Traffic Signal Optimization with Connected Vehicle Trajectories

by

Xingmin Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2023

Doctoral Committee:

Professor Henry Liu, Chair
Associate Professor Neda Masoud
Professor Siqian Shen
Professor Yafeng Yin
Professor Lei Ying

Xingmin Wang

xingminw@umich.edu

ORCID iD: 0000-0003-0435-2786

I dedicate this dissertation to my parents for their support,
encouragement, and unconditional love.

ACKNOWLEDGMENTS

It has been an unbelievable journey for me to pursue the Ph.D. degree at the University of Michigan. First and foremost, I would like to express my heartfelt gratitude to my advisor Prof. Henry Liu for his guidance, support, and encouragement over the past five years. Dr. Liu has been a role model for me in both life and research, especially his extraordinary ability to identify key research problems. He can always guide me in the right direction with great vision whenever I get lost while diving into unnecessary details. I truly enjoy the discussion with him, I learn a lot each time and realize there is more to learn from him. His wisdom, passion, and expertise in research set up a lifetime model for me in my future career path. I feel honored and lucky to work with him, and none of this will be achieved without him.

I also would like to thank my committee members, Prof. Yafeng Yin, Prof. Neda Masoud, Prof. Siqian Shen, and Prof. Lei Ying, for their valuable and constructive suggestions throughout my doctoral study. I appreciate the opportunities to collaborate and learn from them in classes, seminars, and projects. They have been very helpful and supportive.

Many thanks to my collaborators at General Motors, Vivek Vijaya Kumar, Dr. Fan Bai, and Dr. Paul Krajewski, for providing the data. My thanks also go to Gary Piotrowicz, Rachel Jones, Danielle Deneau, and Ahmad Jawad at Road Commission for Oakland County, for helping with the field implementation.

I would like to thank my former and current lab mates in Michigan Traffic Lab, Prof. Shuo Feng, Prof. Yiheng Feng, Prof. Wai Wong, Prof. Zhengxia Zou, Prof. Xiaowei Shi, Dr. Jianfeng Zheng, Dr. Yan Zhao, Dr. Shihong Huang, Dr. Zhen Yang, Dr. Lin Liu, Dr. Rusheng Zhang, Dr. Depu Meng, Dr. Ronan Keane, Dr. Yuanxin Zhong, Dr. Boqi Li, Shengyin Shen, Xintao Yan, Zachary Jerome, Zihao Wang, Haowei Sun, and Haojie Zhu. I have a memorable and enjoyable time with them for research collaborations, group meetings, and dinner parties. Particularly, I feel indebted to Xintao, Shengyin, and Yuanxin, who were with me at both my happiest and hardest times during the past years. I also cannot say enough thanks to Zachary and Zihao for our incredible traffic signal group.

I am thankful to my friends and colleagues at UM, Prof. Zhengtian Xu, Prof. Xiaotong Sun, Prof. Ziyou Song, Prof. Yunlin Pan, Prof. Xian Yu, Dr. Qiuhao Hu, Dr. Niankai Yang, Dr. Kaiwen Liu, Chenhao Zhang, Lu Wen, Xinyu Fei, Zhichen Liu, Minghui Wu, Xiang Wei, Xiao Li, Weifan Zhang, Zheyu Zhang, Sitong Liu, Wenshan Yu, Yuting Cao, and Zaiyi Jiang. I also sincerely thank my dear friends, Prof. Zhaojian Li, Dr. Laixi Shi, Ruxing Fu, and Yiming Fucha, who have given me great support and help over the years.

Last but certainly not least, I owe my deepest gratitude to my parents, Zhiyong Wang and Zhiying Liu, for their unconditional love. Although being in a different timezone, they are almost ready to answer my phone call at any time. I truly thank them for believing in me and respecting my choices. There is no chance that I would have gone so far without them.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

