m$, where π_1 is the start time of Π and π_{m+1} is the end time of Π. The length of each time interval $T_{\text{step}} = \pi_{i+1} - \pi_i$ determines the precision of the BoQ curve. If this size is too large, it might give an inaccurate BoQ. If this size is too small, it might lead to over-fitting. A sensitivity analysis on the time interval length is conducted in Section 3.7.

In this way, the BoQ curve due to the red signal in cycle p can be represented by a continuous piecewise linear curve $BoQ_p(t)$ on the $x - t$ plane as

$$x(t) = BoQ_p(t) = \begin{cases} \alpha_1 t + \beta_1, & \pi_1 \le t < \pi_2 \\ \cdots & \\ \alpha_n t + \beta_n, & t \ge \pi_n \end{cases} \tag{3.9}$$

where $\alpha_i, \beta_i, i = 1, \cdots, n$ are the slopes and intercepts in each time interval.

The estimation of the BoQ curve can be modelled as

$$\min \frac{1}{2} \sum_{j \in B_p}^{m} \epsilon_j^2 + \lambda_1 \sum_{j \in S_p} e_j + \lambda_2 \sum_{j \in V_p} e_j + \lambda_3 \sum_{i=2}^{m} |\alpha_i - \alpha_{i-1}| \tag{3.10}$$

$$\text{s.t. } \alpha_{i-1}\pi_i + \beta_{i-1} = \alpha_i \pi_i + \beta_i,\ i = 2, \cdots, m \tag{3.11}$$

$$x_j - BoQ_p(t_j) = \epsilon_j,\ j \in B_p \tag{3.12}$$

$$x_j - BoQ_p(t_j) \ge -e_j,\ j \in V_p \tag{3.13}$$

$$x_j - BoQ_p(t_j) \le e_j,\ j \in S_p \tag{3.14}$$

$$e_j \ge 0,\ \forall j \in V_p \cup S_p \tag{3.15}$$

$$0 \le \alpha_i \le w \tag{3.16}$$

Constraint Eq.(3.11) guarantees that the $(i-1)$th and the ith pieces of $BoQ_p(t)$ have the same value at π_i, so that the BoQ curve is continuous. Constraint Eq.(3.12) represents the fitting errors of the BoQ curve on the critical points. Constraints Eq.(3.13)-Eq.(3.15) imply that the BoQ curve should separate the stopped trajectory points and the moving trajectory points. A penalty $e_j > 0$ is introduced if the trajectory point (t_j, x_j) cannot be classified into the correct state (i.e. moving or stopped). Constraint Eq.(3.16) is derived from the LWR kinematic wave theory, and guarantees that the slope of each piece of the BoQ curve is within the backward wave speed and 0. The objective function Eq.(3.10) aims to minimize the weighted error of fitting. The first term represents the total fitting error caused by the critical points; the second and the third term represent the total penalty caused by the failure to separate the moving and the stopped states; the fourth term attempts to control the number of linear segments in the BoQ to give a more robust solution. This is done by minimizing non-zero components in $\{\alpha_i - \alpha_{i-1}\}$, i.e. to make α_i and α_{i-1} as close as possible. Since the number of non-zero components in $\{\alpha_i - \alpha_{i-1}\}$ is not a convex function, an l_1 regularization term $\sum_{i=2}^{m} |\alpha_i - \alpha_{i-1}|$ is minimized instead as a convex approximation (1-norm approximation). By minimizing $\sum_{i=2}^{m} |\alpha_i - \alpha_{i-1}|$, the number of non-zero components in the set $\{\alpha_i - \alpha_{i-1}, \forall i\}$ can be approximately minimized. This technique is called $l1$-magic and often used in signal processing to reconstruct sparse signals (signals in which most components are zero). It is shown in Candes and Romberg (2005) that this technique always yields a good approximation.

The regularization factors λ_1, λ_2 and λ_3 control how much weight should be given to each term of the objective function. Large λ_1 and λ_2 represent less tolerance on the misclassification of the moving and stopped states. Different weights λ_1 and λ_2 can also be given, if the importance of misclassification of the moving and stopped states are different. Large λ_3 will result in small number of breaks (i.e. segments) in the BoQ curve. These three parameters can be trained from data (including ground truth queue length) by a cross-validation procedure.

Note that the formulation Eq.(3.10) - (3.16) is convex. Hence, the model can be solved easily with any convex optimization solver using the interior-point method or Newton's method.

3.2.5 Calculation of the queue length

With the BoQ curve and FoQ curve due to the red signal in cycle p, the queuing region can be identified. The remaining queue from cycle p can be then calculated as the difference between them. Then, the queue length at time t is the summation of the remaining queues, i.e.

$$Q(t) = \sum_{p} \max\{FoQ_p(t) - BoQ_p(t), 0\} \tag{3.17}$$

3.3 Reuse of Upstream Departure Information

In an urban network, if the intersections can communicate with each other, the departure information from the upstream intersection, calculated from the FoQ curve, can be beneficial for the queue estimation of the downstream intersection, especially in scenarios with low penetration rates. However, due to consideration of privacy and data transmission, it is not desirable to track the actual vehicle trajectories or share them between intersections. In this section, we extend the general methodology proposed in Section 3.2 to integrate the estimated departure information from the upstream intersections to further improve the estimation accuracy at the network level while retaining vehicle privacy. This extended methodology follows two steps. In the first step, the arrival flow at the downstream intersection is estimated based on a platoon dispersion model (Section 3.3.1); in the second step, such flow information is utilized for queue estimation (Section 3.3.2).

3.3.1 Estimation of the arrival flow

Denote q_i as the flow that would have arrived at the intersection at time interval $\Pi_i = \{j | \pi_i \leq t_j < \pi_{i+1}\}$ in cycle p. We aim to estimate the flow q_i considering the information of the upstream intersection and the effect of platoon dispersion.

There are two cases based on whether the cycle $p - 1$ at the downstream intersection is oversaturated.

In the first case, the flow q_i consists of the vehicles queued in cycle $p-1$, hence q_i is the saturation flow of the intersection.

In the second case, the flow q_i has not queued in the previous cycle. Instead, it comes from the upstream intersection. In an urban network, the discharging line at the upstream intersection provides prior arrival information to the downstream intersections. The departure flow from the upstream intersection can be zero (red time), the saturation flow (discharging), or the arrival flow at the intersection. However, due to the fluctuation of vehicle speeds, the discharging platoons from the upstream intersection tend to disperse over time and space. Hence, the vehicles discharged uniformly from the upstream signal would arrive at the downstream signal in a non-uniform manner, which makes the arrival flow at the downstream intersection different from the departure flow at the upstream intersection.

Many models depict the platoon dispersion (Grace and Potts, 1964; Qiao et al., 2001; Robertson, 1969). In particular, the most widely used platoon dispersion model is Robertson (1969), which has become a virtual universal standard platoon dispersion model and has been implemented in various traffic simulation software, including TRANSYT (Robertson, 1969), SCOOT (Hunt et al., 1981) and TRAFLO (Lieberman and B.J., 1980). In this chapter, the Robertson's platoon dispersion model (Robertson, 1969) is adopted to model the progression of vehicles between the two intersections. Note that other platoon dispersion models can also be incorporated in a similar way.

The Robertson's platoon dispersion model states that the downstream flow should be the linear combination of the previous downstream flow, and the corresponding upstream flow, i.e.

$$q_i = \frac{\rho T_f}{1+\rho T_f} q_{i-1} + \frac{1}{1+\rho T_f} q'_{i-T_f} \tag{3.18}$$

where

- q_i is the arrival flow at the downstream intersection at time step i;
- q'_i is the departure flow from the upstream intersection at time step i;
- T_f is the free flow travel time between the two intersections with unit of time steps;
- ρ is a platoon dispersion factor expressing the degree of the dispersion of the platoon, which can be calibrated with empirical data (Robertson, 1969).

The parameters T_f and ρ are constants if the configuration of the two intersections is given. The flow q'_{i-T_f} is given by the upstream intersection. The arriving flow at the downstream intersection can be calculated by Eq.(3.18).

3.3.2 Estimation of the queue length

Using the estimated arrival flow q_i and the kinematic wave theory, the slope of the BoQ curve at time interval Π_i can be calculated as

$$\hat{\alpha}_i = \frac{q_i}{q_i/u_f - k_{\text{jam}}} \tag{3.19}$$

Then a regularization term can be added to Eq.(3.10) to integrate the information provided by the trajectory data and the flow data.

$$\min \frac{1}{2} \sum_{j\in B_p}^{m} \epsilon_j^2 + \lambda_1 \sum_{j\in S_p} e_j + \lambda_2 \sum_{j\in V_p} e_j + \lambda_3 \sum_{i=2}^{m} |\alpha_i - \alpha_{i-1}| + \gamma \sum_{i=1}^{m} \max\{\hat{\alpha}_i - \alpha_i, 0\} \tag{3.20}$$

where γ is a weighting parameter balancing the trajectory data and the flow information. It represents our belief on the flow information. The more we trust the flow information and the more noisy the trajectory data is, the larger the γ should be. For low penetration rate and large noises in data, a large γ should be utilized. The regularization term aims to ensure that the $\hat{\alpha}_i \leq \alpha_i$. This is because the proposed methodology tends to underestimate the duration of the FoQ curve if the penetration rate is low and the last few vehicles in the queue are not connected. Therefore, the estimated slope $\hat{\alpha}_i$ is usually lower than the actual one α_i.

3.4 Inclusion of the Intermediate States for Cases With Limited Data

This section extends the general methodology to explicitly handle the cases with low penetration rates and low sampling rates. One example of such cases is queue length estimation with probe taxi data or floating car data covering around 10% of the vehicles and reported every 10-30s. The potential problem in these cases is that there might not be enough trajectory points for the FoQ and BoQ estimation strictly following the methodology proposed in Section 3.2. Recall that in order to estimate the FoQ or the BoQ curve in each cycle, we need at least one critical point. When the data is limited, however, there might not be a critical point for some cycles at all. This can be problematic, as the accurate estimation of both curves depends on the number of critical points we have. Thus, it is desirable to get as many critical points as possible. In this section, the data in the intermediate state (i.e. trajectory points with speed between $\underline{u}$ and $\bar{u}$) are used to provide information on acceleration and deceleration. With such information, more critical points can be estimated.

The estimation of critical points using the intermediate trajectory points follows two steps.

1. Identify the set of accelerating and decelerating trajectory points in each signal cycle p, respectively.
2. Estimate the critical points by fitting the acceleration and deceleration curves.

In the first step, denote A_p and D_p as the set of accelerating and decelerating trajectory points in signal cycle p, respectively. These trajectory points in A_p and D_p are determined by traversing each trajectory. Recall that in a normal cycle, the trajectory points are very likely to follow the order of states: accelerating, free flow state, decelerating, stopped state. Hence, considering two consecutive trajectory points in the same cycle where the earlier point j is an intermediate trajectory point ($j \in I_p$), and the later point j' is a free flow trajectory point ($j' \in V_p$) or an accelerating trajectory point ($j' \in A_p$) , we can deduce that the earlier trajectory point j is accelerating and assign it into A_p; Otherwise if an intermediate trajectory point $j \in I_p$ is right after a free flow trajectory point $j' \in V_p$ or a decelerating point $j' \in D_p$ in the same cycle, the trajectory point j should be decelerating and assigned into D_p. In the absence of free flow trajectory points (e.g. when the sampling rate is very low), it is more difficult to distinguish the accelerating and decelerating trajectory points. In such cases, we use the speed information. In cycle p, if the speed of a trajectory point is larger than the previous point along the same trajectory, we classify it as an accelerating trajectory point and assign it into set A_p. Otherwise we regard it as a decelerating trajectory point and assign it into set D_p.

It is assumed that all the vehicles have a uniform and constant acceleration rate a and deceleration rate d. Historical trajectory points can be used to calibrate both a and d. For a historical trajectory point j with speed v_j, denote Δx_j as the distance from the location of this trajectory point to the location where the vehicle stops (if available), then for each trajectory point j, it holds that

$$2a\Delta x_j = v_j^2 \text{ or } -2d\Delta x_j = v_j^2 \tag{3.21}$$

Therefore, a and d can be estimated as

$$a = \frac{\sum_{j \in \bar{A}} v_j^2}{2 \sum_{j \in \bar{A}} \Delta x_j} \tag{3.22}$$

$$d = -\frac{\sum_{j \in \bar{D}} v_j^2}{2 \sum_{j \in \bar{D}} \Delta x_j} \tag{3.23}$$

where $\bar{A}$ and $\bar{D}$ denote the trajectory points in A and D if Δx_j is available, respectively. Note that even in scenarios with very low penetration rates and sampling rates, given a sufficiently long period of time, we can still obtain enough trajectory points for $\bar{A}$ and $\bar{D}$.

In the second step, we estimate the critical points using data points in A_p and D_p. The estimation of the critical points of the FoQ and BoQ curve with intermediate trajectory points are similar. In the rest of this section, for simplicity, we only present the estimation of the FoQ critical points. To adapt the following approach to the estimation of the BoQ curve, we only need to replace A_p and a with D_p and d, respectively.

With the constant acceleration assumption, the accelerating trajectory can be represented as a parabola.

$$x(t) = \frac{1}{2}at^2 + bt + c \tag{3.24}$$

Then the estimation of the FoQ critical points follows two steps. In the first step, the accelerating trajectory Eq.(3.24) is estimated using the intermediate trajectory points in the set A_{p+1}. In the second step, the critical points are determined with the assistance of Eq.(3.24). For the simplicity of presentation, the details will be omitted here and can be found in Appendix A.1.

3.5 Simulation Settings

This section describes the two datasets for evaluation: 1) the Lankershim dataset from the NGSIM project for undersaturated scenarios, and 2) the simulated data for oversaturated scenarios based on an arterial of Wehntalerstrasse, Zurich, Switzerland.

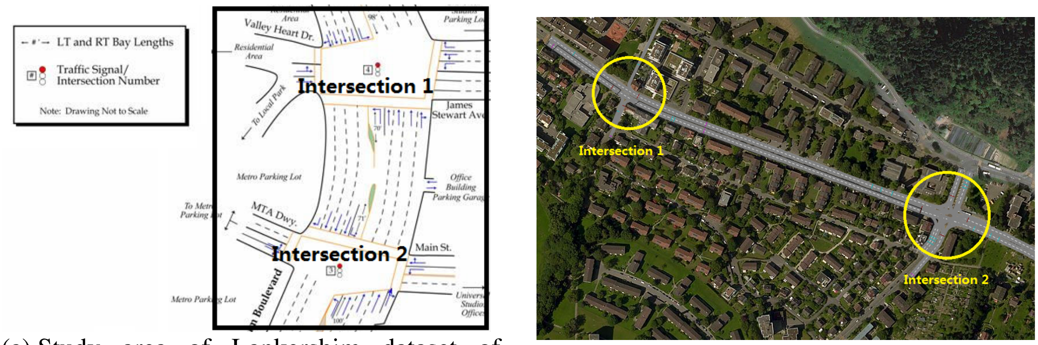
In the Lankershim dataset from the NGSIM project, all lanes with straight movements are considered (2 lanes). The study area is shown in Figure 3.3(a). The vehicle trajectory data correspond to the southbound trajectories from 8:30 a.m. to 8:45 a.m. (10 cycles) on June 16, 2005, see in Figure 3.3(c).

This dataset (thereafter called Wehntalerstrasse dataset) is based on the simulation of an arterial at Wehntalerstrasse, Zurich, Switzerland using VISSIM. We consider two intersections: 1) the intersection between Wehntalerstrasse and Einfangstrasse (Intersection 1, upstream), and 2) the intersection between Wehntalerstrasse and Glaubtenstrasse (Intersection 2, downstream). The considered arterial link has two lanes. The layout of the simulation is shown in Figure 3.3(b). This simulation model is calibrated and validated based on empirical measurements on Nov. 16, 2014. The simulated trajectory is illustrated in Figure 3.3(d) (13 cycles).

The parameters for both scenarios are summarized in Table 3.1.

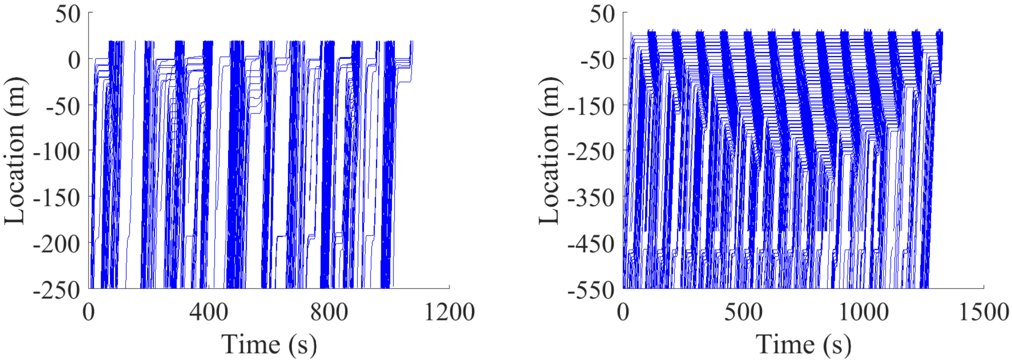
The performance of the proposed methodology is evaluated by the mean estimation error, i.e. the average absolute difference between the estimated queue length and the actual queue length, i.e.

$$MAE = \frac{1}{t_f - t_0} \int_{t_0}^{t_f} |Q(t) - \hat{Q}(t)| \tag{3.25}$$



(a) Study area of Lankershim dataset of NGSIM.

(b) Study area of Wehntalerstrasse dataset.



(c) Vehicle trajectories in the Lankershim dataset.

(d) Vehicle trajectories in the Wehntalerstrasse dataset.

Figure 3.3: Study area and vehicle trajectories of the simulation.

Table 3.1: Parameters for simulation.

Parameter	Lankershim dataset	Wehntalerstrasse dataset
u_f [km/hr]	60	60
w [km/hr]	23.8	16.8
k_{jam} [veh/km]	200	250
T_f	17	40
γ	0.5	0.5
ρ	0.1	0.1
λ_1	1	1
λ_2	1	1
λ_3	0.5	0.5
$\underline{u}$ [m/s]	1	1
$\bar{u}$ [m/s]	5	5
T_{step}[s]	2	2

where $\hat{Q}(t)$ is the estimated queue length at time t and $Q(t)$ is the ground truth queue length.

For the Lankershim dataset, the average queue length over time of the studied area is 3.0 cars, which means that if we estimate the queue length always as zero, we will get an MAE of 3.0 cars. For the Wehntalerstrasse dataset, the average queue length over time is 25.0 cars.

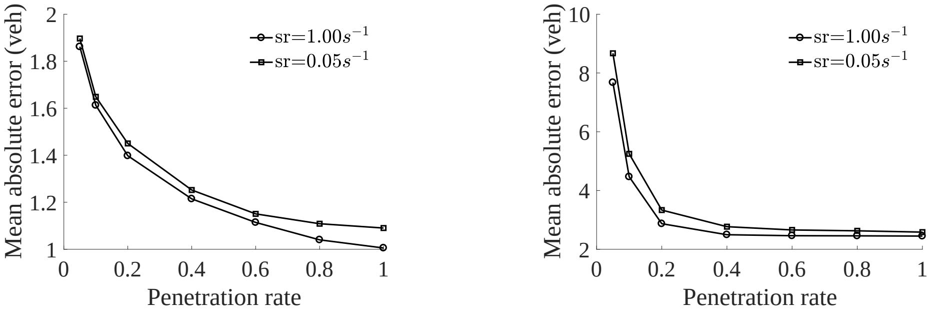
The model is solved using MATLAB with CVX toolbox (Grant and Boyd, 2014, 2008), a solver for convex optimization problems. The algorithm used by CVX tool box is the interior-point method (Kim et al., 2007). 10 random seeds are evaluated for each comparison.

3.6 Case Study and Results

This section shows the performance of the proposed methodology at both sites. Section 3.6.1 shows the results for an isolated intersection. Section 3.6.2 evaluates the benefit of considering a piecewise linear BoQ instead of a linear BoQ. Section 3.6.3 demonstrates the benefit of considering the trajectory points in the intermediate state. The value of considering flow information is discussed in Section 3.6.4.

3.6.1 Results for an isolated intersection

In this section, the proposed methodology is evaluated at Intersection 2 in Figure 3.3(a) and Figure 3.3(b) for the undersaturated case and the oversaturated case, respectively. The resulting MAEs are shown in Figure 3.4. The penetration rate is set to vary between 0.05 and 1. The sampling rate (sr) is chosen to vary between 0.05s^{-1} and 1s^{-1}. We found that the performance of the proposed methodology is not very sensitive to the sampling rate when the sampling rate is higher than 0.2s^{-1}. Therefore for illustration convenience, we only show the results with sampling rates 0.05s^{-1} and 1s^{-1}. We can also see from Figure 3.4 that the proposed methodology is more sensitive to the penetration rate (pr).



(a) Lankershim dataset

(b) Wehntalerstrasse dataset

Figure 3.4: Performance of the proposed methodology for an isolated intersection.

It can be seen from Figure 3.4(a) and Figure 3.4(b) that the estimation error decreases with the increase in penetration rate. The proposed methodology performs well for a relatively low penetration rate (0.1) and a low sampling rate (0.05s^{-1}), with the MAE less than 1.5 car for the Lankershim dataset and 5.2 cars for the Wehntalerstrasse dataset. This is expected, as for the undersaturated scenarios, the average queue length is small. Therefore, the error cannot be too large. For the oversaturated scenarios, however, as the absolute number of vehicles is large, having a penetration rate of 0.1 still gives sufficient critical points. However, the performance of the proposed methodology deteriorates as the penetration rate drops from 0.1 to 0.05, This is expected, because there are very few critical points, or even vehicle trajectories, in many cycles for such a low penetration rate. In such scenarios, it is hard to use deterministic methods to estimate the queue using only connected vehicle data. One solution could be stochastic models that can provide more robust results(Comert, 2013a; Hao and Ban, 2015). We may also need other sources of information for queue length estimation in such scenarios.

The marginal benefit of having more connected vehicles decreases with the increase in the penetration rate. This suggests that the information provided by 40% can already yield satisfactory estimation results. We can also observe an error of on average 1 or 2 cars even for the 100% penetration rate and sampling rate of 1s^{-1}. This is because the driver behaviors in reality are stochastic and heterogeneous. The drivers do not exactly behave according to the traffic models. Note that here we assume the real penetration rate is not available to the proposed methodology. In addition, we perform the queue estimation for both lanes as a whole, which relies on the assumption that the queue length on both lanes are similar. However, the queue length may not be balanced on both lanes in reality. We may adapt the proposed methodology to lane-based queue estimation to improve the accuracy.

Comparing Figure 3.4(a) and Figure 3.4(b), we can see that the difference of the MAE between the two cases is small for moderate penetration rates (larger than 0.4). This is expected, as the estimation error of the queue length usually depends more on the last few vehicles in the queue, rather than the actual queue length. If the last few vehicles are all conventional vehicles, we tend to underestimate the queue length. Therefore, the estimation errors in the undersaturated scenarios and oversaturated scenarios are similar.

To get more detailed understanding of the proposed methodology, we demonstrates the estimation of queue lengths of both sites with penetration rates (pr) of 0.2 and 1.0 for a particular random seed (42) in Figure 3.5. In the time-space diagram (Figure 3.5(a), Figure 3.5(c), Figure 3.5(e), and Figure 3.5(g)), the green dashed lines represent the trajectories that provide the ground truth, whereas the blue solid lines represent the trajectories of connected vehicles that we use for queue estimation. It can be seen that the proposed methodology is able to accurately estimate the FoQ and BoQ curve based on the information available. In scenarios with low penetration rates (Figure 3.5(a), Figure 3.5(b), Figure 3.5(e), and Figure 3.5(f)), the proposed methodology might fail to estimate the last few vehicles due to the lack of information and tend to underestimate the queue length for scenarios with relatively low penetration rate. In fact, even if the queue profiles are perfectly estimated, there could still be errors in the queue estimation due to the imbalance of the queue length on different lanes (Figure 3.5(c), Figure 3.5(d), Figure 3.5(g), and Figure 3.5(h)). Comparing Figure 3.5(b) and Figure 3.5(d) with Figure 3.5(f) and Figure 3.5(h), we can also see that the queue estimation process is more sensitive to low penetration rate in undersaturated scenarios than in oversaturated scenarios. This is The estimation error can be large, if very few vehicles arrive at the intersection, e.g. the fourth and sixth cycle in Figure 3.5(a).

The cycle-by-cycle maximum queue length is shown in Table 3.2. It can be seen that the proposed proposed methodology can in general successfully estimate the FoQs, BoQs and queue lengths, even if the penetration rate is low.

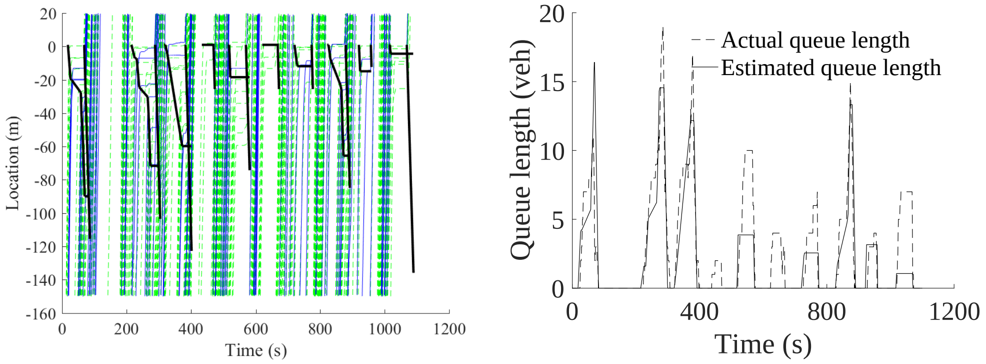
3.6.2 Value of considering piecewise linear BoQ

In order to evaluate the benefit of considering the piecewise linear BoQ curve, we compare the results with a state-of-art method (Ban et al., 2011) that assumes constant demand within a cycle, and thus a linear BoQ between the maximum queue length and the minimum queue length[1]. However, note that the demand in both datasets is not constant within a signal cycle, because it is affected by the traffic signal in the downstream intersection. In scenarios with a more uniform demand within the signal cycle, the two methods yield similar results. We test the scenarios with penetration rates (pr) of 0.2 and 0.8, and sampling rate (sr) of 1s^{-1}, 0.2s^{-1} and 0.05s^{-1}. The results are shown in Figure 3.6. The results for the proposed method and the method in Ban et al. (2011) are shown in the blank bar and the shaded bar, respectively. Note that the method in Ban et al. (2011) is based on sample travel times, assuming that the arrival times on two virtual lines are available. Therefore, the results for different sampling rates are the same.

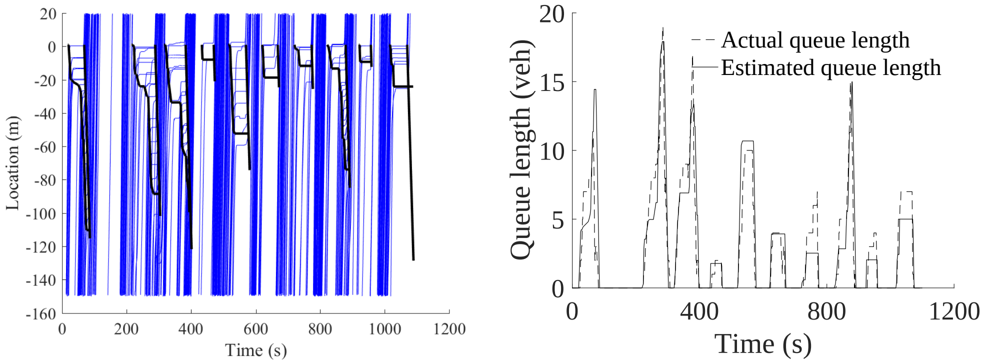
It can be seen that considering a piecewise linear BoQ can improve the accuracy in estimation

---

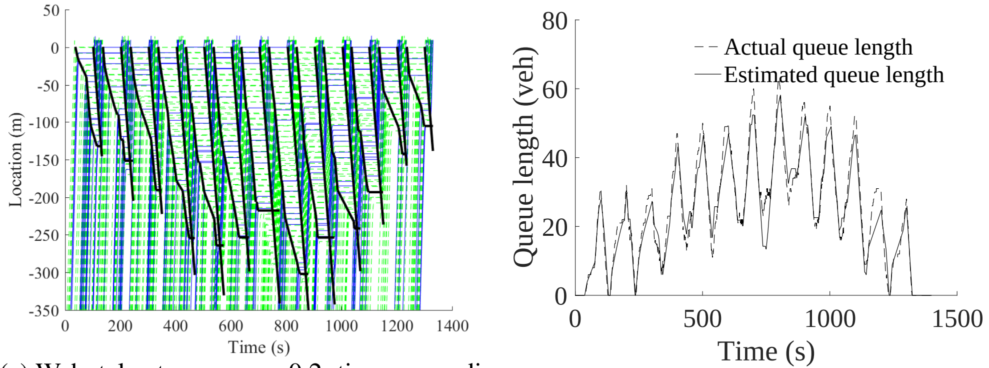
[1] In undersaturated scenarios, the method in Ban et al. (2011) estimates a linear BoQ, as the minimum queue length is 0. While in oversaturated scenarios, the resulting BoQ from Ban et al. (2011) can have two linear segments.



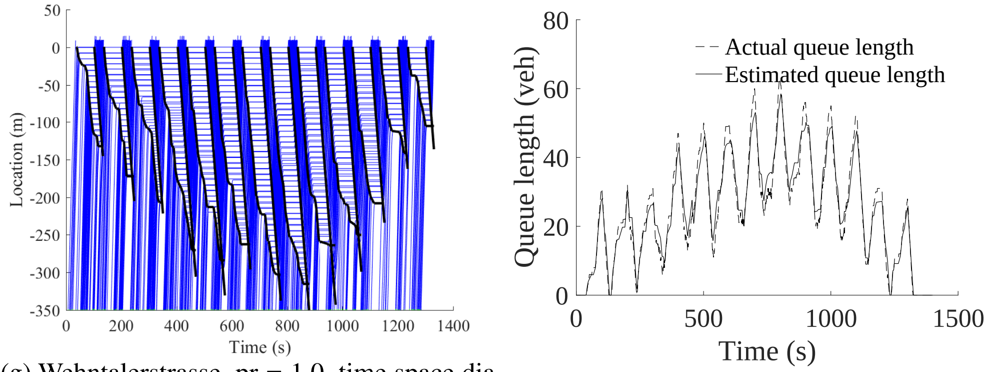
(a) Lankershim, pr = 0.2, time space diagram (b) Lankershim, pr = 0.2, queue length



(c) Lankershim, pr = 1.0, time space diagram (d) Lankershim, pr = 1.0, queue length



(e) Wehntalerstrasse, pr = 0.2, time space diagram (f) Wehntalerstrasse, pr = 0.2, queue length



(g) Wehntalerstrasse, pr = 1.0, time space diagram (h) Wehntalerstrasse, pr = 1.0, queue length

Figure 3.5: Illustration of queue lengths.

Table 3.2: Cycle-by-cycle maximum queue lengths for Lankershim and Wehntalerstrasse datasets (sampling rate is 1s^{-1}).

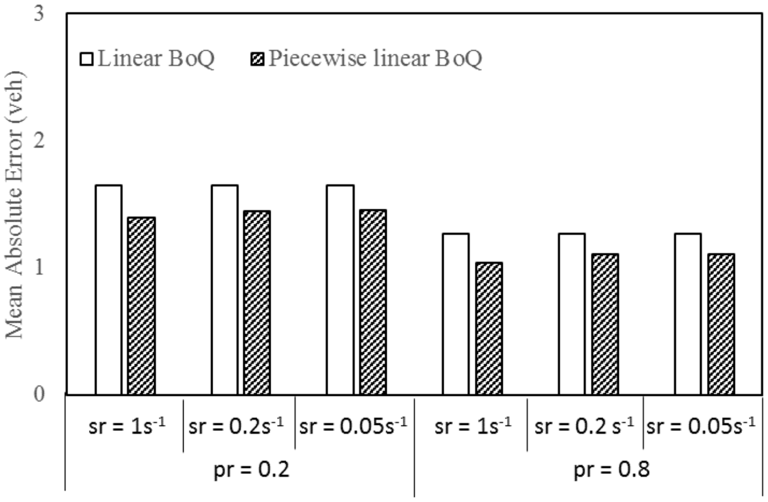
			pr = 0.2		pr = 1.0	
Site	Cycle	Actual	Estimated	Error	Estimated	Error
Lankershim	1	11.0	11.9	0.9	11.4	0.4
	2	19.0	14.6	4.4	17.9	1.1
	3	17.0	12.2	4.8	13.1	3.9
	4	2.0	0.0	2.0	1.8	0.2
	5	10.0	3.9	6.1	10.7	0.7
	6	4.0	0.0	4.0	3.9	0.1
	7	7.0	2.6	4.4	2.5	4.5
	8	15.0	13.3	1.7	15.0	0.0
	9	4.0	3.2	0.8	2.0	2.0
	10	7.0	1.1	5.9	5.0	2.0
Wehntalerstrasse	1	31.0	28.6	2.4	28.9	2.1
	2	30.0	29.9	0.1	29.9	0.1
	3	30.0	27.9	2.1	27.5	2.5
	4	47.0	42.8	4.2	43.4	3.6
	5	50.0	46.6	3.4	46.1	3.9
	6	50.0	47.9	2.1	46.0	4.0
	7	60.0	54.6	5.4	54.4	5.6
	8	63.0	58.6	4.4	58.7	4.3
	9	56.0	52.6	3.4	51.1	4.9
	10	55.0	50.1	4.9	50.6	4.4
	11	52.0	48.5	3.5	49.0	3.0
	12	31.0	25.3	5.7	28.4	2.6
	13	27.0	26.4	0.6	25.9	1.1

(up to 0.6 car and 16%). This shows that we can reduce the systematic errors caused by non-uniform arrivals. It is also shown that the improvement is in general larger for scenarios with higher sampling rates and higher penetration rates. This is because in such scenarios, the BoQ is more likely to be non-linear.

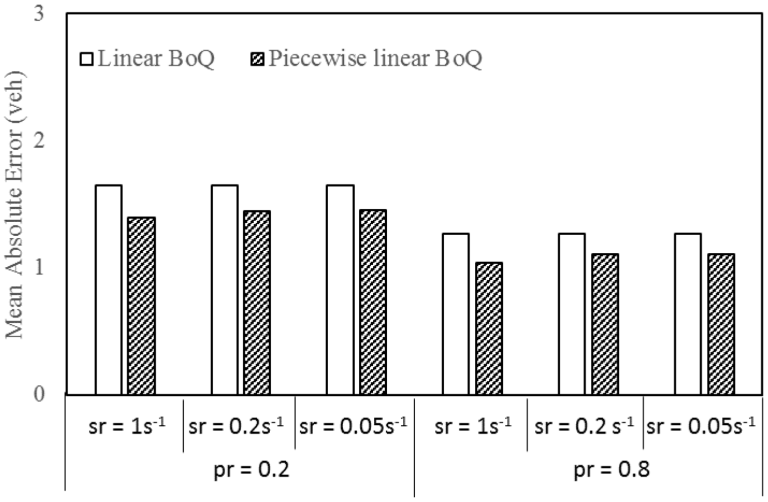
3.6.3 Value of considering acceleration and deceleration

The value of considering the intermediate state is shown in Figure 3.7. The shaded bars represent the estimation results of the extended methodology considering intermediate state proposed in Section 3.4, and the white bars represent the estimation results of the general methodology without including acceleration and deceleration information.

We can conclude that the method proposed in Section 3.4 successfully reduces the estimation error in all scenarios tested. The improvement is up to 0.1 car and 6%. Generally speaking, the benefit of this method is larger in scenarios with low penetration rates and/or low sampling rates. In such scenarios, the reported trajectory points are very few. Hence, to get sufficient critical points, the intermediate state should be taken into account. However, in scenarios with high penetration rates and/or high sampling rates, e.g. $pr = 0.8$ and $sr = 1\text{s}^{-1}$, the benefit of considering the intermediate state is marginal. Therefore, it is advised to consider the intermediate state only if either the sampling rate or the penetration rate is low.



(a) Lankershim dataset



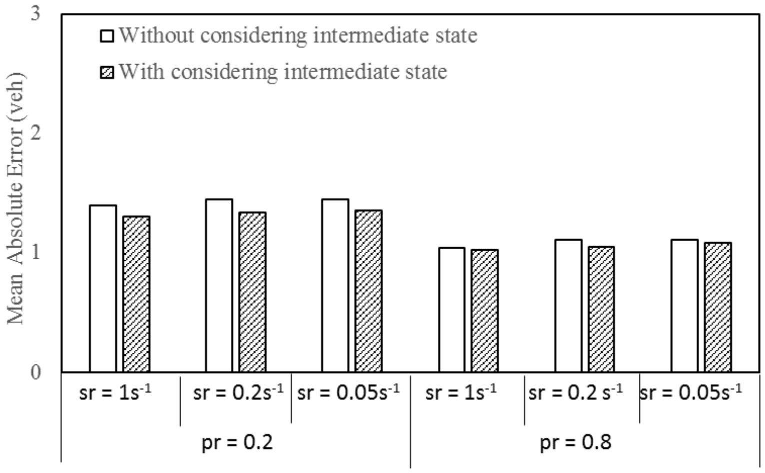
(b) Wehntalerstrasse dataset

Figure 3.6: Value of considering piecewise linear BoQ.

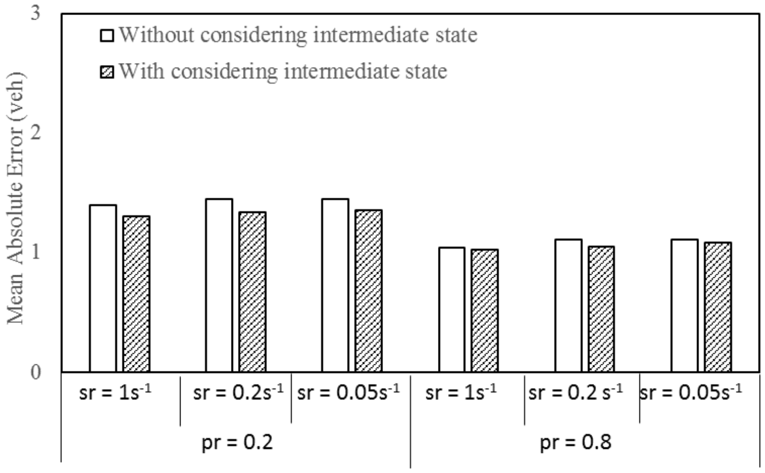
3.6.4 Value of integrating flow information

This section evaluates the value of integrating the upstream departure information by comparing the extended methodology for the network level (Section 3.3) to the general methodology. The departure flow from Intersection 1 at both sites is integrated with a Robertson's platoon dispersion model. The results are shown in Figure 3.8.

Figure 3.8 shows that the flow information from the upstream intersection improves the performance of the methodology in all scenarios tested. The improvement is up to 0.2 car and 16%. It can also be seen that trend is similar in both types of scenarios. However, more benefits can be obtained by considering the flow information in the undersaturated scenarios. This is because in undersaturated scenarios, there are not many trajectories. Therefore, additional information given by the upstream intersection is more valuable.



(a) Lankershim dataset



(b) Wehntalerstrasse dataset

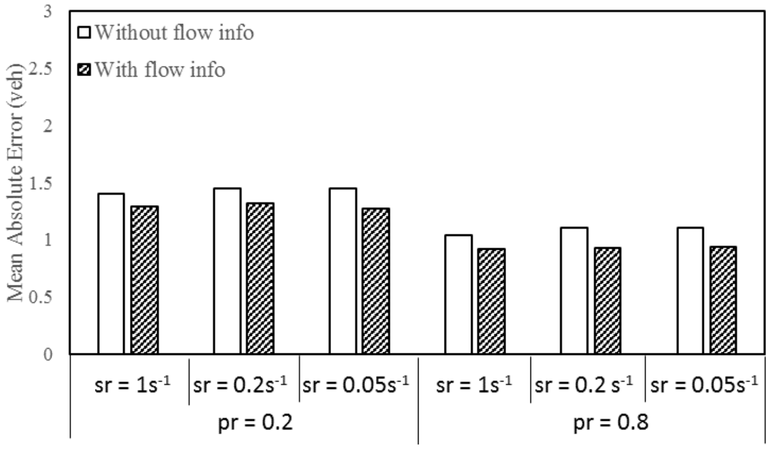
Figure 3.7: Value of considering acceleration and deceleration.

3.7 Sensitivity Analysis of the Proposed Methodology

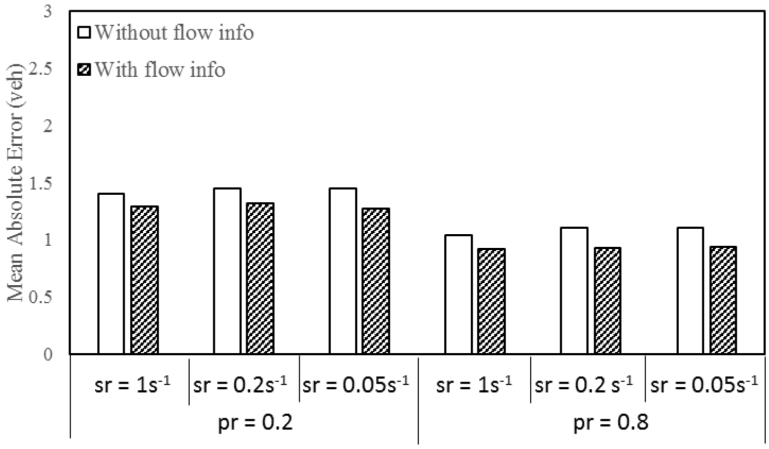
In this section, we analyze various aspects of the proposed methodology. First, the sensitivity to the model parameters is analyzed in Section 3.7.1. Second, the robustness of the proposed methodology is evaluated in Section 3.7.2. For presentation simplicity, the results of this section are only based on the Lakershim dataset. The results of the Wehntalerstrasse dataset exhibit similar properties.

3.7.1 Sensitivity analysis to model parameters

This section discusses the sensitivity of the proposed methodology to the model parameters: the regularization term λ_1, λ_2 and λ_3, the time step T_{step} for the BoQ estimation, the lower speed bound $\underline{u}$, the upper speed bound $\bar{u}$, and the parameters for the fundamental diagram w, u_f and k_{jam}. The aim is to evaluate the impact of each parameter on the performance of the proposed methodology if it deviates from the optimal values. To do this, we use a One-at-a-time sensitivity analysis method. Specifically, we change one parameter at each time and fix the other parameters at their optimal value, and observe how this influences the performance. In this way, we obtain the local sensitivity to this parameter. Note that this local sensitivity analysis is already enough



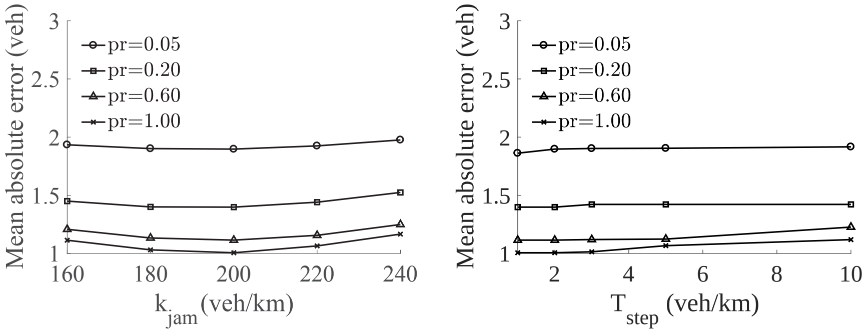
(a) Lankershim dataset



(b) Wehntalerstrasse dataset

Figure 3.8: Value of flow information.

for practical applications, because we do not expect the parameter values to deviate a lot from the optimal values. More complex sensitivity analysis methods (i.e. Ge and Menendez, 2017; Ge et al., 2015) could be used in the future to study the relations between the different parameters.



(a) k_{jam} (b) T_{step}

Figure 3.9: Sensitivity analysis to model parameters. Sampling rate is 1.0s^{-1}.

In the rest of this subsection, λ_1, λ_2 and λ_3 are chosen to vary between 0.5 and 2, T_{step} between 1s and 10s, the lower speed bound $\underline{u}$ between 1m/s and 6m/s, the upper speed bound $\bar{u}$ between 6m/s and 10m/s, u_f between 50km/h and 70km/h, w between 15km/hr and 25km/hr, and k_{jam}

between 160veh/km and 240veh/km.

The results show that the proposed methodology is not sensitive to the regularization factors λ_1, λ_2 and λ_3, the upper and lower speed bounds $\underline{u}$ and $\bar{u}$, the free flow speed u_f and the backward wave speed w, as long as these values are within a reasonable range. This is expected, as the regularization factors λ_1 and λ_2 penalize the FoQ and BoQ curves if they misclassify the stopped state and free-flow states; λ_3 controls the number of pieces in the piecewise linear BoQ curve. In the tested cases, as there are not many trajectory points, the probability of misclassification and the number of pieces in the piecewise linear BoQ curve are both quite small. In scenarios with higher oversaturation or a larger block size, however, these parameters could play a more important role. The free flow speed u_f and the backward wave speed w can be fitted from the trajectory points, therefore their specific values are not important for the model. For the upper and lower speed bounds $\underline{u}$ and $\bar{u}$, the probability of the trajectory points falling into the speed range of [1m/s, 3m/s] and [8m/s, 10m/s] is small if the links are not oversaturated. Therefore the proposed methodology is not sensitive to the specific values of the speed bounds. However, in scenarios with over-saturated links where the vehicles frequently accelerate and decelerate, these two parameters might be important.

Figure 3.9 shows the sensitivity analysis for the sensitive parameters, k_{jam} and T_{step}. As is shown in Figure 3.9(a), the model is sensitive to the jam density k_{jam}. This is reasonable, as the jam density is used to calculate the queue length from the BoQ and FoQ curves. Inaccurate jam density will lead to inaccurate queue length estimation, even if the BoQ and FoQ curves are accurate. It is shown in Figure 3.9(b) that the proposed methodology is also sensitive to T_{step}. If T_{step} is too large, there would be a large discretization error in the BoQ curve. Hence, the queue estimation error increases. It is also observed that the proposed methodology performs similar for a T_{step} of 1s to 3s. This is because these values of T_{step} are already smaller than the vehicle headway. So the marginal benefit of further reducing T_{step} is negligible. This suggests that the value of T_{step} should be chosen as 1-3s, and the value of k_{jam} should be as accurate as possible (e.g. obtained from the real data).

3.7.2 Robustness to measurement errors

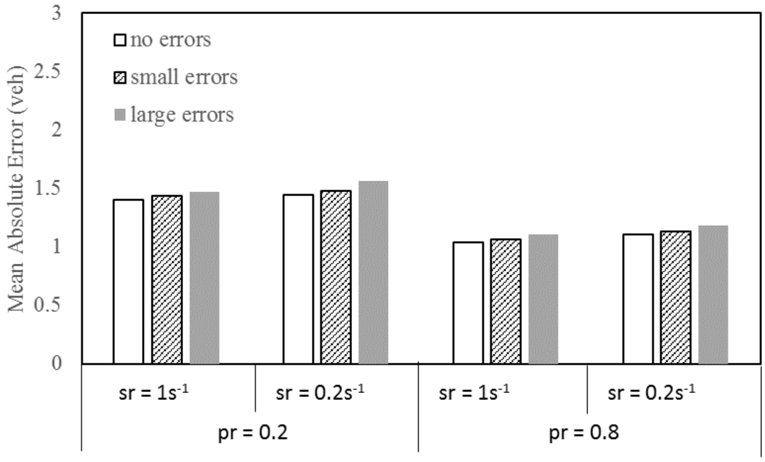


Figure 3.10: Sensitivity to the measurement errors, where small error represents a standard error of 2m for location and 0.5m/s for speed, and large error represents a standard error of 10m for location and 2m/s for speed.