LIST OF ACRONYMS

AACVTE Ann Arbor Connected Vehicle Test Environment

ATSC adaptive traffic signal control system

AV automated vehicles

BSM basic safety message

CAV connected and automated vehicles

CV connected vehicles

CV connected vehicle

DSRC dedicated short-range communication

EM expectation-maximization

GNSS Global Navigation Satellite System

HCM highway capacity manual

IMU inertial measurement unit

LOS level of service

LWR Lighthill-Whitham-Richards

MAPE mean absolute percentage error

MAP maximum a posterior

MCMC Markov chain Monte Carlo

MLE maximum likelihood estimation

MM method of moments

OBU on-board unit

OSaaS Optimizing traffic Signals as a Service

PI performance index

PRT perception-reaction time

PTS probabilistic time-space

QCC queue clearance control

RL reinforcement learning

RMSD root-mean-square derivation

RMSE root-mean-square error

RSD relative standard derivation

RSU road-side unit

SPaT signal phase and timing

SUMO Simulation of Urban MObility

TOD time of day

LIST OF SYMBOLS

q^m	Saturation flow rate of a certain movement
z	Number of lanes of a certain movement
Δu	Unit traffic flow (under the saturation flow rate) per time step Δt
(t, n)	Free-flow arrival time and number of unit traffic flow in Newellian coordinates
(t', s')	Normal Euclidean time and space coordinates
v_f	Free-flow speed
h	Jam space headway
$X^n(t)$	Spatial queue length at time t (number of unit traffic flow Δu, unit: 1)
$X^s(t)$	Spatial queue length at time t (distance, unit: meter)
$X(t)$	Point queue length (under the Newellian coordinates) at time t
$\Psi_t(\cdot), \Psi_t^{-1}(\cdot)$	Mapping function and its inverse function that converts between the point queue $X(t)$ and spatial queue $X^n(t)$
$A(t)$	Arrival at time t
$B(t)$	Departure at time t
$a(t)$	Average arrival rate (expectation of $A(t)$) at time t
$b(t)$	Average departure rate (expectation of $B(t)$) at time t
$S(t)$	Traffic signal state at time t
$x(t, k)$	Pmf of the point queue length ($X(t)$) distribution at time t with k stopped vehicles
$\rho^n(t, n), \rho^t(t, n)$	Probability that there are vehicles traveling on the vertical and horizontal edges in the probabilistic time-space (PTS) diagram
t_c	Start of the red time of cycle c
t_c^r	End of the red time of cycle c

$\mathcal{T}_c$	Set of all time steps of cycle c
ϕ	Penetration rate
Θ	Set of stationary traffic parameters to be estimated
$\mathcal{O}$	Observed vehicle trajectory data
$\mathcal{X}$	Overall traffic state including both observed and unobserved vehicle trajectories
d	Average control delay
$\tilde{A}, \tilde{d}, \tilde{X}^n$	Observed arrival, control delay, and queue length The superscript tilde ($\sim$) in this dissertation denotes that this variable comes from observation
N_c	Total number of cycles that are used to estimate unknown traffic state and parameter
$L(\cdot)$	Log-likelihood function
$\boldsymbol{s}$	Overall traffic signal timing parameters
τ^k	Temporal boundary between TOD k and TOD $k+1$
C^k	Cycle length of TOD k
$\boldsymbol{g}^k$	Green splits of TOD k
$\boldsymbol{o}^k$	Offsets of TOD k
$I(\cdot)$	Performance index function
Δo	Relative offset
t_l	Lag time from the most recently available estimation results to the current time step

ABSTRACT

Traffic signal re-timing is one of the most cost-effective methods for reducing congestion and energy consumption in urban areas based on the existing road infrastructure. However, high installation and maintenance costs of vehicle detectors have prevented the widespread implementation of adaptive traffic signal control system (ATSC). In the past few years, vehicle trajectory data has become increasingly available and offers many advantages over detectors and other infrastructure-based sensors for traffic monitoring. However, one major challenge of using vehicle trajectory data for traffic signal re-timing is the data sparsity and incompleteness caused by the limited penetration rate.

This dissertation aims at providing systematic methods for traffic signal optimization with vehicle trajectory data at the current market penetration rate ($\leq 10\%$). The main contribution is the newly proposed stochastic traffic flow model under Newellian coordinates, which is established based on Newell's simplified car-following model. We show that a point-queue model under the Newellian coordinates can sufficiently capture the whole spatial-temporal traffic state through the PTS diagram. This simplification is made feasible by ignoring the stochastic driving behavior since most of the system uncertainty comes from the stochastic traffic demand as well as sparse observation at a low penetration rate.

The main advantage of the proposed model is that it is a stochastic model with much lower dimensions and can be directly calibrated by taking the vehicle trajectory data as the input. It enables us to apply different statistical estimation algorithms to estimate both stationary traffic parameters (i.e., penetration rate, average arrival rate, etc.) and real-time traffic state (queue length). Based on the estimated traffic state and parameters, we also develop different optimization

programs for the re-timing of fixed-time traffic signals and a rule-based queue clearance control (QCC) for real-time traffic signals.

With the proposed methods, we develop an integrated traffic signal re-timing system called Optimizing traffic Signals as a Service (OSaaS). In April 2022, a citywide field test of OSaaS was conducted in Birmingham, Michigan, with 34 signalized intersections. 2 corridors and 2 isolated intersections were implemented with new fixed-time signal timing plans, resulting in decreases in both the delay and number of stops by up to 20% and 30%, respectively. OSaaS is a closed-loop iterative system including performance evaluation, traffic state estimation, traffic signal diagnosis, and optimization. By not requiring installation or maintenance of vehicle detectors, OSaaS provides a more scalable, sustainable, resilient, responsive, and efficient solution to traffic signal re-timing based on vehicle trajectory, which could be applied to every traffic signal in the world.

CHAPTER 1

Introduction

1.1 Background

Annually, drivers in the United States experience roughly $22.9 billion in direct and indirect congestion costs at signalized intersections (Son, 2019). Much of this delay is the result of outdated or improper traffic signal operations, which the 2019 National Traffic Signal Report Card gave a C+ (Son, 2019). Traffic signal re-timing is widely regarded by traffic engineers as one of the most cost-effective methods for reducing congestion and energy consumption in urban areas as it doesn't require any major changes to the existing infrastructure. Large benefit-to-cost ratios (ranging from 20:1 to 83:1) have been reported by traffic agencies across the country (Sunkari, 2004; Chien et al., 2006; Department and Howard/Stein-Hudson Associates, 2010).

However, a large proportion of the 320,000 signalized intersections in the US do not have detectors and are controlled by fixed-time traffic signals (Son, 2019). Many agencies must rely on in-person data collection to monitor traffic demand at these intersections. This is a time-consuming process that has narrow observation windows (2 days at most), limiting the potential for traffic signal performance improvements. As traffic demand undergoes natural changes or growth, signal timing plans become outdated, which increases congestion and energy costs. Traffic management authorities might feel pressure to recover as much of these costs as possible because of infrequent re-timing opportunities.

Some intersections with detection capability use vehicle-actuated control or ATSC, which are more responsive to the time-varying traffic demand compared to fixed-time traffic signals. While in many cases ATSC has proven to be effective, sometimes improving travel times by up to 50 percent or more, their widespread implementation has been prevented by high installation, maintenance, and software licensing costs (Dobrota et al., 2020). In addition, actuated signal control has also been discouraged by the National Association of City Transportation Officials (NACTO) because of the maintenance requirements and detection upkeep on streets (NACTO, 2015). Since a single unreliable detector can jeopardize the effectiveness of an entire ATSC or actuated system, these

detectors require frequent monitoring and maintenance. Many agencies have also discovered that ATSC cannot simply be installed and left alone for long periods of time. Some, especially those who are not knowledgeable of the inner workings of ATSC, have been forced to switch back to traditional fixed-timed plans after the original ATSC settings are unable to handle long-term changes in traffic patterns (Dobrota et al., 2022). As a result, although the first ATSC systems were developed in the early 1970s, only a small percentage of signalized intersections in the U.S. ($2-5\%$) have been outfitted with this technology (Zhao and Tian, 2012).

1.2 Vehicle trajectory data

1.2.1 Introduction to trajectory data

In the past few years, vehicle trajectory data has become increasingly available and has been explored as an alternative to detector-based traffic management; it can be collected from a variety of existing resources. For example, it can be extracted from basic safety message (BSM) through dedicated short-range communication (DSRC) when the vehicle equipped with on-board unit (OBU) is within the communication range of the road-side unit (RSU) (Bezzina and Sayer, 2014; Wang et al., 2020). It can also be collected from ride-hailing services (Uber, Lyft, etc.) and navigation systems (Google maps). There are also different open-source trajectory data such as NGSIM (Kovvali et al., 2007) and pNEUMA data (Barmpounakis and Geroliminis, 2020) that are frequently used by researchers.

Although different types of trajectory data have different available channels, they all have common essential attributes including unique device or trip ID, timestamp, and GNSS coordinates (latitude and longitude). Accuracy and frequency are the most important metrics of data quality. Unlike safety-related applications that have a high requirement for both accuracy and frequency of the trajectory data, efficiency-related traffic operational applications have a much lower requirement for data quality. It is already sufficient if the time interval is within 5 seconds and the accuracy of GNSS coordinates is not larger than 10 meters.