Due to the errors in the measurement devices, the location and speed information received by the central controller may be inaccurate. This section tests how the errors in location affects the performance of the proposed methodology. It is assumed that both the location and speed errors follow a Gaussian distribution with mean 0, which means that there are no systematic errors. The standard deviation of the location measurement is assumed to vary between 0 and 10m. This is because the normal GPS devices provide an accuracy of 7.8 meters at 95% confidence level (Florin and Olariu, 2015), and the accuracy can also be enhanced by map-matching (Greenfeld, 2002), Kalman filter (see Appendix A.3), or data fusion with other in-vehicle sensors (Caron et al., 2006). The variance of the speed errors is assumed to be between 0 and 2m/s. The location errors and the speed errors are assumed to be independent. The errors at different time steps are also assumed to be independent.

The effects of the measurement errors are shown in Figure 3.10. It can be seen that the mean absolute error increases only slightly with the standard deviation of the measurement error (up to 2% for small errors and 7% for large errors). Therefore, the proposed model is robust to measurement errors.

3.8 Implementation Details

3.8.1 Online implementation

In this section, we present the online implementation details of the proposed methodology for real-time queue estimation. For simplicity of the presentation, we only describe the details for the general methodology in Section 3.2. The extensions in Section 3.3-3.4 can be treated in a similar way.

In an online scenario, let T_e represent the length of the time step to perform queue estimation. At each time instant $t = T_e,\ 2T_e,\ \cdots$, we update the BoQ and FoQ curves up to this time instant using the currently available trajectory information. Notice that at each time instant, we use the calculated BoQ and FoQ curves only to estimate the queue length at the current time instant, rather than update the queue estimation in the past or forecast the queueing process that will take place in the future. This is rather useful as many signal control strategies only require the information of the current queue length.

Recall that the proposed methodology can be used for both undersaturated and oversaturated scenarios. In oversaturated scenarios, there could be more than one segment of BoQ/FoQ curves at each time instant, as the queue can be carried over from previous cycles. Therefore, we first identify which cycles these BoQ/FoQ segments belong to. To this end, let us define a cycle p as an “active cycle”, if the queue due to the red signal in cycle p is still active at time t. In other words, if p is an active cycle, the FoQ and BoQ curves due to the red signal in cycle p, i.e. $FOQ_p(t)$ and $BOQ_p(t)$, respectively, are still active at time instant t. Therefore, the set of active cycles at each time instant can be updated by checking whether the remaining queue of cycle p is positive ($FOQ_p(t) - BOQ_p(t) > 0$). Recall that each vehicle can be included in the queue of multiple cycles. Then, estimating the queue length at time t can be formulated as calculating the FoQ and BoQ curves of all active cycles. Denote the set of all the active cycles at time t as P_t. The general implementation framework is summarized into Algorithm 3.1, as follows.

At each time t, Algorithm 3.1 first checks if there is a new cycle (lines 2-3) and updates the set of active cycles P accordingly. Line 5 represents the general framework, for which there are two methods to implement.

The first method, hereafter named direct implementation method, applies the exact procedure described in Section 3.2 for the trajectory information up to the current time instant. We update the sets of trajectory points S_p and V_p based on Section 3.2.1, calculate the critical points based on Section 3.2.2, and obtain the FoQ curve and BoQ curve by solving Eq.(3.4)-(3.8) (Section 3.2.3) and Eq.(3.10)-(3.16) (Section 3.2.4), respectively. It will be shown in Section 3.8.2 that this implementation method is sufficiently efficient for most real-time traffic control (e.g. signal control) algorithms.

Algorithm 3.1 General Online Implementation of the Methodology.

```
Input: $P_0 = \emptyset$,
1: for $t = T_e, 2T_e, \cdots$ do
2:     if $t - T_e < r_p \leq t$ then
3:         $P_t = P_{t-T_e} \cup \{p\}$
4:     for $p \in P$ do
5:         Update $S_p$, $V_p$, $F_p$ and $B_p$ with newly received trajectory
             points, and obtain $FOQ_p$ and $BOQ_p$
6:         if $FOQ_p(t - T_e) - BOQ_p(t - T_e) > 0$
             and $FOQ_p(t) - BOQ_p(t) \leq 0$ then
7:             $P_{t+T_e} = P_t \backslash \{p\}$
8:     Calculate queue length $Q(t)$ using Eq.(3.17)
```

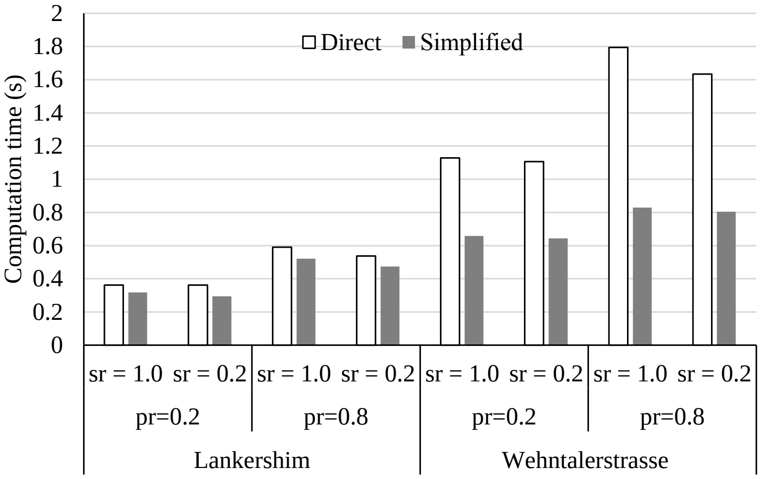
The second method, hereafter named simplified implementation method, further reduces the computation time and the memory requirement. Specifically, we have three simplifications compared to the direct implementation method.

1) We reduce the amount of data required for calculating the critical points (Section 3.2.2). Recall that the identification of critical points is based on linear regression. Therefore, instead of the entire set of trajectory points, we only store the mean and total number of the trajectory points of vehicle n in state s (stopped or free flow) in each cycle p, denoted as $(\bar{t}^s_{m,p}, \bar{x}^s_{m,p})$ and $n^s_{m,p}$, respectively. These values are updated every time when new trajectory information is received. Then, the FoQ and BoQ critical points are calculated in a similar way as in Section 3.2.2 where $J_n \cap S_p$ is replaced by the stopped mean trajectory point and $J_n \cap V_p$ is replaced by the free flow mean trajectory point.
2) We reduce the number of constraints in the optimization problems solving the BoQ and FoQ curves. For constraints Eq.(3.6)-(3.8) and Eq.(3.13)-(3.15), we do not use the entire set of stopped (or free flow) trajectory points S_p (or V_p). Instead, we only keep the constraints associated with first and the last stopped (or free flow) trajectory points of each vehicle in cycle p, which can be seen as the supporting vectors for constraints Eq.(3.6) and Eq.(3.14) (or Eq.(3.7) and (3.13)).
3) We reduce the number of decision variables in the optimization problems solving the BoQ curve. At each time instant t, we only update the BoQ curve shortly before t (i.e. in range $(t - \delta t, t])$. This is because the BoQ curve is mainly influenced by the information around time t. In other words, in the optimization problem Eq.(3.10)-(3.16), we only keep the decision variables (i.e. the slopes α_i and intercepts β_i) associated with this time range. Notice that the constraints can be simplified accordingly, as α_i and β_i before time instant $t - \delta t$ are treated as given parameters calculated in previous time instants. Specifically, we only need to keep the critical points (B_p), stopped (S_p) and free flow (V_p) trajectory points with a time stamp in range $(t - \delta t, t])$ in the optimization problem.

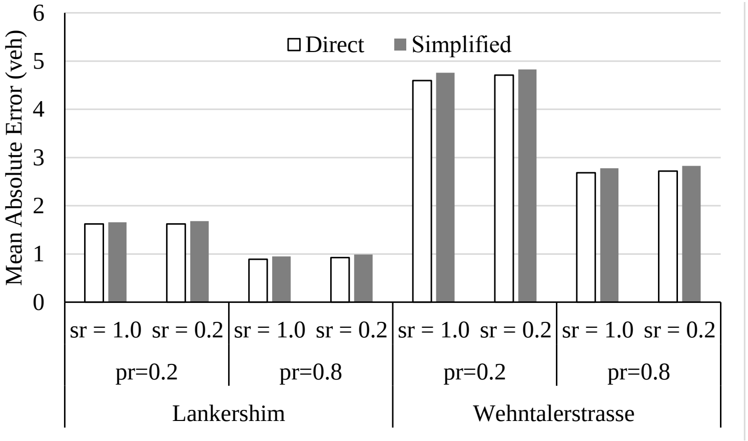
After the FoQ and BoQ curve for cycle p are obtained, we check if the cycle p is still active (i.e. whether the queue of cycle p has fully discharged) by comparing the obtained FOQ and BOQ curve. If cycle p is no longer active, we remove cycle p from the set of active cycles. This procedure is illustrated in lines 6-7 in Algorithm 3.1. After the FOQ and BOQ curves are calculated for all the active cycles, the queue length is calculated using Eq.17 (line 8).

3.8.2 Comparison between the two implementation methods

The two online implementation methods are tested on the Intersection 2 of Lankershim and Wehntalerstrasse datasets. From the description of the general framework and two online implementation methods, the simplified method requires significantly less memory than the direct method, as only a small fraction of trajectory points need to be stored. In this subsection,



(a) Computation time



(b) Mean Absolute Error

Figure 3.11: Comparison between two online implementation methods: the direct method and the simplified method.

we evaluate the average time and the mean absolute error of both methods. The results are summarized in Figure 3.11.

Figure 3.11(a) shows the average computation time in one time step, and Figure 3.11(b) shows the mean absolute error of the online solution of both methods compared to the ground truth. Here, δt is chosen as 10s.

It can be seen from Figure 3.11 that the direct method can yield satisfactory results within 2s and the simplified method takes less than 0.8s. This shows that both methods are sufficiently efficient for most applications (e.g. signal control). The simplified method performs similarly to the direct method in terms of the estimation accuracy (only up to 5% worse). However, the simplified method significantly reduces the computation time. The improvement is up to 18% for Lankershim dataset and 54% for Wehntalerstrasse. This shows that the simplified method performs especially well for oversaturated scenarios. Another observation is that the computation time increases as the penetration rate and the sampling rate increase. This is expected, as the amount of information increases, thus the optimization model has more decision variables. Our experiments also show that the results are not sensitive to the size of the discretization time step, thus we can choose a smaller discretization time for a better accuracy.

3.9 Conclusions and Future Work

In this chapter, we propose a methodology to estimate the queue length in a connected vehicle environment. The output of the proposed methodology is the queue profile, which can be used in trajectory reconstruction, delay evaluation, flow and density estimation, etc. The proposed methodology can be beneficial to designing and evaluating signal control strategies. A convex optimization model is formulated to relax the widely adopted assumption of linear BoQ curve, calculating a linear FoQ curve and a piecewise linear BoQ curve. The queue length can be represented as the difference between the two curves. We further propose a framework to reuse the estimated discharging rate of the upstream intersections, which can better utilize the information provided by connected vehicles and facilitate network level estimation. The cases with low sampling rate are also handled by utilizing the trajectory data in the intermediate state.

A validation experiment with the Lankershim dataset and Wehntalerstrasse dataset shows that the proposed methodology performs well. The average estimation error is within 1.5 cars for undersaturated scenarios and 5.2 cars for oversaturated scenarios, with penetration rates larger than 0.1 and sampling rates higher than 0.05s^{-1}. Compared to a state-of-art method that assumes linear BoQ, the proposed methodology improves the estimation accuracy by up to 16%. For an arterial, the performance of the proposed methodology is further enhanced by the departure flow information at the upstream intersection, with an improvement of up to 16%. It is also shown that the proposed methodology works well under the low sampling rate, and is relatively robust to measurement errors.

This chapter relies on the assumption of a fundamental diagram and a few parameters. It is shown that the proposed methodology is only sensitive to the assumed jam density and the time step for back of queue estimation T_{step}. It is expected that the performance of the proposed methodology is not sensitive to the shape of the fundamental diagram, as long as the jam density is well calibrated.

We further propose two online implementation methods for real-time queue estimation. Results show that the direct method that uses the exact offline method takes less than 2s for one estimation step, and an simplified method takes less than 0.8s for each estimation step. This shows that the convexity of the methodology ensures the efficiency in computation, which is sufficient for most applications.

One limitation of the proposed method is that it requires at least one BoQ critical point and one FoQ critical point in each signal cycle. In scenarios with very sparse data (i.e. very low penetration rates or sampling rates), it is essential to incorporate probabilistic models into the proposed methodology in the future work. Another future direction is to utilize the obtained queue length to perform short term traffic flow prediction.

PART II

Local-level Traffic Control in a Connected and Automated Vehicle Environment

Abstract

The second part of this dissertation is devoted to real-time traffic control at the local level in a connected and automated vehicle environment. In this part, we will focus on the optimization of traffic signals. Traffic signals are essential components for urban traffic management, and have attracted enormous research interest since the late 1950s (Webster, 1958). Traditional traffic control strategies use either historical data (fixed-time) or real-time information provided by loop detectors (actuated or adaptive) to determine departure priority. Loop detectors are usually installed at a fixed location and cannot provide detailed information about the movement of individual vehicles. Therefore, even if those signal control strategies can adapt to the variations in demand, there is still room for improvement.

The recent development in the technologies of connected and automated vehicles makes it possible to track and control the movement of vehicles and thus can be beneficial for traffic control. Connected and automated vehicles are able to provide detailed, real-time, and anticipative information with a better spatial coverage, facilitating more intricate and predictive control strategies. Using such information, we are not only able to optimize the current control actions, but also to take into account the traffic performance in the future. In fact, all the control strategies developed in this part follow the concept of MPC, where an embedded optimization model is solved to maximize the traffic performance in a moving time horizon. Moreover, wireless communication systems and automated driving can help advise drivers or control vehicles, providing a more flexible design of traffic control strategies.

Despite the benefit of the technologies in connected and automated vehicles, the penetration rates will remain low in the near future. Low penetration rates lead to a more uncertain environment, which is challenging for traffic control. We will develop strategies robust enough for the low penetration rates, and in the meanwhile, provide guidelines on how the control strategies should evolve with the increase of the penetration rates. Moreover, urban cities are multimodal systems where various types of vehicles interact with each other, making the control problem more complex. We will differentiate certain groups of vehicles and incorporate priority schemes to the proposed control strategies.

In Chapter 4, we will address the important research gap L1 as established in Section 2.3. To this end, we jointly optimize the traffic signal timing and the trajectory planning of automated vehicles at local intersections in the transition period where various level of technologies coexist in the traffic stream. We aim to optimize the departure sequence in which the vehicles leave the intersection. We also reduce the computational burden with a branch and bound solution algorithm, and improve the robustness against measurement errors using a Kalman filter. Simulation results show an evident decrease in the total number of stops and delay when using the proposed strategy for the tested scenarios with information levels as low as 50%, even in scenarios with measurement noises. The simulation results are used to develop a heuristic to switch between different features of the proposed control strategy, according to the total demand and penetration rate of each technology.

In Chapter 5, we incorporate a priority scheme into the proposed control strategy in Chapter 4 to address research gap L2 as described in Section 2.3. We aim to design a transit signal priority (TSP) algorithm that prioritizes public transport at local intersections. The TSP algorithm can also be adapted to scenarios with near-side or far-side bus stops. Moreover, it can minimize either signal delay or schedule delay for buses while minimizing additional car delays. Simulation results show the total passenger delay is reduced with minimal increase in the delay of cars in

the conflicting approach. It is also shown that this TSP algorithm is not sensitive to the assumed bus passenger occupancy, nor the estimation of bus dwell time, hence does not require accurate information on these parameters.

For the reader's convenience, a list of the most important variables is given below.

The variables below are given sets (inputs)

I	arrival sequence, cars indexed by i
J	departure sequence, cars indexed by j
M	stream, indexed by m
N	current car set at each decision time, cars indexed by c
C	current car set at each decision time
B	current bus set at each decision time

The variables below are given scalars (inputs)

a_n	maximum acceleration rate of vehicle n
k_{jam}	jam density
l	length of the intersection
L_n	length of vehicle n
s	distance from the intersection to the bus stop
S^b (S^c)	saturation flow rate if all vehicles are buses (cars)
S_n	estimated saturation flow rate for vehicle n
ST_{n_b}	time when bus n_b is scheduled to depart from the bus stop
t_{dwell}	dwell time plus lost time due to the acceleration and deceleration of the bus
u_f	free flow speed
u_{min}	lower bound for speed in trajectory design
V_n	virtual departure time of vehicle n at the downstream end of the intersection
w	backward wave speed

The variables below are associated with decision variables

$d_{n,J}$	sum of jam spacings of the vehicles in front of vehicle n in its platoon
$D_{n,J}$	departure time of vehicle n at the downstream end of the intersection for departure sequence J
$E_{n,J}$	entering time of vehicle n to the intersection stop line for departure sequence J
$P_{n,J}$	delay penalty for car c for departure sequence J, this represents the time it takes for car c to cross the intersection
o_n	passenger occupancy of vehicle n
$O_{n,J}$	position of car c within the platoon using departure sequence J
TD_J	total delay for departure sequence J
$u_{n,J}^{init}$	initial speed of car c when entering the intersection
$u_{n,J}^{opt}$	optimal speed in trajectory design
$u_{n,J}^{des}$	design speed in trajectory design
Z_n	departure time of vehicle n from the bus stop

Chapter 4

Intersection Control in a Connected and Automated Vehicle Environment

This chapter is partially based on the following paper.

- Yang, K., S. I. Guler and M. Menendez (2016a) Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles, *Transportation Research Part C: Emerging Technologies*, **72**, 109–129.

4.1 Objectives and Contributions

Connected and automated vehicles have been attracting increasing attention on the control of signalized intersections. Some of the existing works utilize the information provided by connected vehicles to design signal control strategies, optimizing either signal phases (Priemer and Friedrich, 2009; Hu et al., 2015) or the departure sequence (Guler et al., 2014; Wu et al., 2007; Pandit et al., 2013). Optimizing departure sequence can provide more detailed control strategy as each individual vehicle is explicitly considered. Other works integrate trajectory design of automated vehicles into the signal control scheme (Li and Wang, 2006; Lee and Park, 2012; Dresner and Stone, 2004, 2006). However, as is described in Section 2.3 (research gap L1), the existing works assume that all vehicles are automated or connected.

In this chapter, we aim to develop a control strategy that jointly optimizes the signal timing of local intersections and trajectory planning of automated vehicles in a scenario where conventional, connected but non-automated, and automated vehicles co-exist in the traffic systems. The contributions of this chapter are four-fold.

1) We enable bidirectional vehicle-to-infrastructure communication and successfully integrate trajectory design for automated vehicles into the signal control scheme.
2) We propose a scheme to show how the control strategy should evolve with different development stages of connected vehicle technology, considering three different types of vehicles.
3) We increase the computational efficiency with a branch and bound solution algorithm.
4) We employ a Kalman filter to reduce the impact of measurement errors on the final solution.

This chapter is organized as follows. Section 4.2 introduces the developed methodology. Section 4.3 describes the simulation settings. Section 4.4 evaluates the performance of the developed methodology by comparing it to an actuated signal control algorithm. Section 4.5 performs a sensitivity analyses and evaluates the robustness of the proposed control strategy. Section 4.6 proposes a demand responsive control scheme based on the application bounds of the proposed strategy. Section 4.7 concludes this chapter and proposes potential future directions.

4.2 General Methodology

Three categories of vehicles are considered: 1) conventional vehicles, 2) automated vehicles, and 3) connected vehicles. Although all automated vehicles provide information (i.e. are connected), in this chapter, we call "connected vehicles" only those vehicles that send information, but whose trajectory cannot be controlled/modified[2]. Both connected and automated vehicles report their location, speed, and direction to the central controller in real-time through V2I communication systems (DSRC). It is also assumed that the central controller is able to modify the trajectory of the automated vehicles through V2I communication.

The flowchart of the proposed strategy is shown in Figure 4.1.

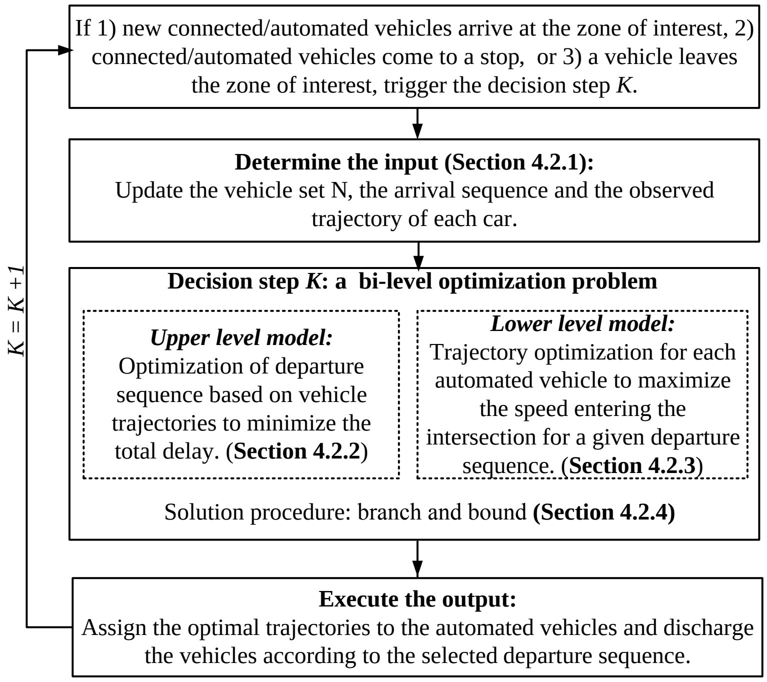

If 1) new connected/automated vehicles arrive at the zone of interest, 2) connected/automated vehicles come to a stop, or 3) a vehicle leaves the zone of interest, trigger the decision step K.
Determine the input (Section 4.2.1):
Update the vehicle set N, the arrival sequence and the observed trajectory of each car.
Decision step K: a bi-level optimization problem
Upper level model:
Optimization of departure sequence based on vehicle trajectories to minimize the total delay. (Section 4.2.2)
Lower level model:
Trajectory optimization for each automated vehicle to maximize the speed entering the intersection for a given departure sequence. (Section 4.2.3)
Solution procedure: branch and bound (Section 4.2.4)
Execute the output:
Assign the optimal trajectories to the automated vehicles and discharge the vehicles according to the selected departure sequence.
K = K +1

Figure 4.1: Flowchart of the proposed strategy.

The inputs to the control strategy include the vehicle set N that consists of all connected and automated vehicles in the zone of interest, plus stopped conventional vehicles ahead of some stopped connected or automated vehicles, as well as the arrival sequence and the observed or estimated trajectory of each vehicle.

The output of the control strategy is the optimal departure sequence for all vehicles in vehicle set N and the optimized trajectory for each automated vehicle in the zone of interest. Here, the same as in Chapter 3, the length of the zone of interest is chosen as the minimum of the city block size and the DSRC transmission range (e.g. around 100m).

The goal of the optimization is to determine the optimal departure sequence of all vehicles and the optimized trajectories for automated vehicles in an on-line manner (i.e. in real time). Each decision step (i.e. the time between every update of the optimal departure sequence and trajectories of automated vehicles) is triggered by one of three events: 1) a connected or automated vehicle enters the zone of interest; 2) a connected or automated vehicle comes to a stop; 3) a connected or automated vehicle leaves the zone of interest. All of these events require the inputs to be updated. The first event adds new connected or automated vehicles to the set N,

---

[2] In cases where connected vehicles also receive instructions, if the drivers follow them, they are deemed as automated vehicles; if the drivers do not change their behaviors accordingly, they fall into the connected vehicle category. For simplicity, the case where the drivers react to the instructions but do not follow them exactly is not considered.

the second event unveils stopped conventional vehicles that might have not been detected before, and the third event removes a vehicle from the vehicle set N.

In each decision step, a bi-level optimization model is proposed to integrate the optimization of departure sequence and the trajectory design for automated vehicles based on the arrival information. The upper level model is adapted from Guler et al. (2014). In the upper level model, we optimize the departure sequence to minimize the total delay based on the revealed vehicle trajectories. In the lower level model, we optimize the trajectory of each automated vehicle for a given departure sequence to maximize the speed entering the intersection. The two levels interact with each other. The lower level model takes the departure sequence and returns vehicle trajectories to the upper level model. Notice that the decisions at both levels are performed by the same decision maker (i.e. the central controller). Examples of similar idea can be found in recent works (e.g. Xu et al. (2018)). This is in contrast to the bi-level optimization problems with a user equilibrium as the lower level model (e.g. Lei and Ouyang (2017)). In this chapter, the bi-level model is solved by a branch and bound algorithm.

In the final step, vehicles are discharged according to the calculated departure sequence. Automated vehicles will be given designed trajectories. It is assumed that the signal turns green for an approach right before the first vehicle in that platoon enters the intersection.

The rest of this section presents the procedure within each decision step. Section 4.2.1 presents how the model inputs are determined. Section 4.2.2 presents the upper level model, i.e. optimization of departure sequence. Section 4.2.3 presents the lower level model, i.e. trajectory design for automated vehicles. Section 4.2.4 proposes a fast branch and bound algorithm to solve the bi-level model.

4.2.1 Model inputs

When the decision step is triggered, the inputs to the proposed strategy, namely the vehicle set, arrival sequence and the trajectory of each vehicle need to be updated. The method to update automated and connected vehicles differs from that for conventional vehicles.

The arrival sequence and trajectories of connected and automated vehicles are obtained directly from the V2I communication systems. It is assumed that connected vehicles and automated vehicles report their location and speed (i.e. trajectory) at any given time after they enter the zone of interest. With such information and assuming free flow speed u_f, the virtual departure time V_n (i.e. the time at which a vehicle would arrive to the downstream end of the intersection if there were no queueing) of an automated or connected vehicle n can be obtained and updated once it enters the zone of interest.

The arrival information (arrival sequence and virtual departure time) of conventional vehicles (i.e. vehicles that do not provide any type of information) can be estimated using the location information of connected and automated vehicles. Once a connected or automated vehicle comes to a stop, the number of conventional vehicles stopping in front of it (i.e. the arrival sequence of conventional vehicles) can be estimated by real-time queue estimation methods (e.g. the method proposed in Chapter 3). The virtual departure times for these conventional vehicles are estimated using a linear interpolation.

4.2.2 Upper level model: optimization of departure sequence

Denote M as the set of approaches. Denote I as the arrival sequence, i.e. the sequence in which vehicles on both approaches in vehicle set N arrive at the zone of interest. For automated and connected vehicles, the arrival sequence and approach are detected from the vehicle-to-infrastructure system once they enter the zone of interest. The arrival sequence and approach for conventional vehicles are estimated using the information provided by connected and automated vehicles.

Denote J as the departure sequence, i.e. the sequence in which vehicles on both approaches in vehicle set N depart from the intersection. Note that departure sequence J is one permutation of the arrival sequence I that satisfies the first-in-first-out principle for each individual approach. To minimize total delay, the optimal departure sequence is found, i.e.

$$\min_{J} TD_J = \sum_{c \in N} (D_{n,J} - V_n) \tag{4.1}$$

where for departure sequence J, $D_{n,J}$ represents the predicted departure time of vehicle n from the downstream end of the intersection. Recall that V_n is the virtual departure time from the downstream line of the intersection for vehicle n. Hence, the difference $D_{n,J} - V_n$ is the delay for vehicle n.

Assume each vehicle is represented by a triple $n = (i, j, m)$ where $i \in I$ is the position of vehicle n in the arrival sequence, $j \in J$ is the position of vehicle n in the departure sequence, and $m \in M$ is the approach. The term "approach" here represents the vehicle flow governed by one traffic signal phase. For a given departure sequence J, and the corresponding optimal trajectories obtained in Section 4.2.3, the predicted departure time for vehicle $n = \{i, j, m\}$, $D_{n,J}$, is calculated as the maximum of the virtual departure time and the next possible departure time, i.e[3].

$$D_{n,J} = \max \left\{ V_n,\ D_{n',J} + \frac{1}{S_m} + P_{n,J} \right\},\ \forall n \in N \tag{4.2}$$

where n' is the previous vehicle in this departure sequence to vehicle n, and S_m is the saturation flow from approach m. The delay penalty $P_{n,J}$ represents the time it takes for each vehicle to cross the intersection, and is given in Eq.(4.3) derived with basic kinematic laws.

$$P_{n,J} = \max \left\{ \frac{l}{u_f},\ \frac{-u_{n,J}^{init} + \sqrt{(u_{n,J}^{init})^2 + 2a_n l}}{a_n} \right\} \tag{4.3}$$

where l is the length of the intersection; a_n is the acceleration rate of vehicle n, and $u_{n,J}^{init}$ is the initial speed of a vehicle n when entering the intersection. The initial speed $u_{n,J}^{init}$ is the result of the trajectory optimization for automated vehicles as Eq.(A.8) and Eq.(A.12). More details are provided in Section 4.2.3 and Appendix A.2. The second term of the right hand side in Eq.(4.3) represents the acceleration time of vehicle n if the initial speed is smaller than the free flow speed u_f. Here, the case where the vehicle reaches the free flow speed in the middle of the intersection is not considered. This might cause a small and systematic error which should not influence the results in any significant manner[4].

As is shown in Figure 4.2, the penalty decreases as the initial speed increases. Notice that as more vehicles discharge from a platoon, the initial speed increases since vehicles have space to accelerate before consecutively entering the intersection. Therefore, the penalty favors platooning instead of alternating departures between approaches. By not requiring vehicles to stop, the initial speed can also be increased. Hence, allowing vehicles to discharge in platoons is favorable for decreasing the penalty.

---

[3] For the very first vehicle entering the system when the system is empty, we expect that it drives at the free flow speed, thus we have $D_{n,J} = V_n$. For the first vehicle in the departure sequence J which is not the first vehicle in the system, Eq.(5) still holds, where n' is the previous vehicle that departs before vehicle n (although n' is not in the departure sequence J but in the departure sequence of an earlier time step).

[4] In the cases where the vehicle accelerates to the maximum speed inside the intersection, the vehicle should enter the intersection with a relatively large speed. In other words, the vehicle starts accelerating from the stopped state quite far upstream from the intersection. As is shown in Figure 4.2, this error is too small to have any significant impact on the results.

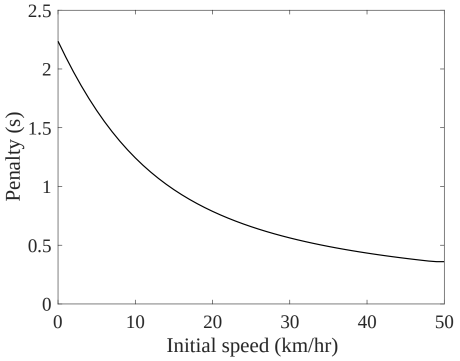
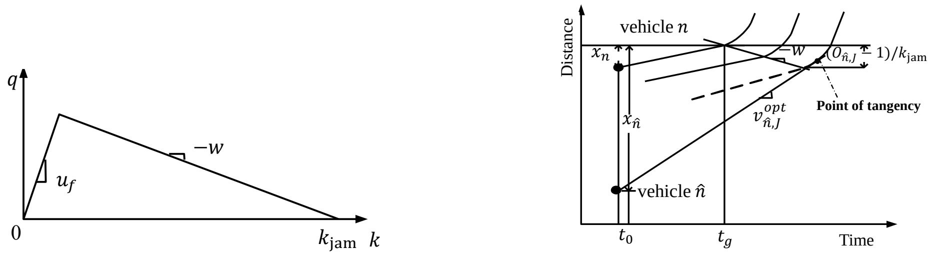


Figure 4.2: Illustration of penalty, where u_f = 50km/h, l = 5m, a_n=2m/s^2.

4.2.3 Lower level model: trajectory design for automated vehicles

For a given departure sequence J, the trajectory of each vehicle can be sequentially determined according to the departure sequence. The trajectory is designed for each individual automated vehicle based on the real or estimated traffic information (departure time, speed, etc.) of the vehicles in front of it. The objective of the trajectory design is to let vehicles pass the intersection at a specific time with the maximum possible speed and, if possible, without stopping. Notice that in this chapter, in order to evaluate the benefit of trajectory planning, we do not distinguish the different reaction times for automated and connected/conventional vehicles. This gives a lower bound of the potential improvement by using automated vehicles.

In order to calculate the optimal trajectories for automated vehicles, the “future” trajectories of connected and conventional vehicles should be predicted using Newell’s vehicle following model. One input to the Newell’s vehicle following model is the departure sequence, which is represented as the signal timing. The other parameters of the Newell’s vehicle following model are determined using kinematic wave theory. For Newell’s vehicle following model, the distance travelled in a time step is calculated as the minimum of how far a vehicle can travel at free-flow speed and what would be permissible due to downstream congestion. Under free-flow conditions, the trajectory of the vehicle no longer depends on the vehicle in front of it. In the case with downstream congestion, the spacing between each pair of vehicles is the inverse of the density. For more information on Newell’s vehicle following model, see Newell (2002).



(a) Triangular fundamental diagram

(b) Time space diagram with sample trajectories

Figure 4.3: Illustration of the speed calculation. vehicle n represents the first vehicle in the platoon; vehicle $\hat{n}$ represents the vehicle that follows other vehicles. The dashed trajectory represents the closest possible trajectory of vehicle $\hat{n}$ to the last vehicle in the platoon (according to Newell’s vehicle following model).

For ease of presentation, the acceleration and deceleration processes are simplified without changing the overall control. To this end, the acceleration and deceleration processes are divided into two types. The first type occurs because the signal changes from red to green. This type of maneuver includes the vehicles that accelerate from the stopped state and cross the intersection (see Figure 4.3(b) for an example). This type involves only acceleration maneuvers and usually occurs close to the intersection. The second type of maneuver consists of all other possibilities, i.e. either decelerating because of red signal or a queue downstream, or informed by the controller to accelerate or decelerate. This type involves both acceleration and deceleration maneuvers and occurs near the back of queue, which can be near the intersection or not. In other words, the first type consists of the accelerating vehicles because the signal switches to green and the second type includes all the other cases. The first type of acceleration process is taken into consideration to properly evaluate the benefits of platooning (see Eq.(4.3)). For simplicity, constant acceleration with acceleration rate a is assumed. For the second type, instantaneous acceleration and deceleration are assumed to simplify the discussion (i.e. $a = \infty$). Notice that this assumption of instantaneous acceleration and deceleration can be made without loss of generality. As long as the time to enter the intersection is fixed, the assumption of instantaneous acceleration and deceleration will only slightly influence the initial speed $u_{n,J}^{init}$ at which the vehicle enters the intersection, and thus change the total calculated delay by Eq.(4.2) and Eq.(4.3). However, as is shown in Figure 4.2, the marginal change in delay penalty decreases as $u_{n,J}^{init}$ increases. Hence, as long as $u_{n,J}^{init}$ is sufficiently large (i.e. over 25 km/hr), the total calculated delay across all vehicles will only be slightly different than in reality. Therefore, it is reasonable to assume instantaneous acceleration and deceleration before a vehicle joins the dispatching platoon or comes to a stop.

By the assumption above, the design of the full trajectory of an automated vehicle n can be simplified into deciding on a particular design speed $u_{n,J}^{des}$ at the time the trajectory design is performed. This is a quasi-optimal solution, which simplifies the discussion with satisfactory approximation. The initial speed entering the intersection $u_{n,J}^{init}$ can be calculated from $u_{n,J}^{des}$.

Intuitively, it is always possible to let a vehicle pass the intersection without stopping, as long as the vehicle drives at a sufficiently low speed. However, it does not make sense if the vehicle crawls to the intersection. Therefore, a lower bound $u_{\min}$ is defined to constrain the design speed, i.e.$u_{\min} \leq u_{n,J}^{des} \leq u_f$, where the second inequality means that the design speed cannot exceed the free flow speed u_f.

Given the trajectory of the previous vehicles, an automated vehicle can be assigned to either wait until the next signal cycle, or follow the previous vehicle. This is equivalent to considering two cases based on whether an automated vehicle is the first vehicle in its platoon or not (this is determined from the departure sequence), i.e. $O_{n,J} = 1$ or $O_{n,J} > 1$ where $O_{n,J}$ represents vehicle n's position in the platoon for a given departure sequence J.

As is shown in Figure 4.3(b), vehicle n represents the case with $O_{n,J} = 1$ and vehicle $\hat{n}$ represents the case with $O_{\hat{n},J} > 1$. In this figure, t_0 is the time the trajectory design is performed. x_n and $x_{\hat{n}}$ are the distance from vehicle n and vehicle $\hat{n}$ to the intersection, respectively; $u_{n,J}^{opt}$ and $u_{\hat{n},J}^{opt}$ are the optimal speeds of vehicle n and vehicle $\hat{n}$ respectively at time t_0; and t_g is the earliest possible time when this approach can be given green signal, which is determined by the passing time of the last vehicle that has departed in the conflicting approach. Here a triangular fundamental diagram is assumed, as is shown in Figure 4.3(a), with free flow speed u_f, backward wave speed w and jam density k_{jam}. For the simplicity of presentation, the detailed calculations can be found in Appendix A.2. The design speed of the automated vehicles can be calculated as Eq.(A.7) and Eq.(A.11).

4.2.4 A branch and bound solution algorithm

In this section, a branch and bound algorithm is proposed to search for the exact optimal departure sequence and the optimal speed for the proposed bi-level optimization model. The solution procedure can be modelled as a tree search problem, as is shown in Figure 4.4.

We construct a branch and bound tree where we fix a vehicle in the departure sequence at each

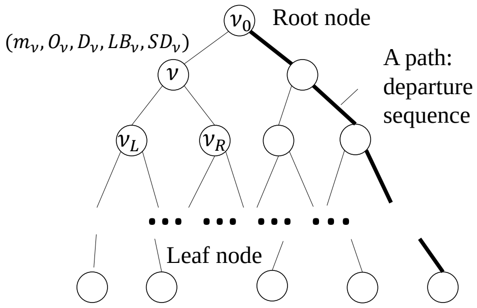


Figure 4.4: Illustration of the branch and bound algorithm

node. In other words, each node consists a vehicle which can be either a connected/automated vehicle or an estimated conventional vehicle. Each node has the following attributes shown in Table 4.1.

Table 4.1: Attributes of a node v.

m_v	arriving approach of the vehicle in node v
O_v	position in the platoon of the vehicle in node v
D_v	departure time of the vehicle in node v
SD_v	sub delay from the root node to the current node v
LB_v	lower bound of the total delay among all the departure sequences that consist node v

The root node of this tree consists of the last vehicle that has departed from the intersection, which gives the initial state (platoon information, departure time of the previous vehicle, etc.). Each child of node v consists of the next vehicle that departs after the vehicle in node v. For this problem, each node v has at most two children (the nodes that are successive to node v), as there are two approaches to discharge from. Therefore, each path from the root node to a leaf node represents a departure sequence. Then finding the optimal departure sequence is equivalent to finding the path that minimizes the total vehicle delay. Denote the set of vehicles as N_m for approach $m = 1, 2$. Then the number of candidate paths is $\begin{pmatrix} |N_1| + |N_2| \\ |N_1| \end{pmatrix} = \frac{(|N_1 + N_2|)!}{|N_1|!|N_2|!}$, where $|\cdot|$ represents the number of element and ! represents factorial.

A branch and bound algorithm is used for the tree search problem. This algorithm is based on the depth-first search, which traverses the nodes of the tree from the root node and searches as far as possible along each path before backtracking. The implementation of the depth-first search algorithm uses a stack $\mathcal{S}$. When visiting each node, the algorithm calculates the departure time of the vehicle in this node (Eq.(4.2)-(4.3)) and the optimal speed if this vehicle is automated (Eq.(A.6)-(A.12)). A trimming mechanism is introduced based on a lower and upper estimated bound for the optimal solution. For this problem, a global upper bound, UB, of the optimal solution is defined as the minimum total delay of all the paths found so far. For each node v, a local lower bound, LB_v, is also defined such that LB_v is smaller than the total delay of any path including node v, which will be determined later in this section. Before checking the children of each node, the node is checked against these bounds, and is disvehicleded if it cannot produce a better solution than the best one found so far. Specifically, if $LB_v > UB$, then node v, together with all of its children and descendants, should be trimmed. By applying this trimming mechanism, the computational complexity is drastically reduced.

The solution procedure is written in pseudo code and described in Algorithm 4.1.

Algorithm 4.1 Branch and bound algorithm for solving the optimal departure sequence and trajectory

```
Input: vehicle sets $N_i$ from approach $i = 1, 2$, including the speed and location information; the
    approach $m_0$, platoon $O_0$, and departure time $D_0$ of the last vehicle that has departed.
Output: the optimal departure sequence $\hat{J}$; the optimal total delay $TD$; and the optimal speed
    $u_{n,\hat{J}}^{opt}$ for all $n$ in $\hat{J}$.
Build the root node $n_0$ using $m_0$, $O_0$ and $D_0$
Calculate the lower bound $LB_0$ for the root node $n_0$ using Eq.(4.9)
For the same departure sequence $J_0$, calculate the global upper bound $UB$ and $u_{n,J_0}^{opt}$ using
    Eq.(4.2), (4.3), (A.6)-(A.12).
Initialize the stack $\mathcal{S}$ and $\mathcal{S}$.push($n_0$)
Total delay $TD \leftarrow UB$
while $\mathcal{S}$ is not empty do
    node $v = \mathcal{S}$.pop();
    if $LB_v > UB$ then
        Continue
    if $TD > UB$ then
        $TD \leftarrow UB$
        Update the optimal departure sequence the optimal speed
    Build the child nodes $v_L$ and $v_R$ of node $v$ with the first vehicles in $N_i, i = 1, 2$.
    Calculate the departure time $D_L$ and $D_R$ using Eq.(4.2) and Eq.(4.3)
    Calculate the sub delay from the root node to $v_L$ and $v_R$, respectively, i.e. $SD_L \leftarrow
        SD_v + D_L$ and $SD_R \leftarrow SD_v + D_R$
    Calculate the lower bound $LB_L$ and $LB_R$, respectively, using Eq.(4.9)
    $\mathcal{S}$.push($v_L$) and $\mathcal{S}$.push($v_R$)
```

The rest of this section finds the local lower bound LB_v for node v.

Note that from Eq.(4.2) and Eq.(4.3), it holds that

$$D_{n,J} \geq D_{n',J} + \frac{1}{S_m} + \frac{l}{v_f} \tag{4.4}$$

where n is any vehicle and n' is the previous vehicle that departs exactly before n.

Then for any departure sequence J, define $D'_{n,J}$ such that

$$D'_{n_v,J} = D_{n_v,J} \tag{4.5}$$

$$D'_{n,J} = D'_{n',J} + \frac{1}{S_m} + \frac{l}{v_f} \tag{4.6}$$

where n_v is the vehicle in node v; n and n' vehicles in later position than n_v in departure sequence J; n' is the previous vehicle to vehicle n.

It is derived from Eq.(4.4)-(4.6) that

$$D'_{n,J} \leq D_{n,J} \tag{4.7}$$

Denote J_n as the ordered set of vehicles in J starting from n_v. Denote SD_n as the sub delay from the root node to node v. Then it holds by Eq.(4.7) that

$$SD_n + \min_{J_n} \sum_{n \in J_n / \{n_v\}} D'_{n,J} \leq \min_{J} TD_J \tag{4.8}$$

By Eq.(4.8), the lower bound LB_ν can be defined as

$$LB_\nu = SD_n + \min_{J_n} \sum_{n \in J_n / \{n_\nu\}} D'_{n_\nu, J} \tag{4.9}$$

Then the following model needs to be solved.

$$\min_{J_n} \sum_{n \in J_n / \{n_\nu\}} D'_{n_\nu, J} \tag{4.10}$$

Notice that only the first term of the penalty in Eq.(4.3) is considered when determining the lower bound. Then, only two possible departure sequences in vehicle set $J_n/\{n_\nu\}$ solves the optimization model Eq.(4.10) and yields the lower bound LB_ν in Eq.(4.9): i) the departure sequence that discharges all vehicles from approach 1 first, then discharges all vehicles from approach 2; ii) the departure sequence that discharges all vehicles from approach 2 first and then discharges all vehicles from approach 1. These two cases are compared to find the value of LB_ν using Eq.(4.9). Note that this lower bound may not be realized as only the first term of the penalty function Eq.(4.3) is considered.

4.3 Simulation Settings

A micro-simulation platform is coded in Java to evaluate the proposed strategy. There are two interacting layers in the simulation framework: 1) the real layer simulates the traffic dynamics using the arrival information and the control policy; 2) the control layer calculates the control policy (departure sequence and trajectory) using the real traffic information.

The real layer consists of the vehicle following behaviors, the vehicle dynamics and the configurations of the intersection and the signal. Many vehicle following models have been proposed over the years, either deterministic (Treiber et al., 2000; Wiedemann, 1974) or stochastic (Chen et al., 2010; Treiber et al., 2006). For simplicity, the vehicle following behavior is assumed as the Intelligent Driver Model (IDM) (Treiber et al., 2000) to represent a more realistic driving behavior. The parameters in IDM are calibrated using the Lankershim Dataset from the Next Generation Simulation (NGSim) project (Alexiadis et al., 2004). The maximum acceleration rate is 1.8m/s^2, the desired deceleration rate 3m/s^2, the minimum spacing is 2.4m, the vehicle length 4.8m, the reaction time 1.4s and the desired speed 60km/hr. The dynamics of the vehicles follows the basic kinematic equations discretized by the time step 0.01s. The length of intersection is l =5m, the maximum green time 60s, and the minimum green time 5s. The minimum and maximum green times are set at the same levels for both the actuated control and the proposed control strategies.

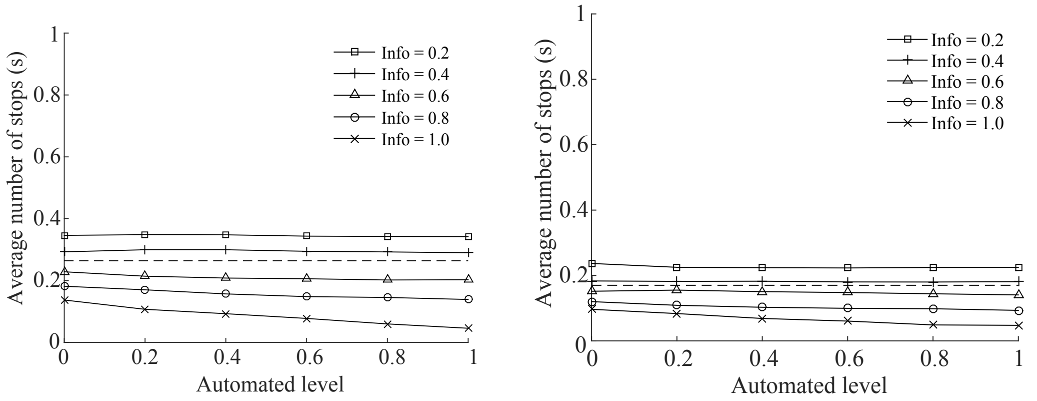
In the control layer, the parameters are defined as: the acceleration rate a = 1.8m/s^2; the saturation flow rate $S_1 = S_2$ = 1800veh/h; free flow speed u_f = 60km/h; backward wave speed w = 20km/h; length of the zone of interest[5] is 100m; minimum speed for trajectory design $u_{\min}$ = 10km/h.

Notice that the vehicle following models used in the simulation and the control strategy are different. This is to show that the simplification in the control strategy performs well in practice.

The total input flow (combined flow of the two approaches) is set to vary between 1000 and 2000veh/h. Notice that the cases with a total flow of 2000 veh/hr represent oversaturated conditions. The demand ratio (ratio of total demand between the two approaches, i.e. flow on approach 1 divided by flow on approach 2) varies between 0.2 and 1. A small demand ratio

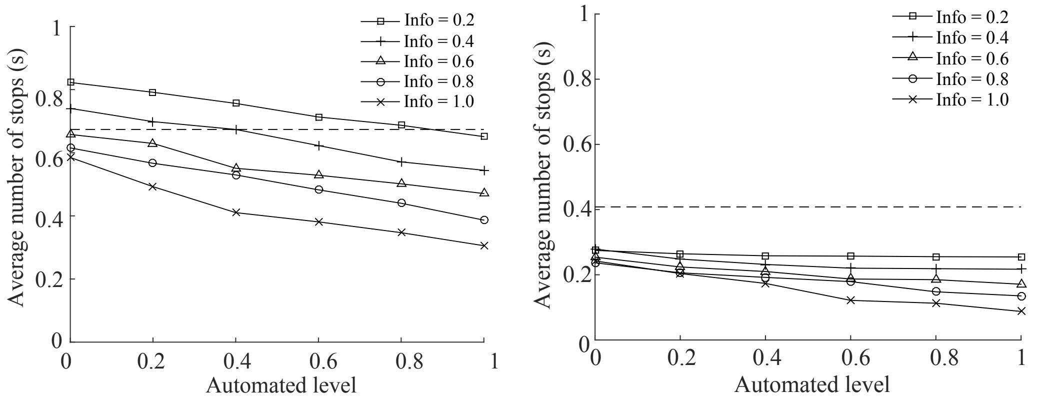
---

[5] A length of 100 meters is chosen because this is the minimum of the typical city block size (100m) and the average DSRC range used in the literature (300m). Also note that this choice is conservative. The performance is expected to improve by increasing this length, as there is more information provided by connected/automated vehicles.



(a) total flow 1000veh/hr, demand ratio 1.0

(b) total flow 1000veh/hr, demand ratio 0.2



(c) total flow 2000veh/hr, demand ratio 1.0

(d) total flow 2000veh/hr, demand ratio 0.2

Figure 4.5: Simulation results for average number of stops per vehicle. Solid lines with different labels represent the results of the proposed strategy under different information levels. Dashed lines represent the results of the actuated signal control algorithm.

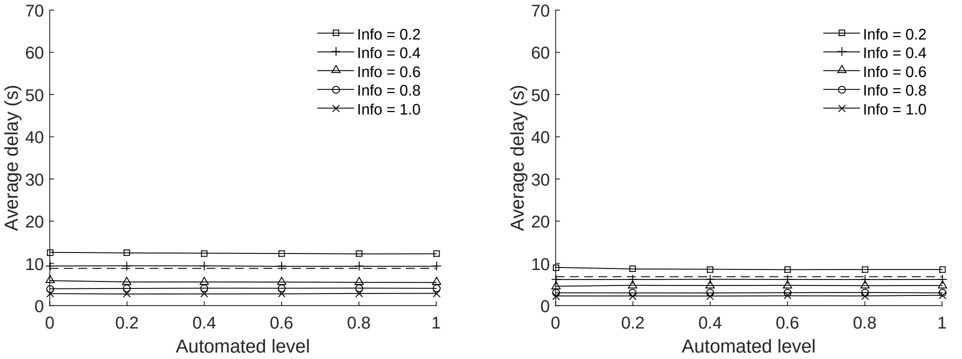
means that the demand is unbalanced. Arrivals are generated randomly assuming an exponential headway distribution. The expected headway equals the inverse of the flow for a given approach.

Note that at least a small fraction of connected and/or automated vehicles are required as they constitute the only information source (as there are no fixed detectors). Therefore, the information level is set to vary between 0.2 and 1 in the simulation. The automated level ranges from 0 to 1. As the automated level decreases to 0, the control strategy will be reduced to the algorithm in Guler et al. (2014) (i.e. without trajectory design).

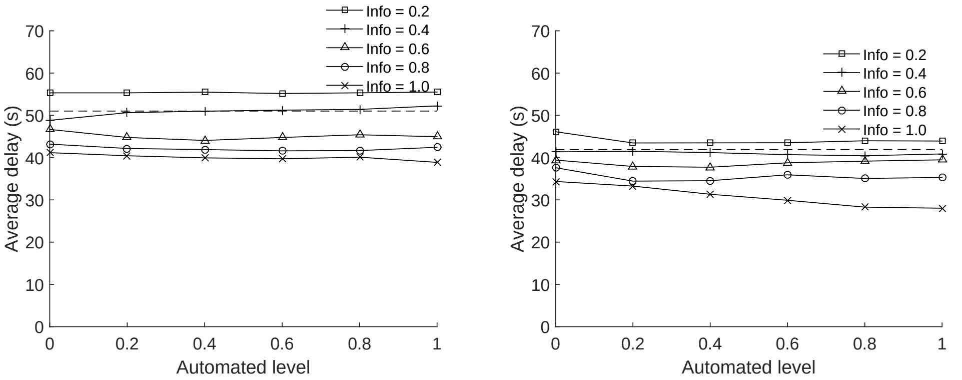
A simulation of 400 vehicles is run for 20 different random seeds for each scenario tested. Average delay and average number of stops per vehicle are recorded.

4.4 Case Study and Results

The performance of the proposed strategy is evaluated by comparing the resulting average vehicle delay and average number of stops to an actuated signal control algorithm. The actuated signal control algorithm assumes the presence of fixed detectors near the intersection on both approaches. It operates as such: the signal switches to red if 1) no vehicles have arrived on the current approach for 5 seconds or 2) the green time exceeds the maximum green time of 60 seconds. The location of the loop detectors is chosen at 65 meters upstream the intersection, which is optimized by a sensitivity analysis. Simulations are conducted for scenarios of different total flows, demand ratios, information levels and automated levels. The results for average



(a) total flow 1000veh/hr, demand ratio 1.0 (b) total flow 1000veh/hr, demand ratio 0.2



(c) total flow 2000veh/hr, demand ratio 1.0 (d) total flow 2000veh/hr, demand ratio 0.2

Figure 4.6: Simulation results for average delay per vehicle. Solid lines with different labels represent the results of the proposed strategy under different information levels. Dashed lines represent the results of the actuated signal control algorithm.

number of stops and average delay per vehicle are shown in Figure 4.5 and Figure 4.6 respectively.

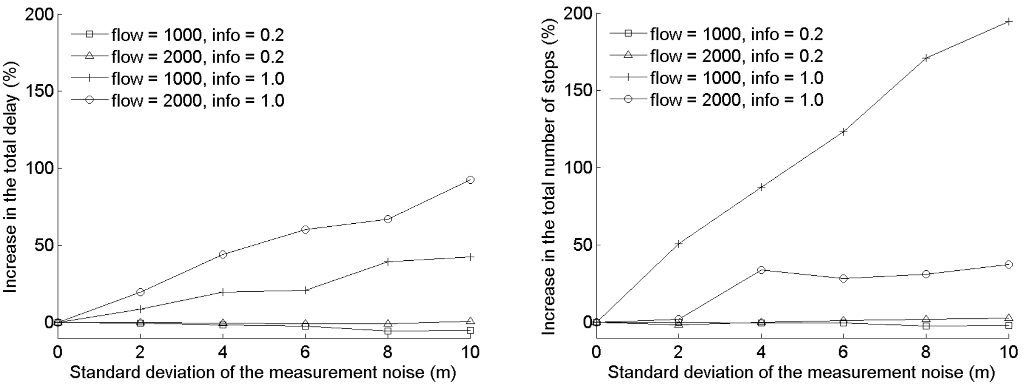
The results show that the proposed strategy performs better for both average number of stops and average delay for a moderate information level. Generally speaking, we require 40% penetration rate of vehicles that provide information to perform better than the actuated signal control algorithm. This is an expected result because the actuated signal control algorithm uses information extracted from infrastructure devices (e.g. loop detectors, video cameras, etc.), whereas the proposed strategy does not utilize any of such information. In the case where the proposed strategy could also use information from other sources (besides the connected and the automated vehicles themselves), it is expected to outperform the actuated signal control algorithm for a much lower penetration rate.

The effects of trajectory design can also be observed from the trend (i.e. monotonicity) of each curve in Figure 4.5 and 4.6. This is because the scenarios with automated level of 0 represent the cases without trajectory design. As is shown in Figure 4.5 and 4.6(a)-(b), the average number of stops in all scenarios tested and the average delay in the low demand scenarios decrease monotonously with the increase of automated level. This means that the trajectory design successfully reduces the average number of stops in all scenarios and the average delay in low demand scenarios. However, the trajectory design does not perform as well for the high demand scenarios for the average delay. In fact, for high demand scenarios, the performance of trajectory design is highly sensitive to the information level. For low information levels, the average delay might increase with the increasing percentage of automated vehicles, as is shown in Figure 4.6(c) and 4.6(d). This happens because in the high demand scenarios, there are vehicles lingering for more than one cycle and forming long queues due to oversaturation. As information drops, the proposed strategy will have insufficient knowledge of the arrivals and the queues. In such cases, although trajectory design is still possible, it is very likely to give an unsatisfactory solution which may give rise to a non-optimal signal timing strategy. Hence, this unsatisfactory trajectory could be detrimental to the system. For high information levels, however, the trajectory design efficiently reduces both number of stops and delay. Therefore a better approach is to use the proposed strategy without trajectory design in the high demand and low information cases.

4.5 Sensitivity Analysis of the Proposed Strategy

4.5.1 Robustness to location errors

Due to the noises in measurement and the data corruption in transmission, the location information received by the central controller may not be accurate. This section tests how the errors in location affects the performance of the proposed strategy. Similar to Section 3.7.2, we assume the location errors and the speed errors follow a joint Gaussian distribution. The errors (location and speed) at different time steps are assumed to be independent. The mean of the Gaussian distribution is assumed as 0 for both location and speed, which represents that there are no systematic errors. The standard deviation of the location measurement is assumed as 15.0m and the standard deviation of the speed measurement is assumed to be 2m/s (Florin and Olariu, 2015). A Kalman filter is implemented to reduce the noise in the location measurements (a detailed description of this algorithm can be found in A.3. The estimation error of the location is not sensitive to the speed measurement errors after the implementation of the Kalman filter (see Figure A.1). Then the covariance matrix is assumed to be $R = [225, 0; 0, 4]$. The other simulation settings are the same as in Section 4.6. The results are shown in Figure 4.7.



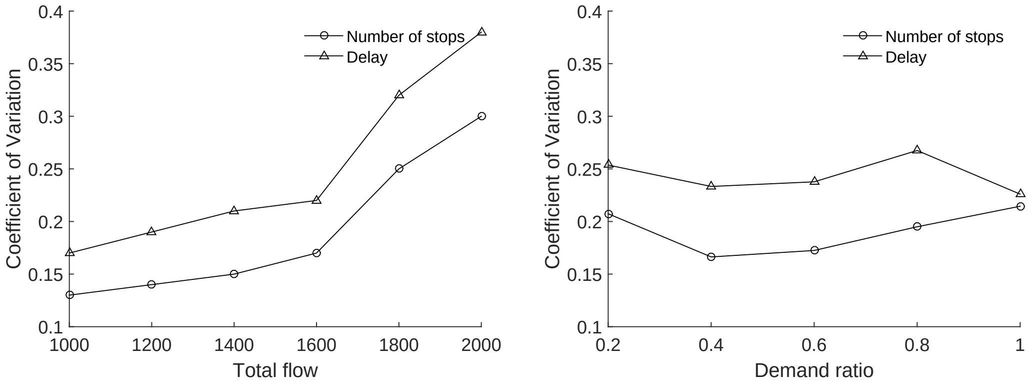
(a) delay

(b) number of stops

Figure 4.7: Robustness to measurement errors. The dashed lines represent the results of the actuated control algorithm.

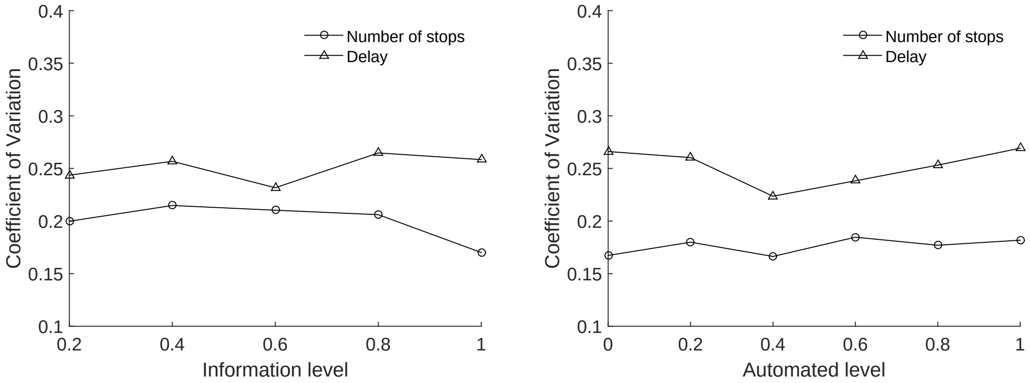
Figure 4.7 shows that the proposed strategy outperforms the actuated algorithm in the cases with full information even when there are measurement errors. It can also be seen that the effect of these measurement errors is larger if the information level and automation level increase. This is because as more vehicles report information and receive trajectory guidance, the system receives more noisy data.

Moreover, the actuated algorithm assumes that the loop detector data is available. If the loop detector data is integrated in the proposed strategy, the proposed strategy can be expected to always outperform the actuated algorithm. Also note that these results are obtained using very conservative assumptions as automated vehicles in reality should not produce large measurement noises. Therefore, the proposed strategy is expected to work properly in reality.



(a) total flow

(b) demand ratio



(c) information level

(d) automated level

Figure 4.8: Average coefficient of variation with different total flows, demand ratios, information levels and automated levels.

4.5.2 Robustness to arrival patterns

This subsection tests how robust the proposed strategy is to different arrival patterns. For each combination of total flow rate, demand ratio, information level and automated level, the measure for robustness is defined as the coefficient of variation, i.e. the ratio between the standard deviation and the mean of the average delay or the average number of stops. The coefficient of variation measures the dispersion of the results in the data set, and is comparable across different data sets even though they do not have the same mean.

Multiple tests show that the coefficient of variation is not sensitive to the information level or the automated level (as is shown in Figure 4.8(b)-4.8(d), but highly related to total flow. Particularly, the coefficient of variation increases drastically with total flow. The trend is very similar for all demand ratios. Therefore, for demonstration purposes, the coefficient of variation is averaged over different demand ratios, information levels and automated levels to show the impacts of total flow on the robustness. Figure 4.8(a) shows the results.

As is shown in Figure 4.8(a), the coefficient of variance increases as the total flow increases. This means that the proposed strategy is more sensitive to the arrival patterns as the total flow increases, and the intersection becomes more saturated. Hence, the arrival patterns will have a large influence on the temporal and spatial extent of the queue, which impacts the performance of the proposed strategy (especially for low information levels). This also helps explain the oscillating pattern of the performance index in the high demand cases in Figure 4.6.

4.5.3 Efficiency of the branch and bound algorithm

This section tests the efficiency of the proposed branch and bound algorithm. A total of 1896 cases are tested. The algorithm finds the optimal solution in all cases tested. The efficiency of this branch and bound algorithm is evaluated by comparing the average number of nodes visited to obtain the optimum departure sequence to an enumeration algorithm. The enumeration algorithm visits all the nodes in the tree. The results are shown in Table 4.2.

Table 4.2: Efficiency of the branch and bound algorithm (nodes visited).

# of vehicles in zone of interest	14	15	16	17	18	19	20	21
Enumeration	28001	54931	123112	174284	223887	290485	437376	1139544
Branch and bound	1385	1616	1832	2236	2306	2096	2088	2404
Ratio	40.4	68.0	134.4	155.9	194.2	277.2	418.9	948.0

Table 4.2 shows the trend of both the enumeration method and the branch and bound method when the number of vehicles in the zone of interest increases. It can be seen from Table 4.2 that the branch and bound method is significantly more efficient than the enumeration method. In contrast to the exponentially increasing complexity of the enumeration method, the computational complexity of the branch and bound increases more linearly with increasing the number of vehicles.

The average run time for each case using the branch and bound algorithm is 0.6 milliseconds, in contrast to 26 milliseconds using the enumeration method. The speed at which the solution can be obtained with the branch and bound algorithm is promising for increasing the complexity of the problem to account for multiple lanes and approaches.

4.6 A Demand Responsive Control Scheme

As discussed in Section 4.3, the proposed strategy has limitations especially for low information availability cases. In this section, a demand responsive control scheme is proposed to adapt to different traffic situations. This strategy dynamically switches between the two (or three) algorithms based on demand and information level: the proposed strategy with and without trajectory design, and the actuated signal control algorithm if the necessary infrastructure (e.g. detectors) is available.

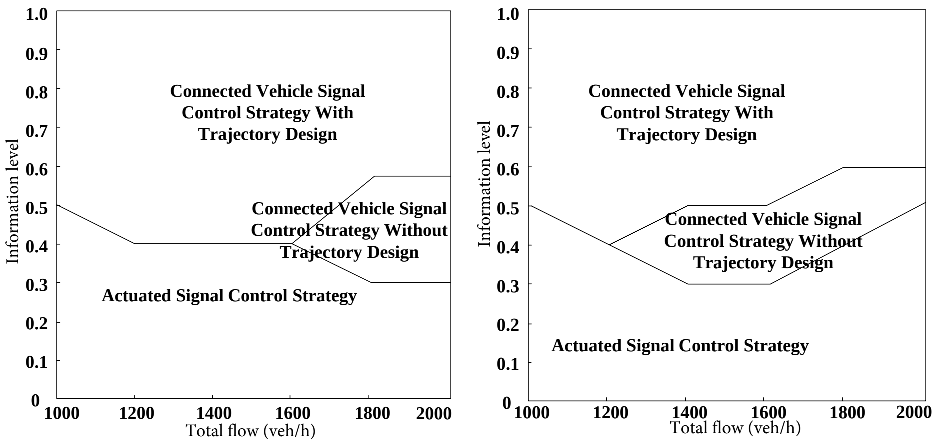
The demand responsive scheme is determined by identifying the bounds for each algorithm. To do so, simulation is conducted for different total flows, demand ratios, information levels and automated levels. Other parameters remain the same. Simulation results are shown in Figure 4.9[6]. Two polylines represent the boundaries for the three algorithms. Notice that those boundaries are conservative because loop detector information is not integrated into the proposed strategy, but is used for actuated signal control.

The upper boundary distinguishes between the proposed strategy with and without trajectory design. The boundary is calculated as the minimum information level such that using trajectory design gives a better result in both average delay and average number of stops. In other words, for a certain combination of information level and total flow, trajectory design is only adopted if both average delay and average number of stops can be improved for all automated levels (≥ 0.2). This is a conservative application bound.

The lower boundary, i.e. the boundary between actuated signal control strategy and connected vehicle signal control strategy, is determined by the minimum information level such that

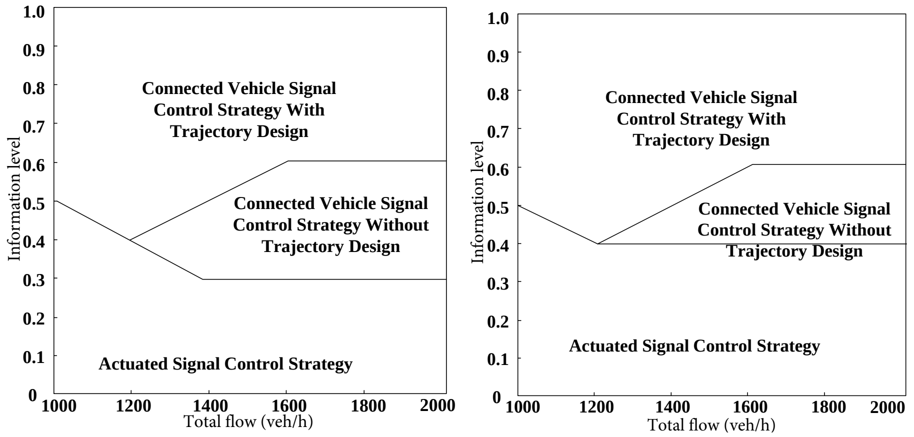
---

[6] The upper boundary determines when the trajectory design should be used. If the system is worse off with the trajectory design, then the algorithm without trajectory design is advised by Figure 4.9.



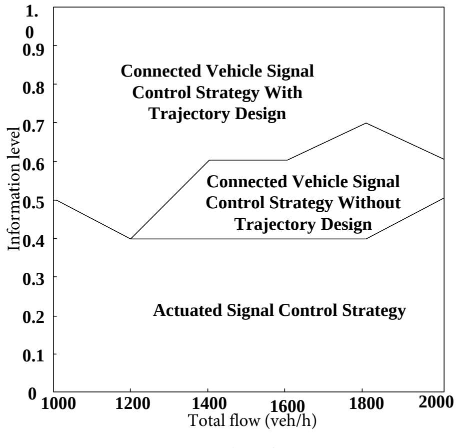
(a) Demand ratio = 0.2

(b) Demand ratio = 0.4



(c) Demand ratio = 0.6

(d) Demand ratio = 0.8



(e) Demand ratio = 1.0

Figure 4.9: Demand responsive control scheme based on connected vehicle technology.

connected vehicle strategy performs better than the actuated signal control strategy in terms of both average delay and average number of stops. For a given information level, demand ratio and total flow, the connected vehicle strategy is chosen only if both performance indices are improved for all automated levels. Similar to the upper one, this is also a conservative application bound.

The two boundaries divide the traffic parameter space into three parts, each of which corresponds to one of the three algorithms considered. When the current traffic situation lies in a certain part of this figure, the algorithm corresponding to that zone will be used. It is shown in

Figure 4.9 that the proposed strategy always requires less than 50% vehicles to provide information in order to outperform the actuated signal control algorithm. This is less strict than most of the existing research works (He et al., 2012; Li et al., 2014b) that require all or a majority of the vehicles to be connected or automated.

The application bounds can be relaxed if loop detector information is integrated into the proposed strategy. Simulations show that the performance is sensitive to the location of the loop detector (Yang et al., 2015). If information from a loop detector installed further than 30 meters upstream of the intersection is used, the proposed strategy can always achieve a better performance than the actuated signal control algorithm. Hence, only the proposed strategy with and without trajectory design should be considered.

4.7 Conclusions and Future Work

In this chapter, we propose a bi-level optimization based strategy that jointly optimizes the trajectory planning and signal control of simple local intersections. A branch and bound algorithm is developed to solve the complex embedded optimization problem. A Kalman filter is also adopted to reduce the impact of the measurement errors. We further integrate three different stages of technology deployment by developing a heuristics to switch the different features of the signal control depending on the stage of the technology.

Various simulations are conducted to evaluate the performance of the proposed strategy. The results show that with information levels greater than 50%, the delay and number of stops can be reduced compared to the actuated signal control algorithm. A demand responsive scheme that combines the proposed strategy with and without trajectory design with a simple actuated signal control algorithm was proposed based on the simulation results. It is also shown that the proposed strategy is not sensitive to measurement noises.

In this chapter, we use Newell's vehicle following model to simplify the calculation of the trajectory design. In principle, any vehicle following model can be used here. As long as the trajectories of previous vehicles are given, the design speed to enter the intersection can be determined.

The results of this chapter can also be generalized to show the influence of package loss (i.e. the loss of information when a vehicle communicates to the central controller). Such loss is equivalent to reducing information level or automated level. As discussed in Section 4.4, the performance indices would be worse if information level is lower. However, for package losses within a certain range, there would not be much deterioration in performance.

In this chapter, we consider simple intersections as an initial building block. Future work includes further generalization of the proposed strategy to a complex intersection. The increase in complexity can be driven by two factors: more modes or more intersection movements. In multimodal scenarios, there might be some traffic modes requesting priority, e.g. public transport vehicles, emergency vehicles, etc. One way to deal with the priority requests would be to change the objective function, e.g. total passenger delay (public transport vehicles), total value of time (emergency vehicles), etc. An initial step towards multimodal control systems will be shown in Chapter 5. On the other hand, the proposed fast branch and bound algorithm makes it possible to generalize the model for more intersection movements. It is expected that for a larger scale of intersection, the model can be solved with reasonable computational time. A difficulty that might arise from considering a more complex intersection is, however, the planning of the turning trajectories and lane changings in scenarios where conventional, connected and automated vehicles coexist in the traffic streams. Another possible extension of this work is coordinated control of multiple intersections, which involves the cooperation not only between vehicles and intersections, but also among intersections. Ongoing efforts are being made in this direction.

Chapter 5

Intersection Control with Transit Signal Priority in a Connected Vehicle Environment

This chapter is partially based on the following papers.

- Yang, K., M. Menendez and S. I. Guler (2018b) Implementing transit signal priority in a connected vehicle environment with and without bus stops, *Transportmetrica B: Transport Dynamics*, 1–23.

- Yang, K., S. I. Guler and M. Menendez (2015) A transit signal priority algorithm under connected vehicle environment, paper presented at the *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 66–70.

- Yang, K., M. Menendez and S. I. Guler (2016b) Using connected vehicle technology to optimize transit signal priority, paper presented at the *Transportation Research Board 95th Annual Meeting*.

5.1 Objectives and Contributions

Public transport vehicles are the most important type of vehicles requesting priority at urban intersections. With the potential ability to move a large number of passengers, public transport services can use the road space more effectively than private cars and thus relieve traffic congestion in urban areas. Existing research aims to provide TSP to buses without over-penalizing the cars in the conflicting approaches in a connected vehicle environment (He et al., 2012, 2014; Hu et al., 2015, 2016; Lee et al., 2017). As established in Section 2.3 (research gap L2), all of these works focus on signalized intersections, without considering the coordination between traffic signals and bus infrastructure (bus stops) and/or bus operations (bus schedule).

This chapter aims to address this research gap and extend Chapter 4 by incorporating TSP. The contributions are two-fold.

1) We introduce buses and develop a TSP algorithm that minimizes person delay in a connected or semi-connected environment. The proposed methodology does not need platooning information, but utilizes the platooning effect by optimizing the departure sequence.
2) We further coordinate the traffic signal operation with both the bus stops and the bus schedule. This further minimizes passenger delay by only providing priority when it is optimal from a system's perspective.

This chapter is organized as follows. Section 5.2 presents the developed algorithm. Section 5.3 describes the simulation framework. Section 5.4 shows the simulation results and evaluates the performance of the algorithm. Section 5.5 conducts sensitivity analysis to analyse the robustness of the algorithm. Section 5.6 concludes the chapter.

5.2 General Methodology

In this section, a TSP algorithm is proposed by integrating public transport into the control strategy proposed in Chapter 4, using the information obtained from connected vehicles. Notice that for the simplicity of presentation, we do not consider automated vehicles and the planning of their trajectories. This, however, can be incorporated by following similar approaches as in Section 4.2.3.

Similar to Chapter 4, we consider a simple scenario of an isolated intersection with two one-way traffic streams as an initial building block, as is shown in Figure 5.1. The intersection can be either with no bus stop, a far-side bus stop, or a near-side bus stop. The framework proposed in this chapter can be generalized into a more complicated intersection.

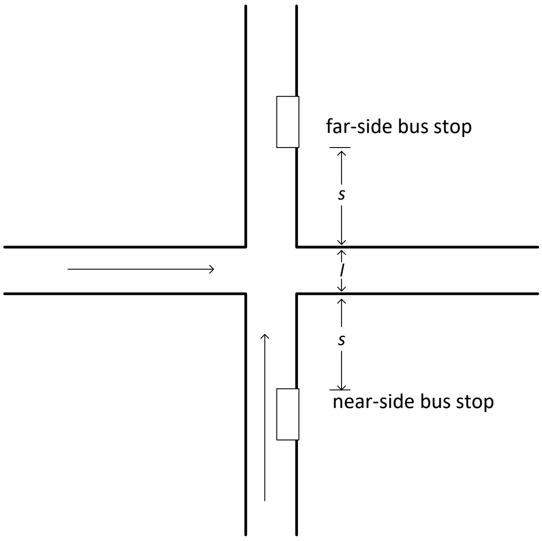


Figure 5.1: Intersection layout.

Two vehicle modes are considered: cars and buses. The buses are assumed to travel on a mixed lane with the cars. Note that this work can be easily extended to include other modes. However, this work focuses on the bus mode due to the unique challenges it presents (i.e. multiple passengers and stops). All buses and a certain percentage of cars are assumed to be connected vehicles.

As before, each vehicle n (bus or car) is represented by a triple indicator (i, j, m) where $i \in I$ is the arrival sequence, $j \in J$ is the departure sequence, and $m \in M$ is the approach. The arrival information is obtained similar to Section 4.2.1 based on the location and speed information provided by each connected vehicle via V2I communication system. Note that for a bus, the virtual departure time would include the dwell time at a near-side bus stop (if exists).

Similar to Chapter 4, each decision step is triggered by one of three events: 1) a bus or connected car enters the zone of interest[7]; 2) a bus or connected car comes to an unscheduled stop (i.e. not at the bus stop for the purpose of passengers boarding o alighting); or 3) a bus or a connected car leaves the downstream line of the intersection.

The rest of this section presents the methodology to determine the optimal departure sequence as well as some modifications to address different scenarios. Section 5.2.1 presents the algorithm without considering bus stops. Section 5.2.2 adapts the algorithm to the scenarios with bus stops while still considering signal delay. Section 5.2.3 further modifies the algorithm to minimize schedule delay for buses.

---

[7] Note that the algorithm is triggered by any bus entering the zone of interest even if there is a near-side bus stop. Predicting the bus arrival time to the near-side bus stop can help optimize the operations of cars.

5.2.1 Algorithm without considering bus stops

This subsection presents the general algorithm without considering bus stops. In each decision step, the algorithm optimizes the total passenger delay locally based on the arrival information. The optimization horizon is the time such that all these vehicles can be discharged. The solution of the optimization can be based on the branch and bound algorithm proposed in Section 4.2.4. All possible departure sequences are generated for all vehicles in set N. In each departure sequence J, the first-in-first-out queueing system holds for each individual approach.

For each departure sequence J, the resulting total delay of all vehicles currently in the system is predicted in an iterative manner with a similar idea as in Chapter 4. For vehicle n, the virtual departure time V_n is defined based on the free flow travel time from its detection point to the downstream end of the intersection. Then for a given departure sequence J, the predicted time for car $n = \{i, j, m\}$ to enter the intersection, $E_{n,J}$, is calculated as Eq.(5.1)

$$E_{n,J} = \max\{V_n - \frac{l}{u_f},\ E_{n',J} + \frac{1}{S_{n',J}}\} \tag{5.1}$$

where l is the length of the intersection; u_f is the free flow speed; n' the previous vehicle in the departure sequence; and $S_{n',J}$ is the saturation flow rate for the previous vehicle n', which is defined as

$$S_{n',J} = \begin{cases} S^b, & \text{if } n' \text{ is a bus} \\ S^c, & \text{if } n' \text{ is a car} \end{cases} \tag{5.2}$$

where S^b is the saturation flow for buses; S^c is the saturation flow for cars. Note that the buses and cars usually have different saturation flows, thus the different saturation headways. Here it is assumed in Eq.(5.1) that the headway between two vehicles depends only on the leader (as the headway is computed based on the front vehicle) . This is an approximation to simplify the calculation, and does not affect the results significantly[8].

Based on Eq.(5.1), the predicted departure time for car $n = \{i, j, m\}$ from the downstream end of the intersection, $D_{n,J}$, is calculated as the maximum of the virtual departure time and the next possible departure time, i.e.

$$D_{n,J} = \max\left\{V_n, E_{n',J} + \frac{1}{S_{n',J}} + P_{n,J}\right\} \tag{5.3}$$

where n' is the previous vehicle in the departure sequence on either approach. $S_{n',J}$ is the saturation flow rate for the previous vehicle n' according to Eq.(5.2). Eq.(5.3) means that the intersection is empty before allowing a new car to enter it for safety purposes. The delay penalty $P_{n,J}$ represents the time it takes for each bus or car to cross the intersection, and is given in Eq.(5.4) derived with basic kinematic theory[9].

$$P_{n,J} = \max\left\{\frac{l + L_n}{u_f},\ \frac{\sqrt{2a_n(d_{n,J} + L_n + l)} - \sqrt{2a_n d_{n,J}}}{a_n}\right\} \tag{5.4}$$

where u_f is the free flow speed of vehicle n; l is the length of the intersection; L_n is the length of vehicle n; a_n is the acceleration rate of vehicle n (if vehicle n follows some vehicles with a

---

[8] This holds because of two reasons. First, as the position of the vehicles is denoted as the position of their front end, the vehicle length of the preceding vehicle would have an impact on the saturation headway. Second, the error caused by such simplification is expected to have a larger impact on the case where a bus follows a car. However, the results show that we successfully provide buses with priority and reduce their delay.

[9] In this chapter, the acceleration related to the signal is considered to account for the benefit of platooning. However, the acceleration and deceleration related to bus stops are not considered for model simplicity. In the simulation, all the acceleration and deceleration maneuvers are considered for a realistic environment.

smaller acceleration rate, a_n would adopt the acceleration rate of the leader); $d_{n,J}$ is the distance of vehicle n to the stop line calculated as the sum of jam spacings of all the vehicles in front of n in its platoon. The first term of the right hand side in Eq.(5.4) represents the time when vehicle n crosses the intersection with free flow speed. The second term represents the acceleration time of vehicle n if the initial speed is smaller than the free flow speed, u_f, and the vehicle n accelerates across the intersection. Similar to Eq.4.3, we neglected the case where vehicle n accelerates to the maximum speed in the middle of the intersection. This might cause a small and systematic error which should not affect the results significantly. Notice that Eq.(5.4) is different from Eq.(4.3) for two reasons. First, a bus is usually much longer than a car, and thus the length of the bus cannot be neglected. Second, as buses and cars have different length, we cannot use the position in a platoon to estimate the progression time of a vehicle.

Then, the objective is to find a departure sequence J that minimizes the total delay.

$$\min_J \, TD_J = \sum_{n \in N} o_n (D_{n,J} - V_n) \tag{5.5}$$

where o_n is the occupancy (number of passengers) of vehicle n.

In the final step, vehicles are discharged according to the calculated departure sequence. It is assumed that the signal turns green for an approach right before the first vehicle in that platoon enters the intersection.

5.2.2 Algorithms considering bus stops and signal delay

In this subsection, the aforementioned algorithm is adapted to consider bus stops and a minimization of the signal delay. Two scenarios are studied according to the location of the bus stop: far-side bus stop and near-side bus stop. In both cases, curbside bus stops are assumed.

5.2.2.1 Far-side bus stop

The far-side bus stop is located after the intersection. When a bus dwells at the far-side bus stop, it creates a temporary bottleneck for vehicles that have passed the intersection. If the queue created by the dwelling bus spills back to the intersection, it blocks vehicles from crossing the intersection even if they are given priority. In this case, the green time would be wasted. Hence, the road space, s, between the far-side bus stop and the intersection should be taken into account to avoid spillbacks and signal inefficiency at the intersection. To this end, the formulation of departure times should be adapted to better capture this scenario.

Define $Z_{n_b,J}$ as the time when bus n_b departs from the bus stop, i.e.

$$Z_{n_b,J} = D_{n_b,J} + \frac{s}{u_f} + t_{\text{dwell}} \tag{5.6}$$

where t_{dwell} is the dwell time of the bus. In this chapter, it is assumed that the dwell time of a bus includes the dwell time plus lost time due to the deceleration and acceleration of the bus when entering and leaving the bus stop.

Let δn be the maximum number of vehicles that the road space between the far-side bus stop and the departure end of the intersection can accommodate. Then the departure time of the δnth car, $\hat{n}$, following the bus in the same approach should be modified to

$$D_{\hat{n}} = \max\left\{V_{\hat{n}},\; D_{n'} + \frac{1}{S_{n',J}} + P_{n',J},\; Z_{n_b,J} + \frac{s}{w}\right\} \tag{5.7}$$

where n' is the previous vehicle in the departure sequence on either approach and n_b is the last bus dispatched in the same approach. Notice that the third term implies that a car following the bus cannot depart until the queue formed due to the dwelling bus clears. The first two terms in Eq.(5.7) represent the case where the car $\hat{n}$ is not affected by the queue caused by the bus. The third term in Eq.(5.7) represents the case where the car $\hat{n}$ is blocked by the spillback (note that only one lane roads are considered in this chapter); w is the backward wave speed. The departure time in this case should be the departure time of the bus plus the time it takes for the queue between the far side bus stop and the intersection to dissipate.

5.2.2.2 Near-side bus stop

The near-side bus stop is located before the intersection. In this situation, the bus might be blocked by the traffic before arriving at the bus stop, which would create extra delay. Alternatively, green time might be wasted if given to the approach while all cars are retained upstream of the bus stop. Given our optimization algorithm, this second scenario gets automatically eliminated[10]. Therefore, we focus on the first one.

To adapt to the case with near-side bus stops, the departure time of each car (Eq.(5.3)) in the approach with the bus stop should be modified. Let s be the road space between the bus stop and the stopline of the intersection, and δn be the maximum number of vehicles this road space can accommodate. Let $E_{n,J}$ represent the time vehicle n enters the intersection and $Z_{n,J}$ represent the time vehicle n passes (or leaves) the bus stop. For a car, $Z_{n,J}$ is the time of passing the bus stop; for a bus, $Z_{n,J}$ is the time of leaving the bus stop after the dwell time.

Then for each car n, it passes the bus stop at time

$$Z_{n,J} = \max\{V_n - \frac{s + l + L_n}{u_f}, Z_{n'',J} + \frac{1}{S_{n'',J}},\ E_{\hat{n},J} + \frac{s}{w}\} \tag{5.8}$$

where s is the distance from the bus stop to the stop line at the intersection, n'' is the vehicle preceding n in the same approach, $\hat{n}$ is the δnth vehicle in front of car n in the same approach. Here, the first term is the virtual departure time from the bus stop, the second term is the earliest possible passing time due to the previous vehicle, the third term is the earliest possible passing time due to the signal. Notice that this third term determines at what time the queue due to a dwelling bus would clear the nth car. Hence, if a queue spillback happens, this limits the time at which the car can proceed to the bus stop.

Similarly, for the bus n_b, the time when it starts to leave the bus stop can be calculated as

$$Z_{n_b,J} = \max\{V_{n_b} - \frac{s + l + L_{n_b}}{u_f},\ Z_{n'',J} + \frac{1}{S^b} + t_{\text{dwell}},\ E_{\hat{n},J} + \frac{s}{w}\} \tag{5.9}$$

where the definitions of n'' and $\hat{n}$ are the same as in Eq.(5.8). Note that the virtual departure time V_{n_b} for a bus includes the dwell time at the bus stop. The first term of Eq.(5.9) is the virtual departure time for bus n_b from the bus stop if there is no congestion; the second term represents the departure time of the bus from the bus stop if the bus suffers congestion before it arrives at the bus stop, but is not affected by the queue due to the traffic signal; the third term represents the earliest possible time for the bus to leave the bus stop because of the traffic signal. Notice that here the third term determines when the queue due to the signal would clear the bus stop location. Hence, the queue formed at the signal limits the time at which the bus can leave the bus stop.

---

[10] Consider a scenario where in Approach 1, a bus dwells at the near-side bus stop and cars are blocked behind it. If the priority is given to Approach 1, every vehicle both in Approach 1 and Approach 2 would be behind the bus in the departure sequence, suffering from the delay of the dwell time of the bus. Alternatively, if we give priority to Approach 2 until the bus reaches the intersection, then some vehicles in Approach 2 no longer suffer from delay because of the bus, and the delay of cars queueing behind the bus remains the same. It is obvious that the second possibility is more efficient. Therefore the first possibility gets automatically eliminated.

Then the time vehicle n (car or bus) enters the intersection can be calculated as

$$E_{n,J} = \max\{Z_{n,J} + \frac{s}{u_f},\ E_{n',J} + \frac{1}{S_n}\} \tag{5.10}$$

where n' is the previous vehicle in the departure sequence on either approach. The first term represents the case where vehicle n enters the intersection freely after passing the bus stop. The second term represents the time when vehicle n is allowed to enter the intersection after passing the bus stop as controlled by the signal.

The departure time can be calculated as

$$D_{n,J} = \max\{Z_{n,J} + \frac{s + l + L_n}{u_f},\ E_{n',J} + \frac{1}{S_n} + P_{n,J}\} \tag{5.11}$$

where $P_{n,J}$ is the penalty as defined in Eq.(5.4).

5.2.3 Algorithm considering bus stops and schedule delay

In this subsection, the algorithm is adapted to the scenario where the bus has to maintain a schedule. Assume that bus n_b is scheduled to depart from the bus stop at time $ST_{n_b,J}$. However, due to fluctuations in travel time the bus may depart early, on-time, or late from the bus stop. If the bus departs from the stop earlier than scheduled, no benefit would be achieved for the bus by giving it priority while the performance of the cars might be sacrificed. Hence, with this algorithm the early bus will not receive a priority, only late and on-time buses will receive priority based on their schedule delay. Therefore, the objective function of Eq.(5.5) should be modified to:

$$\min \sum_{n_c \in C} o_{n_c}(D_{n_c,J} - V_{n_c}) + \sum_{n_b \in B} o_{n_b} \max\{Z_{n_b,J} - ST_{n_b,J}, 0\} \tag{5.12}$$

where C is the car set and B is the bus set. The first term represents the total delay for car passengers and the second term represents the total schedule delay for bus passengers. Note that the schedule delay includes the signal delay, if applicable. It is assumed that all buses have a schedule, otherwise signal delay for the unscheduled buses should also be considered.

This objective function is used regardless of whether the bus stop is a near-side or a far-side stop using the definitions of $Z_{n_b,J}$ as presented in Section 5.2.2.

5.3 Simulation Framework

In this section, a simulation is used to evaluate the performance of the algorithm. The simulation framework consists of two interacting layers, the real time traffic simulation layer and the control algorithm layer. The real traffic is simulated by a microscopic simulation package SUMO (Simulation of Urban MObility) (Krajzewicz et al., 2012; SUMO, 2015), to evaluate the performance of the algorithm. SUMO is an open-source, highly portable simulation package that has already been used by research works in traffic control and connected vehicle environment (Pandit et al., 2013; Baiocchi et al., 2015; Ma et al., 2014; Nagle and Gayah, 2015). Flow dynamics in SUMO are based on the car-following model by Krauss et al. (1997). The corresponding parameters are calibrated and validated using the trajectories in the Lankershim Boulevard Dataset (8:30 a.m.-8:45 a.m.) of the Next Generation Simulation project (NGSim, Alexiadis et al., 2004). The calibrated parameters are maximum speed 60km/hr, desired acceleration rate 1.7m/s^2, desired deceleration rate 1.7m/s^2, and minimum gap 2.0m. Notice that the car following model we used here is different from that in Chapter 4. Also, in order to account for buses, we have used different pairs of vehicles to calibrate the car following models. Therefore, the parameters in Chapter 5 and Chapter 4 are sometimes different. The real time vehicle location and speed are simulated by SUMO and sent by its traffic control interface module TraCI to the control layer. In the control layer, the proposed algorithm is coded in Python. The control layer calculates the signal timing using the real traffic information and sends it to the SUMO simulator.

The parameters in the simulation are defined as follows. Different saturation levels are considered by setting the volume to capacity ratio (v/c ratio) to vary between 0.1 and 1.1. We do not consider the lost time of the intersection, thus the capacity of the intersection is taken as 1800veh/hr. Note that a v/c ratio larger than 1 represents the intersection is oversaturated. For simplicity, the capacity is computed based on the cars. Bus flows are set to vary between 10 veh/hr and 30 veh/hr (corresponding to bus frequencies of 6 min and 2 min respectively). Buses are assumed to arrive only from one approach. Arrivals of both buses and cars are generated randomly assuming an exponential headway distribution. The expected headway equals the inverse of the flow for a given approach and a given mode. Car occupancy is assumed to be 1.2 passengers per vehicle, and bus occupancy is assumed to be 50 passengers per vehicle. This assumption is later relaxed and its effects are studied in Section 5.5.2.

The penetration rate of connected cars (i.e. the number of connected cars divided by the total number of cars) is set to vary between 0 and 1 in the simulation (equivalent to the information level evaluated in Chapter 4). All buses are assumed to be connected. For a penetration rate of 0, the only information source is the arrival of buses, therefore the buses will always be given priority. In this case, the algorithm is reduced to a fixed-time signal control algorithm with TSP. If no bus is detected the traffic signal operates with two phases equal to the maximum green time of each approach. Then, when a bus is detected upon its entrance to the zone of interest, it is provided with absolute priority.

The evaluation is based on a simple intersection with two one-way one-lane roads. Turning is not permitted. The length of the intersection is l = 5m. The end of near-side bus stop is assumed to be s = 30m from the stopline of the intersection; and the beginning of the far-side bus stop is assumed to be s = 30m from the end of the intersection. The minimum green time of the signal is 5s. The maximum green time is optimized using the Webster's formula (Webster, 1958). In scenarios with balanced demand, the maximum green time is 60s for both approaches, based on the insights generated in Chapter 4. In scenarios with unbalanced demand (demand ratio of 0.2), the maximum green time is 100s for the main road, and 20s for the side road. Other inputs to the simulation are assumed as: the saturation flow for the buses is S^b = 600veh/hr; the saturation flow for the cars is S^c = 1800veh/h; free flow speed u_f = 50km/h for cars and buses in the control strategy; vehicle length of all buses L_b=12.5m; vehicle length of all cars L_c=5m; length of the zone of interest 100m; acceleration rate (in the control strategy) for all cars 2.0m/s^2; acceleration rate for all buses (in the control strategy) 0.8m/s^2; dwell time of the bus 20s. The assumption regarding the location of the bus stops and the dwell time are later relaxed, and their effects are studied in Section 5.5.1 and Section 5.5.3, respectively. The traffic demand on both approaches is assumed to be either balanced (i.e. none of the approaches have a demand drastically higher than the other) or unbalanced. As in Chapter 4, the demand ratio is 1.0 for scenarios with balanced demand and 0.2 for scenarios with unbalanced demand. A simulation of 400 cars is run 10 times for each scenario tested. This is regardless of the v/c ratio. Although a fixed total number of vehicles implies different number of buses, the delay savings are evaluated in exactly the same scenarios (same car flow, bus flow and random seeds). Therefore the results are still comparable across different variations of algorithms and different penetration rates. For different v/c ratios, the simulation time is different. The total delay across both approaches is recorded.

The simulation is conducted on a PC with Windows 7 system and Intel Core i7 CPU (2.9GHz). Computational cost is recorded for each simulation run. The average running time for a single simulation of 400 cars is less than 1 min. Note that the algorithm is triggered every time an connected car or bus enters the zone of interest or stops, the average computation time for each decision stage is 0.15s, far less than the normal headway between two cars.

5.4 Simulation Results and Algorithm Evaluation

This section evaluates the performance of the proposed algorithms: 1) the algorithm with general TSP (without considering bus stops), 2) the algorithm considering bus stops, and 3) the algorithm considering schedule delay. The evaluation framework is shown in Figure 5.2. The results are summarized in Table 5.1, where the shaded number represents the best algorithm in each scenario. Section 5.4.1 shows the value of TSP in a connected vehicle environment. Section 5.4.2 shows the value of TSP considering bus stops. Section 5.4.3 shows the value of TSP considering bus schedule.