1.2.2 Characteristics of vehicle trajectory data

Monitoring traffic through vehicle trajectory data offers many advantages over detectors and sensors (Saldivar-Carranza et al., 2021; Wang et al., 2022a). It has a much larger coverage area than detector data, and is available at almost every intersection, especially those with higher traffic volumes. For example, Figure 1.1 shows the comparison of the coverage between vehicle trajectory data and vehicle detectors: vehicle trajectory data covers the whole road network while detectors

are only installed at certain locations.

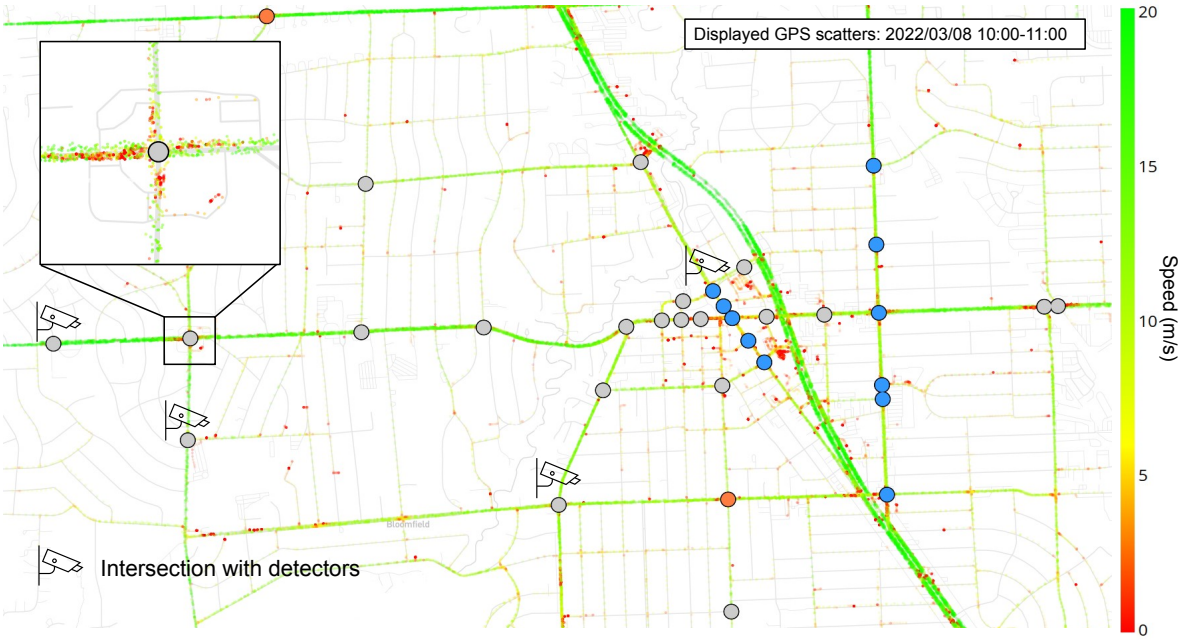


Figure 1.1: Scatter plots of vehicle trajectories and locations of detectors (Birmingham, Michigan).

Detector data and vehicle trajectory data provide different traffic measurements. While detector data provides more complete information at certain locations including traffic counts and speed, vehicle trajectory data spans the spatial-temporal space and provides more enriched information (Figure 1.2). Accurate travel time and delay measurements including stop durations and locations can be easily extracted. Besides, trajectory data can also provide path information that is not directly available in detector data.

In general, vehicle trajectory data provides a more economical solution to traffic monitoring compared to detectors by not requiring any new additional equipment for complete monitoring ability across an entire urban network. A monitoring system made up of vehicle trajectory data is more resilient to equipment failure as it will not completely lose traffic monitoring capability at any specific location if one vehicle's transponder malfunctions. With the continued advancement of connected and automated vehicles, it can be foreseen that more vehicle trajectories will be available in the future and will be a more sustainable and scalable solution to urban traffic monitoring and management.

Although vehicle trajectory data has many advantages over traditional detectors and sensors, there are some difficulties in using the currently available vehicle trajectory data for traffic signal optimization, including sparse observation caused by the limited penetration rate and the lack of a suitable stochastic traffic flow model. More details will be discussed in Section 1.4.