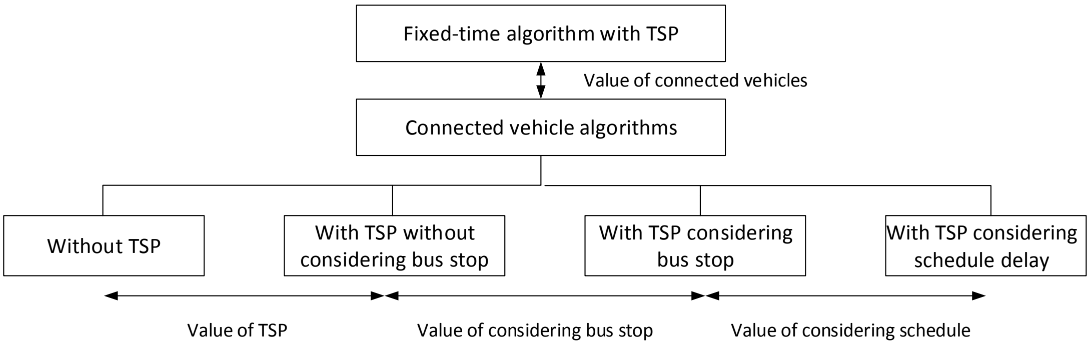

Fixed-time algorithm with TSP
Value of connected vehicles
Connected vehicle algorithms
Without TSP
With TSP without considering bus stop
With TSP considering bus stop
With TSP considering schedule delay
Value of TSP
Value of considering bus stop
Value of considering schedule

Figure 5.2: Evaluation framework.

5.4.1 Value of connected vehicles and value of TSP

The value of TSP is evaluated by comparing the average delay savings per passenger of the connected TSP algorithm without considering bus stops[11] and two algorithms: 1) fixed-time algorithm with TSP; 2) the basic connected vehicle algorithm proposed in Guler et al. (2014) which does not consider the difference between buses and cars. Simulation is conducted with different v/c ratios, bus flows, penetration rates and bus stops. Both balanced demand and unbalanced demand are considered. For presentation simplicity, only results with bus flow of 30veh/hr are shown in this subsection. The results with lower bus flow and other v/c ratios exhibit similar properties.

Figure 5.3 and Figure 5.4 show the average delay savings for the algorithm with general TSP compared to the fixed time strategy and the connected vehicle algorithm without TSP, respectively, including the confidence interval, as a function of the penetration rate. Figure 5.3 shows the results in scenarios with balanced demand, and Figure 5.4 shows the results in scenarios with unbalanced demand (demand ratio is 0.2). It can be seen from Figure 5.3 and Figure 5.4 that the results for balanced demand and unbalanced demand follow a similar trend. The algorithm that use information provided by connected vehicles perform better compared to the fixed-time algorithm in nearly all scenarios, especially the scenarios with the undersaturated or unbalanced demand.

The value of connected vehicles can be seen from the clear bars in Figure 5.3 and Figure 5.4. As is shown in Figure 5.3(a), 5.3(c) and 5.3(e) (also Figure 5.4(a), Figure 5.4(c) and Figure 5.4(e), the connected TSP algorithm without considering bus stop preforms better than the fixed time algorithm with TSP in undersaturated scenarios (v/c ratio of 0.7) with/without bus stops. Same

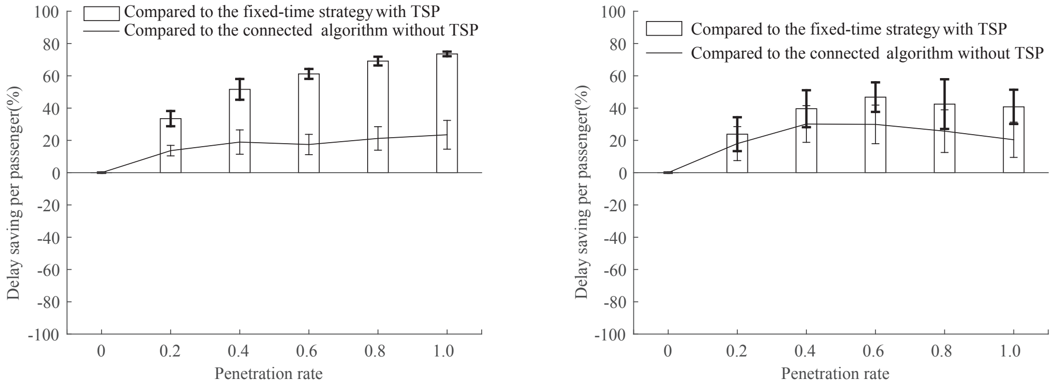
---

[11] There are scenarios with bus stops, but bus stops are not taken into account by the algorithm.

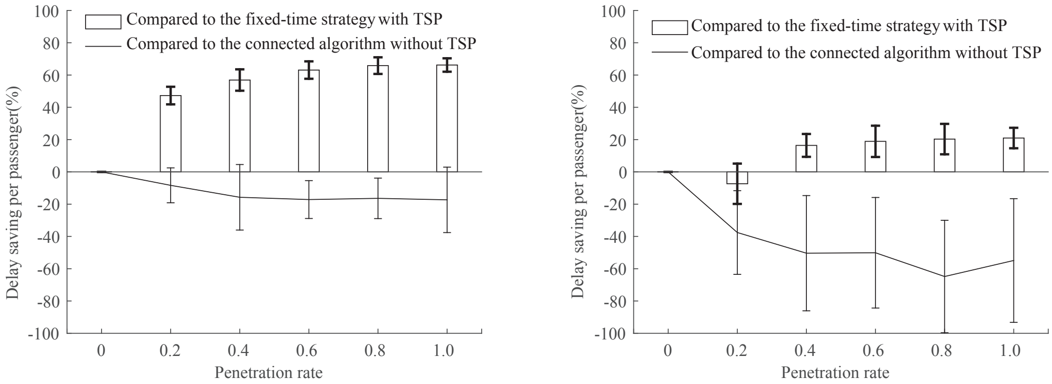
Table 5.1: Summary of results (passenger delay in seconds). The shaded number represents the best algorithm in each scenario.

			balanced				unbalanced			
stop	v/c	pr	fixed-time	Without TSP	TSP without stop	TSP with stop	fixed-time	Without TSP	TSP without stop	TSP with stop
no stop	0.7	0	22.7	22.7	22.7	22.7	12.9	12.9	12.9	12.9
		0.2	22.7	17.5	15.1	15.1	12.9	10.5	9.7	9.7
		0.4	22.7	13.6	11.0	11.0	12.9	9.6	7.9	7.9
		0.6	22.7	10.7	8.8	8.8	12.9	6.8	6.0	6.0
		0.8	22.7	8.9	7.0	7.0	12.9	6.7	5.8	5.8
		1	22.7	7.9	6.0	6.0	12.9	5.8	5.0	5.0
	1.1	0	32.8	32.8	32.8	32.8	16.7	16.7	16.7	16.7
		0.2	32.8	30.7	24.9	24.9	16.7	13.6	11.2	11.2
		0.4	32.8	29.0	19.9	19.9	16.7	10.2	8.3	8.3
		0.6	32.8	25.5	17.5	17.5	16.7	8.3	6.8	6.8
		0.8	32.8	25.4	18.9	18.9	16.7	8.0	6.3	6.3
		1	32.8	24.5	19.5	19.5	16.7	8.5	6.5	6.5
near-side	0.7	0	41.0	41.0	41.0	41.0	25.2	25.2	25.2	25.2
		0.2	41.0	20.0	21.6	16.0	25.2	24.6	25.1	20.1
		0.4	41.0	15.4	17.6	11.5	25.2	19.2	21.3	17.3
		0.6	41.0	12.9	15.0	9.7	25.2	15.9	17.3	14.8
		0.8	41.0	12.0	14.0	8.7	25.2	13.4	14.3	11.5
		1	41.0	11.9	13.9	8.2	25.2	9.6	11.3	8.1
	1.1	0	73.1	73.1	73.1	73.1	35.6	35.6	35.6	35.6
		0.2	73.1	58.3	78.7	32.2	35.6	28.6	32.3	12.3
		0.4	73.1	43.1	61.3	25.7	35.6	21.9	25.1	10.5
		0.6	73.1	40.9	59.0	22.1	35.6	13.2	15.1	9.9
		0.8	73.1	36.8	58.2	21.0	35.6	11.6	14.2	8.5
		1	73.1	39.9	57.6	19.1	35.6	9.0	13.6	8.4
far-side	0.7	0	23.1	23.1	23.1	23.1	16.4	16.4	16.4	16.4
		0.2	23.1	18.1	17.4	15.8	16.4	16.1	12.8	11.4
		0.4	23.1	14.8	13.2	11.4	16.4	15.4	12.1	7.8
		0.6	23.1	11.5	10.8	9.1	16.4	13.1	10.6	5.7
		0.8	23.1	10.8	9.3	7.5	16.4	11.8	9.9	5.5
		1	23.1	9.2	8.2	6.6	16.4	4.4	3.7	3.2
	1.1	0	34.4	34.4	34.4	34.4	19.3	19.3	19.3	19.3
	34.4	0.2	34.4	49.5	39.4	31.2	19.3	20.4	18.3	14.3
	34.4	0.4	34.4	44.8	31.6	23.7	19.3	16.5	14.3	8.1
	34.4	0.6	34.4	43.4	31.6	23.8	19.3	16.1	13.4	6.5
	34.4	0.8	34.4	41.5	34.1	22.5	19.3	15.0	11.2	6.3
	34.4	1	34.4	44.3	32.0	24.0	19.3	10.5	7.2	5.4

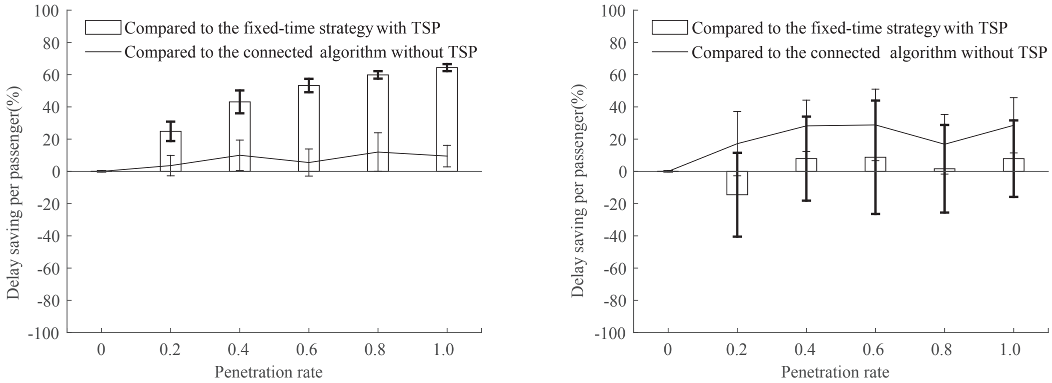
holds for Figure 5.3(b) and Figure 5.4(b), the oversaturated scenario (v/c ratio of 1.1) without bus stops. In these scenarios, the average delay savings per passenger compared to the fixed-time algorithm with TSP can be up to 40% if penetration rate is greater than 0.4. This means that the connected TSP algorithm without considering bus stops successfully reduces the average delay per passenger compared to the fixed-time algorithm with TSP in these scenarios. In other words, there is value in using connected vehicle technology and the information it provides for undersaturated intersections and intersections without bus stops.



(a) v/c ratio = 0.7, without bus stop (b) v/c ratio = 1.1, without bus stop



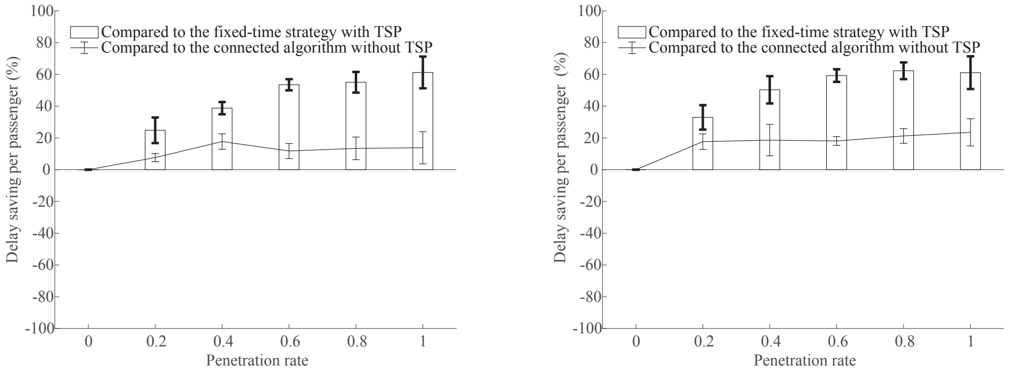
(c) v/c ratio = 0.7, near-side bus stop (d) v/c ratio = 1.1, near-side bus stop



(e) v/c ratio = 0.7, far-side bus stop (f) v/c ratio = 1.1, far-side bus stop

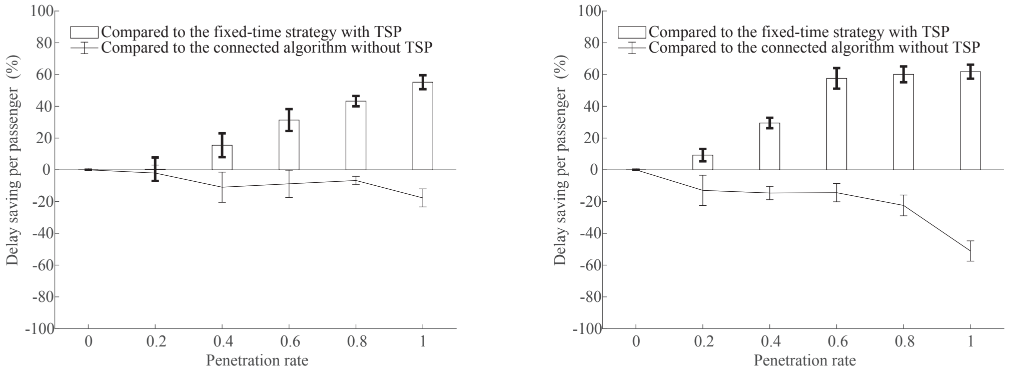
Figure 5.3: Average delay savings per passenger (both cars and buses) for algorithm with general TSP compared to the fixed-time strategy with TSP and the connected vehicle algorithm without TSP. The demand is balanced. Bus flow is 30 veh/hr. Error bars represent the confidence interval of the average delay savings (confidence level 95%).

However, in oversaturated scenarios (v/c ratio of 1.1) with bus stops and balanced demands (Figure 5.3(d), 5.3(f), the general TSP does not necessarily reduce the average delay per passenger compared to the fixed-time strategy, especially for low penetration rates. This is because the algorithm does not consider the bus stops. Here in some scenarios, the green time might be wasted as cars are blocked by buses dwelling at the stop. The results for the unbalanced demand scenarios (Figure 5.4(d) and Figure 5.4(f) are different, as the fixed signal control can yield very poor performance. Besides, in contrast to the balanced demand scenarios, the additional



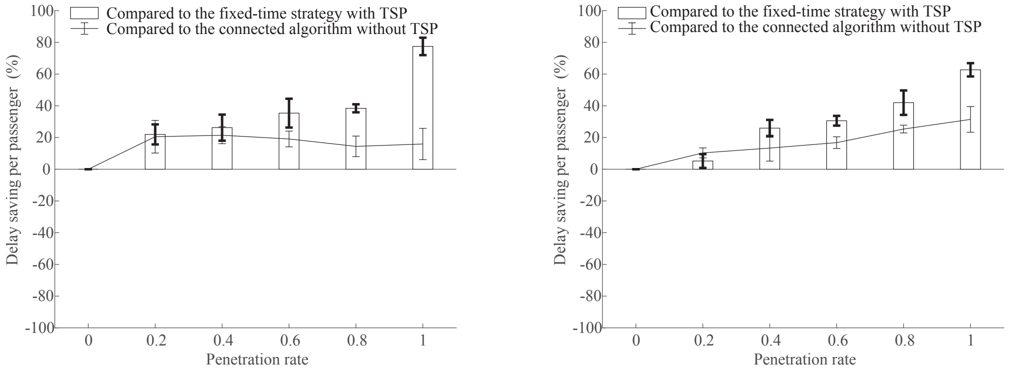
(a) v/c ratio = 0.7, without bus stop

(b) v/c ratio = 1.1, without bus stop



(c) v/c ratio = 0.7, near-side bus stop

(d) v/c ratio = 1.1, near-side bus stop



(e) v/c ratio = 0.7, far-side bus stop

(f) v/c ratio = 1.1, far-side bus stop

Figure 5.4: Average delay savings per passenger (both cars and buses) for algorithm with general TSP compared to the fixed-time strategy with TSP and the connected vehicle algorithm without TSP. The demand is unbalanced. Bus flow is 30 veh/hr. Error bars represent the confidence interval of the average delay saving (confidence level 95%).

information received with the general TSP becomes very valuable.

The value of TSP can be observed by the solid line in Figure 5.3 and Figure 5.4. In scenarios without a bus stop, the connected TSP algorithm without considering bus stops performs better than the connected vehicle algorithm without TSP. This means that giving bus priority in scenarios without bus stop saves average delay per passenger. Hence, providing bus priority is valuable in those cases.

However, providing TSP without considering bus stops may not be desirable if bus stops do exist. In scenarios with far-side bus stops, although the connected TSP algorithm without considering bus stop outperforms the connected vehicle algorithm on average, the variability of the results is large. This means that in the worst cases, providing bus priority without considering stops may be harmful. Similarly, in scenarios with near-side bus stops, the performance of the general TSP algorithm is worse than not considering TSP. This is because the general TSP does not coordinate the bus stop with the signal. For near-side bus stops, the signal might add green time for a dwelling bus to pass the intersection even if there is a long queue in the competing approach. For far-side bus stops, the queue spilled back from the bus stops might block the intersection. Hence, by deciding on signal timings without considering the bus stops, the algorithm greatly sacrifices the performance of the competing approach. Therefore, the algorithm should always consider bus stops when giving bus priority at intersection with bus stops, especially if they are near-side.

Note that in Table 5.1, the delay of the scenarios with the near-side bus stop is higher compared to the scenarios with far-side bus stop. One potential reason is that in the scenarios with the near-side bus stop, both cars and buses are affected by both the traffic signal and the buses. Therefore, the total delay includes both the signal delay and the delay caused by the bus stop. In scenarios with the far-side bus stop, however, the vehicle delay at the intersection is only caused by the traffic signal, thus includes only signal delay.

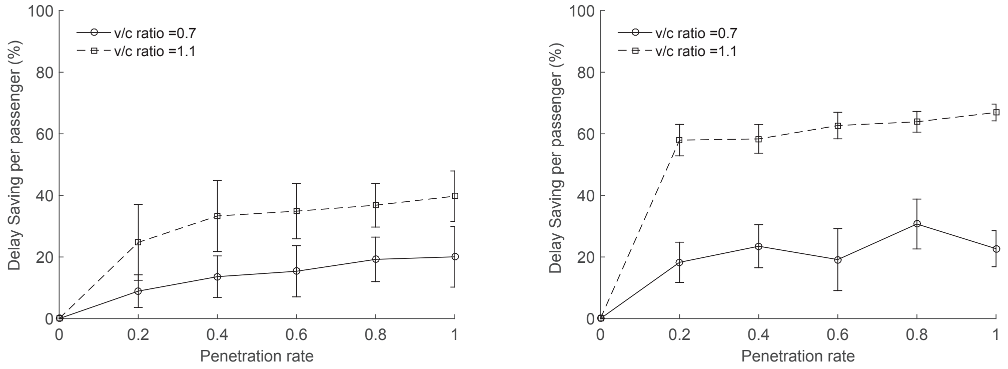
5.4.2 Value of TSP considering bus stops

To evaluate the value of considering bus stops, the delay savings obtained by the connected TSP algorithm considering bus stop compared to the connected TSP algorithm without considering bus stop are shown in Figure 5.5. Note that when there are bus stops, the connected TSP algorithm considering bus stops always outperforms the fixed-time algorithm with TSP, as is shown in Table 5.1. The delay savings are observed for all penetration rates tested for both low and high, balanced and unbalanced demand.

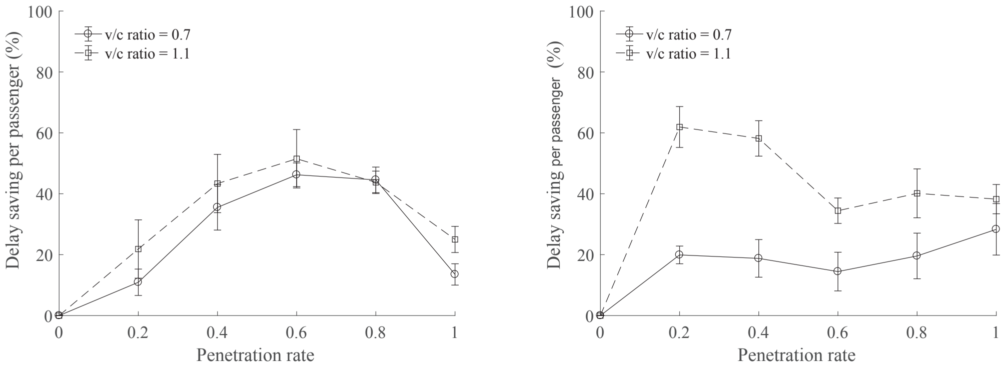
As is shown in Figure 5.5, the connected TSP algorithm considering bus stops outperforms the connected TSP algorithm without bus stops in all scenarios tested, and the results are statistically significant. In scenarios with near-side bus stops and both balanced demand and unbalanced demand (i.e. Figure 5.5(a) and (c), the connected TSP algorithm considering bus stops generally saves 20% more delay for penetration rates larger than 0.2 in undersaturated scenarios, and 40% in oversaturated scenarios. In scenarios with far-side bus stops (i.e. Figure 5.5(b) for balanced demand and Figure 5.5(d) for unbalanced demand), the connected TSP algorithm saves 30% extra delay compared to the general TSP algorithm in oversaturated scenarios and 15% in undersaturated scenarios. It is also shown in Figure 5.5 that even in the worst case, the connected TSP algorithm considering bus stop still outperforms the connected TSP algorithm without considering bus stops. This shows that considering the coordination of bus stops and signals successfully improves the performance, especially for scenarios with near-side bus stops. It can be further seen from Table 5.1 (the shaded numbers) that the best algorithm with the lowest passenger delay is always the TSP considering bus stops in all scenarios tested. In the case without bus stops, the algorithm from Section 2.1 yields a similar performance.

5.4.3 Value of the algorithm considering schedule delay

To evaluate the performance regarding schedule delay, simulations are conducted to compare the algorithm considering schedule delay to the connected TSP algorithm considering bus stops and signal delay. Both scenarios with near-side and far-side stops are simulated. Buses are assumed to have a schedule. The simulation is set up so that buses have the possibility to arrive either early, on time, or late. Two types of such scenarios are tested: 1) 30% of buses are early; 2) 60% of buses are early. The results are summarized in Figure 5.6. For ease of comprehension, the results are shown without confidence intervals.

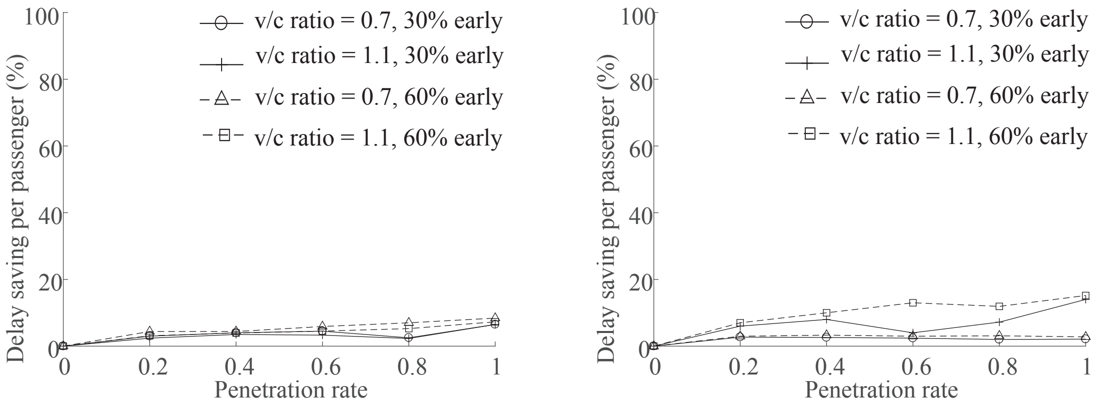


(a) Far side bus stop, balanced demand (b) Near side bus stop, balanced demand



(c) Far side bus stop, unbalanced demand (d) Near side bus stop, unbalanced demand

Figure 5.5: Average delay savings per passenger for the connected algorithm with TSP considering stops compared to algorithm with general TSP. Bus flow is assumed as 30 veh/hr. Error bars represent the confidence interval of the average delay saving (confidence level 95%).



(a) Far side bus stop (b) Near side bus stop

Figure 5.6: Average delay savings per passenger (signal delay for cars and schedule delay for buses) for algorithm with TSP considering schedule delay compared to algorithm considering signal delay. Bus flow is assumed as 30 veh/hr.

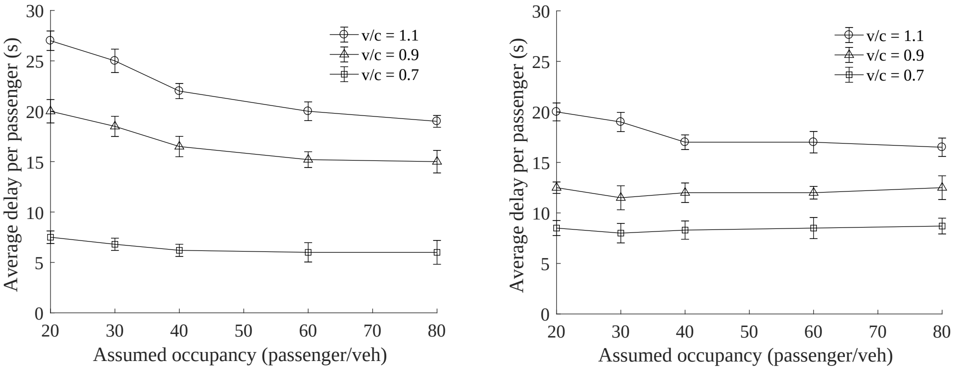
As is shown in Figure 5.6, on average the delay savings are positive compared to the connected TSP algorithms considering signal delay, although they are typically below 10% if 30% of buses are early, and 20% if 60% of buses are early. It can be seen that by increasing the percentage of early buses, the benefit of considering the schedule delay increases. This is because the buses do not receive priority when they are early if schedule delay is considered. The remaining intersection capacity can be given to cars, which reduces the average delay of cars, especially in scenarios with a near-side bus stop.

However, the benefit of considering the bus schedule is not statistically significant compared to the connected TSP algorithm considering signal delay (i.e. most confidence intervals contain zero). The variability of the results is very large, which means that in some cases, by considering the schedule delay, the algorithm performs even worse than by considering signal delay. This is an expected result. The connected TSP algorithm considering signal delay already fully utilizes the intersection capacity and successfully coordinates the intersection and bus stops. Only the cars in the conflicting approach might benefit from accounting for bus schedule. Nevertheless, extra delays might be imparted onto vehicles following the bus (including cars and buses) due to the approach not receiving priority. Therefore, it is expected that the additional delay savings are not significant. In practice, taking into consideration the trade off between implementation cost and the delay savings, it may not be worthwhile to consider bus schedule.

5.5 Robustness of the Algorithm

Below, the robustness of the algorithm is evaluated. Section 5.5.1 discusses the sensitivity to location of bus stops, Section 5.5.2 discusses the sensitivity to assumed occupancy, and Section 5.5.3 discusses the robustness to estimation errors in the bus dwell time.

5.5.1 Sensitivity to location of bus stops



(a) Far-side bus stops

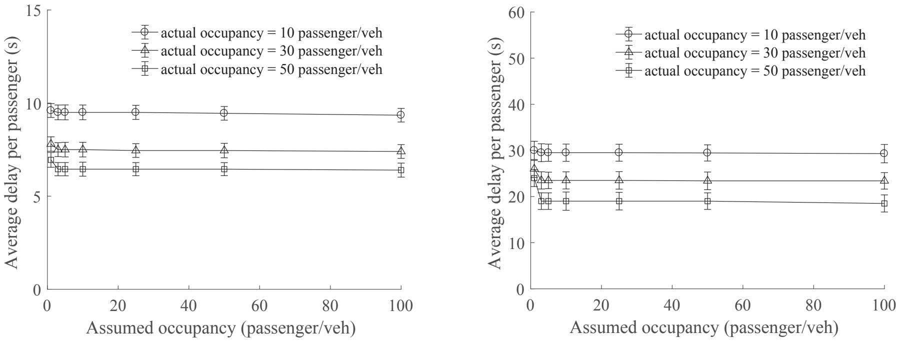
(b) Near-side bus stops

Figure 5.7: Sensitivity to location of stops where bus flow = 30 veh/hr, bus occupancy = 50 passenger/veh and penetration rate = 1.0. The performance index is per passenger delay, and includes both cars and buses. Error bars represent the confidence interval of the average delay saving (confidence level 95%).

The sensitivity of the algorithm introduced in Section 2.2 to the location of bus stops is discussed here. Scenarios with different v/c ratios and bus stop locations are tested. The results are shown in Figure 5.7. For near-side bus stops, the performance of the algorithm is only sensitive to the location of the bus stops in oversaturated cases (v/c = 1.1). For far-side bus stops, the performance of the algorithm is sensitive to the location of the bus stops in near oversaturated or oversaturated cases (v/c ratio > 0.9). This is because as the v/c ratio increases, spillback is more likely to occur either from the bus stop to the intersection (far-side bus stop), or from the intersection to the bus stop (near-side bus stop). Also, note that typically if a far-side bus stop is used, the delay will be greater than if a near-side bus stop were used.

For undersaturated cases, the average passenger delay does not change with the location of stops. This shows that the algorithm provides good coordination between the intersection and the bus stop.

5.5.2 Sensitivity to assumed bus occupancy



(a) v/c ratio = 0.7, without considering bus stop (b) v/c ratio = 1.1, without considering bus stop

Figure 5.8: Sensitivity to assumed occupancy, where penetration rate=1.0, and bus flow=30 veh/hr. The performance index is per passenger delay that includes both cars and buses. Error bars represent the confidence interval of the average delay saving (confidence level 95%).

In reality, it is usually not feasible to obtain an accurate estimation of bus occupancy. However, a certain occupancy should be assumed for a bus when requesting priority. Here the sensitivity of the algorithm to the assumed bus occupancy is analyzed using the connected TSP algorithm introduced in Section 2.2. Different sets of parameters (v/c ratios, bus arrivals, bus stops, and penetration rates) are tested. For presentation simplicity, Figure 5.8 shows the results of some particular cases. Other results with different inputs are similar.

As is shown in Figure 5.8, as long as the assumed bus occupancy is greater than 6, the influence of assumed bus occupancy on average passenger delay is marginal. Hence, a priori information of bus occupancy is not required by this algorithm. Note that by assigning buses an occupancy larger than 6, they can obtain full priority. This is achieved without sacrificing the performance of the conflicting approach using the connected algorithms (more details are given in Section 5.5.4).

It can also be seen from Figure 5.8 that the average delay per passenger slightly decreases as the assumed bus occupancy increases. This again highlights the fact that this algorithm performs better than the original connected vehicle algorithm (treating buses as ordinary vehicles) in multi-modal scenarios.

5.5.3 Sensitivity to bus dwell time estimation error

The dwell time in reality might be stochastic, due to the variation of passenger behaviors. Hence, we may not have accurate information on the bus dwell time. This subsection evaluates the robustness of the algorithm in scenarios with bus dwell time estimation error. It is assumed that the errors follow a Gaussian distribution with mean 0. The cases with moderate errors (with a standard deviation of 2s), medium errors (with a standard deviation of 6s) and large errors (with a standard deviation of 10s) are tested, respectively. The minimum dwell time is set as 5s. The bus flow is chosen as 30veh/hr. Note that this is a relatively high bus flow. It is expected that the dwell time errors have a larger influence on the system performance in such cases. The results are shown in Figure 5.9.

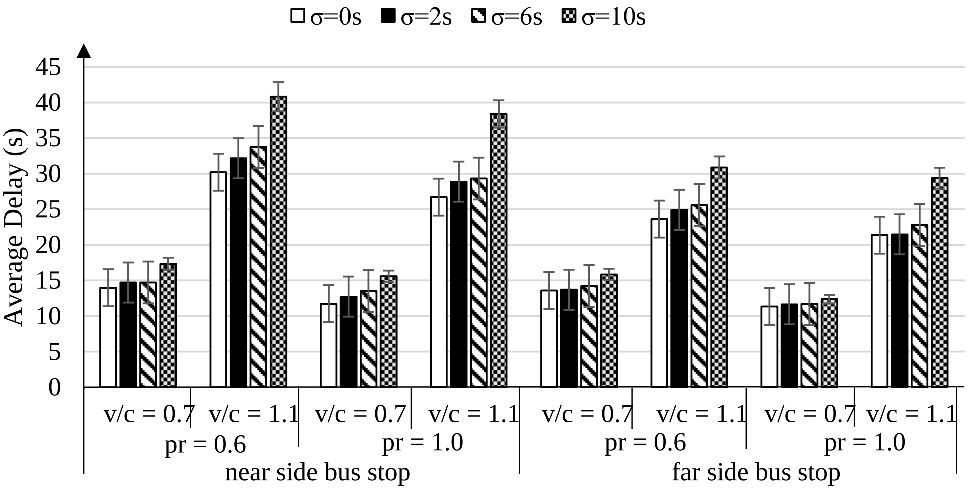


Figure 5.9: Sensitivity to bus dwell time estimation error. The bus flow is assumed to be 30veh/hr. Error bars represent the confidence interval of the average delay saving (confidence level 95%).

It can be seen that the small and medium estimation errors in bus dwell time only have marginal impact on the performance of the algorithm , with an increase of average delay smaller than 10% compared to the scenarios with perfect information. Thus, we can conclude that the algorithm is not sensitive to the estimation error on the bus dwell time for small and medium errors.

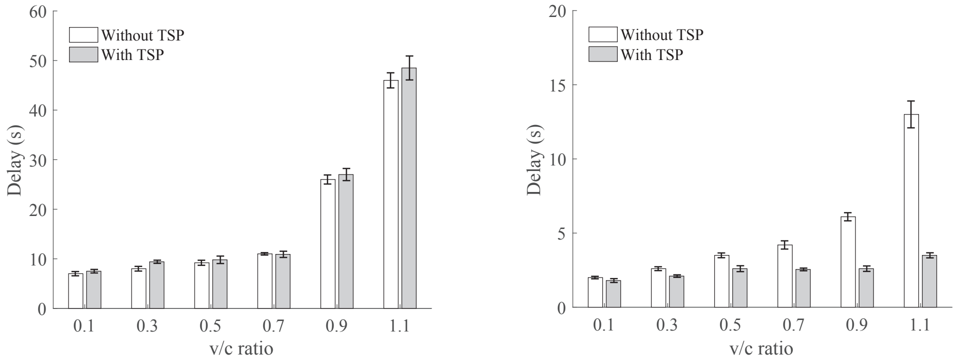
In scenarios with large errors in the estimation of the dwell time, the deterioration of the algorithm is marginal for cases with low demand, but more significant for cases with high demand. Recall that the average dwell time of the buses is 20s, a standard deviation of 10s is already quite large, and is expected to deteriorate the performance of the algorithm. In scenarios with low demand, the errors in the dwell time only affect the algorithm locally (i.e. for the vehicles close to the bus). However, in scenarios with high demand, the congestion can propagate and the errors might affect more vehicles.

5.5.4 Trade-off between bus priority and the conflicting approach delay

This subsection tests whether giving priority to buses sacrifices the performance of the conflicting approach. To do so, analysis is conducted to compare the resulting average vehicle delay of buses, and cars in the conflicting approach with and without TSP in the scenarios without bus stops. Figure 5.10(a) shows the average vehicle delay of cars in the conflicting approach and Figure 5.10(b) shows the average vehicle delay of buses.

It can be seen from both Figure 5.10(a)- 5.10(b) that the algorithm only slightly increases the average delay per car of the conflicting approach while significantly reducing the delay of buses.

Particularly, as is shown in Figure 5.10(a), the car delay increases by less than 5% for v/c ratios lower than 0.9 and less than 10% for v/c ratio of 1.1. For v/c ratios lower than 0.9 where the intersection is undersaturated, vehicles can be discharged from the intersection soon after their arrival, thus giving bus priority will only locally influence several vehicles in the conflicting approach. For v/c ratio of 1.1, it is expected that giving priority to buses will sacrifice the performance of the other approach. However, as there would be no wasted green time, the conflicting approach will be given priority as soon as possible. Therefore, the increased delay is not significant.



(a) Average car delay in the conflicting approach (b) Average bus delay

Figure 5.10: Average vehicle delay of buses and cars in the conflicting approach where penetration rate = 1.0 and bus occupancy = 50 passenger/veh. Error bars represent the confidence interval of the average delay saving (confidence level 95%).

Notice that the simulation is conducted with a bus flow of 30veh/hr, which is usually a high frequency for bus arrivals. If buses arrive less frequently, it is expected that bus priority would have even less impact on the performance of the conflicting approach.

5.6 Conclusions and Future Work

In this chapter, we integrate TSP into the signal control strategy proposed in Chapter 4 considering different bus infrastructure (layouts of bus stops) and bus operations (schedule), and propose three TSP algorithms in a connected vehicle environment: 1) TSP algorithm without considering bus stop; 2) TSP algorithm considering bus stops; and 3) TSP algorithm considering bus stops and schedule.

Various simulations are conducted to evaluate the performance of the proposed algorithm. Simulation results show that the algorithm successfully reduces average delay per passenger compared to a fixed time signal control algorithm with TSP and the basic connected vehicle algorithm without TSP. The value of TSP, the value of considering bus stops, and the value of considering bus schedule are discussed. In scenarios without bus stops, average passenger delay can be reduced by giving bus priority. In scenarios with bus stops, bus stops should be considered when providing bus priority. The findings shed light on the methods and significance of coordinating the TSP algorithms with the bus stops. However, the benefit of considering bus schedule is marginal compared to the benefit of considering the bus stops.

Sensitivity analysis is conducted to show the robustness of the algorithm. It is shown that the algorithm does not require accurate information of bus occupancy, nor dwell time. This is valuable in cases where bus occupancy is not available, or there are estimation errors for the dwell time. It is also shown that the performance of the algorithm is not sensitive to the location of bus stops in undersaturated scenarios. In summary, the algorithm provides a good coordination between bus stops and the intersection.

Future work includes considering a more complex intersection where computational issues should be taken into account (e.g. by formulating an integer programming problem). The proposed scheme can also be generalized to multiple lane scenarios, where the effects of the bus stop can be modelled by the kinematic wave theory (Luthy et al., 2016; Arnet et al., 2015). Another possible extension is to incorporate dedicated bus lanes or pre-signals (Guler and Menendez, 2014a,b; Guler et al., 2016), which provide spatial priority to buses.

PART III

Network-level Traffic Control in a Connected Vehicle Environment

Abstract

In this part, we investigate traffic control of large-scale urban networks in a connected vehicle environment. In particular, we study perimeter control to carefully restrict the incoming flow into a pre-defined region by manipulating the traffic signals at the boundary intersections in order to prevent congestion inside. Perimeter control is a promising control scheme for large-scale urban networks, as it requires the optimization of a small number of intersections. The methodological framework of perimeter control is typically based on the aggregated traffic model of the MFD (or NFD), which relates the trip completion flow to the traffic accumulation inside a network.

Chapter 6 addresses two important research gaps (N1 and N2) in the perimeter control literature established in Section 2.3. First, unlike most of the existing works that focus only on the network level control decisions, we propose the multi-scale perimeter control strategy based on MPC that distributes optimally the aggregated control decision to individual perimeter intersections and accounts for the competing objective functions at both levels. To solve the challenging embedded optimization problem that involves many decision variables, we develop an approximation framework to convert the problem into a linear programming problem, ensuring an efficient solution. Numerical analysis shows that by applying the proposed controller, the protected network can operate around the desired state as expressed by the MFD, while the total delay at the perimeter is minimized as well. Second, we consider a connected vehicle environment, where connected vehicles serve as the only data source. This is a first attempt to develop a network-level traffic control methodology by using the emerging connected vehicles technology. In order to handle the strong noises due to low penetration rates of connected vehicles, we extend the proposed multi-scale perimeter control to a stochastic MPC scheme explicitly considering the stochasticity in traffic state estimation, prediction, and the shape of the MFD. Simulation analysis demonstrates the robustness of the proposed stochastic controller, showing that efficient controllers can indeed be designed with this newly-spread vehicle technology even in the absence of other data collection schemes (e.g. loop detectors).

Chapter 7 extends the multi-scale perimeter control strategy proposed in Chapter 6 to address research gap N3 (established in Section 2.3) by introducing the heterogeneity of vehicles. The heterogeneity is represented by the difference across vehicles in their value-of-time (VOT). This allows us to differentiate across modes (buses are expected to carry more passengers, hence have a higher VOT), across occupancy levels (as with buses, cars with more passengers can increase passenger throughput), across emergency levels (as emergency vehicles typically have a higher VOT), etc. We then employ a pricing scheme to prioritize vehicles with high VOT such that they enter the protected network with less delay through priority lanes at the perimeter intersections where a fee could be charged if necessary. These lanes receive more favorable signal times than the other lanes to ensure a fast service. The proposed approach is tested in a simulated network which resembles the main features of the central area of the city of Zurich, Switzerland. By using the proposed strategy, the traffic accumulation inside the network is still stabilized, and the monetary cost due to delay is significantly reduced at the perimeter. The distribution of the combined cost (including cost due to delay and tolls) is more evenly distributed among VOT groups than that resulting from the multi-scale strategy. The proposed strategy also has the potential to enhance passenger mobility, as vehicles with more passengers typically have a higher VOT.

For reader's convenience, the most important variables are given below.

The variables below are given sets or indices (inputs)

I	set of intersections.
I_{pri}	set of priority intersections (with priority lanes installed)
I_{non}	set of non-priority intersections (without priority lanes installed)
J	vehicle groups, indexed by j
M^i	set of streams of an intersection i, $i \in I$, indexed by m
M^i_{in}	set of streams of an intersection i, $i \in I$
M^i_{out}	set of streams of an intersection i, $i \in I$
P^i	set of phases of intersection i, $i \in I$, indexed by p
k	index of the current cycle
l	index of predicted cycle from the current cycle
V^i	set of all movements (traffic moving in the same direction) at intersection i, indexed by ξ
V^i_{pri}	set of priority movements (with priority lanes installed) at intersection i
V^i_{non}	set of non-priority movements (without priority lanes installed) at intersection i
T^i	set of priority streams at intersection i
N^i	set of normal streams (streams associated with priority movements other than priority streams) at intersection i
Y^i	set of general streams (streams associated with non-priority movements) at intersection i

The variables below are functions or notations (inputs)

$G(\cdot)$	MFD that relates the completion flow to the traffic accumulation in a network
$\phi(\cdot)$	function that relates a stream to a movement, i.e. $\xi = \phi(m)$ represents stream m is associated with movement ξ
$\rho_m(\cdot)$	logistic function calculating the probability for the drivers to choose stream m
$U_m(\cdot)$	utility function associated with stream m

The variables below are given scalars or can be estimated using real data (inputs)

C	cycle length of each intersection
L	prediction horizon (number of cycles)
σ_j	value of time for vehicle group j
π_j	probability that a vehicle belong to the vehicle group j
ζ	parameters to the VOT distribution
v	absolute value of the slope of the left-branch of the MFD (free flow states)
w	absolute value of the slope of the right-branch of the MFD (congested states)

n_{cr}	critical accumulation in the MFD
$\hat{n}_{ab}(k)$	measured traffic accumulation for vehicles in Region a with destination in Region b at cycle k, $a, b = 1, 2$, where 1 represents the center region, and 2 represents the periphery region. It is the initial traffic accumulation for the proposed MPC model
$\hat{x}^i_m(k)$	measured queue length of stream m at intersection i during cycle k, $i \in I$, $m \in M^i$. $\hat{x}^i_m(k)$ is considered as the initial value for the proposed MPC model
$\hat{x}^i_{mj}(k)$	measured queue length of stream m in vehicle group j at intersection i during cycle k, $i \in I$, $m \in M^i$, $j \in J$. $\hat{x}^i_m(k)$ is considered as the initial value for the proposed MPC model
s^i_{mp}	maximum discharging flow of stream m in cycle p at intersection i, $i \in I$, $m \in M^i$, $p \in P^i$
$D_{ab}(k + l \mid k)$	predicted newly generated demand, generated from region a with destination of region b at cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $a, b = 1, 2$
α^i_m	percentage of outflow vehicles on stream m at intersection i, $i \in I$, $m \in M^i$
$g^i_{\max}$	maximum allowed total green time ratio at intersection i, $i \in I$
$g^i_{p,\min}$	minimum allowed green time ratio for phase p at intersection i, $i \in I$, $p \in P^i$
$\lambda^i_\xi(k + l \mid k)$	predicted arrival flow for movement $\xi \in V^i$ at intersection i in cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $j \in J$
$\lambda^i_\xi(k + l \mid k)$	predicted arrival flow for movement $\xi \in V^i$ in vehicle group $j \in J$ at intersection i in cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $j \in J$
$X^i_{m,\max}$	storage capacity for stream i at intersection i

The variables below are associated with decision variables

$n_{ab}(k + l \mid k)$	predicted accumulation of vehicles in Region a with destination in Region b at cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $a, b = 1, 2$
$\beta_{ab}(k + l \mid k)$	predicted controlled flow transferring from Region a to Region b at cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $a, b = 1, 2$
$\mu^i_m(k + l \mid k)$	predicted departure flow of stream m at intersection i in cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $i \in I$, $m \in M^i$
$q^i_m(k + l \mid k)$	predicted arrival flow of stream m at intersection i in cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $i \in I$, $m \in M^i$ (note that $q^i_m(k + l \mid k)$ is treated as given parameters in Chapter 6, but can be associated with decision variables in Chapter 7)
$q^i_{mj}(k + l \mid k)$	predicted arrival flow of stream m in vehicle group j at intersection i in cycle $k + l$ based on information available at cycle k, $0 \leq l \leq L$, $i \in I$, $m \in M^i$, $j \in J$

$x_m^i(k+l\|k)$	predicted queue length of stream m at intersection i in cycle $k+l$ based on information available at cycle k, $0 \le l \le L$, $i \in I$, $m \in M^i$
$x_{mj}^i(k+l\|k)$	predicted queue length of stream m in vehicle group j at intersection i in cycle $k+l$ based on information available at cycle k, $0 \le l \le L$, $i \in I$, $m \in M^i$, $j \in J$
$g_p^i(k+l)$	green time ratio of phase p at intersection i at cycle $k+l$, $0 \le l \le L$, $i \in I$, $p \in P^i$
$z_{mj}^i(k+l\|k)$	predicted probability for vehicle group j to take stream $m \in T^i \cup N^i$ at intersection i in cycle $k+l$ based on information available at cycle k, $0 \le l \le L$, $j \in J$
$\tau^i(k+l\|k)$	predicted toll for using the priority lanes at intersection i in cycle $k+l$ based on information available at cycle k, $0 \le l \le L$, $j \in J$

Chapter 6

Multi-scale Perimeter Control in a Connected Vehicle Environment

This chapter is partially based on the following papers.

- Yang, K., N. Zheng and M. Menendez (2018d) Multi-scale perimeter control approach in a connected-vehicle environment, *Transportation Research Part C: Emerging Technologies*, **94**, 32 – 49.

- Yang, K., N. Zheng and M. Menendez (2017b) Multi-scale perimeter control approach in a connected-vehicle environment, paper presented at the *22nd International Symposium on Transportation and Traffic Theory (ISTTT)*, vol. 23, 101–120.

6.1 Objectives and Contributions

Perimeter control is an efficient and effective traffic control scheme, which has been receiving enormous research attention, ranging from the control of single/two regions (Haddad and Geroliminis, 2012; Keyvan-Ekbatani et al., 2012; Geroliminis et al., 2013; Haddad et al., 2013; Aboudolas and Geroliminis, 2013; Keyvan-Ekbatani et al., 2015; Haddad, 2015) to multiple regions (Aboudolas and Geroliminis, 2013; Ramezani et al., 2015; Kouvelas et al., 2016). However, although some works attempt to consider the local intersections (Keyvan-Ekbatani et al., 2012; Geroliminis et al., 2013; Haddad et al., 2013; Aboudolas and Geroliminis, 2013; Hajiahmadi et al., 2015; Haddad, 2017), there are no methodological works that 1) balance the competing objectives of the network and local levels, and 2) investigate how the aggregated control decisions can be translated to the decision variables for the local control problem. Moreover, existing perimeter control strategies do not account for the promising connected vehicle technology. Connected vehicles provide detailed and diverse information, capable of performing more intricate and anticipative control. However, it is challenging to handle the uncertainties due to the limited penetration rates during the transition period. Some existing works rely on robust control schemes and may yield a conservative control decision. Moreover, the measurement and prediction noises are typically ignored, which are crucial for the deployment of the connected vehicles.

In this chapter, we propose control strategies for large-scale urban networks utilizing the information provided by connected vehicles. In particular, we aim to address research gaps N1 and N2 as established in Section 2.3. We will integrate the optimal control of perimeter intersections (i.e. to minimize local delay) into the perimeter control scheme (i.e. to optimize traffic performance at the network level). We will also handle the challenging issues brought by the low penetration rates of connected vehicles. The contributions of this chapter are three-fold.

1) We propose a multi-scale control strategy that optimizes traffic performance at both the network and the local level (i.e. perimeter intersections). This is the first theoretical perimeter control framework that considers the local perimeter intersections.
2) We apply the proposed controller in a connected-vehicle environment, for which the robustness of the control is enhanced. This is, to the best of our knowledge, the first attempt to develop a network-level methodology using such technology.

3) We develop a stochastic model predictive controller to handle the uncertainties resulting from the limited penetration rates of the connected vehicle technology.

The rest of this chapter is organized as follows. Section 6.2 introduces the general framework of a dynamic traffic system model and the design of the proposed control strategy, i.e. a multi-scale perimeter flow controller. Section 6.3 describes the simulations settings. Section 6.4 analyzes the performance of the proposed control strategy and carries out a comparative analysis to the classical control strategies. Section 6.4 extends the proposed multi-scale controller to a stochastic controller that takes into account system uncertainties brought by the limited information from connected vehicles. Section 6.5 further investigates the robustness of the proposed control strategy and demonstrates the importance of the multi-scale treatment. Section 6.6 concludes this chapter and provides future research directions.

6.2 General Methodology

6.2.1 Problem presentation

Consider a typical urban city of single-center structure, with two regions. Let us denote the center region as “1” and the outside region as “2”, where the perimeter between the two regions is the perimeter where we apply the control via traffic signals. The two levels of control under consideration are (i) the city center network which attracts large demand (i.e. network level); and (ii) the intersections at the perimeter, where the perimeter control is implemented (i.e. local level). At the network level, the dynamics of the system are described by an MFD $O = G(n)$ where the aggregated traffic completion flow O (outflow, vehs per time unit) is a function $G(\cdot)$ of the traffic accumulation n (number of vehicles, vehs). The MFD is assumed to be known for the given network.

The change in traffic accumulations of the region, reflecting the network-level traffic state, can be represented by the evolution of the accumulation n, which is captured by mass conservation equations without the need for detailed traffic information (such as routing at link level) within the network. Eq.(6.1)-Eq.(6.2) show the time-discretized dynamics at the network level. The cycle length is denoted as C [hr].

$$n_{11}(k+1) = n_{11}(k) + D_{11}(k)C + \beta_{21}(k)C - \frac{n_{11}(k)}{n_{11}(k) + n_{12}(k)} G(n_{11}(k) + n_{12}(k))C \tag{6.1}$$

$$n_{12}(k+1) = n_{12}(k) + D_{12}(k)C - \beta_{12}(k)C \tag{6.2}$$

In the equations, $n_{ab}(k)$ [veh] represents the traffic accumulation in region a with destination in region b at cycle k (independently of the origin of the individual trips). Hence, $n_{11}(k)$ denotes the traffic accumulation in the city center with a destination in the city center; $n_{12}(k)$ is the traffic accumulation in the city center which will end trips in the outside region; likewise, $n_{21}(k)$ is the traffic accumulation in the outside region that will end their trips in the city center. $D_{ab}(k)$ [veh/hr] denotes the newly generated demand from region a, with a destination to region b at cycle k. Therefore $D_{11}(k)$ is the internal demand having destinations inside the city center; $D_{21}(k)$ is the demand coming from outside while ending their trips inside of the center center; $D_{12}(k)$ is the demand generated in the center region with destination to the outside region; and $D_{22}(k)$ is not considered here. $\beta_{21}(k)$ and $\beta_{12}(k)$ [veh/hr] are the controlled flow, entering and leaving the center region, respectively (these are important variables to the proposed control strategy). The last term in Eq.(6.1) represents the uncontrolled trip completion flow inside the city center. Since the MFD function G is known and the demand $D_{ab}(k)$ can be learned without requiring detailed Origin-Destination (OD) information, traffic accumulation Eq.(6.1)-Eq.(6.2) can be monitored in real time. Uncertainties in the demand $D_{ab}(k)$ will be discussed in Section 6.5. In Eq.(6.1) and Eq.(6.2), it is assumed that the network is homogeneous, which is also the requirement of a well-defined MFD. If the network is not homogeneous, it can be divided into a few homogeneous sub-networks using clustering techniques (Ji and Geroliminis, 2012).

Initially, we assume that the network is equipped with a sufficient number of connected vehicles (i.e. high penetration rate) so that we can properly measure the states (i.e. traffic accumulation and queue lengths) of the network. This assumption is relaxed in Section 6.5, and its impact on the traffic performance will be discussed later. Theoretically, the method proposed in this chapter can also use the data from other sources such as video cameras, probe vehicles, and historical data. However, in practice, the method is expected to work with better accuracy with connected vehicle data, as we assume real-time information is available on the dynamic origin-destination (OD) at the network level and the arrival flow at the intersection level (a priori knowledge on which intersection each vehicle uses to enter the city center).

For the local level, we track the dynamics of the queues at the perimeter intersections. Denote the set of the intersections as I, the set of streams for each intersection i as M^i, and the phases for each intersection as P^i. We assume a constant cycle length for all intersections C [hr]. This assumption reduces complexity in formulating the control problem, allowing an extensive numerical analysis with moderate computational burden. Alternatively, cycle length and other signal settings can be dynamically formulated, and easily integrated into the proposed model.

We refer to the departure flow for each direction m of intersection i at cycle k as $\mu^i_m(k)$ [veh/hr]. $\mu^i_m(k)$ is estimated based on the arrival flow $q^i_m(k)$ [veh/hr], the queue length $x^i_m(k)$ [veh], the discharging flow s^i_{mp} [veh/hr], the cycle length C [hr], and the allocated green time ratio $g^i_p(k)$ [-] (Liu et al., 2008; Zhao et al., 2011; Li et al., 2013; Tan et al., 2017). For each intersection i, the maximum discharging flow rate s^i_{mp} is a time-invariant parameter which takes one of the following two values: 1) the saturation flow rate, if green signal is given to stream m in phase p at intersection i; 2) 0, otherwise. s^i_{mp} only represents the ability of stream m at intersection i to discharge vehicles in phase p. It is independent of the arrival flow of vehicles. The actual departure rate $\mu^i_m(k)$ is, on the other hand, bounded by the demand at this stream. Note that we assume that the arrival flow $q^i_m(k)$ [veh/hr], the queue length $x^i_m(k)$ [veh], the maximum discharging flow s^i_{mp} [veh/hr], and the allocated green time ratio $g^i_p(k)$ can have different values among intersections. The evolution of the queue length $x^i_m(k+1)$ can be captured by the mass conservation law, as well, i.e. Eq.(7.4).

$$x^i_m(k+1) = x^i_m(k) + q^i_m(k)C - \mu^i_m(k)C,\ m \in M^i,\ i \in I \tag{6.3}$$

For the proposed framework at the local level, we take into account not only the directions for accommodating flows $\beta_{21}(k)$ and $\beta_{12}(k)$, but also the general directions of a typical signalized intersection.

6.2.2 An MPC approach for multi-scale control

An MPC approach is proposed to the multi-scale controller which integrates the network level perimeter control and the local level intersection control through detailed signal optimization. The proposed MPC relies on a dynamic traffic model that couples both the network and the intersections. It calculates the green time ratios at each intersection (i.e. decision variables) to optimize the total travel cost in the current signal cycle, while taking into consideration the costs in the future cycles. This is achieved by solving an optimization problem with a finite-time horizon (N cycles), where we obtain the evolution of traffic accumulations and the predicted optimal green ratios for each cycle, but only execute the green ratios for the current cycle. These features make the proposed MPC approach fundamentally different, while more complicated to solve than the ones in previous studies.

There are two types of inputs to the MPC: the current traffic states (i.e. traffic accumulation and queue lengths at each intersection), and the traffic demands. The traffic accumulation and

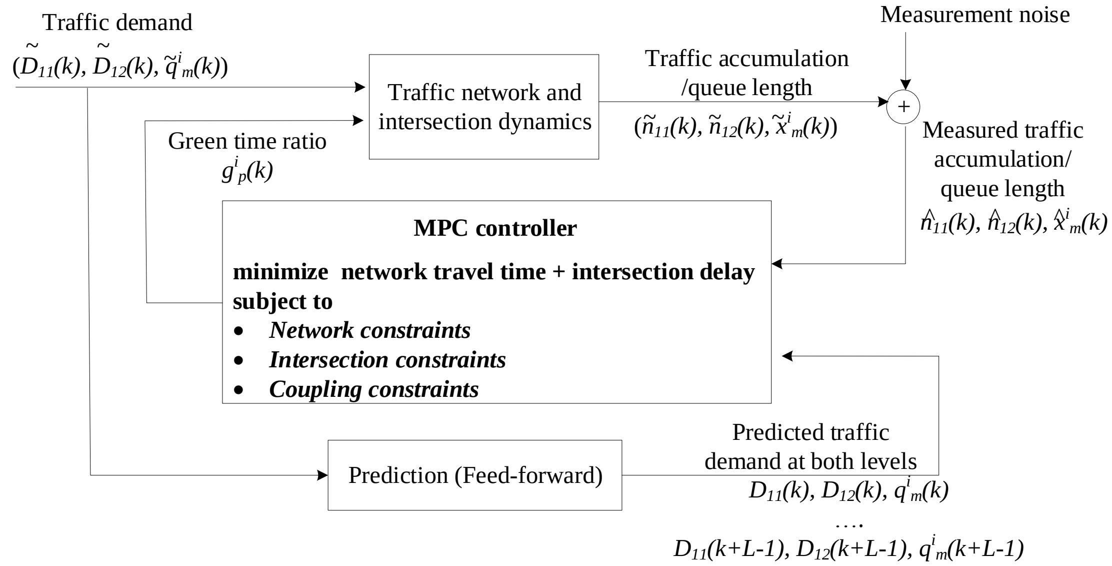

Traffic demand
($\tilde{D}_{11}(k)$, $\tilde{D}_{12}(k)$, $\tilde{q}^i_m(k)$)
Traffic network and intersection dynamics
Green time ratio $g^i_p(k)$
Traffic accumulation /queue length
($\tilde{n}_{11}(k)$, $\tilde{n}_{12}(k)$, $\tilde{x}^i_m(k)$)
Measurement noise
+
Measured traffic accumulation/ queue length
$\hat{n}_{11}(k)$, $\hat{n}_{12}(k)$, $\hat{x}^i_m(k)$
MPC controller
minimize network travel time + intersection delay
subject to
Network constraints
Intersection constraints
Coupling constraints
Prediction (Feed-forward)
Predicted traffic demand at both levels
$D_{11}(k)$, $D_{12}(k)$, $q^i_m(k)$
....
$D_{11}(k+L-1)$, $D_{12}(k+L-1)$, $q^i_m(k+L-1)$

Figure 6.1: Control diagram for MPC.

queue lengths are measured using information communicated from the connected vehicles. The traffic demands and the region-level ODs can be predicted in real time through the CVs alone or combined with historical data.

The control diagram in Figure 6.1 illustrates graphically the control process. Note that we separate the real system dynamics (at both network and local level) and the MPC prediction models in Figure 6.1. The MPC prediction model can be different from the system dynamics. $g^i_p(k)$ are the decision variables (i.e., output of the MPC controller). $\tilde{D}_{ab}(k)$ and $\tilde{q}^i_m(k)$ represent the current real demand at the network and local level at cycle k, respectively; $D_{ab}(k+l)$ and $q^i_m(k+l)$, $0 \le l \le L-1$ represent the predicted demand at the network level and the intersection level, respectively. $\tilde{n}_{ab}(k)$ and $\tilde{x}^i_m(k)$ represent the current real traffic accumulation and queue length in the traffic system, respectively; $\hat{n}_{ab}(k)$ and $\hat{x}^i_m(k)$ represent the measured traffic accumulation and queue length at cycle k, respectively, both are considered to be obtained through connected vehicles. Notice that no measurement noise nor prediction error is incorporated at this stage, i.e. $D_{ab}(k) = \tilde{D}_{ab}(k)$, $q^i_m(k) = \tilde{q}^i_m(k)$, $\hat{n}_{ab}(k) = \tilde{n}_{ab}(k)$, $\hat{x}^i_m(k) = \tilde{x}^i_m(k)$. This assumption will be relaxed later where additional treatment is integrated.

At each cycle k, we solve the MPC optimization model over a moving time horizon of the next L cycles. The detailed formulation of the optimization problem is displayed in Eq.(6.4)-(6.15). The inputs to the model are (a) the predicted demand in the next L cycles at both the network and local level, (b) the measured traffic accumulation, and (c) the queue length. The output of the model are the optimal green time ratios for each phase at each intersection during the current cycle, which serve as a feedback to the traffic system.

The objective function Eq.(6.4) minimizes the total travel cost J_D of the whole system, including the total delay incurred at the perimeter intersections and the total travel time within the controlled network subject to constraints Eq.(6.5)-Eq.(6.15). The first two terms in Eq.(6.4), multiplied by the cycle length, represent the network travel time in the city network during a given cycle; the third term (also multiplied by the cycle length) represents the total travel delay at the perimeter intersections also during a given cycle. This is computed for all future L cycles (the horizon). Notice that the last term excludes the outflow streams from the network M^i_{out} (explanation on the exclusion is given below). In Eq.(6.4), the two objective criteria (travel time in the city network and delay at the perimeter intersections) have the same weights. However, the weights can be adjusted for different policy-oriented purposes.

$$\min J_D = C \sum_{l=1}^{L} \Big(n_{11}(k+l|k) + n_{12}(k+l|k) + \sum_{i \in I} \sum_{m \in M \backslash M^i_{\text{out}}} x^i_m(k+l|k) \Big) \tag{6.4}$$

$$\text{s.t. } n_{11}(k+l+1|k) = n_{11}(k+l|k) + D_{11}(k+l|k)C + \beta_{21}(k+l|k)C$$
$$- \frac{n_{11}(k+l|k)}{n_{11}(k+l|k) + n_{12}(k+l|k)} G\Big(n_{11}(k+l|k) + n_{12}(k+l|k)\Big)C, \ \forall 0 \le l \le L-1 \tag{6.5}$$

$$n_{12}(k+l+1|k) = n_{12}(k+l|k) + D_{12}(k+l|k)C - \beta_{12}(k+l|k)C, \ \forall 0 \le l \le L-1 \tag{6.6}$$

$$\beta_{21}(k+l|k) = \sum_{i \in I} \sum_{m \in M^i_{\text{in}}} \mu^i_m(k+l|k), \ \forall 0 \le l \le L-1 \tag{6.7}$$

$$\beta_{12}(k+l|k) = \sum_{i \in I} \sum_{m \in M^i_{\text{out}}} \mu^i_m(k+l|k), \ \forall 0 \le l \le L-1 \tag{6.8}$$

$$\mu^i_m(k+l|k) = \min\Big\{ \frac{\alpha^i_m}{|I|} \frac{n_{12}(k+l|k)}{n_{11}(k+l|k) + n_{12}(k+l|k)} G\Big(n_{11}(k+l|k) + n_{12}(k+l|k)\Big),$$
$$\sum_{p \in P^i} s^i_{mp} g^i_p(k+l|k) \Big\}, \ \forall m \in M^i_{\text{out}}, \ \forall i \in I, \ \forall 0 \le l \le L \tag{6.9}$$

$$\mu^i_m(k+l|k) = \min\{ x^i_m(k+l|k)/C + q^i_m(k+l|k), \sum_{p \in P^i} s^i_{mp} g^i_p(k+l|k) \},$$
$$\forall m \in M^i \backslash M^i_{\text{out}}, \ \forall i \in I, \ \forall 0 \le l \le L-1 \tag{6.10}$$

$$x^i_m(k+l+1|k) = x^i_m(k+l|k) + q^i_m(k+l|k)C - \mu^i_m(k+l|k)C,$$
$$\forall m \in M^i \backslash M^i_{\text{out}}, \ \forall i \in I, \ \forall 0 \le l \le L-1 \tag{6.11}$$

$$\sum_{p \in P} g^i_p(k+l|k) \le g^i_{\max}, \ \forall i \in I, \ 0 \le l \le L-1 \tag{6.12}$$

$$g^i_p(k+l|k) \ge g^i_{p,\min}, \ \forall p \in P^i, \forall i \in I, \forall 0 \le l \le L-1 \tag{6.13}$$

$$n_{1b}(k|k) = \hat{n}_{1b}(k), \ b = 1, 2 \tag{6.14}$$

$$x^i_m(k|k) = \hat{x}^i_m(k), \ \forall m \in M^i \backslash M^i_{\text{out}}, \ \forall i \in I \tag{6.15}$$

Constraints Eq.(6.5) and Eq.(6.6) represent the mass conservation of vehicle accumulations at the network level (they are modelled after Eq.(6.1) and Eq.(6.2)). Eq.(6.7) and Eq.(6.8) are "coupling" constraints which link the network and the local level. Here M^i_{in} and M^i_{out} are two disjoint subsets of set M, which represent the set of inflow streams to the network and the set of outflow streams from the network at intersection i, respectively. Eq.(6.7) reflects that the vehicles departing from streams in M^i_{in} are the inflow entering into region 1. Likewise, Eq. (6.8) holds because vehicles departing from streams in M^i_{out} are the outflow leaving from region 1. Eq.(6.9) and Eq.(6.10) ensure that the departure flow of each stream is taken as the minimum of the demand and the capacity determined by the allocated green time. The first term in Eq.(6.9) represents the "demand" leaving the network. It is the flow arriving at the outflow streams M^i_{out} of each intersection, the sum of which is determined by the outflow of the network. In the prediction model, we assume that the outflow of the network is evenly distributed across all intersections. For each intersection, there might exist multiple outflow streams accommodating the flows that leave the network. α^i_m represents the percentage of outflow vehicles on each of them. The value of α^i_m can be dynamically updated by online estimation (e.g. a Kalman filter) from CV data while integrating historical information. Note that obtaining α^i_m from traditional data sources (such as fixed loop detectors) may require more effort while resulting in less accuracy. The first term in Eq.(6.10) is the total demand for each stream at each intersection, determined jointly by the queue length and the arrival flows. Here, $q^i_m(k+l|k)$ represents the newly arriving flow for stream m at cycle $k+l$ at intersection i, predicted at cycle k. The second terms in both Eq.(6.9) and Eq.(6.10) represent the capacity, determined by the allocated green time ratios. Eq.(6.11) describes the queue dynamics at the local level using basic conservation law. It does not include the outflow streams of the network, as 1) the total arrival to these streams is decided by the outflow of the network; 2) in the objective function Eq.(6.4), these outflow vehicles are already included in $n_{12}(t)$. Eq.(6.12) and Eq.(6.13) are the physical constraints imposed on the green time ratio. $g^i_{\max}$ is the maximum allowed total green ratio across all phases, defined as $1-\eta/C$ where η is the total lost time. $g^i_{p,\min}$ is the minimum duration of the green signal sufficient to discharge a given number of vehicles (e.g. 2 vehicles). Both $g^i_{\max}$ and $g^i_{p,\min}$ are given parameters, which can be determined by the configurations of the intersections. Constraints Eq.(6.14) and Eq.(6.15) define the initial traffic accumulations, $n_{1b,init}$, and the initial queue lengths, $x^i_{m,init}$.

As $n_{11}(k+l|k)$, $n_{12}(k+l|k)$, $\beta_{12}(k+l|k)$, $\beta_{21}(k+l|k)$, $\mu^i_m(k+l|k)$, $x^i_m(k+l|k)$ are all functions of $g^i_p(k+l|k)$, the only decision variables are the green time ratios $g^i_p(k+l|k)$ for $p \in P^i$, $i \in I$, $\forall 0 \le l \le L$.

It can been seen that the optimization problem defined by Eq.(6.4)-(6.15) is highly complex. To reduce the non-convexity and facilitate the solution searching, constraints Eq.(6.5) and Eq.(6.9) are converted into a piecewise linear function. Details are presented in Section 6.2.3.

6.2.3 An approximation framework

In this subsection, we reformulate the optimization problem Eq.(6.4)-Eq.(6.15) through linearization techniques. For presentation simplicity, we neglect the cycle index k in this subsection.

A triangular-shape MFD (used also in Haddad and Geroliminis, 2012; Haddad et al., 2013) is employed in the prediction model. Note that for the numerical simulation, an empirical MFD (with scatters and noises) is utilized. This makes the prediction model different from the simulation model (used later in Section 4, Section 5.2, and Section 5.3). This is an important treatment for the MPC-based approach. Also note that choosing alternative MFD shapes for the prediction model is straightforward in our framework. Our results show, however, that the triangular MFD represents a good compromise between accuracy and efficiency, and works sufficiently well.

The MFD is represented as

$$G(n) = \min\{vn, (w+v)n_{cr} - wn\} \tag{6.16}$$

where $v,\ w > 0$ are the absolute value of the slopes of the left and the right branch of the MFD, respectively; n_{cr} is the critical vehicle accumulation.

The proposed MPC with its current form Eq.(6.4)-Eq.(6.15) is highly nonlinear and nonconvex due to the following terms in Eq.(6.5) and Eq.(6.9), respectively.

$$O_{1b} = \frac{n_{1b}}{n_{11}+n_{12}} G(n_{11}+n_{12}), b = 1,2 \tag{6.17}$$

Hence, we rewrite $O_{1b}, b = 1,2$ as follows.

$$O_{1b} = \min\left\{vn_{1b}, \frac{(w+v)n_{cr}n_{1b}}{n_{11}+n_{12}} - wn_{1b}\right\},\ b = 1,2 \tag{6.18}$$

It can be seen from Eq.(6.18) that the only nonlinear terms are $\frac{n_{1b}}{n_{11}+n_{12}}, b = 1,2$. To solve the problem efficiently, we apply linearization technique to this term with a first-order Taylor polynomial approximation. Denote the initial accumulation as $(\hat{n}_{11}, \hat{n}_{12})$. Specifically, we linearize the second term of O_{11} at $(\hat{n}_{11}, n_{cr} - \hat{n}_{11})$ and the second term of O_{12} at $(n_{cr} - \hat{n}_{12}, \hat{n}_{12})$.

With simple calculations, we can then rewrite O_{11} and O_{12} as follows.

$$O_{1b} \approx \bar{O}_{1b} = \min\left\{vn_{1b}, \gamma_{a0} + \gamma_{a1}n_{11} + \gamma_{a2}n_{12}\right\},\ a = 1,2 \tag{6.19}$$

where

$$\gamma_{10} = (w+v)\hat{n}_{11},\ \gamma_{11} = v - (w+v)\frac{\hat{n}_{11}}{n_{cr}},\ \gamma_{12} = -(w+v)\frac{\hat{n}_{11}}{n_{cr}} \tag{6.20}$$

$$\gamma_{20} = (w+v)\hat{n}_{12},\ \gamma_{21} = -(w+v)\frac{\hat{n}_{12}}{n_{cr}},\ \gamma_{22} = v - (w+v)\frac{\hat{n}_{12}}{n_{cr}} \tag{6.21}$$

Notice that we linearize O_{11} and O_{12} at $(\hat{n}_{11}, n_{cr} - \hat{n}_{11})$ and $(n_{cr} - \hat{n}_{12}, \hat{n}_{12})$ instead of the original initial accumulation $(\hat{n}_{11}, \hat{n}_{12})$. The reason is summarized in Lemma 1.

Lemma 1. By linearizing O_{1b} as $\bar{O}_{1b}$ (Eq.(6.19), we obtain

$$\bar{O}_{1b} = O_{1b},\ \text{if}\ n_{11}(k) + n_{12}(k) \le n_{cr},\ b = 1,2 \tag{6.22}$$

The proof of the lemma is trivial. We omit here for brevity. Lemma 1 indicates that the linear approximation does not change O_{11} and O_{12} in the free flow state.

The linearization of O_{11} and O_{12} can greatly accelerate the solution procedure of the proposed MPC model. Constraints Eq.(6.5) and Eq.(6.9) can be reformulated as follows.

$$\begin{aligned} n_{11}(k+1) = n_{11}(k) + D_{11}(k)C + \beta_{21}(k)C - \min\{vCn_{11}(k), \\ \gamma_{10}C + \gamma_{11}Cn_{11}(k) + \gamma_{12}Cn_{12}(k)\},\ \forall 0 \le l \le L-1 \end{aligned} \tag{6.23}$$

$$\begin{aligned} \mu^i_m(k) = \min\left\{\frac{\alpha^i_m}{|I|}vn_{12}(k), \frac{\alpha^i_m}{|I|}\big(\gamma_{20} + \gamma_{21}n_{11}(k) + \gamma_{22}n_{12}(k)\big), \right. \\ \left. \sum_{p\in P^i} s^i_{mp}g^i_p(k+l|k)\right\},\ \forall i \in I,\ \forall 0 \le l \le L \end{aligned} \tag{6.24}$$

By replacing Eq.(6.5) and Eq.(6.9) with Eq.(6.23) and Eq.(6.24), the highly nonlinear and nonconvex MPC can be reformulated into a piecewise linear optimization problem. Though still non-convex, the problem can be solved using the existing algorithms (for example the branch and cut algorithm in Keha et al. (2006)) by introducing binary decision variables which represent whether these constraints are chosen. Since there will not be too many binary decision variables, the problem can be solved in a reasonable time (30s for our simulated cases).

We notice that the only constraint which still makes the problem non-convex is Eq.(6.10) for $m \in M^i_{in}$, as the sign before $\beta_{21}(k+l|k)$ in Eq.(6.23) is positive. To accelerate the procedure, we approximate $\mu_m(k+l|k) = \sum_{p\in P^i} s^p_m g^p(k+l)$, for $m \in M^i_{in}$. Two remarks should be made regarding for this approximation.

R1) This approximation might overestimate $\mu_m(k+l|k)$ if $x^i_m(k+l|k)/C + q^i_m(k+l|k) < \sum_{p\in P^i} s^p_m g^p(k+l)$, i.e. when demand to enter the network is low. However, in scenarios where there are always high demand, e.g. during peak hour, the approximated model is practically equivalent to the original model. Also note that by overestimating the network inflows, the obtained control policy can be more conservative at the network level.

R2) If the signal timing is very flexible (i.e. $g^i_{p,\min}$ is sufficiently small and each green phase discharges only one stream), $g^i_p(k+l|k)$ can be set as small as possible. In such cases, the approximated model is also equivalent to the original model.

With this approximation, we replace Eq.(11) with two constraints Eq.(6.25) and Eq.(6.26).

$$\mu^i_m(k+l|k) = \min\{x^i_m(k+l|k)/C + q^i_m(k+l|k), \sum_{p\in P^i} s^i_{mp}g^i_p(k+l|k)\},$$

$$\forall m \in M^i \backslash M^i_{\text{out}} \backslash M^i_{\text{in}},\ \forall i \in I,\ \forall 0 \leq l \leq L-1 \tag{6.25}$$

$$\mu^i_m(k+l|k) = \sum_{p \in P^i} s^i_{mp} g^i_p(k+l|k),\ \forall m \in M^i_{\text{in}},$$

$$\forall i \in I,\ \forall 0 \leq l \leq L-1 \tag{6.26}$$

Eq.(12) is further replaced with Eq.(6.27) to guarantee that $x^i_m(k+l|k) \geq 0$.

$$\begin{aligned} x^i_m(k+l+1|k) = &\max\{0, x^i_m(k+l|k) + q^i_m(k+l|k)C - \mu^i_m(k+l|k)C\}, \\ &\forall m \in M^i \backslash M^i_{\text{out}},\ \forall i \in I,\ \forall 0 \leq l \leq L-1 \end{aligned} \tag{6.27}$$

Having been transformed to its current form, the approximated model becomes a convex piecewise linear model. Both the traffic accumulations n and the queue length x can be represented as a sum of max functions of the green ratios g. This convex piecewise linear model can be equivalently reformulated into a linear programming problem by relaxing the min and max operator with the corresponding inequalities. The model can be solved efficiently with the classic simplex method (Nelder and Mead, 1965) or an interior-point method (Kojima et al., 1989) in a commercial solver, such as Cplex. The average running time for solving the multi-scale problem in each cycle is 0.5s on a PC with one CPU core.

6.3 Simulation Settings

We simulate a typical morning-peak period. The studied region is shown in Figure 6.2(a), where a similar type of perimeter control is implemented (Ortigosa et al., 2014; Ambuhl et al., 2018). The MFD of this network are designed to mimic the aggregated traffic features of the city center of Zurich, Switzerland, shown in Figure 6.2(b) (Ambühl et al., 2016). Note that the simulation is macroscopic. The system dynamics are realized using the MFD-based aggregated dynamics, rather than a micro-simulation.

The total demand for entering the network during a 1.5 hour period is shown in Figure 6.2(c). Note that this demand profile generates a large amount of traffic accumulation for testing the controllers. The resulting MFD without the application of any type of perimeter control reaches oversaturated states (i.e. the network gets congested).

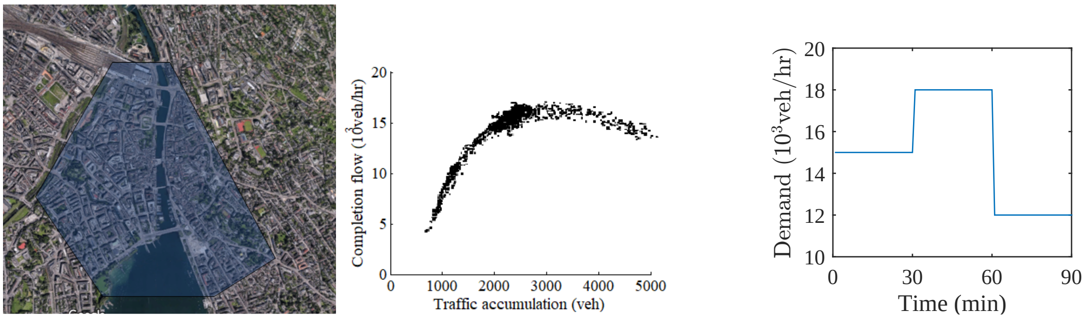
This network consists of 20 intersections at the perimeter. For the prediction model, we use a triangular MFD with $v = 5\text{hr}^{-1}$, $\text{w} = 2.5\text{hr}^{-1}$ and $n_{cr} = 3000\text{veh}$. The maximum flow of the network is 15000veh/hr. The initial traffic accumulation in the onset of the simulation is assumed to be 2000veh, below the critical accumulation. Each intersection is assumed to have the phases and streams shown in Figure 6.2(d) and Figure 6.2(e)[12]. The saturation flow is assumed to be different across intersections. The cycle length C is 1 min. The maximum allowed green ratio for each intersection $g^i_{\max}$ is 0.9, and the minimum green ratio $g^i_{p,\min}$for each phase is 0.1 (which means 6s). There is no offset between intersections.

6.4 Case Study on the Multi-scale Controller

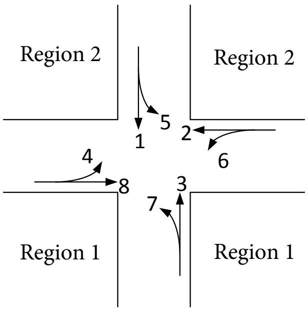
To evaluate the proposed strategy, we compare three types of controllers in the following case study: 1) the proposed multi-scale MPC-based controller, 2) a classical controller developed in the traffic control literature using Proportional, Integral and Differentiation (PID) method (Åström and Hägglund, 2006), and 3) a bang-bang type of controller which is also one of the most classical benchmarks in the control theory literature (Bellman et al., 1956). For both the

---

[12] Right turn is not included in this case study, as it will complicate our signal plan with limited added value. It is not difficult for our model to handle the cases with either protected or unprotected right turns.



(a) Studied region (b) MFD of the studied region (c) Network level demand



Streams	Phases
1,3	1
5,7	2
2,4	3
6,8	4

(d) Intersection streams (e) Intersection phases

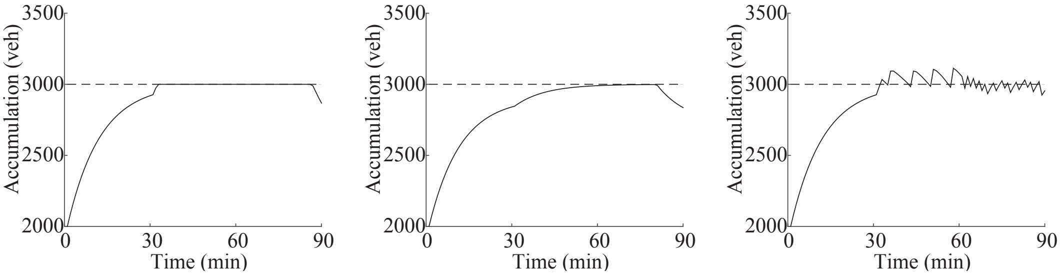
Figure 6.2: Simulations settings. At the intersection level (d and e), streams 1 and 6 are the inflow to the city network, and streams 3 and 7 are the outflow from the city network.

PID controller and the bang-bang controller, we first determine the total inflow and outflow of the network. These flows are then distributed to each intersection based on the local queue information. A system with perfect information is considered in Section 6.4.1; Systems with moderate noises and large noises are considered in Section 6.4.2 and Section 6.4.3, respectively. The numerical simulation is conducted with Python, with integration of a Cplex C++ API for solving the model. The average solution time for one cycle is less than 1s. The moving horizon for the proposed multi-scale MPC is chosen as 20 cycles.

6.4.1 Performance of the multi-scale controller

Let us define two performance measures, the traffic accumulation in the center and the queue length at the perimeter intersections. The queue values are normalized and dimensionless (rescaled in relation to the maximum and minimum queue values of the whole network), given that the intersections have different physical configurations (e.g. the discharging flow, arrival flow, etc). The resulting time series of the two measures are displayed in Figure 6.3, Figure 6.4 and Figure 6.5 for the three controllers.

As shown in Figure 6.3, all three controllers succeed in effectively operating the city center network, by maintaining the accumulation around the critical value. It appears that the bang-bang controller generates oscillations in the system dynamics. The classical bang-bang controller follows an all-or-nothing rule, which is as expected to cause aggressive control actions towards the change of the system states. The oscillations exist even in scenarios without noises in the control system (the type of noises considered later will be detailed in Section 4.3). The PID controller, regulates the system in a relatively less aggressive manner, and maintains an the accumulation slightly smaller than the critical accumulation. This is because the inflow and the outflow of the network, given by the PID controller, may not satisfy the constraints at the intersections, as the local level dynamics and signal settings are not explicitly considered. The



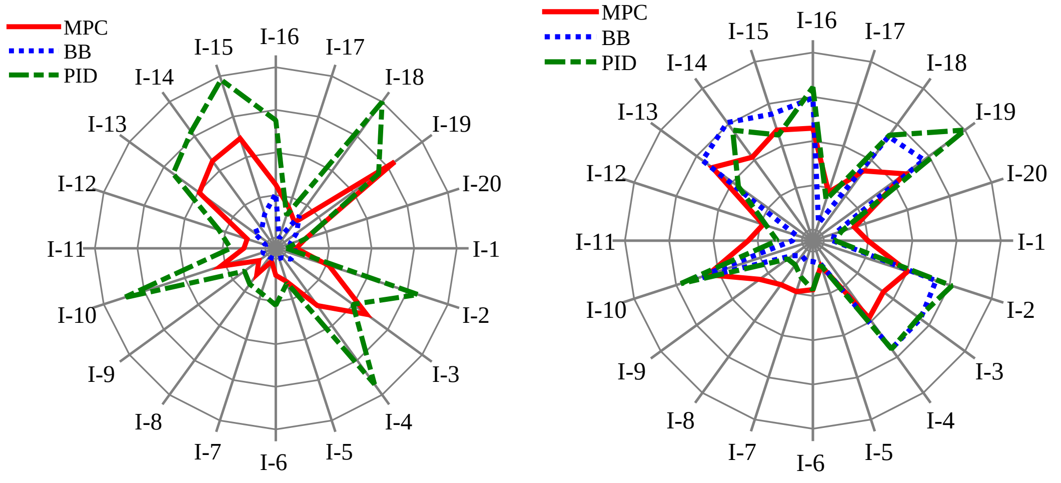
(a) Multi-scale MPC (b) PID controller (c) Bang-bang controller

Figure 6.3: Accumulation comparison between the MPC-based controller, the classical PID controller and the bang-bang controller in scenarios without noises.

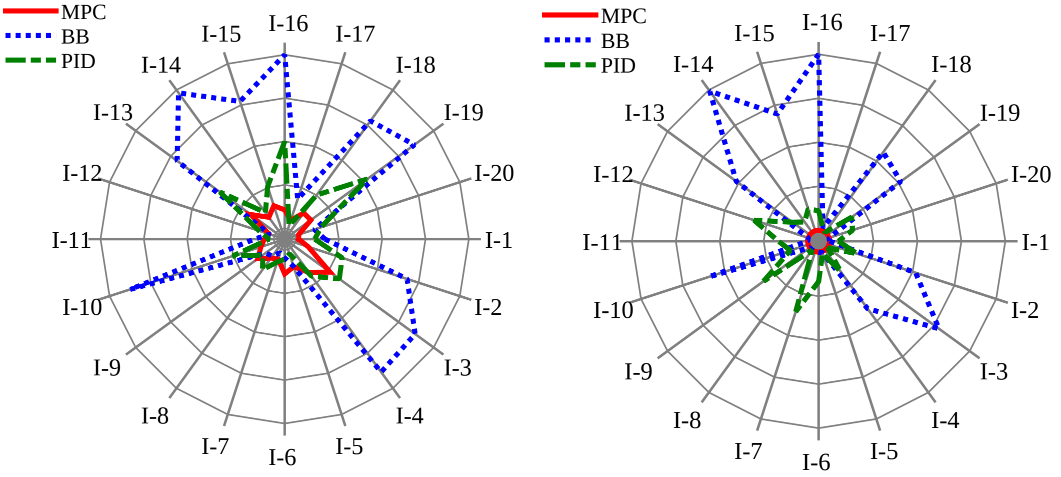
proposed MPC controller appears to produce promising results, exhibiting high stability and efficiency at the network level. First, in comparison with the bang-bang controller, the MPC produces oscillation-free system dynamics. Second, in comparison with the PID controller, the MPC assures a higher network utilization, accommodating traffic at the critical accumulation level for a much longer period of time (roughly 30 minutes). Regarding the global performance, the three controllers seem to be equivalently functioning. After a careful analysis on the local level, however, the real advantage of the proposed controller is revealed. The differences are indeed found at the local intersections.

Figure 6.4 displays radar-shape plots of queue length distribution at the intersections. These plots illustrate the average queues in the inflow (streams 1 and 6) and side (streams 2, 4 ,5 and 8) direction to the network, during two time periods: the beginning of the peak hour (Figure 6.4(a) and Figure 6.4(c)), and the middle of the peak hour (Figure 6.4(b) and Figure 6.4(d). For the outflow direction, the demand is served by all of the three controllers, and there is almost no queue at any intersection.

The solid radial lines in gray represent the 20 intersections, while the rings are the queue length contour references. The radar pattern connects the length of the queue of the adjacent intersections, aiming at providing a clear illustration on the resulting queues under the different control strategies. Recall that the queue values have been normalized for comparable illustration. As a general remark, the proposed multi-scale MPC controller handles the queue well for the concerned directions during both time periods. Comparing to the other two controllers, it generates shorter queues at the most intersections. From Figure 6.4(a), it seems that the bang-bang controller works the best for managing the inflow to the network in the onset of the peak hour. This is not surprising, because the bang-bang controller gives the highest priority to the inflow streams by sacrificing the majority of traffic from the side streams (see Figure 6.4(c)). Moreover, the bang-bang controller may release queues of the inflow direction drastically which increases the network accumulation and negates the network-level performance (e.g. it can generate a traffic accumulation larger than the critical accumulation). In the middle of the peak hour, it can be observed that the results are totally different. The MPC controller clearly works the best among the three controllers. As displayed in Figure 6.4(b) , it reduces significantly the queues at most intersections. Furthermore it looks that the queues are more evenly distributed among intersections. The MPC controller outperforms the classical PID controller in both scenarios for the inflow queues. This is expected, as the PID controller does not take into account the detailed configuration and the queue of the intersections. Furthermore, as observed in Figure 6.4(c) and Figure 6.4(d), the proposed control improves the traffic operation of side directions, as well. It is found that the proposed controller significantly reduces the local queues, by up to 60% compared to the bang-bang and the PID controllers. The bang-bang controller, not surprisingly, can result in much longer queues, e.g. for intersection 16 (I-16) the queue length is double under bang-bang



(a) Onset of the peak hour (35 min) at inflow streams (b) Middle of the peak hour (70 min) at inflow streams



(c) Onset of the peak hour (35 min) at side streams (d) Middle of the peak hour (70 min) at side streams

Figure 6.4: Queue length comparison between the MPC-based controller, the bang-bang controller, and the classical PID controller in scenarios without noises. Queue lengths are normalized with the minimum and maximum queue length in the network.

as the one under the MPC or the PID.

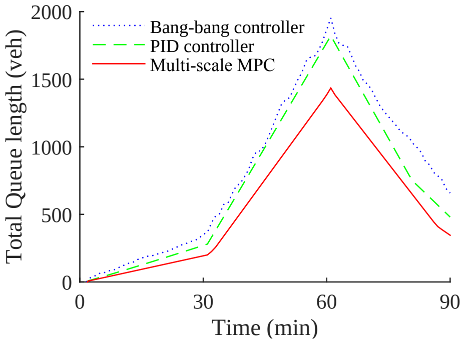


Figure 6.5: Total queue length for all streams and intersections without noises.

Figure 6.5 shows the total queue length at each cycle for the three controllers. It is evident that the proposed controller renders smaller queues than the other two controllers. This is because the proposed controller optimizes the combined cost of both the network and the intersection level.

We have also carried out analysis for scenarios where the demand is low. Under such conditions, the performance is similar for all controllers, e.g. the difference in the resultant total delays is ignorable. This is reasonable, since the magnitude of the impact of local queue on global performance is directly determined by the level of demand flowing into the network.

6.4.2 System performance under moderate noises

In this section, we test the performance of the proposed controller in a noisy environment. Precisely, we consider the stochasticity in the MFD, the measurement noises, and the prediction uncertainties[13]. This makes the simulated MFD essentially different from the prediction MFD in the MPC optimization model. Recall that we assume the connected vehicles are the only information source. In a network fully equipped with connected vehicles, perfect information can be provided at both network and local levels for the controller. However, when the penetration rate is insufficient, an inaccurate state measurement (i.e. traffic accumulation and queue lengths) would be expected. These noises affect the accuracy of the initial state in the MPC controller (see Figure 6.1), i.e. $\hat{n}_{11}(k)$, $\hat{n}_{12}(k)$, $\hat{x}^i_m(k)$ in the proposed model Eq.(6.4)-Eq.(6.15), thus control efficiency could be hindered. Furthermore, as the traffic dynamics can be highly stochastic, it is difficult to predict the future arrivals and departures with complete accuracy.

It is then important to evaluate how less than perfect information regarding these parameters affects the control controller.

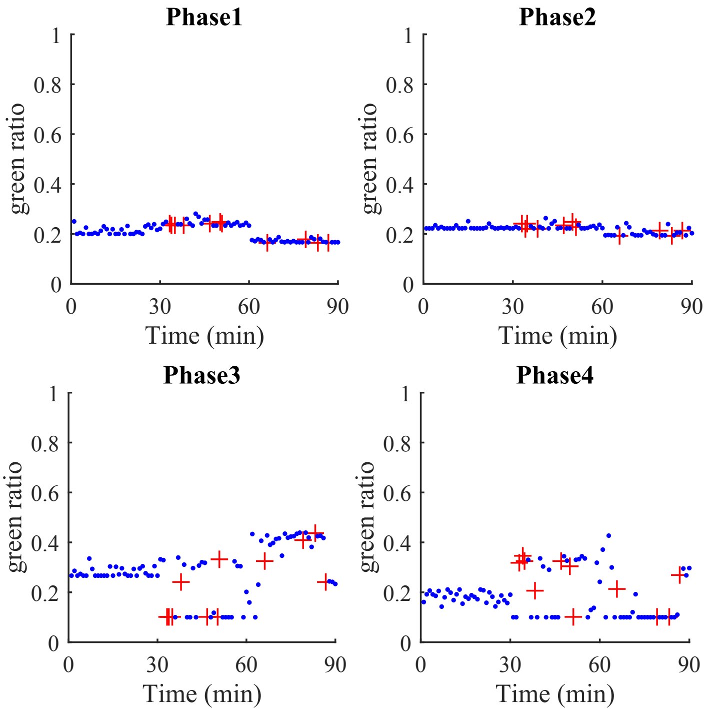
In this section, we look at the system performance for moderate noise level. The error for the traffic accumulation and queue length measurement is assumed to follow a Gaussian distribution with a mean of 0 and a standard deviation of 5% of the mean value. Furthermore, we simultaneously consider the stochasticity existing on the MFD, so the outflow from the network has e.g. a 10% fluctuation (i.e, the actual outflow from the network follows a uniform distribution between $0.9O(t)$ and $1.1O(t)$). The errors in the demand prediction are assumed to follow a Gaussian distribution with mean 0 and a standard deviation of 10% of the value. These three types of noises are assumed to be independent.

Figure 6.6(a) displays the resulting green ratio over time, for the four different phases of a representative intersection operating under perimeter control. The performance of the other intersections looks similar even though they have different demand patterns and capacities. Recall that Phases 2 and 4 accommodate the network inflow and outflow respectively, while Phase 3 serves both directions. Phase 1 regulates the directions other than the network inflow and outflow. It receives green time allocation up to 25% of the cycle time. Such result is obtained because the controller aims to minimize the total delay of all traffic. A simplification of the intersections with only inflow and outflow direction cannot reflect the impact of control on local queues, and possibly overestimates the transfer flows between the two regions. Treating the entire intersection is thus important. From the time series of the Phase 2, it can be observed that the restriction of the inflow to the network occurs during the peak hour between minutes 30 and 90.

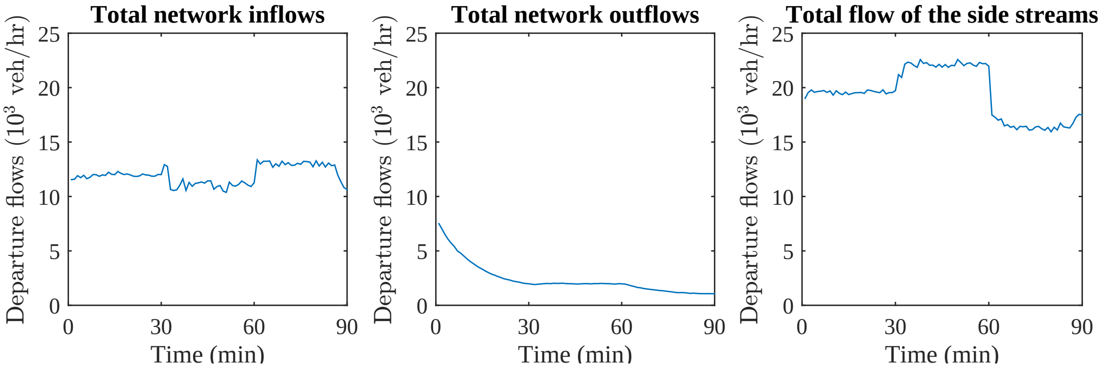
Figure 6.6(b) shows the controlled inflow (Streams 1 and 6), outflow of the center region (Streams 3 and 7), and other controlled departure flows (Streams 2, 4, 5 and 8) at the aggregated level for the whole network. It is evident that all the departure flows are quite smooth under moderate noises. Figure 6.6(c) displays the time series of traffic accumulation. The changes of flow restriction and traffic accumulation at the network level appear to be stable and smooth.

---

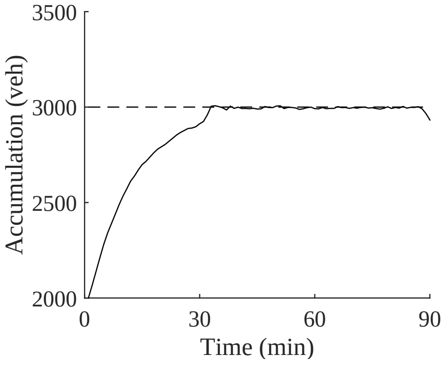
[13] Note that the noisy environment also includes scenarios with uncontrolled intersections. The noises brought by the uncontrolled intersections (as in Keyvan-Ekbatani et al., 2012) can be combined with the internal demand.



(a) Green ratios at a representative intersection



(b) Departure flows (across all intersections)



(c) Network traffic accumulation

Figure 6.6: System performance with moderate noises.

6.4.3 Performance deterioration with strong noises in the system

We investigate now the performance when the system error increases. The corresponding noises on the measurement of accumulation and queue length reach to 15% (instead of 5% in Section 6.4.2). The errors in the demand prediction are assumed to follow a Gaussian distribution with mean 0 and a standard deviation of 30% (in contrast to 10% in Section 6.4.2) of the value. The outflow calculated from the MFD is assumed to have a 20% fluctuation (in contrast to 10% in Section 6.4.2) . All the other settings remain unchanged.

We reproduce the same graphs from Figure 6.6 in Figure 6.7. The accumulation time series in Figure 6.7(c) clearly indicates a strong oscillation after the control is activated. Oscillations are also observed in the controlled flows. After a first glance, it may seem the proposed controller works in a bang-bang style, as up-and-downs of departure flow can be found at local intersections and at the network.

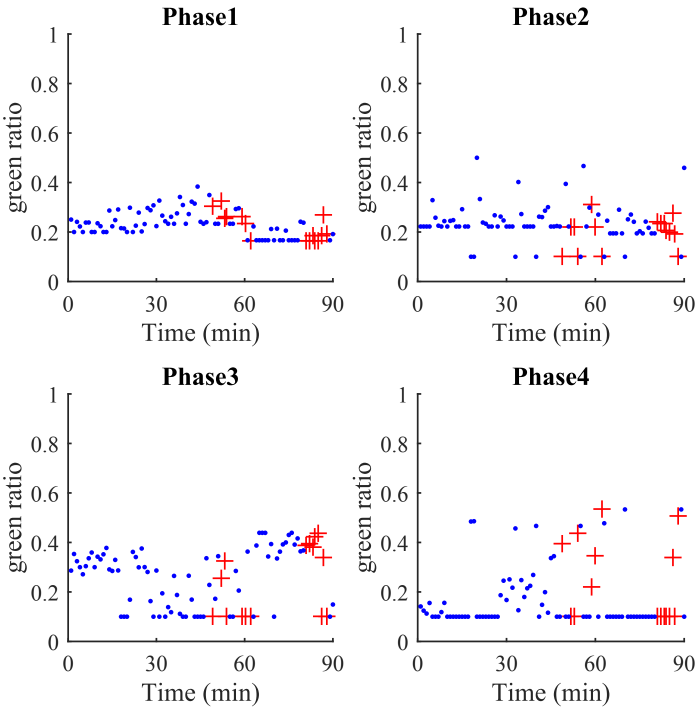
To investigate in detail the difference in mechanism between the proposed controller and the bang-bang controller, we plot in Figure 6.7(a) the time series of the green ratio with scatters and colors. Red color represents the time intervals when the accumulation is larger than the critical accumulation. It can be seen that the proposed control does not follow a "green-or-red" logic which is employed by the bang-bang controller. The proposed controller sometimes even restricts the inflow, when the accumulation is below the critical one (for example at minute 44). As the proposed controller predicts the possible increase in that may cause queues later, inflow is restricted in advance. This reflects the fundamental difference of the proposed control from the bang-bang which takes only myopic decisions.

Table 6.1: Comparison between the two MPC, the PID controller and the Bang-bang controller. Numbers within parenthesis reflect the change with respect to the multi-scale MPC for each of the scenarios.

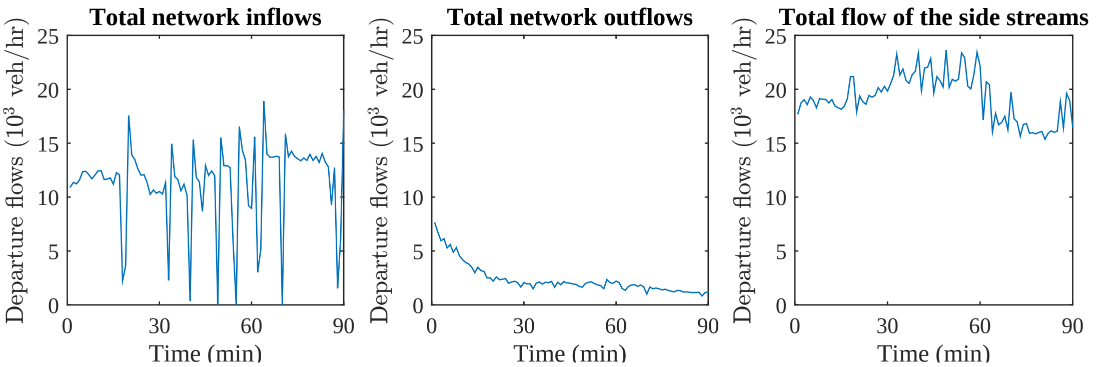
		Large noise	Moderate noise	No noise
Total travel cost (hr)	Multi-scale MPC	7263	6122	5443
	PID controller	7722 (+6.3%)	6663 (+8.8%)	5708 (+4.9%)
	Bang-bang controller	8928 (+22.9%)	7481 (+22.2%)	6211 (+14.1%)
Total intersection delay (hr)	Multi-scale MPC	2536	1495	1082
	PID controller	2950 (+16.3%)	1972 (+31.9%)	1341 (+23.9%)
	Bang-bang controller	3953 (+55.9%)	2624 (+75.5%)	1485 (+37.2%)

A quantitative comparison between the proposed multi-scale MPC, the PID controller, and the bang-bang controller in scenarios with large noises, moderate noises and no noises is shown in Table 6.1. It is shown that the hybrid controller outperforms both the PID and the bang-bang controller, especially with noises. Specifically, the savings in network travel times is more than 4.9% compared to the PID controller and more than 14% compared to the Bang-bang controller. The savings in intersection delay is more than 16% compared to the PID controller and more than 37% compared to the Bang-bang controller.

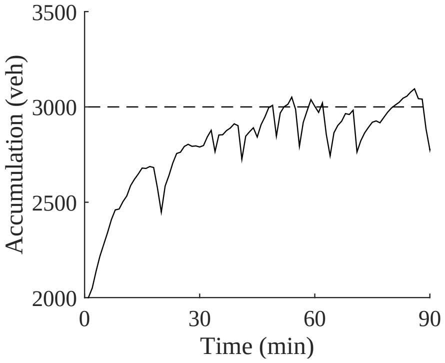
In summary, we have demonstrated that although the proposed strategy performs well under moderate noises, strong noises and uncertainties in the system due to the low penetration rates of CVs can create strong fluctuations in system dynamics. This type of fluctuation indicates an over-correction of the controller, which hinders the performance of the system. To address this issue, we extend the multi-scale perimeter control strategy to a stochastic MPC in the next section.



(a) Green ratios at a representative intersection



(b) Departure flows (across all intersections)



(c) Network traffic accumulation

Figure 6.7: System performance with strong noises.

6.5 Stochastic Controller

6.5.1 Design of the stochastic MPC

Given the results of Section 6.4.3, it is evident that the controlled performance of the system can become worse when perfect and accurate information on the traffic accumulations and demand is not available. Furthermore, utilizing deterministic traffic models (e.g. the non-scattered MFD) overlooks the stochastic nature of traffic dynamics, thus might make the control actions given by Eq.(6.4)-Eq.(6.15) non-optimal.

To this end, we extend the developed controller to account for the noises. We explicitly consider two types of noises: demand prediction errors and measurement errors in traffic accumulation and queue length. In other words, we consider the predicted demand ($D_{ab}(k+l|k)$, $q^i_m(k+l|k)$), and measurement ($\hat{n}_{ab}(k)$, $\hat{x}^i_m(k)$) to be random variables following known distributions. In fact, the distributions of these random variables can be estimated from real-time or historical CV data with Bayesian filters (e.g. Kalman filter) or machine learning techniques. The details are beyond the scope of this thesis. Interested readers can refer to Yuan et al. (2012), Gayah and Dixit (2013), Ramezani and Geroliminis (2015), or Ambühl et al. (2018).

In contrast with existing robust controllers which optimize the worst-case performance of the controller in a noisy environment (Haddad and Shraiber, 2014; Haddad, 2015), we employ a stochastic MPC based methodology. For the definition of stochastic MPC, interested readers can refer to Kouvaritakis et al. (2004) and Couchman et al. (2006). The reasons for proposing a stochastic MPC in this case are two-fold. First, the robust control methods normally assume bounded noises, which does not necessarily apply for this system. The actual internal demand at the network level and the arrival flows at the intersection level can be in a large range, which could make the resulting control action too conservative due to the strong stochasticity in the system. Second, applying the robust control methods to this problem could render a complex, or even intractable optimization problem, considering the scale and complexity of the formulated model.

Therefore, we propose a two-stage stochastic MPC-based methodology, which employs stochastic programming. This type of control method has been extensively applied in many disciplines (e.g. Farina et al., 2016; Parisio et al., 2016; Tong et al., 2015). Recall that for the MPC-based controller, only the control action (green ratios) in the first cycle is executed (the others are for prediction purposes). Hence, we aim to determine the green ratios in the first optimized cycle (i.e. the current cycle) such that the optimal green ratios minimize the average cost in the L future cycles. In the rest of this section, we establish the two stages of the stochastic MPC. The first stage master problem considers the green ratios to be executed (i.e. $g^i_p(k)$) subject to the physical constraints of the green ratios. The second stage subproblem estimates the total travel cost resulting from the chosen green ratios for each scenario arrival flow, internal demand, traffic accumulation, and queue lengths for each sample in the sample space R.

The first stage master problem is formulated as Eq.(6.28)-Eq.(6.30).

$$\min E_r J_{D,r} \tag{6.28}$$

$$\text{s.t.} \sum_{p \in P} g^i_p(k) \leq g^i_{\max}, \ \forall i \in I \tag{6.29}$$

$$g^i_p(k) \geq g^i_{\min,p}, \ \forall i \in I, p \in P^i \tag{6.30}$$

where r is one sample in the sample space R. $J_{D,r}$ is the optimal objective value of the second stage problem resulting from the decision variable $g^i_p(k)$ in sample r.

The second stage subproblem is formulated as follows.

$$\begin{aligned} \min\ J_{D,r} &= C\sum_{l=1}^{L}\Big(n_{11,r}(k+l|k)+n_{12,r}(k+l|k)\Big) \\ &\quad + C\sum_{l=1}^{L}\sum_{i\in I}\sum_{m\in M^i\setminus M^i_{\text{out}}} x^i_{m,r}(k+l|k) \end{aligned} \tag{6.31}$$

where the decision variables of the second stage problem in each sample r are $g^i_{p,r}(k+l)$, $1\leq l\leq L$. The corresponding $n_{11,r}(k+l|k)$, $n_{12,r}(k+l|k)$, $x^i_{m,r}(k+l|k)$, together with $\mu_{m,r}(k+l|k)$, $\beta_{11,r}(k+l|k)$ and $\beta_{12,r}(k+l|k)$, are regarded as functions of $g^i_{p,r}(k+l)$. These variables satisfy constraints Eq.(6.5)-Eq.(6.15). Note that for the following cycles (starting from $k+1, l\geq 1$), the green ratios $g^i_{p,r}(k+l)$ are different for each sample r. In contrast, the green ratios in the current cycle, $g^i_p(k)$, are given as known parameters for the second stage problem, and they are the same for each sample, i.e.

$$g^i_{p,r}(k)=\hat{g}^i_p(k),\ \forall p\in P^i, \forall i\in I, \forall 0\leq l\leq L-1 \tag{6.32}$$

We can simplify the second stage problem into a linear programming problem with the same approximation as utilized in Section 3.2.

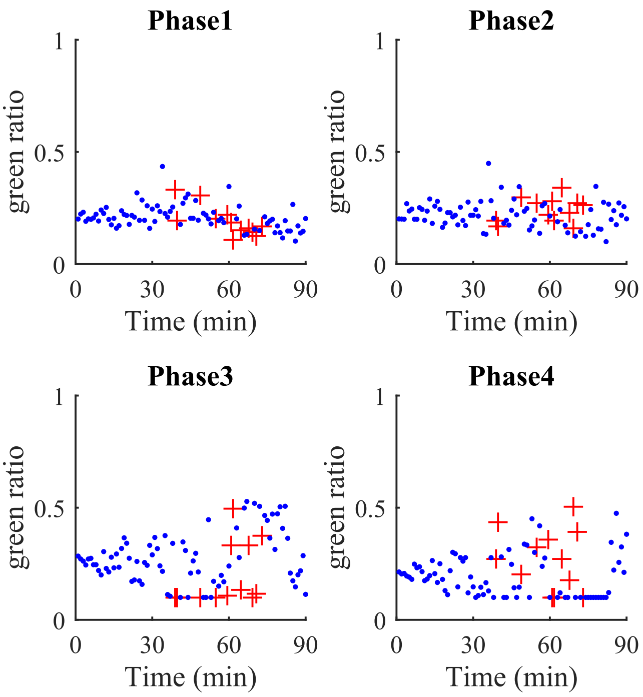
The two stages interact with each other. The connection between the first stage problem and the second stage problem is two-fold. First, the objective function of the first stage problem is calculated by all the second stage problems. Second, the solution to the first stage problem is the initial solution for the second stage problem.

However, as the sample space R is continuous, it is not possible to enumerate all the samples to calculate the expected travel cost. One general method to handle this problem is the so-called Sample Path Optimization (Robinson, 1996), i.e. to sample a finite number of scenarios, R_0, following the given distribution. Specifically, we sample the arrival flow, internal demand, traffic accumulation, and queue lengths as $q^i_{m,r}(k+l|k)$, $D_{ab,r}(k+l|k)$, $\hat{n}_{ab,r}(k)$ and $\hat{x}^i_{m,r}(k)$. Ideally, we should also adopt a large sample size (i.e. number of scenarios with different realizations of the same noise level) to estimate the expected travel cost more accurately. However, for the sake of computational efficiency, we limit the sample size to 20-50, as this number of samples is enough to cover the sample space. For information on how to generate representative samples, interested readers can refer to Sobol (1967) and Ge and Menendez (2014). It is shown in Section 5.2 and Section 5.3 that the stochastic MPC performs sufficiently well with a sample size of 20 for the cases with similar city size to Zurich. Empirical simulations also indicate that the benefit of having a larger sample size than 20 is marginal.

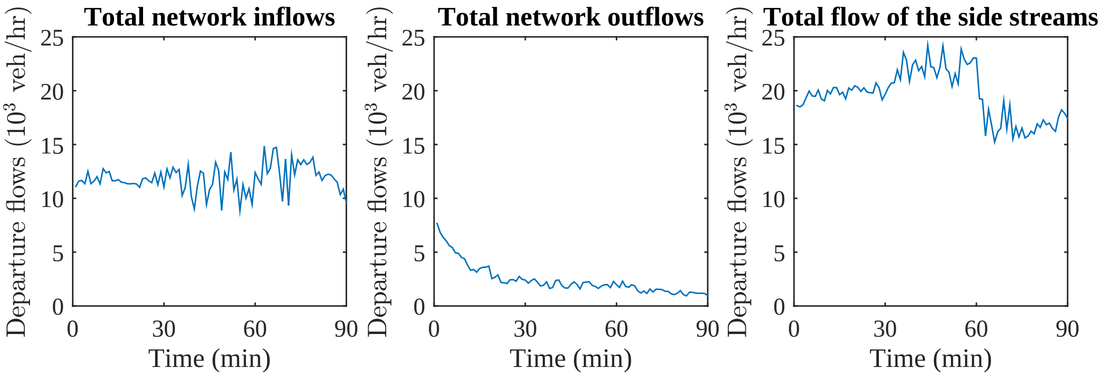
We combine the two stages into one linear programming problem and solve it with the simplex method in Cplex. This is proved to be more efficient than the $L-$shape algorithm (Higle and Sen, 1991) which iteratively solves the two stage problem.

6.5.2 Performance of the stochastic MPC

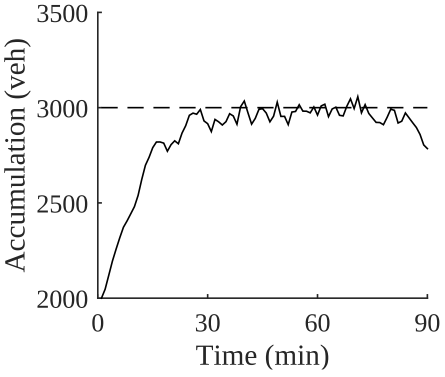
We implement the stochastic controller using the same simulation settings (with strong noises) as in Section 6.4.3 and evaluate the performance of the controller. Figure 6.8(c) shows the comparison of the accumulation time series after applying the stochastic controller. The sample size is chosen as 20 and the computational time for each time step is 15s (in contrast with less than 1s for the multi-scale MPC based controller). Compared to Figure 6.7, it is evident that the system changes in a more stable manner thanks to the stochastic controller. Compared to the oscillations of the multi-scale controller, the stochastic controller successfully maintains the accumulation around the critical one (Figure 6.8(b)). The fluctuation of the network inflows is



(a) Green ratios at a representative intersection



(b) Departure flows (across all intersections)



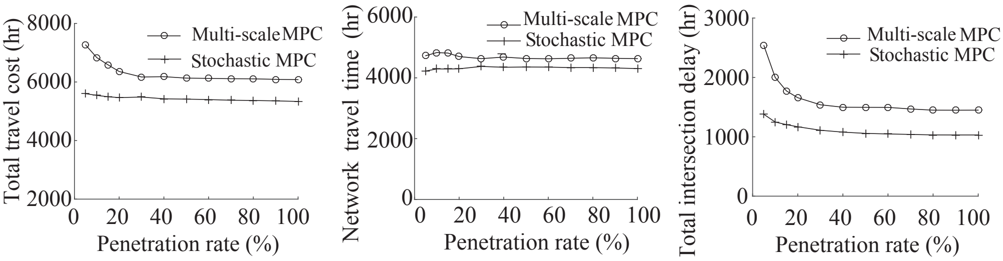
(c) Network traffic accumulation

Figure 6.8: System performance of the stochastic MPC with strong noises.

also reduced (Figure 6.8(c)).

6.5.3 Value of connected vehicles

We evaluated the value of connected vehicles in this subsection (i.e. the value of the information level introduced in Chapter 4). The penetration rate of connected vehicles affects the measurement of the traffic accumulation and the queue lengths. Existing works identified the relationship between the measurement errors and the penetration rates, e.g. in Ramezani and Geroliminis (2015) and Gayah and Dixit (2013). To avoid randomness, we select 10 random seeds for executing the controllers, we then compute the average value of each available performance measure. The resulting performance measures from both the multi-scale controller and the stochastic controller are illustrated in Figure 6.9.



(a) Total travel cost (b) Network travel time (c) Intersection delay

Figure 6.9: Performance of the stochastic MPC in different penetration rates of connected vehicles.

Figure 6.9(a) shows the total travel costs in the whole system over different levels of penetration rates. It is observed that the stochastic MPC always outperforms the multi-scale MPC for all the penetration rates. It can be seen that the total travel costs resulting from the stochastic MPC is less sensitive to the penetration rates than that of the multi-scale MPC. This is expected, as the proposed stochastic MPC controller aims to maintain the system at a desired state given that uncertain noises exist in the system. Regardless of the different levels of the noise, the control actions eventually brings the system towards the desired state. Furthermore, Notice that there are still noticeable gaps between the two curves even at a penetration rate of 100%. One possible explanation is that the system has uncertainty in demand and MFD. Such uncertainty exists even for 100% penetration rate.

Next, let us look into more details at both the network level and the local level. At the network level (Figure 6.9(b)), it is shown that the network travel time is not sensitive to the penetration rates for both the multi-scale and stochastic MPC. This is expected, as both controllers can stabilize the system. Note that the total network travel time for the scenarios with low penetration rates is even slightly smaller than that of the scenarios with high penetration rates. This is because there are less vehicles in the network for the scenarios with low penetration rates, as fewer vehicles are let in at the perimeter. At the local level (Figure 6.9(c)), the total intersection delay is strongly influenced by the penetration rate for both the multi-scale and stochastic MPCs. By increasing the penetration rate from 5% to 100%, the total delay at the perimeter intersections resulting from the multi-scale MPC and the stochastic MPC is reduced by 41.1% and 27.3%, respectively. The reduction in delay clearly indicates that for both controllers, the penetration rate is more important for the intersection control than the network control. We can also see that the stochastic controller is less sensitive to the penetration rates than the multi-scale controller. This indicates that the stochastic controller exhibits more robustness to uncertainties in the system.

Even if the stochastic MPC tends to outperform the multi-scale MPC, the multi-scale MPC has the strong advantage of low computational cost. Hence, under higher penetration rates and low system stochasticity, it might be advisable to apply the multi-scale control in reality.

6.6 Conclusions and Future Work

This chapter proposes a multi-scale MPC based strategy to integrate the network level perimeter control and the local level signal control in a traffic system with one center region and one periphery. The model calculates the optimal green ratios that minimize the total travel costs for both levels in a moving time horizon. Connected vehicles provide not only information on the current traffic state in both network (traffic accumulation) and local level (queue length), but also provide information on possible future arrivals. The predictive nature of the connected vehicles enables a more efficient application of the MPC based strategy. This MPC based strategy is further extended into a two-stage stochastic MPC to cope with large system noises due to the stochasticity of the system and the lack of information. The stochastic controller optimizes the expected travel cost resulting from the current control action.

Case studies show that the multi-scale MPC based strategy successfully stabilizes the traffic accumulation in the center network with the least impact on the perimeter intersections. The total travel cost is minimized for the vehicles both in the network and on the perimeter. The proposed controller is also robust to moderate noises. In scenarios with large noises, the stochastic controller is shown to successfully reduce the oscillations in the system. We looked at the applicability of the proposed two controllers under different penetration rates of connected vehicles. For scenarios with very low penetration rate and/or large system noises, the stochastic controller is highly recommended. The sample size in the stochastic optimization model should be no less than 20 and compatible with the computational power of the system. For scenarios with high penetration rate and low system noises, multi-scale control already gives satisfactory results.

In the proposed framework, we assume that conventional and connected vehicles exist in the traffic system. We will further investigate how the proposed control strategies can be extended to consider the scenarios where automated vehicles are present in the traffic system, with or without dedicated infrastructure (e.g. in Yang et al. (2019)). Furthermore, we consider the inflow to the network entering from a fixed intersection and neglect the possible reroute to its adjacent intersections when queue length is large. This can be important, especially in the scenarios where automated vehicles exist in the traffic systems. Incorporating this effect will be a future work, which requires significant amount of research efforts on the modelling of the system dynamics. This work does not assume the outside network to be homogeneous with a single MFD. The arrival rate to the perimeter intersections is assumed to be predicted with real-time connected vehicles information. The impact of the control policy on the outside network and integration of the network and periphery for multi-region perimeter control are considered also as future research directions. Additionally, it would be interesting to study how the proposed model can be extended to consider the scenarios where the perimeter of the each region changes dynamically due to real-time partitioning of the multi-region network. Moreover, the model can be adapted to address multimodal networks by incorporating multimodal MFDs (Geroliminis et al., 2014; Loder et al., 2017) and flexible sharing strategies (He et al., 2018a). As a matter of fact, the algorithm presented in this chapter has already been extended as an alternative to dedicated bus lanes (Chiabaut et al., 2018). Another potential limitation of the proposed approach is that perimeter control may not be able to handle the scenarios with very high demand. As a future work, we will test the performance of the proposed strategy in such scenarios and integrate incentives so that passengers switch their transportation mode or departure time.

Chapter 7

Perimeter Control with Priority Lanes in a Connected Vehicle Environment

This chapter is partially based on the following papers.

- Yang, K., N. Zheng and M. Menendez (2018e) A perimeter control approach integrating dedicated express toll lanes, paper presented at the *Transportation Research Board 97th Annual Meeting*.
- Yang, K., N. Zheng and M. Menendez (2017a) Integrating perimeter control with dedicated express toll lanes, paper presented at the *Traffic and Granular Flow Conference (TGF 2017)*.

7.1 Objectives and Contributions

Although extensive efforts have been made to study the perimeter control for large-scale urban networks (Haddad and Geroliminis, 2012; Keyvan-Ekbatani et al., 2012; Geroliminis et al., 2013; Haddad et al., 2013; Aboudolas and Geroliminis, 2013; Keyvan-Ekbatani et al., 2015; Haddad, 2015), priority schemes are rarely incorporated to differentiate the incoming vehicles to the network. Some works have analyzed multimodal networks and considered the interactions between public transport vehicles and cars (Ampountolas et al., 2014; Chiabaut, 2015; Ampountolas et al., 2017), without providing, however, any priority to public transport. It would be beneficial for the entire traffic system if priority is provided to not only public transport vehicles, but also other types of vehicles, such as emergency vehicles, ride-sharing vehicles, etc.

In this chapter, we extend the perimeter control approach proposed in Chapter 6 to address the research gap (N3) described in Section 2.3. We dynamically provide priority to certain groups of vehicles to maximize the social welfare and enhance passenger mobility. The idea is to integrate priority lanes at the local perimeter intersections where incoming vehicles can choose the different dispersing lanes upon arrival. Such priority lanes can enable faster service subject to a toll. The contributions of this chapter are two-fold.

1) We integrate a priority scheme into the perimeter control to improve both the performance and the welfare of all vehicles.
2) We propose an algorithm to dynamically recalibrate the VOT distribution.

The rest of the chapter is organized as follows. Section 7.2 presents the general methodology. It discusses the system dynamics, formulates the integration of perimeter control and priority lanes into a single control problem, and introduces an online VOT recalibration algorithm. Section 7.3 describes the simulation settings. Section 7.4 presents a case study inspired by the city of Zurich to evaluate the performance of the proposed strategy considering vehicle heterogeneity and the value of the recalibration algorithm. Section 7.5 performs a sensitivity analysis to evaluate the robustness of the proposed algorithm. Section 7.6 concludes the chapter and presents some ideas for future work.

7.2 General Methodology

7.2.1 Problem presentation

Following Chapter 6, we consider a city with a single center (Region 1) and a suburban area (Region 2) with a large demand from the suburban area to the city center. For the simplicity of presentation, we consider only cars in this chapter, which can include private cars, ride-sharing cars, taxis, etc. The proposed strategy can, nevertheless, be generalized to account for public transport vehicles and emergency vehicles and provide free priority services to them. We further assume that a certain portion of vehicles are connected vehicles, and are able to provide information on the traffic systems. We perform perimeter control at a set of perimeter intersections I. The dynamics of the city center can be described using an MFD, $O = G(n)$, in the same manner as in Eq.(6.1) and Eq.(6.2).

Without the loss of generality, we consider the heterogeneity in car VOTs, defined as the sum of the VOTs of all passengers in a car. Notice that the proposed strategy also applies to other priority metrics characterizing the passengers; welfare. It is assumed that none of the connected vehicles report their VOTs, as it is private information and it can be hard to tell. Instead, we assume certain initial knowledge of the distribution of car VOTs, i.e. the combined distribution of person VOT and car occupancy. Such knowledge does not have to be accurate, as we develop a recalibration algorithm to update this distribution in an online manner (see Section 7.2.4). We also assume the same VOT distribution across all the perimeter intersections. The proposed modeling and control strategies can be easily generalized to account for the spatial heterogeneity of the VOT distribution (or other priority metrics).

To provide priority to the high valued cars, we dedicate certain lanes at some of the perimeter intersections as priority lanes (e.g. in the inflow direction to the network). Such priority lanes, subject to a dynamic toll, are given more favorable signal timings to ensure a fast service for the vehicles that use them. We hereafter name the perimeter intersections with priority lanes as priority intersections, and the ones without priority lanes as non-priority intersections, denoted as I_{pri} and I_{non}, respectively. We further denote the set of movements[14] at intersection i as V^i, indexed by ξ, which can be further divided as $V^i = V^i_{\text{pri}} \cup V^i_{\text{non}}$. Here, V^i_{pri} represents the set of movements where priority lanes are installed and V^i_{non} represents the other movements. Priority movements only exist at priority intersections. At non-priority intersections $V^i_{\text{pri}} = \emptyset$. For each priority intersection i, we further classify the set of streams as $M^i = T^i \cup N^i \cup Y^i$ where priority streams (T^i) are the streams on the priority lanes, normal streams (N^i) are the streams associated with the same movement as the priority streams but using normal (i.e. non-priority) lanes, and general streams (Y^i) are the rest of the streams (e.g. side streams). The sets T^i, N^i, and Y^i are mutually exclusive. For a non-priority intersection i, $N^i = T^i = \emptyset$ and $Y^i = M^i$, where M^i is the set of all streams at intersection i. The relation between a movement ξ and a stream m can be characterized using the function $\xi = \phi(m)$, representing that stream m belongs to movement ξ. One possible layout for the priority lanes, normal lanes, and the general lanes at a priority intersection is shown in Figure 7.1. Notice that the priority lanes are installed after a certain point, hereafter called the entrance.

Cars that belong to a priority movement make lane decisions (i.e. whether to take priority lanes or the normal lanes) when they are shortly before the entrance of the priority lanes. We assume that these cars will not change the type of lanes after they enter the chosen lane. This assumption is reasonable, as changes in the type of lanes after the entrance can be restricted by the infrastructure. Cars will receive the information of the current toll $\tau^i(k)$ for taking the priority lanes at priority intersection i in the cycle k when they make decision. This toll is only charged to the cars when they take the priority lanes, i.e. once for each car. It is assumed that

---

[14] Recall that a movement represents the traffic moving in a certain direction (defined in Section 2.1.1).

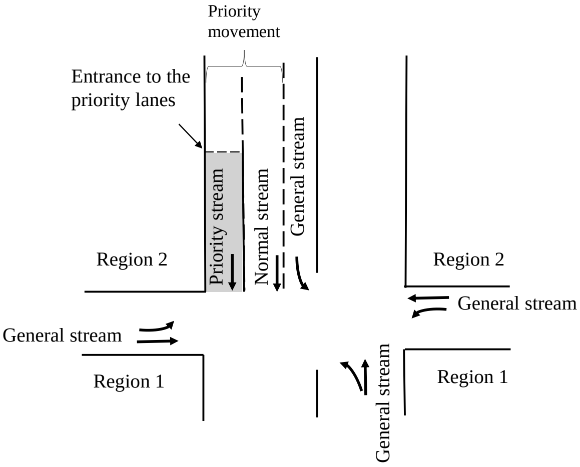


Figure 7.1: Illustration of one possible layout at priority intersection i.

cars will change to the priority lanes as soon as they make the decision. Also notice that we have intersection specific tolls, as different perimeter intersections might have different traffic conditions. We assume that the decision of the cars relies on the toll information and its estimation of the traffic conditions. Collectively, this can be modelled by a discrete choice model, which is one of the most commonly employed in this context (Zheng and Geroliminis, 2013, 2016). The choice function is assumed as a logistic function, represented as Eq.(7.1). Notice that we define $\rho_m(\cdot)$ as a function, and thus we use generic variables rather than those related to a certain cycle or intersection (the same holds for Eq.(7.2)).

$$\rho_m(\tau,\sigma,\boldsymbol{x},\boldsymbol{\mu}) = \frac{\exp\Big(-\omega U_m(\tau,\sigma,x_m,\mu_m)\Big)}{\sum\limits_{m'\in M^i,\phi(m')=\phi(m)} \exp\Big(-\omega U_{m'}(\tau,\sigma,x_{m'},\mu_{m'})\Big)},\ m\in M^i,\ \phi(m)\in V^i_{\text{pri}} \tag{7.1}$$

where ω is a coefficient which can be calibrated from survey data. A larger ω represents higher rationality. The scenarios where $\omega \to \infty$ imply that all drivers are rational, i.e. they choose the type of lane that maximizes their utility when entering the network $U_m(\tau,\sigma,x,\mu)$. Here, the utility function is estimated by the drivers using the observed queue length x, the estimated discharging flow μ, the VOT of a car σ, and the current toll τ, as in Eq.(7.2).

$$U_m(\tau,\sigma,x,\mu) = \begin{cases} -\tau - \dfrac{x}{\mu}\sigma, & m\in T^i,\ i\in I_{\text{pri}} \\ -\dfrac{x}{\mu}\sigma, & m\in N^i,\ i\in I_{\text{pri}} \end{cases} \tag{7.2}$$

Notice that it is not required that drivers have perfect information. In other words, the observed queue length and discharging flows may not be exactly the same as the actual ones. Error terms could exist in the observation. Also notice that Eq.(7.2) can be more complex to include more factors. In this dissertation, we use the simple utility function as an example.

For the ease of modeling, we discretize the VOT distribution, and categorize the cars wishing to enter the network into $|J|$ groups as a set J. The cars within each group have similar VOTs. Denote the average VOT of each group as σ_j, and the probability that a car falls into VOT group j as $\pi_j \geq 0$, satisfying

$$\sum_{j\in J}\pi_j = 1 \tag{7.3}$$

Then the cars within each vehicle group satisfy the following mass conservation equation.

$$x^i_{mj}(k+1) = x^i_{mj}(k) + q^i_{mj}(k)C - \mu^i_{mj}(k)C,\ m \in M^i,\ i \in I,\ j \in J \tag{7.4}$$

where $x^i_{mj}(k)$, $q^i_{mj}(k)$, and $\mu^i_{mj}(k)$ represent the queue length, arrival flow, and the departure flow in stream $m \in M^i$ for vehicle group j at intersection i during time step k. C is the cycle length. Unlike in Chapter 6 where $q^i_{mj}(k)$ is treated as a given parameter, in this chapter, $q^i_{mj}(k)$ is a function of the tolls. Here, we calculate $q^i_{mj}(k)$ as

$$q^i_{mj}(k) = \begin{cases} \lambda^i_{\xi j}(k) z^i_{mj}(k), & m \in T^i \cup N^i,\ \xi = \phi(m),\ j \in J,\ i \in I \\ \lambda^i_{\xi j}(k), & m \in Y^i,\ \xi = \phi(m),\ j \in J,\ i \in I \end{cases} \tag{7.5}$$

where $\lambda^i_{\xi j}(k)$ represents the incoming demand in vehicle group j in movement $\xi \in V^i$ to the perimeter intersection $i \in I$, which is treated as a given parameter in this chapter. Notice that we need additional differentiation between the flow of a stream m, $q^i_{mj}(k)$, and the flow in the movement $\xi = \phi(m)$, $\lambda^i_{\xi j}(k)$, because a priority movement can include multiple streams. Here, $z^i_{mj}(k) = \rho_m(\tau^i(k), \sigma_j, \tilde{x}^i_T(k), \tilde{\mu}^i_T(k), \tilde{x}^i_N(k), \tilde{\mu}^i_N(k))$ represents the probability for cars in vehicle group j driving in the priority movement $\xi = \phi(m)$ to choose stream $m \in M^i$, where $\tilde{x}^i_m(k)$ and $\tilde{\mu}^i_m(k)$ represent driver's estimation of the queue length and departure flow.

In the rest of the chapter, we denote the aggregated variables for $x^i_{mj}(k)$, $q^i_{mj}(k)$, $\mu^i_{mj}(k)$, and $\lambda^i_{\xi j}(k)$ as $x^i_m(k)$, $q^i_m(k)$, $\mu^i_m(k)$, and $\lambda^i_\xi(k)$, respectively. For each pair of them, the aggregated variable equals to the sum of the variables across all vehicle groups. It is also assumed that the incoming demand to movement $\xi \in V^i$ follows the overall VOT distribution, i.e.

$$\lambda^i_{\xi j}(k) = \pi_j \lambda^i_\xi(k),\ \xi \in V^i,\ j \in J,\ i \in I \tag{7.6}$$

7.2.2 MPC approach

In this section, we extend the multi-scale MPC proposed in Chapter 6 to account for the heterogeneity of cars and to maximize social welfare. We do not only properly determine the signal timings at the perimeter intersections, but also optimize the toll. This is achieved by solving an embedded optimization model where we obtain the optimal green ratios at each perimeter intersection and the tolls on each link with priority lanes for the future cycles, but only execute the control of the current cycle k. The MPC follows a similar process as in Figure 6.1.

In each control cycle k, there are two types of inputs to the MPC, the measured traffic states and the predicted demand. The measured traffic states include the measured total queued vehicles outside the perimeter and the traffic accumulation inside the network. The predicted traffic demand consists of the predicted internal demand and the predicted entering demand for each car group, each stream, and each intersection. The outputs from the MPC, i.e. the control variables, are the green ratios at each intersection and the optimal toll. The formulation of the embedded optimization problem is described as follows.

$$\begin{aligned} \min J_D = {} & C\bar{\sigma} \sum_{l=1}^{L} \Big(n_{11}(k+l|k) + n_{12}(k+l|k) + \sum_{i \in I} \sum_{m \in Y^i \setminus M^i_{\text{out}}} x^i_m(k+l|k) \Big) \\ & + C \sum_{l=1}^{L} \sum_{i \in I} \sum_{m \in T^i \cup N^i} \sum_{j \in J} \sigma_j x^i_{mj}(k+l|k) \end{aligned} \tag{7.7}$$

s.t. Eq.(6.5) – Eq.(6.9) , Eq.(6.12) – (6.14), and the followings

$$\mu^i_m(k+l|k) = \min\left\{x^i_m(k+l|k)/C + q^i_m(k+l|k),\ \sum_{p\in P^i} s^i_{mp} g^i_p(k+l|k)\right\},$$
$$\forall m \in Y^i \backslash M^i_{\text{out}},\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.8}$$

$$x^i_m(k+l+1|k) = x^i_m(k+l|k) + q^i_m(k+l|k)C - \mu^i_m(k+l|k)C,$$
$$\forall m \in Y^i \backslash M^i_{\text{out}},\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.9}$$

$$\mu^i_{mj}(k+l|k) = \min\Big\{x^i_{mj}(k+l|k)/C + q^i_{mj}(k+l|k),$$
$$\sum_{p\in P^i} s^i_{mp} g^i_p(k+l|k)\frac{x^i_{mj}(k+l|k)}{x^i_m(k+l|k)}\Big\},\ \forall m \in T^i \cup N^i,\ \forall i \in I,$$
$$\forall j \in J,\ \forall 0 \le l \le L-1 \tag{7.10}$$

$$x^i_{mj}(k+l+1|k) = x^i_{mj}(k+l|k) + q^i_{mj}(k+l|k)C - \mu^i_{mj}(k+l|k)C,$$
$$\forall m \in T^i \cup N^i,\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.11}$$

$$x^i_m(k+l+1|k) \le X^i_{m,\max},\ \forall m \in M^i,\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.12}$$

$$q^i_{mj}(k+l|k) = \pi_j \lambda_\xi(k+l|k) z^i_{mj}(k+l|k),\ \forall m \in T^i \cup N^i,\ \xi = \phi(m),\ \forall j \in J,$$
$$\forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.13}$$

$$z^i_{mj}(k+l|k) = \rho_m(\tau^i(k+l|k), \sigma_j, \boldsymbol{x}_\xi(k+1|k), \boldsymbol{\mu}_\xi(k+1|k),\ \forall j \in J,$$
$$\forall m \in T^i \cup N^i,\ \xi = \phi(m),\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.14}$$

$$\mu^i_m(k+l|k) = \sum_{j\in J} \mu^i_{mj}(k+l|k),\ \forall m \in T^i \cup N^i,\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.15}$$

$$x^i_m(k+l|k) = \sum_{j\in J} x^i_{mj}(k+l|k),\ \forall m \in T^i \cup N^i,\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.16}$$

$$x^i_m(k|k) = \hat{x}^i_m(k),\ \forall m \in Y^i \backslash M^i_{\text{out}},\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.17}$$

$$x^i_{mj}(k|k) = \hat{x}^i_{mj}(k),\ \forall j \in J,\ \forall m \in T^i \cup N^i,\ \forall i \in I,\ \forall 0 \le l \le L-1 \tag{7.18}$$

where the objective function Eq.(7.7) represents the social welfare both inside the network and at the perimeter. Notice that we track each individual vehicle group and use individual VOTs to calculate the social welfare for the priority and normal streams. However, we calculate the social welfare based on an assumed VOT (e.g. the mean VOT from the VOT distribution), $\bar{\sigma}$, for inside the network and in the general streams. The reasons are three-fold. First, it is necessary to track each individual vehicle group in order to find the optimal toll for the priority and normal streams. On the other hand, the distribution of VOT in these streams can be recalibrated based on the lane choice of the connected cars. Notice that we assume that the connected cars are representative for the entire car population. Second, we cannot, however, obtain accurate measurements on the evolution of the accumulation or queue length of general streams within each vehicle group. Third, using the assumed VOT does not require the tracking of individual vehicle groups, and is able to reduce the number of decision variables in this optimization problem. A sensitivity analysis to the assumed VOT is conducted in Section 7.5.1. For the same reasons, in the constraints, we only track each individual vehicle group for the priority and normal streams. The network level constraints directly follow the conservation constraints from Eq.(6.5)-Eq.(6.9). At the local level, Eq.(7.8) and Eq.(7.9) follow the conservation constraints Eq.(6.10) and Eq.(6.11) and calculate the departure flow and queue length for the general streams, respectively. On the other hand, Eq.(7.10) and Eq.(7.11) calculate the departure flow and queue length for the priority and normal streams, respectively. Notice that in Eq.(7.10), in order to simplify the formulation, we assume that each group of vehicles is uniformly distributed in the queue. In the simulation, however, the vehicles follow a first-in-first-out rule. Eq.(7.12) sets the maximum queue length for each stream, where $X^i_{m,\max}$ is a given parameter related to the infrastructure constraints. Notice that we formulate the storage capacity as hard constraints. This can also be converted to soft constraints by introducing penalty terms in the objective function (e.g. Eq.(3.10) and Eq.(3.4)) in cases where hard constraints cannot lead to feasible solutions. Eq.(7.13) represents the allocated incoming flow for the priority and normal streams, derived from Eq.(7.5) and Eq.(7.6), where the lane choice probability is calculated in Eq.(7.14) using the utility function Eq.(7.2) and logistic function Eq.(7.1). Here, notations $\boldsymbol{x}^i_\xi(k+l|k)$ and $\boldsymbol{\mu}^i_\xi(k+l|k)$ represent vectors $[x^i_m(k+l|k)|m \in T^i \cup N^i, \phi(m) = \xi]$ and $[\mu^i_m(k+l|k)|m \in T^i \cup N^i, \phi(m) = \xi]$, respectively. Eq.(7.15) and Eq.(7.16) are the conservation of the departure flow for the streams associated with priority movements. The bounds on the green ratios and the initial conditions at the network level are calculated using Eq.(6.12)-(6.14). Eq.(7.17) and Eq.(7.18) are the initial conditions at the local level for links with and without priority lanes, respectively.

This optimization model is a non-linear model due to the network level constraints Eq.(6.5) and (6.9), the local level constraint Eq.(7.10), and especially the lane choice probability Eq.(7.14). Therefore, we develop an approximation framework in Section 7.2.3.

7.2.3 Approximation framework

Similar to Section 6.2.3, we assume a triangular MFD and perform the approximation by linearizing Eq.(6.5) and (6.9) as Eq.(6.23) and Eq.(6.24). However, the local level constraint Eq.(7.10) cannot be linearized in the same way, as the number of queued vehicles is not stabilized around a critical value. Instead, we eliminate the nonlinearity by assuming a constant percentage of queued vehicles given by the initial values. Denote the initial proportion of vehicle group j as ϵ^i_{mj}, then

$$\epsilon^i_{mj} = \frac{\hat{x}^i_{mj}}{\hat{x}^i_m}, \ \forall j \in J, \ \forall m \in T^i \cup N^i, \ \forall i \in I \tag{7.19}$$

where $\hat{x}^i_{mj}$ represents the initial values of the queue length for vehicle group j.

This can transform Eq.(7.10) to

$$\mu^i_{mj}(k+l|k) = \min\left\{x^i_{mj}(k+l|k)/C + q^i_{mj}(k+l|k), \sum_{p\in P^i} s^i_{mp} g^i_p(k+l|k)\epsilon^i_{mj}\right\},$$
$$\forall m \in T^i \cup N^i, \ \forall i \in I, \ \forall j \in J, \ \forall 0 \le l \le L-1 \tag{7.20}$$

Eq.(7.20) can work properly for streams where queues are carried over for a few cycles. For streams that are not very oversaturated, Eq.(7.20) might lead to large errors. In these cases, we can use an alternative relaxed formulation, i.e.

$$\mu^i_m(k+l|k) = \min\{x^i_m(k+l|k)/C + q^i_m(k+l|k), \sum_{p\in P^i} s^i_{mp} g^i_p(k+l|k)\Big\},$$
$$\forall m \in T^i \cup N^i, \ \forall i \in I, \ \forall 0 \le l \le L-1 \tag{7.21}$$

$$x^i_{mj}(k+l|k) \ge 0, \ \forall m \in T^i \cup N^i, \ \forall i \in I, \ \forall j \in J, \ \forall 0 \le l \le L-1 \tag{7.22}$$

where Eq.(7.21) and Eq.(7.22) relax the constraint Eq.(7.10), and therefore calculate an upper bound for the optimization problem.

Notice that in this perimeter control problem, streams associated with priority movements might have some overlapping phases, which violates Remark R2 in Section 6.2.3. Therefore, Eq.(7.20) and Eq.(7.21) cannot be further simplified in the same way as Eq.(6.25) and Eq.(6.26).

For the lane choice probability Eq.(7.14), we let the parameter $\omega \to \infty$ in Eq.(7.2). Recall this implies an all-or-nothing approximation for the lane choice behavior of the car riders, i.e. $z^i_{mj}(k+l|k) = 1$ or $z^i_{mj}(k+l|k) = 0$. This makes $z^i_{mj}(k+l|k)$ a binary decision variable. In addition, we conserve the monotonicity of $z^i_{mj}(k+l|k)$ by adding a constraint

$$z^i_{mj}(k+l|k) \geq z^i_{mj}(k+l|k),\ \forall j, j' \in J,\ \text{if } \sigma_j \geq \sigma_{j'},\ \forall m \in T^i \cup N^i,\ \forall i \in I \tag{7.23}$$

With the approximation mentioned above, the optimization model Eq.(7.7)-Eq.(7.18) can be simplified as a mixed integer linear programming (MILP) problem. The decision variables are the green ratios $g^i_p(k+l|k)$ and the binary decision variables $z^i_{mj}(k+l|k)$. This MILP can be solved efficiently using the standard branch and cut algorithm in CPLEX.

A toll $\tau^i(k)$ can be calculated from this binary decision variable $z^i_{mj}(k|k)$ as the smallest τ satisfying

$$\begin{aligned} &U_m(\tau, \sigma_j, \hat{x}^i_m(k), \mu^i_m(k|k)) > U_{m'}(\tau, \sigma_j, \hat{x}^i_{m'}(k), \mu^i_{m'}(k)), \\ &\qquad \forall j \in J,\ \forall m \in T^i,\ \forall m' \in N^i,\ \phi(m) = \phi(m'),\ \text{if } z_j(k|k) = 1 \end{aligned} \tag{7.24}$$

$$\begin{aligned} &U_m(\tau, \sigma_j, \hat{x}^i_m(k), \mu^i_m(k|k)) \leq U_{m'}(\tau, \sigma_j, \hat{x}^i_{m'}(k), \mu^i_{m'}(k|k)), \\ &\qquad \forall j \in J,\ \forall m \in T^i,\ \forall m' \in N^i,\ \phi(m) = \phi(m'),\ \text{if } z_j(k|k) = 0 \end{aligned} \tag{7.25}$$

where $\mu^i_m(k|k)$ represents the departure flow at the current cycle calculated using the decision variable $z^i_{mj}(k|k)$.

The left hand side of Eq.(7.24)-(7.25) represents the cost when taking the priority lanes, while the right hand side represents the cost when taking the normal lanes. Eq.(7.24)-(7.25) guarantee that the assignment obtained from the MPC is beneficial for all the car groups. We choose the lowest toll to ensure the benefits of the car riders.

7.2.4 Online recalibration for the VOT distribution

In this subsection, we use the information provided by connected vehicles to develop an online recalibration method for the VOT distribution for vehicles that wish to enter the city network. Although we assume that connected vehicles do not directly report their VOTs, the lane decision of connected vehicles reveals valuable information on their VOTs. Notice that in this subsection, we estimate a single VOT distribution for the entire network. The same method, nevertheless, can be applied to estimate the VOT distribution in a smaller scale (e.g. intersection, subnetwork, etc.). The estimation results serve as a crucial input to the model developed in Section 7.2.2.

The propose online estimation method is based on the recursive Bayesian filtering (Särkkä, 2013). Denote the VOT probability density function as a continuous function $f(\sigma|\boldsymbol{\zeta})$ characterized by parameter $\boldsymbol{\zeta}$, which can be a vector or scalar. The goal is to use a recursive formula to estimate $\boldsymbol{\zeta}$.

In the beginning of the MPC-based strategy, we assume a certain initial knowledge of $\boldsymbol{\zeta}$, represented as a probability density function $\mathcal{P}(\boldsymbol{\zeta})$ where $\mathcal{P}(\cdot)$ is the notation for probability. The estimated value of $\boldsymbol{\zeta}$ (usually the mean or the value with the maximum probability), i.e. $\boldsymbol{\zeta}^0$, serves as the initial input to the embedded optimization problem Eq.(7.7)-Eq.(7.18). The value and the distribution of $\boldsymbol{\zeta}^0$ can be estimated using prior information, or an educated guess. If the traffic operator is confident about the estimation of $\boldsymbol{\zeta}$, $\mathcal{P}(\boldsymbol{\zeta})$ can be assumed to have a small variance. Otherwise, $\mathcal{P}(\boldsymbol{\zeta})$ can be assumed to be uniformly distributed within a large region.

Given a connected car h that makes a lane decision, we can use this information to recalibrate the estimation of $\boldsymbol{\zeta}$. Denote the corresponding intersection as i^h and the cycle it makes the decision as k^h. Let y^h represent the stream it chooses.The probability of y^h can be calculated as

$$\mathcal{P}(y^h|\boldsymbol{\zeta}) = \int_\sigma \rho_{y^h}(\tau^{i^h}(k^h), \sigma, \hat{\boldsymbol{x}}_\xi)^{i^h}(k^h), \hat{\boldsymbol{\mu}}_\xi)^{i^h}(k^h)) f(\sigma|\boldsymbol{\zeta}) \mathrm{d}\sigma \tag{7.26}$$

We simplify the notation of function $\rho_{y^h}(\cdot)$ as $\rho_{y^h}(\sigma)$ and obtain Eq.(7.27) based on Bayes'

Theorem.

$$\mathcal{P}(\boldsymbol{\zeta}|y^h) = \frac{\mathcal{P}(\boldsymbol{\zeta}) \int_\sigma \rho_{y^h}(\sigma) f(\sigma|\boldsymbol{\zeta}) \mathrm{d}\sigma}{\int_{\boldsymbol{\zeta}} \mathcal{P}(\boldsymbol{\zeta}) \int_\sigma \rho_{y^h}(\sigma) f(\sigma|\boldsymbol{\zeta}) \mathrm{d}\sigma} \tag{7.27}$$

Then, by replacing $\boldsymbol{\zeta}$ with the conditional variable $\boldsymbol{\zeta}^{h-1} = \boldsymbol{\zeta}|y^1, \cdots, y^{h-1}$, we derive

$$\mathcal{P}(\boldsymbol{\zeta}^h) = \frac{\mathcal{P}(\boldsymbol{\zeta}^{h-1}) \int_\sigma \rho_{y^h}(\sigma) f(\sigma|\boldsymbol{\zeta}^{h-1}) \mathrm{d}\sigma}{\int_{\boldsymbol{\zeta}} \mathcal{P}(\boldsymbol{\zeta}) \int_\sigma \rho_{y^h}(\sigma) f(\sigma|\boldsymbol{\zeta}) \mathrm{d}\sigma} \tag{7.28}$$

where $\mathcal{P}(\boldsymbol{\zeta}^0)$ is the prior knowledge of $\boldsymbol{\zeta}$. The impact of $\mathcal{P}(\boldsymbol{\zeta}^0)$ would be small after observing a large number of samples.

Eq.(7.28) can be solved using Monte Carlo sampling (e.g. as a particle filter (Haykin, 2004)). After observing h samples, $\boldsymbol{\zeta}$ can be estimated as

$$\hat{\boldsymbol{\zeta}}^h = \mathrm{argmax} \mathcal{P}(\boldsymbol{\zeta}^h) \tag{7.29}$$

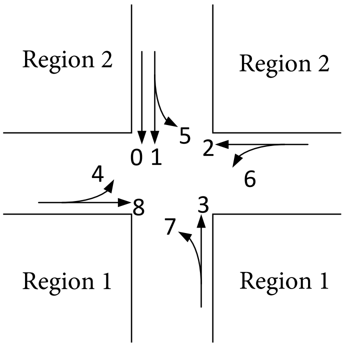
Using the estimated $\hat{\boldsymbol{\zeta}}^h$, we can obtain the VOT distribution as $f(\sigma|\hat{\boldsymbol{\zeta}}^h)$. σ_j can be obtained after discretization.

7.3 Simulation Settings

We simulate a scenario inspired by the city center of Zurich, Switzerland, as described in Chapter 6. The studied region is the same as in Figure 6.2(a), where a simple version of perimeter control is implemented (Ortigosa et al., 2014; Ambuhl et al., 2018). The MFD used in this simulation is derived from loop detector and floating car data in Zurich (Ambühl et al., 2016), shown in Figure 6.2(b). The demand profile mimics the morning commute when the majority of the demand comes from the suburban network to the center network. The total demand for entering the center network during a 1.5 hour period is shown in Figure 6.2(c). Note that this demand profile can generate a large amount of traffic. The resulting traffic accumulation without the application of any type of perimeter control reaches the oversaturated states on the MFD.

There are 20 intersections at the perimeter. The signal streams and phases are shown in Figure 7.2(a) and Figure 7.2(b) for priority intersections (10 intersections), respectively, and Figure 7.2(c) and Figure 7.2(d) for the non-priority intersections. The priority stream is Stream 0 in each priority intersection. Notice that we separate stream 0 and stream 1 for perimeter control rather than for the operations of intersections, as we might be restricting flow from these inflow streams. The cycle length is C =1min. The minimum green time is assumed to be $g^i_{p,\min} = 0.1$ for Phase 1-4, and $g^i_{p,\min}$ =0 for Phase 0. The maximum allowed total green time is $g^i_{\max}$ =0.9, implying a lost time of 6s. The storage capacity at the intersections are assumed to be $X^i_{m,\max} = 40$ vehicles per lane, i.e. around 300 meters. Sensitivity analysis is performed for the storage capacity in Section 7.5.2. Notice that the simulation settings in this chapter are slightly different from those of Chapter 6. Except for the definition of streams and phases, all the intersections are assumed to be identical (e.g. same link capacities). This is to reduce the number of factors that might influence the results. Chapter 6, however, aims to evaluate the benefits of adopting a multi-scale controller in general scenarios, and thus assumes heterogeneity across intersections. Also, due to the inclusion of storage capacity constraints, we consider lower saturation levels in the side streams to avoid violation of the constraints.

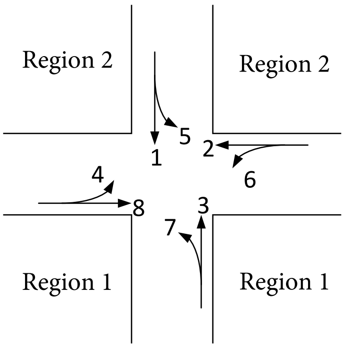
For the optimization model, we use a triangular MFD with α = 5hr^{-1}, β = 2.5hr^{-1} and n_{cr} = 3000veh to approximate the empirical MFD. The maximum flow of the network is 15000veh/hr. The initial traffic accumulation in the beginning of the simulation is set as 2000veh, below the



(a) Streams at priority intersections

Streams	Phases
0,3	0
0,1,3	1
5,7	2
2,4	3
6,8	4

(b) Phases at priority intersections



(c) Streams at non-priority intersections

Streams	Phases
1,3	1
5,7	2
2,4	3
6,8	4

(d) Phases at non-priority intersections

Figure 7.2: Layout of the intersections.

critical accumulation. The initial queue length at all the streams are assumed as 0. The control cycle is set as 1 minute and the moving time horizon includes $L = 20$ cycles. As in Chapter 6, it is assumed that there is a prediction noise of 10% and a measurement noise of 10%.

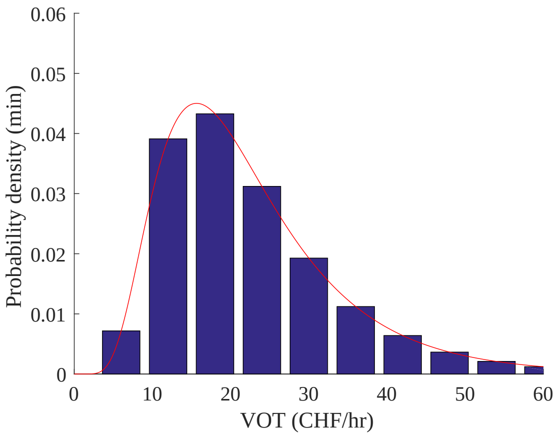


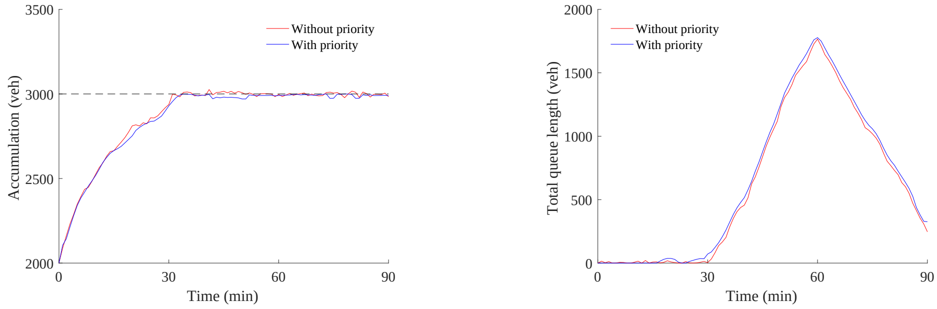
Figure 7.3: Distribution of car VOTs

Figure 7.3 shows the VOT distribution for the car demand. We assume that the VOT of each car follows a log-normal distribution, with the median VOT of the entire population being 16CHF, as an extension to Axhausen et al. (2007) considering passenger occupancies. Here, the parameters for the log-normal distribution are set as $\zeta = [3, 0.5]$. The assumed VOT $\bar{\sigma}$ is 22CHF/hr, which is close to mean of the VOT distribution (22.7CHF). Sensitivity analysis is conducted for $\bar{\sigma}$ in Section 7.2.4. It is assumed that the internal demand D_{ab} and the entering demand λ^i_{mj} follow the above distribution. The assumed distribution is recalibrated in real-time using the Bayesian filter developed in Section 7.2.4. Drivers are expected to choose lane following the simple logistic function Eq.(7.1)with utility as in Eq.(7.2) using parameter $\tilde{\omega} = 2$, whereas in the control, we assume $\omega = \infty$. Sensitivity analysis is also performed for ω in Section 7.5.3.

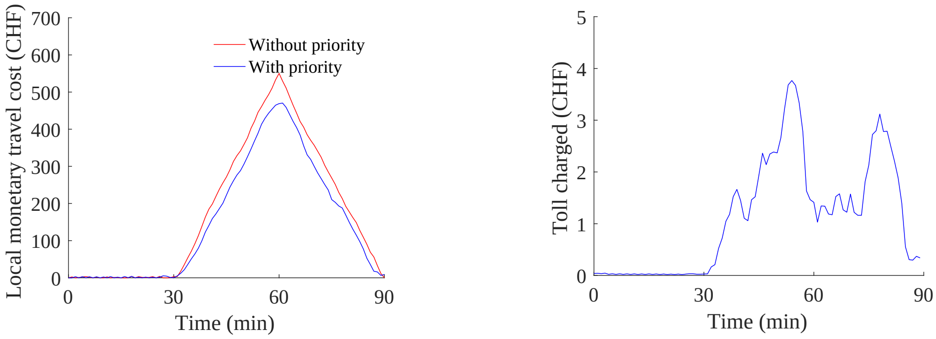
7.4 Case Study and Results

7.4.1 Overall performance

We evaluate the performance of the proposed strategy and show the value of dedicating priority lanes by comparing two strategies: 1) the proposed strategy presented in Eq.(7.7)-(7.18), i.e. with priority, and 2) the multi-scale MPC strategy developed in Chapter 6, i.e. without priority.



(a) Network accumulation (b) Total queue length at perimeter intersections



(c) Monetary cost at perimeter intersections (d) Toll charged at priority lanes

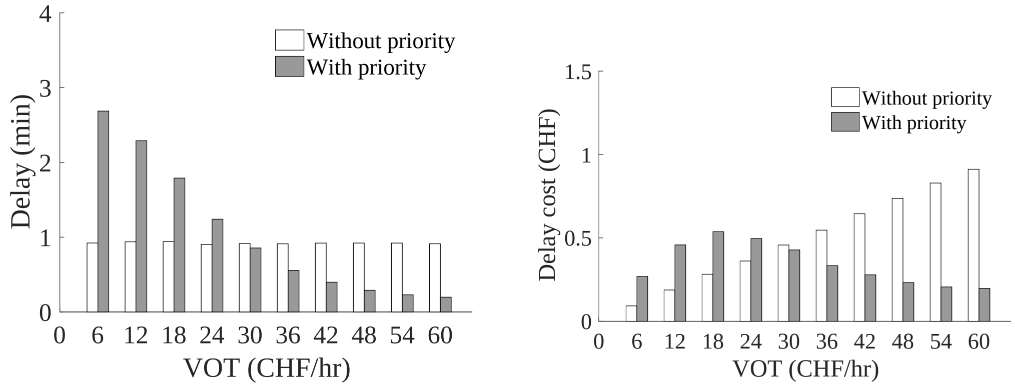
Figure 7.4: Performance of the algorithm.

Figure 7.4 shows the performance of the proposed controller and the multi-scale controller at both the network and local levels. At the network level (Figure 7.4(a)), both controllers successfully stabilize the traffic accumulation around the critical accumulation, and yield similar queue profiles (Figure 7.4(b)). The total travel costs are also very similar (both 96.6k CHF). This is reasonable, since both approaches ensure that the center network operates at its optimal state.

However, implementing the priority lanes improves the values of the passengers at the perimeter intersections. We can also observe from Figure 7.4(c) that the proposed controller improves the local performance throughout the entire simulation period. The total cost using the multi-scale controller is 20.8k CHF, while the total cost using the proposed controller is 18.4k CHF (a saving of 11.6%). This shows that it is beneficial to provide priority to the high-valued vehicles at the perimeter. This improvement is rather promising, as only half of the intersections are installed with priority lanes.

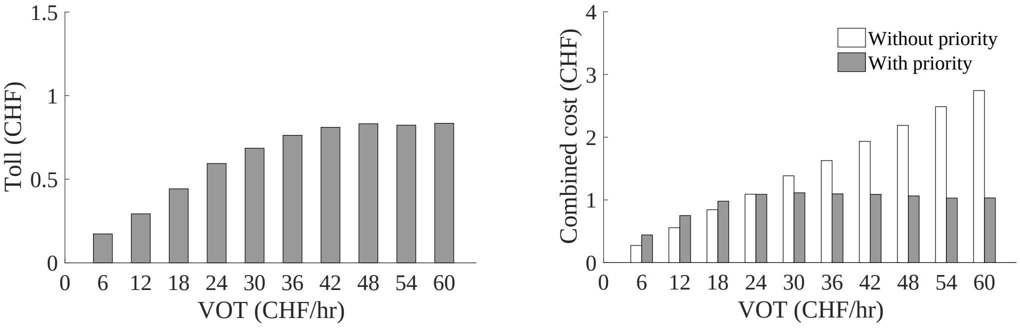
The average toll charged at priority lanes is shown in Figure 7.4(d). Notice that this toll is average across all the intersections. We can see that the toll is almost zero before the system reaches oversaturated states (before 30 min). Note that the highest toll is not charged at the time with the longest queue (60 min). This is because as the system becomes very congested, in order to avoid spillbacks, the proposed strategy will attract more vehicles to the priority lanes by reducing the toll. This shows that the proposed strategy is able to handle the infrastructure constraints while providing priority.

7.4.2 Performance of individual groups of cars



(a) Average delay

(b) Average cost due to delay



(c) Average toll

(d) Average combined cost

Figure 7.5: Performance of individual group of cars in the priority movement

Figure 7.5 shows the performance of individual group of cars in the priority movement. For each vehicle group, we calculate the average delay (cost) across the entire simulation period. We can see from Figure 7.5(a) that the proposed strategy successfully reduces delay for cars with higher values (VOT > 30CHF/hr), but increases the delay for cars with lower values (VOT < 30CHF/hr). This is because priority is given to the high valued vehicles taking the priority lanes. Figure 7.5(a) also shows that the average delay without priority is constant irrelevant of VOTs, as there is no differentiation across groups of cars. On the contrary, the average delay with priority in general decreases with the VOT. This is because we do not provide priority to certain pre-defined groups of cars, but determine the priority according to the real time traffic states. The chance of being prioritized is higher for high valued cars than for the low valued cars. Note that the average delay, though very low, is not necessarily 0 even for high-valued cars. This shows that the proposed strategy allows the cars in the priority lanes to be delayed as well. In other words, we still implement some control to discharge the cars on the priority lanes, and we may hold them if this benefits the system.

Figure 7.5(b) shows the cost due to delay suffered by the cars entering the center network. For the multi-scale strategy, the resulting cost increases almost linearly with the increase of VOT.

This is because the average delay for all the user groups is almost the same. For the proposed strategy, the delay increases if the VOT is smaller and decreases if the VOT is larger.

Figure 7.5(c) shows the average toll each group of cars pay at the perimeter, calculated as the average toll across all the cars in a group, across all the intersections, and across the entire simulation period. Notice that in each cycle, we charge the same toll for the cars using the priority lanes irrespective of their VOT. However, as the toll is dynamic (see Figure 7.4(d)), the percentage of cars using the priority lanes in each vehicle group is dynamic as well. Therefore, cars in the same group may pay different toll if they arrive at the perimeter at different times. The average toll increases with the VOT, as the high valued users are willing to pay more and get priority more often. We can also see that the marginal increase of the toll decreases as the VOT increases. This is because the high valued vehicles mostly take the priority lanes, meaning that they pay the same toll in most cycles.

Figure 7.5(d) demonstrates the combined costs of each group, including the costs due to delay and the tolls. Comparing the results of the two approaches, we observe that by implementing the express toll lanes, cars with a VOT larger than 24CHF/hr benefit significantly, and these benefits increase with the VOT level. Cars with lower VOTs end up slightly worse-off, as the delay of these cars at the perimeter increases. Overall, the combined costs are more evenly distributed among groups of cars than the resulting costs from the multi-scale controller.

7.4.3 Value of the recalibration algorithm

In this subsection, we evaluate the recalibration algorithm, which updates $\boldsymbol{\zeta} = [\zeta_1, \zeta_2]$ in real time using the lane decision information of connected vehicles, where ζ_1 and ζ_2 are the two parameters for the log-normal distribution. Here, we assume that there are adequate number of connected vehicles (100%). We assume that the initial $\hat{\boldsymbol{\zeta}}^0$ to be in the region $\{(\zeta_1, \zeta_2) | \zeta_1 \in [2, 4],\ \zeta_2 \in [0.4, 0.6]\}$. Notice that although the errors in the parameter $\hat{\boldsymbol{\zeta}}^0$ may not seem to be large, their impact on the VOTs, however, is exponential due to the assumption of log-normal distribution. We assume that the prior distribution of parameter $\hat{\boldsymbol{\zeta}}$ is a Gaussian distribution with the mean $\hat{\boldsymbol{\zeta}}^0$. ζ_1^0 and ζ_2^0 are assumed to be mutually independent, both with a standard deviation of 1/3 of their mean to avoid being negative. Figure 7.6 shows the delay savings at all perimeter intersections when starting from different $\hat{\boldsymbol{\zeta}}^0$. Recall that the true $\boldsymbol{\zeta} = [3, 0.5]$. Notice that the network performance is similar for all scenarios tested.

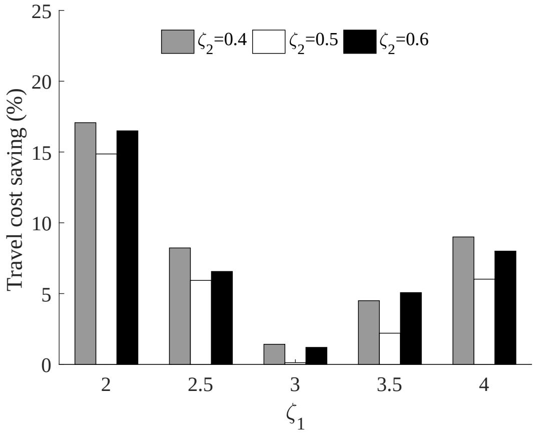


Figure 7.6: Delay savings of the recalibration algorithm at the perimeter intersections. Penetration rate of connected vehicles is 100%.

Figure 7.6 shows the cost savings of the proposed strategy with recalibration compared to that without recalibration at the local level. It can be seen that the cost savings are positive in all scenarios tested. This shows that the proposed recalibration algorithm has successfully improved the performance at the perimeter intersections. It can also be seen that the cost saving is larger if the initial $\hat{\zeta}^0$ deviates more from the true value of ζ. This is reasonable, as there is more room for improvement. This also shows that the accurate estimation of the VOT distribution is important for the operations of the priority lanes.

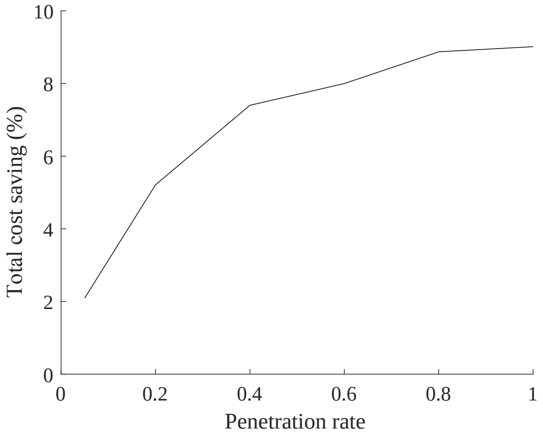


Figure 7.7: Value of information provided by connected vehicles, $\zeta^0 = [4, 0.4]$.

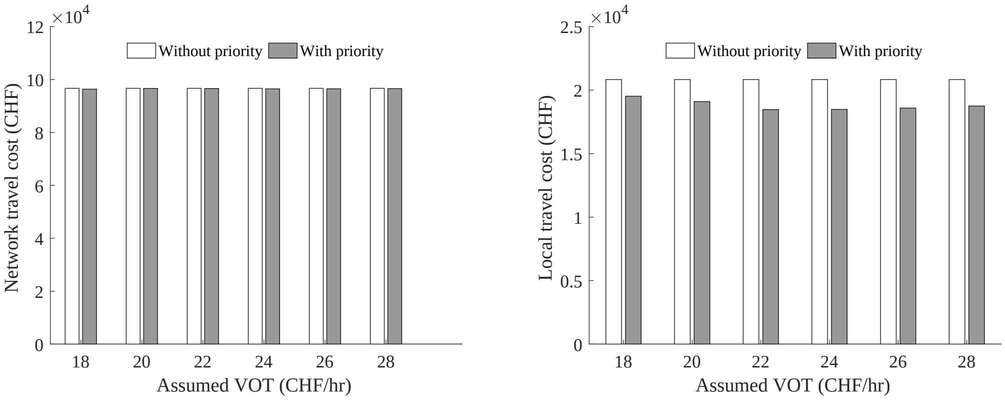
The recalibration algorithm relies on the information of connected vehicles. Unlike in queue estimation (e.g. in Chapter 3 where the queue length evolves with time, in this chapter, the VOT distribution is assumed to be invariant. Therefore, given sufficient number of lane decisions reported by the connected vehicles, the estimation results of the recalibration algorithm will eventually converge to the true parameters of the VOT distribution. However, the penetration rate of the connected vehicles influences the time it takes for the recalibration algorithm to converge. Figure 7.7 illustrates the impact of the penetration rates on the performance of the recalibration algorithm during the simulation period (90min). Initially, ζ^0 is set to be $[4, 0.4]$. We can see from Figure 7.7 that the cost savings of the proposed strategy with recalibration compared to that without recalibration are always positive and monotonously increasing. This shows that the recalibration algorithm can successfully utilize the information provided by connected vehicles. It can also be seen that the marginal benefit of having a higher penetration rate decreases as the penetration rate increases. This is because for higher penetration rates (i.e. larger than 0.6), the recalibration algorithm is able to converge before the simulation ends. Hence, their performance is similar to the scenario where perfect information about the VOT distribution is assumed.

7.5 Sensitivity Analysis

In this section, we perform a sensitivity analysis to validate some treatment in the modeling and solution of the proposed strategy. Section 7.5.1 evaluates the sensitivity to the assumed VOT in the objective function Eq.(7.7). Section 7.5.2 investigates the impact of the links storage capacity. Section 7.5.3 validates the all-or-nothing approximation of the logistic function.

7.5.1 Sensitivity to the assumed VOT

In the objective function Eq.(7.7), we do not track each individual vehicle group inside the network nor in the non-priority movements at the perimeter intersections. The main reason is that the VOT distribution for these vehicles is not readily available. Instead, we use the assumed VOT $\bar{\sigma}$ to calculate the social welfare for these vehicles. In this subsection, we evaluate the impact of the assumed VOT on both the network and local level performance with σ varying between



(a) Network travel cost

(b) Local travel cost

Figure 7.8: Sensitivity to the assumed VOT $\bar{\sigma}$.

18 and 28. Recall that the mean of the VOT distribution is 22.7. The results are summarized in Figure 7.8.

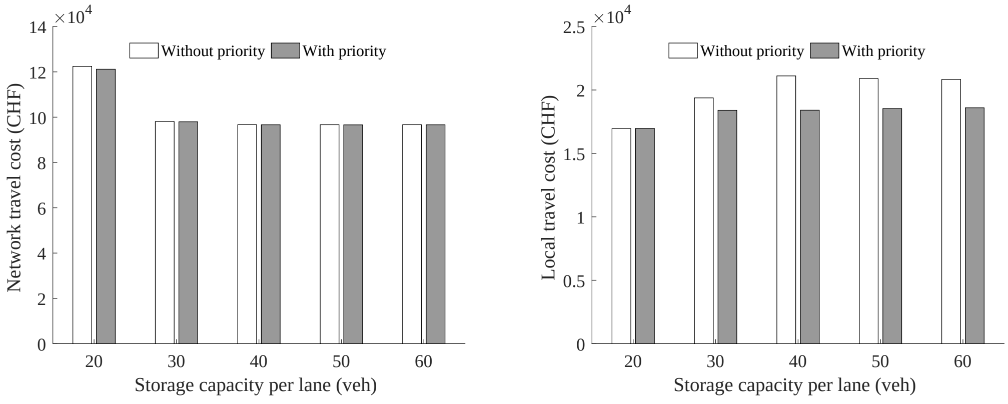
The performance at the network level is illustrated in Figure 7.8(a). We can see that the network travel cost is not sensitive to the assumed VOT. This is true for both strategies with or without priority. One potential reason is that it is typically beneficial to stabilize the accumulation inside the network, for both the vehicles inside the network and vehicles wishing to enter. Therefore, the network level performance does not rely on the assumed VOT.

On the other hand, the performance at the local level, as demonstrated in Figure 7.8(b), is sensitive to the assumed VOT. If the assumed VOT is below certain value (22CHF/hr), the local level performance deteriorates. However, the local level travel cost does not change much if the assumed VOT is larger than this value. One explanation is that if the assumed VOT is too low, the general streams (i.e. streams associated with non-priority movements) at the intersections will be sacrificed in the optimization, as they are given a lower weight. However, higher assumed VOT will not have a significant impact on the local performance, as the streams associated with the priority movements are typically inflow to the network, and therefore coupled with the network performance. Hence, the assumed VOT can be set to be slightly higher than the mean VOT calculated from the VOT distribution without any penalties in terms of system performance.

7.5.2 Sensitivity to storage capacity at the perimeter intersections

Perimeter control might lead to long queues at the perimeter intersections. However, due to the physical constraints of the infrastructure, the perimeter intersections may not be able to hold a large number of vehicles. Especially if the priority lanes are dedicated at some intersections, there could be a higher chance of spillbacks. In the optimization problem, we include the links storage capacity at a constraint to prevent spillbacks (i.e. Eq.(7.12)). In this subsection, we investigate how storage capacity influences the performance at the center network and the perimeter intersections. The storage capacity is assumed to vary between 20veh/lane and 60veh/lane, representing a queue of 150-500m.

It can be seen from Figure 7.9(a) that the network travel cost is sensitive to the storage capacity X_m^i if is smaller than a certain threshold value (e.g. $X_m^i \leq 20$veh). This is expected, because the storage capacity is set as a hard constraint in the optimization problem . Therefore, if the links storage capacity is too small, the intersections will not be able to hold sufficient incoming vehicles. These incoming vehicles will be allowed to enter the network, resulting in the increase of the network accumulation. This can also be reflected by Figure 7.9(b) where the local travel cost is very low for smaller storage capacity. This, however, may not be optimal for the system.



(a) Network travel cost

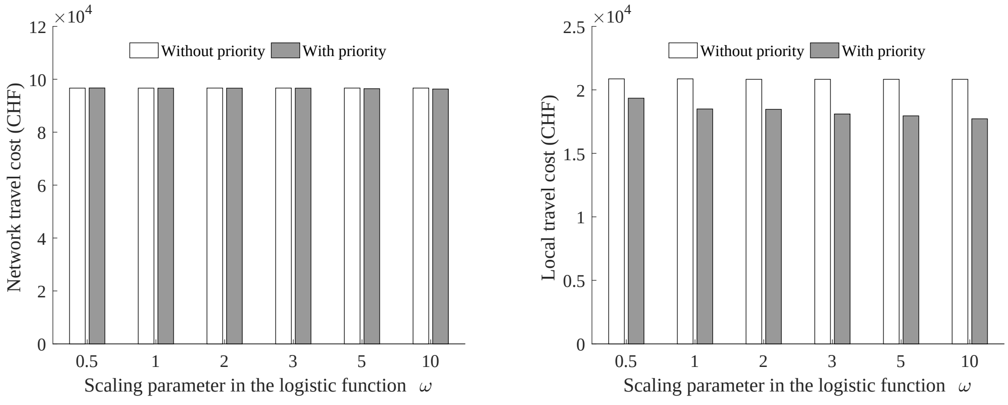
(b) Local travel cost

Figure 7.9: Sensitivity to storage capacity at the perimeter intersections $X^i_{m,\max}$

In this scenario, the storage capacity can be introduced to the optimization problem as a soft constraint to balance the penalty at both levels. If the storage capacity is larger than the threshold value (e.g. $X^i_m > 40$veh), both the network and local level performance become not sensitive to the storage capacity. In such scenario, the storage capacity will influence the distribution of queue length across intersections and streams, but will not impact the overall performance in any significant manner. It can also be seen that although the proposed strategy and the multi-scale strategy follow the same trend, the proposed strategy always perform better than the multi-scale strategy.

7.5.3 Sensitivity to scaling parameter in the logistic function

In this subsection, we investigate the sensitivity to the scaling parameter in the logistic function. The goal of this sensitivity analysis is to validate the all-or-nothing approximation of the logistic function, described in Section 7.2.3 as part of the approximation framework for a fast solution. Recall that in the approximation framework, we set the scaling parameter $\omega \to \infty$, but in the simulation, we set the scaling parameter as $\omega \neq \infty$.



(a) Network travel cost

(b) Local travel cost

Figure 7.10: Sensitivity to scaling parameter ω in the logistic function.

We assume that this parameter ω ranges between 0.5 and 10. The results are summarized in Figure 7.10. Note that the resulting travel costs for the multi-scale strategy are invariant to this scaling parameter, because priority lanes are not provided and vehicles are not differentiated. Here, the results for the multi-scale strategy can be used as a benchmark to evaluate the sensitivity.

We can see from Figure 7.10(a) that the network level travel costs are not sensitive to the scaling parameter. This is because the network performance is always guaranteed by both controllers. At the local level, the performance is slightly sensitive to the choice parameter ω. The local travel cost decreases as ω increases, as large ω produces choice probabilities more similar to the all-or-nothing approximation. The cost savings of the proposed strategy compared to the multi-scale strategy is 7% if $\omega = 0.5$, and can be up to 15% if $w = 10$. Nevertheless, the proposed controller always yields satisfactory cost savings compared to the multi-scale controller.

7.6 Conclusion and Future Work

In this chapter, we extend the multi-scale perimeter control strategy proposed in Chapter 6 by incorporating a priority scheme to enable fast service to certain groups of vehicles. This is achieved by dedicating priority lanes at the perimeter intersections with more favorable signal timings and subject to a dynamic toll. The signal timings at the perimeter intersections and the toll for using the priority lanes are determined dynamically using an MPC controller based on the distribution of VOT. Results show that the proposed strategy successfully stabilizes the center network and yields similar better social welfare than the multi-scale controller proposed in Chapter 6. The benefits of the proposed controller, however, lie in the local perimeter intersections, improving the performance of the intersection by up to 11.6%. It is also demonstrated that the proposed controller is able to handle the storage capacity constraints. It will reduce the toll in very oversaturated scenarios to avoid the spillbacks in particular lanes. Results further show that the proposed approach leads to a more even distribution of costs (tolls and delay costs) across vehicle groups.

We further develop an online recalibration algorithm using the information provided by connected vehicles to obtain more accurate VOT distribution. The knowledge of the VOT distribution is updated via a Bayesian filter if a connected vehicle makes a decision on whether or not to take the priority lanes. It can be seen from the results that the VOT recalibration algorithm successfully improves the performance of the algorithm if there is not sufficient knowledge of the VOT distribution.

Sensitivity analyses are performed to evaluate the robustness of the proposed strategy. It is shown that overall performance of the proposed strategy is not sensitive to the assumed VOT for vehicles inside the network and in non-priority movements. Results further show that the storage capacity, if not too small, will have marginal impact on the traffic performance at both levels. The proposed strategy can also yield satisfactory performance in various scenarios with different lane choice behaviors. This shows that the all-or-nothing approximation framework does not have significant impact on the control performance.

The proposed strategy, to the best of our knowledge, is the first attempt to combine perimeter control and pricing. By adopting such strategy, cars have the option to pay for a fast service, but can also queue to enter the network for free. This strategy does not only apply to the private cars, but can also be extended to emergency vehicles, such as ambulances, if they are allowed to take the priority lanes without paying tolls. The proposed strategy can also help enhance mobility for shared vehicles and public transport vehicles (e.g. He et al., 2018b; Dakic et al., 2019), as the occupancies on these vehicles are high, thus the resulting VOTs are large. A future research direction includes the consideration of spatial heterogeneity of cars (e.g. cars in certain region may generally have a higher VOT). Another potential direction is to include a bidding scheme to use the priority lanes (e.g. using the methodology developed in Yang et al. (2018c)). Such scheme would also provide a fair assignment while yielding information on the VOT distribution.

Chapter 8

Conclusions and Outlook

8.1 Summary

This doctoral research is devoted to develop novel traffic estimation and control strategies at both the network and local levels in a connected and automated vehicle environment, with a particular focus on the transition period with various levels of technology provision. Efforts are therefore made to handle the challenges associated with the transition period and the large-scale multi-modal urban traffic systems. The main contributions and findings of this dissertation are summarized below.

1. Handling the challenges due to the low penetration rates in the transition period

The first main contribution of the dissertation is the development of algorithms to properly handle the challenges due to the low penetration rates of connected and automated vehicles during the transition period. Low penetration rates of the technology result in an uncertain environment due to limited information. As we assume connected and automated vehicles as the only information source, the existence of the conventional vehicles can only be deduced using this information. In order to address this issue, we conduct research in the following three categories.

First, we propose accurate traffic estimation methods to better utilize the information provided by connected and automated vehicles (Part I, Chapter 3). We pay particular attention to queue estimation (trajectory reconstruction), which provides a holistic picture of the urban traffic systems and serves as an essential input to multiple traffic control strategies. We substantially enhance the existing works in two directions: i) relaxing the widely adopted assumption of constant demand in a signal cycle via a computationally efficient convex optimization based methodology, and ii) reusing the estimation results from the upstream intersections by exploiting the relation between the departure flow at the upstream intersections and the arrival flow at the downstream intersection. Simulation has shown that both directions yield significant improvement in the estimation accuracy. Moreover, despite the fact that we consider a more complex problem, the proposed algorithm is based on a convex optimization, and thus ensures computational efficiency. We further propose two online implementation approaches that have proven to be solvable within a reasonable time, sufficient for real-time applications.

Second, we develop robust control strategies against the traffic estimation errors. Chapter 6 in Part III is the first work that applies connected vehicle technology to the macroscopic control of large-scale urban networks and resolves the challenging issue of low penetration rate of this technology. We employ a stochastic MPC to explicitly handle the uncertainties caused by low penetration rates of connected and automated vehicles without yielding too conservative control decisions. It is shown that such strategy can significantly reduce the total travel costs if the uncertainty is large.

Third, we provide a guideline for how the control strategies should evolve with the penetration rates of connected and automated vehicles (Part II, Chapter 4). Based on the simulation results, we develop a heuristic to switch between the different features of the proposed control strategy, according to the demand and penetration rate of each technology.

2. Developing a control framework for both the network and local levels and a multi-scale control scheme to bridge them

The second main contribution of this dissertation is the development of a novel multi-scale control framework for large-scale urban traffic networks. At the local level, we proposed a joint optimization scheme to integrate the traffic signal control with the trajectory planning of automated vehicles (Part II, Chapter 4). At the network level, we focus on the promising perimeter control scheme for large-scale urban networks. We bridge the control of the two levels by introducing a multi-scale perimeter control strategy, with a particular focus on i) how the macroscopic control decisions can be translated into microscopic variables; and ii) how the control objectives at different levels can be synthesized (Part III, Chapter 6). This is the first methodological effort to integrate the individual control of the perimeter intersections and the overall perimeter control of large-scale urban networks. An approximation framework is proposed to enable the feasible and efficient solution of the embedded optimization problem for real-time control. Simulation results show that the proposed multi-scale perimeter control has optimized both the network performance and the intersection delay at the perimeter, outperforming the classical controllers.

3. Integrating priority schemes into the control strategies

The third main contribution of this dissertation is the explicit inclusion of priority schemes into the proposed control strategies. This dissertation differentiates different groups of vehicles (i.e. vehicles corresponding to different transportation modes, or with different occupancies, values of time, priority levels, etc.) and considers their interactions.

Chapter 5 (Part II) integrates transit signal priority into the control strategy proposed in Chapter 4 in order to minimize person delay. The traffic signal is coordinated with both bus infrastructure (bus stops) and bus operation (bus schedule). Simulation results show that it is valuable to consider bus stops, especially the near-side bus stops, when performing signal control. This further provides guidelines on which signal control strategies to be implemented for different scenarios.

Chapter 7 (Part III) extends the multi-scale perimeter control strategy proposed in Chapter 6 by integrating a priority scheme to improve both the performance and the social welfare of all vehicles. This is achieved by dedicating tolled priority lanes for vehicles with high VOTs. An MPC-based control strategy is proposed to dynamically adjust the toll to account for varying traffic conditions. This is the first work that combines perimeter control and congestion pricing. We further propose an online updating algorithm to estimate and recalibrate the distribution of VOT based on the choice of users. Simulation results show 11.6% increase in social welfare by applying the proposed method.

8.2 Limitations

This dissertation investigates how the emerging technologies can be beneficial for the multimodal urban transportation systems. Efficient traffic estimation and control strategies are proposed and validated using real-world inspired case studies. Nevertheless, there are a few limitations that require further investigation.

1) In Part I, we require that there are at least one BoQ critical point and one FoQ critical point within a signal cycle for the proposed method to work. In scenarios with very sparse data (i.e. very low penetration rates or sampling rates), the proposed method might not be able to detect queues. It is essential to incorporate probabilistic models into the proposed methodology in the future work.
2) In Part II, we consider simple intersections as an initial building block to design signal control algorithms. For more complex intersections, few difficulties might arise, including 1)

the planning of the turning and lane changing trajectories in scenarios where conventional, connected, and automated vehicles coexist; 2) computational costs, which could potentially be addressed by generalizing branch and bound algorithms or by using approximation algorithms (e.g. Li and Ban (2018)); and 3) multiple priority requests from vehicles with different modes and competing movements.

3) In Part III, there are mainly two limitations. First, we neglect the possible reroute when queue length at a particular intersection is large. This can be important, especially in the scenarios where automated vehicles exist in the traffic systems. Second, perimeter control may not be able to handle the scenarios with very high demand. One potential solution is to integrate incentives so that passengers can switch their transportation mode or departure time.

8.3 Outlook

Traffic control in a connected and automated vehicle environment is a relatively new and promising research area. Given the complexity in the traffic system, there are many directions to extend and explore. Specifically, potential future research directions are summarized as follows

1) **Comprehensive real-world test.** In this dissertation, the proposed traffic estimation and control strategies have been mainly evaluated with the aid of simulation and limited field data (such as NGSIM data). A possible future research direction is to conduct more comprehensive testing and validation of the methods in a real-world testbed and/or using real data from connected and automated vehicles.
2) **Data fusion algorithms for traffic estimation.** Throughout the dissertation, we assume that the connected and automated vehicles are the only information source. This is a conservative assumption that provides a lower bound for the benefit of these technologies. In the future smart mobility systems, there will be multiple sensors installed in a city (Jiang et al., 2017). Therefore, further efforts can be made to investigate how the information provided by these sensors can be integrated and synthesized to yield a more accurate traffic estimation. This would also be beneficial for the transition period of the technologies.
3) **Control of more complex intersections.** In this dissertation, we focus on simple intersections (Chapter 4 and Chapter 5). Future research can be devoted to extending the proposed strategies to complex intersections with multiple signal phases, multiple lanes, and multiple movements. The difficulty lies in the trajectory planning of turning trajectories and the lane changing in a scenario where vehicles with various technological levels coexist in the traffic streams.
4) **Traffic control with special infrastructure.** In Part II and Part III of this dissertation, we design control strategies to maximize the benefits of connected and automated vehicles during the transition period. Another approach is through dedicated infrastructure (dedicated lanes or dedicated zones). However, it is still not clear how the dedicated infrastructure can be managed with the existing traffic systems. Efforts are currently underway to propose an aggregated control strategy for dedicated zones with automated vehicles and to develop local control algorithms with separated infrastructure (lanes) for automated vehicles (Yang et al., 2018a, 2019).
5) **Combination with other systems.** In this dissertation, we only focus on the estimation and control of signalized intersections in urban cities. Future research can be devoted to integrate the developed framework in this dissertation with some of the authors' works in other systems, e.g. parking (Cao and Menendez, 2018), logistics (Yang et al., 2018c), and highways (He et al., 2017; Tilg et al., 2018).
6) **Integration of new mobility concepts.** In this dissertation, we consider the flexibility of

automated vehicles in traffic flow. However, automated vehicles are often coupled with new mobility services, such as flexible public transport (Nourbakhsh and Ouyang, 2012; Ruch et al., 2018), car-sharing (Becker et al., 2017), and ride-sharing services (Daganzo and Ouyang, 2018). One example is the Autonomous Mobility on-demand systems (e.g. Hyland and Mahmassani, 2018), which provide remarkable flexibility and convenience to travelers and present enormous potential to mitigate traffic congestion. Existing works on the Autonomous Mobility on-demand systems typically assume a congestion-free urban area and known travel times, or rely on steady-state formulations. Research on the impact of the Autonomous Mobility on-demand systems on the traffic systems is often simulation-based. Moreover, to the best of our knowledge, there are few existing methodological works on the integration between public transportation and the the Autonomous Mobility on-demand systems. Designing traffic control strategies considering new mobility concepts such as the Autonomous Mobility on-demand systems is essential for the operations of future multi-modal mobility systems.

7) **Cyber-secure traffic control** It is assumed in this dissertation and most existing works that the information provided by the connected and automated vehicles is faithful. However, it is possible that these vehicles are hacked, as they are computers. Owners of the vehicles might also forge the information for their own benefits. It would be an interesting yet practical direction to design cyber-secure traffic control strategies in a connected and automated vehicle environment against potential attacks.

Bibliography

Aboudolas, K. and N. Geroliminis (2013) Perimeter and boundary flow control in multi-reservoir heterogeneous networks, *Transportation Research Part B: Methodological*, **55**, 265–281.

Alexiadis, V., J. Colyar, J. Halkias, R. Hranac and G. McHale (2004) The next generation simulation program, *Institute of Transportation Engineers. ITE Journal*, **74** (8) 22.

Allsop, R. E. (1971) Delay-minimizing settings for fixed-time traffic signals at a single road junction, *IMA Journal of Applied Mathematics*, **8** (2) 164–185.

Ambühl, L., A. Loder, M. C. Bliemer, M. Menendez and K. W. Axhausen (2018) Introducing a re-sampling methodology for the estimation of empirical macroscopic fundamental diagrams, paper presented at the *97th Annual Meeting of the Transportation Research Board (TRB 2018)*.

Ambühl, L., A. Loder, M. Menendez and K. W. Axhausen (2016) Empirical macroscopic fundamental diagrams: New insights from loop detector and floating car data, *Transportation Research Record*, **17**.

Ambuhl, L., A. Loder, M. Menendez and K. W. Axhausen (2018) A case study of zurich's two-layered perimeter control. 7th Transport Research Arena (TRA 2018); Conference Location: Vienna, Austria; Conference Date: April 16-19, 2018.

Ambühl, L. and M. Menendez (2016) Data fusion algorithm for macroscopic fundamental diagram estimation, *Transportation Research Part C: Emerging Technologies*, **71**, 184–197.

Amini, Z., R. Pedarsani, A. Skabardonis and P. Varaiya (2016) Queue-length estimation using real-time traffic data, paper presented at the *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, 1476–1481.

Ampountolas, K., N. Zheng and N. Geroliminis (2014) Perimeter flow control of bi-modal urban road networks: A robust feedback control approach, paper presented at the *Control Conference (ECC), 2014 European*, 2569–2574.

Ampountolas, K., N. Zheng and N. Geroliminis (2017) Macroscopic modelling and robust control of bi-modal multi-region urban road networks, *Transportation Research Part B: Methodological*, **104**, 616–637.

Andrews, S. and M. Cops (2009) Final report: Vehicle infrastructure integration proof of concept executive summary–vehicle, *US DOT, IntelliDrive (sm) Report FHWA-JPO-09-003*.

Arnet, K., S. I. Guler and M. Menendez (2015) Effects of multimodal operations on urban roadways, *Transportation Research Record: Journal of the Transportation Research Board*, (2533) 1–7.

Åström, K. J. and T. Hägglund (2006) *Advanced PID control*, ISA-The Instrumentation, Systems and Automation Society.

Au, T.-C. and P. Stone (2010) Motion planning algorithms for autonomous intersection management, paper presented at the *Bridging the Gap Between Task and Motion Planning*.

Axhausen, K., S. Hess, A. König, G. Abay, J. Bates and M. Bierlaire (2007) State of the art estimates of the swiss value of travel time savings, paper presented at the 86th annual meeting of the transportation research board.

Baiocchi, A., F. Cuomo, M. D. Felice and G. Fusco (2015) Vehicular ad-hoc networks sampling protocols for traffic monitoring and incident detection in intelligent transportation systems, *Transportation Research Part C: Emerging Technologies*, **56** (0) 177 – 194.

Bakıcı, T., E. Almirall and J. Wareham (2013) A smart city initiative: the case of barcelona, *Journal of the Knowledge Economy*, **4** (2) 135–148.

Balke, K. N., H. A. Charara and R. Parker (2005) Development of a traffic signal performance measurement system (tspms), *Technical Report*, Texas Transportation Institute, Texas A & M University System.

Ban, X., R. Herring, P. Hao and A. Bayen (2009) Delay pattern estimation for signalized intersections using sampled travel times, *Transportation Research Record: Journal of the Transportation Research Board*, (2130) 109–119.

Ban, X. J., P. Hao and Z. Sun (2011) Real time queue length estimation for signalized intersections using travel times from mobile sensors, *Transportation Research Part C: Emerging Technologies*, **19** (6) 1133–1156.

Batty, M., K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis and Y. Portugali (2012) Smart cities of the future, *The European Physical Journal Special Topics*, **214** (1) 481–518.

Becker, H., F. Ciari and K. W. Axhausen (2017) Modeling free-floating car-sharing use in switzerland: A spatial regression and conditional logit approach, *Transportation Research Part C: Emerging Technologies*, **81**, 286–299.

Bellman, R., I. Glicksberg and O. Gross (1956) On the" bang-bang" control problem, *Quarterly of Applied Mathematics*, 11–18.

Bösch, P. M., F. Becker, H. Becker and K. W. Axhausen (2018) Cost-based analysis of autonomous mobility services, *Transport Policy*, **64**, 76–91.

Candes, E. and J. Romberg (2005) l_1 − norm magic: recovery of sparse signals via convex programming, *Technical Report*, Techical Report, Caltech.

Cao, J. and M. Menendez (2018) Quantification of potential cruising time savings through intelligent parking services, *Transportation Research Part A: Policy and Practice*, **116**, 151–165.

Caron, F., E. Duflos, D. Pomorski and P. Vanheeghe (2006) Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects, *Information fusion*, **7** (2) 221–230.

Chang, T.-H. and J.-T. Lin (2000) Optimal signal timing for an oversaturated intersection, *Transportation Research Part B: Methodological*, **34** (6) 471 – 491.

Chen, X., L. Li and Y. Zhang (2010) A markov model for headway/spacing distribution of road traffic, *Intelligent Transportation Systems, IEEE Transactions on*, **11** (4) 773–785.

Cheng, Y., X. Qin, J. Jin and B. Ran (2012) An exploratory shockwave approach to estimating queue length using probe trajectories, *Journal of Intelligent Transportation Systems*, **16** (1) 12–23.

Chiabaut, N. (2015) Evaluation of a multimodal urban arterial: The passenger macroscopic fundamental diagram, *Transportation Research Part B: Methodological*, **81**, 410–420.

Chiabaut, N., C. Buisson and L. Leclercq (2009) Fundamental diagram estimation through passing rate measurements in congestion, *IEEE Transactions on Intelligent Transportation Systems*, **10** (2) 355–359.

Chiabaut, N., M. Küng, M. Menendez and L. Leclercq (2018) Perimeter control as an alternative to dedicated bus lanes: A case study, *Transportation Research Record*, 0361198118786607.

Comert, G. (2013a) Effect of stop line detection in queue length estimation at traffic signals from probe vehicles data, *European Journal of Operational Research*, **226** (1) 67–76.

Comert, G. (2013b) Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals, *Transportation Research Part B: Methodological*, **55**, 59–74.

Comert, G. and M. Cetin (2009) Queue length estimation from probe vehicle location and the impacts of sample size, *European Journal of Operational Research*, **197** (1) 196 – 202.

Couchman, P. D., M. Cannon and B. Kouvaritakis (2006) Stochastic mpc with inequality stability constraints, *Automatica*, **42** (12) 2169–2174.

Daganzo, C. F. (2007) Urban gridlock: Macroscopic modeling and mitigation approaches, *Transportation Research Part B: Methodological*, **41** (1) 49–62.

Daganzo, C. F. and Y. Ouyang (2018) A general model of ridesharing services, *arXiv preprint arXiv:1807.01140*.

Dakic, I. and M. Menendez (2018) On the use of lagrangian observations from public transport and probe vehicles to estimate car space-mean speeds in bi-modal urban networks, *Transportation Research Part C: Emerging Technologies*, **91**, 317–334.

Dakic, I., K. Yang and M. Menendez (2019) Evaluating the effects of passenger occupancy dynamics on a bi-modal perimeter control, paper presented at the *TRB Annual Meeting Online*.

Darwish, T. and K. A. Bakar (2015) Traffic density estimation in vehicular ad hoc networks: A review, *Ad Hoc Networks*, **24**, 337–351.

Dresner, K. and P. Stone (2004) Multiagent traffic management: A reservation-based intersection control mechanism, paper presented at the *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS ’04, 530–537.

Dresner, K. and P. Stone (2006) Human-usable and emergency vehicle-aware control policies for autonomous intersection management, paper presented at the *AAMAS 2006 Workshop on Agents in Traffic and Transportation*, May 2006.

Farina, M., L. Giulioni and R. Scattolini (2016) Stochastic linear model predictive control with chance constraints – a review, *Journal of Process Control*, **44**, 53 – 67.

Feng, Y., K. L. Head, S. Khoshmagham and M. Zamanipour (2015) A real-time adaptive signal control in a connected vehicle environment, *Transportation Research Part C: Emerging Technologies*, **55**, 460 – 473. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.

Florin, R. and S. Olariu (2015) A survey of vehicular communications for traffic signal optimization, *Vehicular Communications*, **2** (2) 70 – 79.

Gayah, V. and V. Dixit (2013) Using mobile probe data and the macroscopic fundamental diagram to estimate network densities: Tests using microsimulation, *Transportation Research Record: Journal of the Transportation Research Board*, (2390) 76–86.

Ge, Q., B. Ciuffo and M. Menendez (2015) Combining screening and metamodel-based methods: An efficient sequential approach for the sensitivity analysis of model outputs, *Reliability Engineering & System Safety*, **134**, 334–344.

Ge, Q. and M. Menendez (2014) An efficient sensitivity analysis approach for computationally expensive microscopic traffic simulation models, *International Journal of Transportation*, **2** (2) 49–64.

Ge, Q. and M. Menendez (2017) Extending morris method for qualitative global sensitivity analysis of models with dependent inputs, *Reliability Engineering & System Safety*, **162**, 28–39.

Geroliminis, N. and C. F. Daganzo (2008) Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings, *Transportation Research Part B: Methodological*, **42** (9) 759–770.

Geroliminis, N., J. Haddad and M. Ramezani (2013) Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach, *IEEE Transactions on Intelligent Transportation Systems*, **14** (1) 348–359.

Geroliminis, N., N. Zheng and K. Ampountolas (2014) A three-dimensional macroscopic fundamental diagram for mixed bi-modal urban networks, *Transportation Research Part C: Emerging Technologies*, **42**, 168–181.

Godfrey, J. W. (1969) The mechanism of a road network, *Traffic Engineering and Control*, **11** (7) 323–327.

Gómez, P., M. Menéndez and E. Mérida-Casermeiro (2015) Evaluation of trade-offs between two data sources for the accurate estimation of origin–destination matrices, *Transportmetrica B: Transport Dynamics*, **3** (3) 222–245.

Goodall, N., B. Smith and B. Park (2013) Traffic signal control with connected vehicles, *Transportation Research Record: Journal of the Transportation Research Board*, **2381**, 65–72.

Grace, M. J. and R. B. Potts (1964) A theory of the diffusion of traffic platoons, *Operations Research*, **12** (2) 255–275.

Grand View Research, Inc. (2018) Smart cities market size worth usd 2.57 trillion by 2025.

Grant, M. and S. Boyd (2008) Graph implementations for nonsmooth convex programs, *Recent advances in learning and control*, 95–110.

Grant, M. and S. Boyd (2014) CVX: Matlab software for disciplined convex programming, version 2.1, `http://cvxr.com/cvx`, March 2014.

Greenfeld, J. S. (2002) Matching gps observations to locations on a digital map, paper presented at the *Transportation Research Board 81st Annual Meeting*.

Guler, S. I., V. V. Gayah and M. Menendez (2016) Bus priority at signalized intersections with single-lane approaches: A novel pre-signal strategy, *Transportation Research Part C: Emerging Technologies*, **63**, 51–70.

Guler, S. I. and M. Menendez (2014a) Analytical formulation and empirical evaluation of pre-signals for bus priority, *Transportation Research Part B: Methodological*, **64** (0) 41 – 53.

Guler, S. I. and M. Menendez (2014b) Evaluation of presignals at oversaturated signalized intersections, *Transportation Research Record: Journal of the Transportation Research Board*, (2418) 11–19.

Guler, S. I., M. Menendez and L. Meier (2014) Using connected vehicle technology to improve the efficiency of intersections, *Transportation Research Part C: Emerging Technologies*, **46**, 121–131.

Guo, J. and N. Balon (2006) Vehicular ad hoc networks and dedicated short-range communication, *Technical Report*, Technical Report, University of Michigan - Dearborn.

Guo, Q., L. Li and X. J. Ban (2019) Urban traffic signal control with connected and automated vehicles: A survey, *Transportation Research Part C: Emerging Technologies*.

Haddad, J. (2015) Robust constrained control of uncertain macroscopic fundamental diagram networks, *Transportation Research Part C: Emerging Technologies*, **59**, 323–339.

Haddad, J. (2017) Optimal perimeter control synthesis for two urban regions with aggregate boundary queue dynamics, *Transportation Research Part B: Methodological*, **96**, 1–25.

Haddad, J. and N. Geroliminis (2012) On the stability of traffic perimeter control in two-region urban cities, *Transportation Research Part B: Methodological*, **46** (9) 1159–1176.

Haddad, J., M. Ramezani and N. Geroliminis (2013) Cooperative traffic control of a mixed network with two urban regions and a freeway, *Transportation Research Part B: Methodological*, **54**, 17–36.

Haddad, J. and A. Shraiber (2014) Robust perimeter control design for an urban region, *Transportation Research Part B: Methodological*, **68**, 315–332.

Hajiahmadi, M., J. Haddad, B. De Schutter and N. Geroliminis (2015) Optimal hybrid perimeter and switching plans control for urban traffic networks, *IEEE Transactions on Control Systems Technology*, **23** (2) 464–478.

Hao, P. and X. Ban (2015) Long queue estimation for signalized intersections using mobile data, *Transportation Research Part B: Methodological*, **82**, 54–73.

Hao, P., X. Ban, K. P. Bennett, Q. Ji and Z. Sun (2012) Signal timing estimation using sample intersection travel times, *IEEE Transactions on Intelligent Transportation Systems*, **13** (2) 792–804.

Hao, P., X. J. Ban, D. Guo and Q. Ji (2014) Cycle-by-cycle intersection queue length distribution estimation using sample travel times, *Transportation research part B: methodological*, **68**, 185–204.

Harrison, C., B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak and P. Williams (2010) Foundations for smarter cities, *IBM Journal of Research and Development*, **54** (4) 1–16.

Haykin, S. (2004) *Kalman filtering and neural networks*, vol. 47, John Wiley & Sons.

He, H., M. Menendez and S. I. Guler (2018a) Analytical evaluation of flexible-sharing strategies on multimodal arterials, *Transportation Research Part A: Policy and Practice*, **114**.

He, H., M. Menendez and K. Yang (2017) Autonomous vehicle for weaving sections, paper presented at the *Traffic and Granular Flow Conference (TGF 2017)*.

He, H., K. Yang, H. Liang, M. Menendez and S. I. Guler (2018b) Providing public transport priority at urban network perimeters: A bi-modal perimeter control approach.

He, Q., K. L. Head and J. Ding (2012) Pamscod: Platoon-based arterial multi-modal signal control with online data, *Transportation Research Part C: Emerging Technologies*, **20** (1) 164–184.

He, Q., K. L. Head and J. Ding (2014) Multi-modal traffic signal control with priority, signal actuation and coordination, *Transportation Research Part C: Emerging Technologies*, **46** (0) 65–82.

Herrera, J. C., D. B. Work, R. Herring, X. J. Ban, Q. Jacobson and A. M. Bayen (2010) Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment, *Transportation Research Part C: Emerging Technologies*, **18** (4) 568–583.

Higle, J. L. and S. Sen (1991) Stochastic decomposition: An algorithm for two-stage linear programs with recourse, *Mathematics of Operations Research*, **16** (3) 650–669.

Hiribarren, G. and J. C. Herrera (2014) Real time traffic states estimation on arterials based on trajectory data, *Transportation Research Part B: Methodological*, **69**, 19 – 30.

Hu, J., B. B. Park and Y.-J. Lee (2015) Coordinated transit signal priority supporting transit progression under connected vehicle technology, *Transportation Research Part C: Emerging Technologies*, **55**, 393–408.

Hu, J., B. B. Park and Y.-J. Lee (2016) Transit signal priority accommodating conflicting requests under connected vehicles technology, *Transportation Research Part C: Emerging Technologies*, **69**, 173–192.

Huang, X., S. Mehrkanoon and J. A. Suykens (2013) Support vector machines with piecewise linear feature mapping, *Neurocomputing*, **117**, 118–127.

Hunt, P., D. Robertson, R. Bretherton and R. Winton (1981) Scoot-a traffic responsive method of coordinating signals, *Technical Report*, Transport and Road Research Laboratory.

Hyland, M. and H. S. Mahmassani (2018) Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign avs to immediate traveler demand requests, *Transportation Research Part C: Emerging Technologies*, **92**, 278–297.

Izadpanah, P., B. Hellinga and L. Fu (2009) Automatic traffic shockwave identification using vehicles' trajectories, paper presented at the *Proceedings of the 88th Annual Meeting of the Transportation Research Board (CD-ROM)*.

Ji, Y. and N. Geroliminis (2012) On the spatial partitioning of urban transportation networks, *Transportation Research Part B: Methodological*, **46** (10) 1639–1656.

Jiang, Z., X. M. Chen and Y. Ouyang (2017) Traffic state and emission estimation for urban expressways based on heterogeneous data, *Transportation Research Part D: Transport and Environment*, **53**, 440–453.

Kalman, R. E. (1960) A new approach to linear filtering and prediction problems, *Journal of basic Engineering*, **82** (1) 35–45.

Kamal, M., J.-I. Imura, T. Hayakawa, A. Ohata and K. Aihara (2015) A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights, *Intelligent Transportation Systems, IEEE Transactions on*, **16** (3) 1136–1147, June 2015.

Keha, A. B., J. Ismael R. de Farias and G. L. Nemhauser (2006) A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization, *Operations Research*, **54** (5) 847–858.

Keyvan-Ekbatani, M., X. Gao, V. Gayah and V. Knoop (2016) Combination of traffic-responsive and gating control in urban networks: Effective interactions. Paper presented at the 95th Annual Meeting of Transportation Research Board. Washington D.C., USA.

Keyvan-Ekbatani, M., A. Kouvelas, I. Papamichail and M. Papageorgiou (2012) Exploiting the fundamental diagram of urban networks for feedback-based gating, *Transportation Research Part B: Methodological*, **46** (10) 1393–1403.

Keyvan-Ekbatani, M., M. Yildirimoglu, N. Geroliminis and M. Papageorgiou (2015) Multiple concentric gating traffic control in large-scale urban networks, *IEEE Transactions on Intelligent Transportation Systems*, **16** (4) 2141–2154.

Kim, S.-J., K. Koh, M. Lustig, S. Boyd and D. Gorinevsky (2007) An interior-point method for large-scale-regularized least squares, *IEEE journal of selected topics in signal processing*, **1** (4) 606–617.

Kojima, M., S. Mizuno and A. Yoshise (1989) A primal-dual interior point algorithm for linear programming, in *Progress in mathematical programming*, 29–47, Springer.

Kouvaritakis, B., M. Cannon and V. Tsachouridis (2004) Recent developments in stochastic mpc and sustainable development, *Annual Reviews in Control*, **28** (1) 23–35.

Kouvelas, A., M. Saeedmanesh and N. Geroliminis (2016) Enhancing feedback perimeter controllers for urban networks by use of online learning and data-driven adaptive optimization, paper presented at the *Transportation Research Board 95th Annual Meeting*, no. 16-3873.

Krajzewicz, D., J. Erdmann, M. Behrisch and L. Bieker (2012) Recent development and applications of sumo–simulation of urban mobility, *International Journal On Advances in Systems and Measurements*, **5** (3&4).

Krauss, S., P. Wagner and C. Gawron (1997) Metastable states in a microscopic model of traffic flow, *Phys. Rev. E*, **55**, 5597–5602, May 1997.

Lee, J. and B. Park (2012) Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment, *Intelligent Transportation Systems, IEEE Transactions on*, **13** (1) 81–90.

Lee, J., B. Park and I. Yun (2013) Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment, *Journal of Transportation Engineering*, **139** (10) 1020–1029.

Lee, Y.-J., S. Dadvar, J. Hu and B. B. Park (2017) Transit signal priority experiment in a connected vehicle technology environment, *Journal of Transportation Engineering, Part A: Systems*, **143** (8) 05017005.

Lei, C. and Y. Ouyang (2017) Dynamic pricing and reservation for intelligent urban parking management, *Transportation Research Part C: Emerging Technologies*, **77**, 226–244.

Leung, K. Y. K., T.-S. Dao, C. M. Clark and J. P. Huissoon (2006) Development of a microscopic traffic simulator for inter-vehicle communication application research, paper presented at the *2006 IEEE Intelligent Transportation Systems Conference*, 1286–1291.

Li, L. and F.-Y. Wang (2006) Cooperative driving at blind crossings using intervehicle communication, *Vehicular Technology, IEEE Transactions on*, **55** (6) 1712–1724.

Li, L., D. Wen and D. Yao (2014a) A survey of traffic control with vehicular communications, *Intelligent Transportation Systems, IEEE Transactions on*, **15** (1) 425–432.

Li, L., K. Yang, Z. Li and Z. Zhang (2013) The optimality condition of the multiple-cycle smoothed curve signal timing model, *Transportation Research Part C: Emerging Technologies*, **27**, 46–57.

Li, W. and X. Ban (2018) Connected vehicles based traffic signal timing optimization, *IEEE Transactions on Intelligent Transportation Systems*.

Li, Z., L. Elefteriadou and S. Ranka (2014b) Signal control optimization for automated vehicles at isolated signalized intersections, *Transportation Research Part C: Emerging Technologies*, **49**, 1–18.

Lieberman, E. and A. B.J. (1980) Traflo: A new tool to evaluate transportation system management strategies, *Transportation Research Record*, **772**, 9–15.

Lighthill, M. and G. Whitham (1955a) On kinematic waves. i. flood movement in long rivers, paper presented at the *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 229, 281–316.

Lighthill, M. J. and G. B. Whitham (1955b) On kinematic waves. ii. a theory of traffic flow on long crowded roads, paper presented at the *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 229, 317–345.

Liu, H., K. N. Balke and W.-H. Lin (2008) A reverse causal-effect modeling approach for signal control of an oversaturated intersection, *Transportation Research Part C: Emerging Technologies*, **16** (6) 742–754.

Loder, A., L. Ambühl, M. Menendez and K. W. Axhausen (2017) Empirics of multi-modal traffic networks–using the 3d macroscopic fundamental diagram, *Transportation Research Part C: Emerging Technologies*, **82**, 88–101.

Luthy, N., S. I. Guler and M. Menendez (2016) Systemwide effects of bus stops: Bus bays vs. curbside bus stops, paper presented at the *Transportation Research Board 95th Annual Meeting*, no. 16-0596.

Ma, X., J. Jin and W. Lei (2014) Multi-criteria analysis of optimal signal plans using microscopic traffic models, *Transportation Research Part D: Transport and Environment*, **32** (0) 1 – 14.

Mahmassani, H., J. C. Williams and R. Herman (1987) Performance of urban traffic networks, paper presented at the *Transportation and Traffic Theory (Proceedings of the Tenth International on Transportation and Traffic Theory Symposium, Cambridge, Massachusetts), NH Gartner, NHM Wilson, editors, Elsevier*, 1–18.

McCarthy, N. (2015) Connected cars by the numbers. Forbes.

Mehran, B., M. Kuwahara and F. Naznin (2012) Implementing kinematic wave theory to reconstruct vehicle trajectories from fixed and probe sensor data, *Transportation research part C: emerging technologies*, **20** (1) 144–163.

Menendez, M. and C. F. Daganzo (2007) Effects of hov lanes on freeway bottlenecks, *Transportation Research Part B: Methodological*, **41** (8) 809–822.

Mori, U., A. Mendiburu, M. Álvarez and J. A. Lozano (2015) A review of travel time estimation and forecasting for advanced traveller information systems, *Transportmetrica A: Transport Science*, **11** (2) 119–157.

Nagle, A. S. and V. V. Gayah (2015) Comparing the use of link and probe data to inform perimeter metering control, paper presented at the *Transportation Research Board 94th Annual Meeting*, no. 15-0621.

Nelder, J. A. and R. Mead (1965) A simplex method for function minimization, *The Computer Journal*, **7** (4) 308–313.

Newell, G. F. (2002) A simplified car-following theory: a lower order model, *Transportation Research Part B: Methodological*, **36** (3) 195–205.

NHTSA (2018) Automated vehicles for safety, `https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety`. Accessed on Oct. 10, 2018.

Nourbakhsh, S. M. and Y. Ouyang (2012) A structured flexible transit system for low demand areas, *Transportation Research Part B: Methodological*, **46** (1) 204–216.

Ortigosa, J., M. Menendez and H. Tapia (2014) Study on the number and location of measurement points for an mfd perimeter control scheme: a case study of zurich, *EURO Journal on Transportation and Logistics*, **3** (3-4) 245–266.

Pandit, K., D. Ghosal, H. M. Zhang and C.-N. Chuah (2013) Adaptive traffic signal control with vehicular ad hoc networks., *IEEE T. Vehicular Technology*, **62** (4) 1459–1471.

Parisio, A., E. Rikos and L. Glielmo (2016) Stochastic model predictive control for economic/environmental operation management of microgrids: An experimental case study, *Journal of Process Control*, **43**, 24 – 37.

Pavone, M., S. L. Smith, E. Frazzoli and D. Rus (2012) Robotic load balancing for mobility-on-demand systems, *The International Journal of Robotics Research*, **31** (7) 839–854.

Priemer, C. and B. Friedrich (2009) A decentralized adaptive traffic signal control using v2i communication data, paper presented at the *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, 1–6, Oct 2009.

Qiao, F., H. Yang and W. H. Lam (2001) Intelligent simulation and prediction of traffic flow dispersion, *Transportation Research Part B: Methodological*, **35** (9) 843–863.

Rahman, M. S. and M. Abdel-Aty (2018) Longitudinal safety evaluation of connected vehicles' platooning on expressways, *Accident Analysis & Prevention*, **117**, 381–391.

Ramezani, M. and N. Geroliminis (2015) Queue profile estimation in congested urban networks with probe data, *Computer-Aided Civil and Infrastructure Engineering*, **30** (6) 414–432.

Ramezani, M., J. Haddad and N. Geroliminis (2015) Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control, *Transportation Research Part B: Methodological*, **74**, 1–19.

Richards, P. I. (1956) Shock waves on the highway, *Operations research*, **4** (1) 42–51.

Robertson, D. I. (1969) Transyt - a traffic network study tool, *RRL Report, No. LR253, Transport and Road Research Laboratory, Growthorne, UK.*

Robinson, S. M. (1996) Analysis of sample-path optimization, *Mathematics of Operations Research*, **21** (3) 513–528.

Ruch, C., L. Sieber, S. Hörl, K. W. Axhausen and E. Frazzoli (2018) Autonomous mobility-on-demand providing superior public transportation in rural areas, *Transportation Research. Part C, Emerging Technologies*.

SAE (2014) Definitions for terms related to on-road motor vehicle automated driving systems, *Technical Report*, Technical report, Technical report, SAE International.

Särkkä, S. (2013) *Bayesian filtering and smoothing*, vol. 3, Cambridge University Press.

Sobol, I. (1967) On the distribution of points in a cube and the approximate evaluation of integrals, *USSR Computational Mathematics and Mathematical Physics*, **7** (4) 86 – 112.

SUMO (2015) Sumo - simulation of urban mobility, `http:// dlr.de/ts/sumo`. [Online; accessed 01-July-2015].

Sun, Z. and X. J. Ban (2013) Vehicle trajectory reconstruction for signalized intersections using mobile traffic sensors, *Transportation Research Part C: Emerging Technologies*, **36**, 268–283.

Talebpour, A., H. S. Mahmassani and S. H. Hamdar (2015) Modeling lane-changing behavior in a connected environment: A game theory approach, *Transportation Research Procedia*, **7**, 420–440.

Tan, J., X. Shi, Z. Li, K. Yang, N. Xie, H. Yu, L. Wang and Z. Li (2017) Continuous and discrete-time optimal controls for an isolated signalized intersection, *Journal of Sensors*, **2017**.

Tilg, G., K. Yang and M. Menendez (2018) Evaluating the effects of automated vehicle technology on the capacity of freeway weaving sections, *Transportation Research Part C: Emerging Technologies*, **96**, 3–21.

Tong, Y., L. Zhao, L. Li and Y. Zhang (2015) Stochastic programming model for oversaturated intersection signal timing, *Transportation Research Part C: Emerging Technologies*, **58, Part C**, 474 – 486. Special Issue: Advanced Road Traffic Control.

Treiber, M., A. Hennecke and D. Helbing (2000) Congested traffic states in empirical observations and microscopic simulations, *Physical Review E*, **62** (2) 1805.

Treiber, M., A. Kesting and D. Helbing (2006) Understanding widely scattered traffic flows, the capacity drop, and platoons as effects of variance-driven time gaps, *Physical Review E*, **74** (1) 016123.

Vahidi, A. and A. Sciarretta (2018) Energy saving potentials of connected and automated vehicles, *Transportation Research Part C: Emerging Technologies*.

Webster, F. (1958) Traffic signal settings, road research laboratory techn. rep. no. 39.

Wiedemann, R. (1974) Simulation des strassenverkehrsflusses, *Schriftenreihe des Instituts fÃ¼r Verkehrswesen der UniversitÃ¤t Karlsruhe, Band 8, Karlsruhe, Germany*.

Wu, J., A. Abbas-Turki, A. Correia and A. El Moudni (2007) Discrete intersection signal control, paper presented at the *Service Operations and Logistics, and Informatics, 2007. SOLI 2007. IEEE International Conference on*, 1–6, Aug 2007.

Xie, K., D. Yang, K. Ozbay and H. Yang (2018) Use of real-world connected vehicle data in identifying high-risk locations based on a new surrogate safety measure, *Accident Analysis & Prevention*.

Xu, B., S. E. Li, Y. Bian, S. Li, X. J. Ban, J. Wang and K. Li (2018) Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections, *Transportation Research Part C: Emerging Technologies*, **93**, 322–334.

Yang, K., S. I. Guler and M. Menendez (2015) A transit signal priority algorithm under connected vehicle environment, paper presented at the *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 66–70.

Yang, K., S. I. Guler and M. Menendez (2016a) Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles, *Transportation Research Part C: Emerging Technologies*, **72**, 109–129.

Yang, K., H. He and M. Menendez (2018a) Bi-modal automated highway lanes. control strategy and evaluation, paper presented at the *CICTP 2018. Intelligence, connectivity, and mobility: proceedings of the 18th COTA International Conference of Transportation Professionals, July 5-8, 2017, Beijing, China*, 124 – 133.

Yang, K. and M. Menendez (2017) A convex model for queue length estimation in a connected vehicle environment, paper presented at the *Transportation Research Board 96th Annual Meeting*.

Yang, K. and M. Menendez (in press) Queue estimation in a connected vehicle environment: A convex approach, *IEEE Transactions on Intelligent Transportation Systems*. DOI: `10.1109/TITS.2018.2866936`.

Yang, K., M. Menendez and S. I. Guler (2016b) Using connected vehicle technology to optimize transit signal priority, paper presented at the *Transportation Research Board 95th Annual Meeting*.

Yang, K., M. Menendez and S. I. Guler (2018b) Implementing transit signal priority in a connected vehicle environment with and without bus stops, *Transportmetrica B: Transport Dynamics*, 1–23.

Yang, K., M. Roca-Riu and M. Menendez (2018c) An auction-based approach for prebooked urban logistics facilities, *Omega*. DOI: `10.1016/j.omega.2018.10.005`.

Yang, K., N. Zheng and M. Menendez (2017a) Integrating perimeter control with dedicated express toll lanes, paper presented at the *Traffic and Granular Flow Conference (TGF 2017)*.

Yang, K., N. Zheng and M. Menendez (2017b) Multi-scale perimeter control approach in a connected-vehicle environment, paper presented at the *22nd International Symposium on Transportation and Traffic Theory (ISTTT)*, vol. 23, 101–120.

Yang, K., N. Zheng and M. Menendez (2018d) Multi-scale perimeter control approach in a connected-vehicle environment, *Transportation Research Part C: Emerging Technologies*, **94**, 32 – 49.

Yang, K., N. Zheng and M. Menendez (2018e) A perimeter control approach integrating dedicated express toll lanes, paper presented at the *Transportation Research Board 97th Annual Meeting*.

Yang, K., N. Zheng and M. Menendez (2019) On the design and operation of special mobility zones in large urban networks. space allocation and traffic control, TRB Annual Meeting Online.

Yuan, Y., J. Van Lint, R. E. Wilson, F. van Wageningen-Kessels and S. P. Hoogendoorn (2012) Real-time lagrangian traffic state estimator for freeways, *IEEE Transactions on Intelligent Transportation Systems*, **13** (1) 59–70.

Zhao, L., X. Peng, L. Li, Z. Li et al. (2011) A fast signal timing algorithm for individual oversaturated intersections, *IEEE Transactions on Intelligent Transportation Systems*, **12** (1) 280–283.

Zheng, N. and N. Geroliminis (2013) On the distribution of urban road space for multimodal congested networks, *Transportation Research Part B: Methodological*, **57**, 326 – 341.

Zheng, N. and N. Geroliminis (2016) Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing, *Transportation Research Part B: Methodological*, **83**, 36–58.

Appendix A

Detailed Calculations

A.1 Inclusion of the Intermediate States for Cases With Limited Data

This appendix elaborates the procedure for estimating the FoQ critical points when the sampling rates are low. Particularly, we consider three cases:

1. There are stopped trajectory points but no free flow trajectory points, i.e. set $V_{p+1} \cap J_n = \emptyset$ and set $S_p \cap J_n \neq \emptyset$;
2. There are free flow trajectory points but no stopped trajectory points, i.e. set $V_{p+1} \cap J_n \neq \emptyset$ and set $S_p \cap J_n = \emptyset$;
3. There are neither free flow trajectory points nor stopped trajectory points, i.e. set $V_{p+1} \cap J_n = \emptyset$ and set $S_p \cap J_n = \emptyset$.

For case 1) and 2), we assume that there is at least one trajectory point in the set A_{p+1}. For case 3), we assume that there are at least two trajectory points in $A_{p+1} \cap J_n$ for at least one vehicle trajectory m. This is the least requirement to guarantee that we can reconstruct at least one critical point.

The procedure has two steps. In the first step, we estimate the accelerating trajectory, i.e. the value of b and c in Eq.(3.24). In the second step, the critical points are obtained based on the accelerating trajectory. In the rest of this section, we explain the procedure for the three cases, respectively.

For case 1), we denote the line fitted from the stopped trajectory points in set $J_n \cap S_p$ as $x(t) = l_n^p$. This line has a slope of zero, as it represents the stopped state. Then the following two properties should hold. First, Eq.(3.24) should be tangent to line $x(t) = l_n^p$. This is because both the location and the speed are continuous. Second, the distance between Eq.(3.24) and the trajectory points in set $J_n \cap A_{p+1}$ should be minimized.

Based on these two properties, we can build an optimization model for trajectory m.

$$\min \sum_{j \in J_n \cap A_{p+1}} \left(\frac{1}{2}at_j^2 + bt_j + c - x_j\right)^2 \tag{A.1}$$

$$\text{s.t. } 2ac - b^2 = 2al_n^p \tag{A.2}$$

where the constraints and the objective function correspond to the two properties, respectively. The decision variables are b and c. Although Eq.(A.1) and Eq.(A.2) are not convex, the optimal solution of b and c can be represented in a closed form. Therefore, it is not necessary to solve the model Eq.(A.1) and Eq.(A.2) in practice.

With the estimated trajectory in the intermediate state, we can reconstruct the trajectory in the free flow state as a tangent line to the parabola Eq.(3.24) with slope u_f. With a basic calculation, the critical point can be represented as $\left(\frac{u_f - 2b}{2a}, l_n^p\right)$.

For Case 2), let us denote the trajectory fitted from the free-flow trajectory points in set $J_n \cap V_{p+1}$ as the line $x = u_n^p t + \eta_n^p$. Similarly to Case 1) in Section 3.4, the parabola Eq.(3.24) should also satisfy the following two requirements.

1) Eq.(3.24) should be tangent to line $x = u_n^p t + \eta_n^p$. This is because both the location and the speed are continuous.
2) The distance between Eq.(3.24) and the trajectory points in set $J_n \cap A_{p+1}$ should be minimized.

Then we formulate the following optimization model

$$\min \sum_{j \in J_n \cap A_{p+1}} \left(\frac{1}{2}at_j^2 + (b + u_n^p)t_j + c - x_j\right)^2 \tag{A.3}$$

$$\text{s.t. } 2ac - b^2 = 2a\eta_n^p \tag{A.4}$$

Model Eq.(A.3) and Eq.(A.4) can also be transformed into a cubic equation and solved analytically. Then the critical point can be represented as $\left(\frac{u_f - 2b}{2a}, c - \frac{b^2}{2a}\right)$.

For Case 3), as there are no trajectory points in either the free flow state or the stopped state, we only require that the distance between Eq.(3.24) and the trajectory points in set $J_n \cap A_{p+1}$ be minimized. Hence we have the following non-constrained optimization problem.

$$\min \sum_{j \in J_n \cap A_{p+1}} \left(\frac{1}{2}at_j^2 + bt_j + c - x_j\right)^2 \tag{A.5}$$

Model Eq.(A.5) is a convex model, which can be transformed into a set of linear equations. Then the critical point can be represented as $\left(\frac{u_f - 2b}{2a}, c - \frac{b^2}{2a}\right)$.

A.2 Calculation of the Optimal and Entering Speed

1) First vehicle in platoon

When vehicle n which is the first vehicle in the platoon, the optimal trajectory would let it pass the intersection exactly at time t_g and then accelerate. This assumption which is made for simplicity purposes, could cause a small and systematic error, as in theory, the vehicle could try to accelerate before and enter the intersection with the free flow speed. This error would affect the model in a conservative way: the observed delay in the simulation is probably slightly larger than what could be realized.

The optimal speed of vehicle n before accelerating can be found by Eq.(A.6).

$$u_{n,J}^{opt} = \frac{x_n - \sigma_x}{t_g - t_0} \tag{A.6}$$

where σ_x is the estimated standard deviation for location errors. σ_x is estimated with the aid of a Kalman filter. The detailed explanation of the Kalman filter and σ_x can be found in A.3. Eq.(A.6) designs a conservative speed when there are measurement errors. In cases without measurement errors, the Kalman filter would give a $\sigma_x = 0$.

The optimal speed may not satisfy the speed lower and upper bound and hence should be further adjusted.

$$u_{n,J}^{des} = \begin{cases} \max\{\min\{u_{n,J}^{opt}; u_f\}; u_{\min}\}, & t_g > t_0 \\ u_f, & t_g \le t_0 \end{cases} \tag{A.7}$$

In case $t_g > t_0$, the speed is constrained by the lower speed bound and the free flow speed. In case $t_g \le t_0$, vehicle n arrives at the intersection after the signal switches, thus vehicle n can drive with the free flow speed on the approach link.

Then, the initial speed at which vehicle n enters the intersection is calculated as

$$u_{n,J}^{init} = \begin{cases} u_{n,J}^{des}, & u_{n,J}^{des} > u_{\min} \\ 0, & u_{n,J}^{des} = u_{\min} \end{cases} \tag{A.8}$$

where in the second case ($u_{n,J}^{des} = u_{\min}$), the vehicle would arrive at the intersection before the signal turns green. This means that the vehicle would simply stop to avoid crawling and then accelerate across the intersection. Therefore the initial speed is 0 in this case.

2) vehicle $\hat{n}$ following previous vehicles

If vehicle $\hat{n}$ follows the previous vehicles (i.e. it is not the platoon leader), the optimal trajectory should be tangent to the closest possible trajectory to the last vehicle in the platoon (based on Newell's vehicle following model (Newell, 2002); see the dash line in Figure 4.3b) . This means that vehicle $\hat{n}$ will join the platoon and then accelerate across the intersection. The following equation is derived by basic kinematic law and traffic flow theory.

$$t_g + \frac{u_{\hat{n},J}^{opt} - u_{n,J}^{des}}{a_n} + \frac{(O_{\hat{n},J} - 1)/k_{\text{jam}}}{w} = t_0 + \frac{x_{\hat{n}}}{u_{\hat{n},J}^{opt}} - \frac{(O_{\hat{n},J} - 1)/k_{\text{jam}} + \frac{(u_{\hat{n},J}^{opt})^2 - (u_{n,J}^{des})^2}{2a_n}}{u_{\hat{n},J}^{opt}} \tag{A.9}$$

where both sides of the equation represent the time when the actual trajectory of vehicle $\hat{n}$ intersects with the closest possible trajectory (marked as point of tangency in Figure 4.3b). The left hand side of Eq.(A.9) corresponds to this intersecting time in the closest possible trajectory while the right hand side corresponds to this intersecting time in the designed trajectory.

The optimal speed $u_{\hat{n},k}^{opt}$ can then be obtained by solving Eq.(A.9).

$$\begin{aligned} u_{\hat{n},J}^{opt} &= u_{n,J}^{des} - a_n \left(t_g - t_0 + \frac{O_{\hat{n},J} - 1}{w k_{\text{jam}}} \right) \\ &+ a_n \sqrt{\left(t_g - t_0 + \frac{O_{\hat{n},J} - 1}{w k_{\text{jam}}} \right)^2 + 2 \frac{1}{a_n} \left(x_{\hat{n}} - \frac{(O_{\hat{n},J} - 1)(u_{n,J}^{des} + w)}{w k_{\text{jam}}} \right)} \end{aligned} \tag{A.10}$$

The design speed $u_{\hat{n},J}^{des}$ is obtained by adjusting $u_{\hat{n},J}^{opt}$.

$$u_{\hat{n},J}^{des} = \max\{\min\{u_{\hat{n},J}^{opt}; u_f\}; u_{\min}\} \tag{A.11}$$

The initial speed when vehicle $\hat{n}$ enters the intersection can be found as

$$u_{\hat{n},J}^{init} = \max \left\{ u_{\hat{n},J}^{des}; \sqrt{(u_{\hat{n},J}^{des})^2 + 2a_n (O_{\hat{n},J} - 1)/k_{\text{jam}}} \right\} \tag{A.12}$$

where the first term of the right hand side represents the case where this vehicle joins the accelerating platoon after entering the intersection and the second term represents the case where this vehicle enters the intersection with the accelerating platoon (as in Figure 4.3b).

A.3 Kalman Filter for Location and Speed Estimation With Measurement Noises

Kalman filter (Kalman, 1960) is an algorithm that produces estimates of unknown variables from a time series of noisy measurements. The estimation results are more precise than a single measurement. In this section, a Kalman filter is applied to estimate the speed x and location u, as well as the standard deviation of the location σ_x. The signal control algorithm uses this information as an input.

The Kalman filter is applied in an online manner. Every time the vehicle sends information, the estimated location $\hat{x}(k)$, speed $\hat{v}(k)$ and their covariance matrix $\mathbf{\Sigma}(k)$ are updated. In this section, all the variables with bold font represent vectors or matrices.

A Kalman filter consists of system equations and measurement equations. The system equations are derived from the basic vehicle motion equations.

$$x(k) = x(k-1) + u(k-1)C + q_x(k) \tag{A.13}$$

$$u(k) = u(k-1) + q_u(k) \tag{A.14}$$

where $x(k)$ and $u(k)$ represent the real location and speed at time step k; C is the duration of each time step, which is chosen as 0.2s in this paper (according to Gómez et al. (2015) and Leung et al. (2006), a resolution between 1 and 5 Hz is reasonable for GPS data). $q_x(k)$ and $q_u(k)$ are the system errors at time step k, which are introduced due to two reasons: 1) acceleration and driving behavior are not considered; 2) time is discretized. The joint distribution of $q_x(k)$ and $q_u(k)$ is assumed to follow a two dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{Q}$.

$$[q_x(k), q_v(k)]^T \sim N(\mathbf{0}, \boldsymbol{Q}) \tag{A.15}$$

where x^T represents the transpose of vector/matrix x. Covariance matrix $\boldsymbol{Q}$ can be calibrated from real data, and is related to the sampling interval of the GPS data. Here, we use $\boldsymbol{Q} = [6.8061, 0.0382; 0.0382, 0.3819]$.

The measurement equations are written as

$$y(k) = x(k) + r_x(k) \tag{A.16}$$

$$v(k) = u(k) + r_u(k) \tag{A.17}$$

where $y(k)$ and $v(k)$ are the measurement of location and speed, respectively. $r_x(k)$ and $r_u(k)$ are measurement errors of location and speed, respectively. The joint distribution of $r_x(k)$ and $r_u(k)$ is assumed to follow a two dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{R}$, i.e. $[r_x(k), r_v(k)]^T \sim N(\mathbf{0}, \boldsymbol{R})$. Covariance matrix $\boldsymbol{R}$ is related to the accuracy of the measurement devices.

To solve the Kalman filter, the system equations and measurement equations are rewritten as

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} x(k-1) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} q_x(k) \\ q_u(k) \end{bmatrix} \tag{A.18}$$

$$\begin{bmatrix} y(k) \\ v(k) \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} y(k-1) \\ v(k-1) \end{bmatrix} + \begin{bmatrix} r_x(k) \\ r_u(k) \end{bmatrix} \tag{A.19}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & C \\ 0 & 1 \end{bmatrix}, \ \boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.20}$$

Then the location, speed and their covariance can be estimated using the following iterative equations.

$$\boldsymbol{\mu}(k) = \boldsymbol{A}\begin{bmatrix}\hat{x}(k-1)\\ \hat{u}(k-1)\end{bmatrix} \tag{A.21}$$

$$\boldsymbol{P}(k) = \boldsymbol{A}\boldsymbol{\Sigma}(k-1)\boldsymbol{A}^T + \boldsymbol{Q} \tag{A.22}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}(k)\boldsymbol{H}^T(\boldsymbol{H}^T\boldsymbol{P}(k)\boldsymbol{H} + \boldsymbol{R})^{-1} \tag{A.23}$$

$$\begin{bmatrix}\hat{x}(k-1)\\ \hat{u}(k-1)\end{bmatrix} = \boldsymbol{\mu}(k) + \boldsymbol{K}(k)\begin{bmatrix}y(k)\\ v(k)\end{bmatrix} \tag{A.24}$$

$$\boldsymbol{\Sigma}(k) = (\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{H})\boldsymbol{P}(k) \tag{A.25}$$

where $\boldsymbol{\mu}(k)$ is the predicted state variables, $\boldsymbol{P}(k)$ is the predicted covariance, $\boldsymbol{K}(k)$ is the Kalman gain. Then, by rewriting $\boldsymbol{\Sigma}(k)$ as $\boldsymbol{\Sigma}(k) = [\Sigma_{xx}, \Sigma_{xu}; \Sigma_{xu}, \Sigma_{uu}]$, the standard deviation of the location can be found as

$$\sigma_x = \sqrt{\Sigma_{xx}}. \tag{A.26}$$

The initial covariance is chosen as $\boldsymbol{R}$ and the initial speed and location are chosen as the initial measurement.

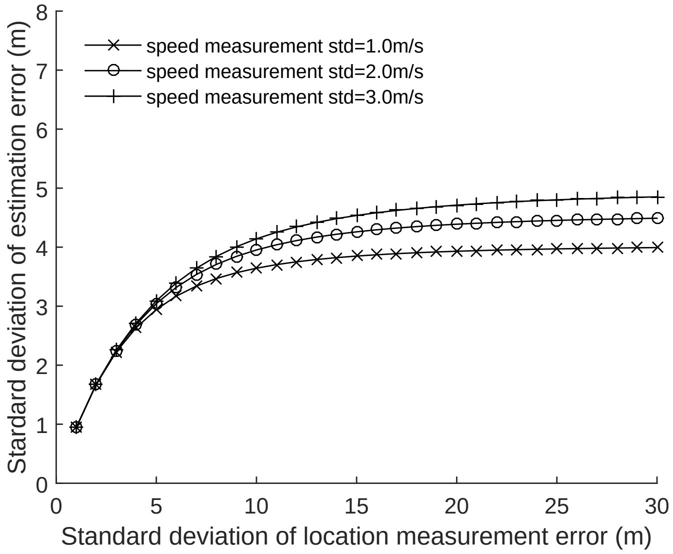


Figure A.1: Effect of the Kalman filter.

The effect of the Kalman filter can be seen in Fig. A.1. The horizontal axis is the standard deviation of the measurement errors, whereas the vertical axis represents the standard deviation of the location estimation after implementing the Kalman filter. While this paper focuses more so on the error in the location measurement since the algorithm relies more on this information, three different standard errors for speed measurements are also shown in this figure. Fig.A.1 shows that the estimation error of the location is smaller than the measurement error. In other words, by using the Kalman filter, the error related to location measurement is reduced. It is also shown in Figure Fig.A.1 that the estimation of the Kalman filter is not very sensitive to the standard deviation of the error in speed measurement. Also notice that adding the Kalman filter would not influence the running time, as only one iteration of the Kalman equations is required in each time step.