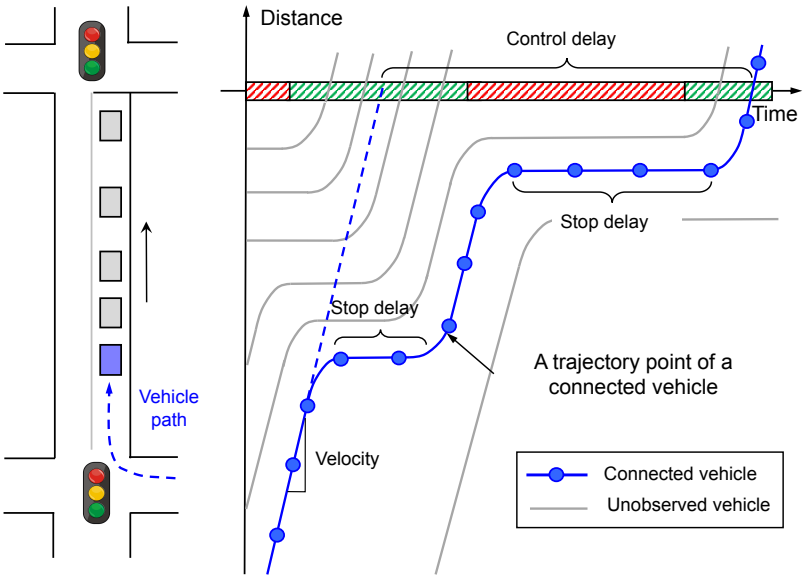


Figure 1.2: Illustration of vehicle trajectory data in the time-space diagram.

1.2.3 Trajectories used in this dissertation

Two different trajectory data sets are used in this dissertation: 1) trajectory data collected by Ann Arbor Connected Vehicle Test Environment (AACVTE) (Wang et al., 2020) and 2) vehicle telemetry data collected by General Motors. AACVTE trajectory data is extracted from BSM; it has a higher frequency of 10 Hz but the penetration rate is low (less than 3%). GM trajectory data is collected from General Motors vehicles, which are equipped with GNSS receivers and inertial measurement unit (IMU). The time interval of GM trajectory data is around 3 seconds and the penetration rate is estimated between 5% and 10%. Another major difference between these two data sets is that AACVTE trajectory data is only available when the vehicle is within the communication range of intersections equipped with RSUs while GM trajectories have long continuous trips from origin to destination. Due to their different characteristics, AACVTE trajectory data is more suitable for safety-related applications while GM data is better for efficiency-related applications. With both data available for this dissertation, GM data is used more frequently.

However, both AACVTE data and GM data do not have the ground truth: they only have the connected vehicle data without knowing the background traffic. Although there are still cross-validation methods to verify the proposed methods without knowing the ground truth, it is better to have the overall traffic for verification purposes. Therefore, this dissertation also contains some experiments utilizing the simulated data generated from Simulation of Urban MObility (SUMO) (Lopez et al., 2018).

1.3 Literature review

1.3.1 Trajectory and map data preprocessing

Trajectory data preprocessing is the basis for all advanced applications based on vehicle trajectory data. Many existing studies have explored different aspects of trajectory processing including data cleaning (Wang et al., 2013; Fazzinga et al., 2014; Li et al., 2020), map matching (Newson and Krumm, 2009; Quddus and Washington, 2015; Yang and Gidofalvi, 2018), encoding, and compression (Zheng, 2015; Wang et al., 2021; Chen et al., 2019). In addition to the vehicle trajectory data, map and network data is another important part. OpenStreetMap (2019) is the most frequently used open-source map data but it cannot be easily used without further cleaning and reformatting. There are some existing tools that can be used to extract, clean, and reformat the OpenStreetMap data including OSMnx (Boeing, 2017) and OSM2GMNS (Lu and Zhou, 2022). However, neither constructs a movement-level network which is essential for many traffic operational applications.

1.3.2 Traffic models for urban traffic networks

Traffic flow models play an important role in traffic state estimation, prediction, and traffic control. Although data-driven methods have received much attention in recent years (Avila and Mezić, 2020; Cui et al., 2020), model-based methods are more reliable and interpretable, particularly with missing data or incomplete observations. Commonly available traffic data cannot provide a complete observation of the overall traffic state: detector-based data and vehicle trajectory data are limited by detection availability (installation) and penetration rate, respectively. In this case, it is important to incorporate traffic flow models as the prior knowledge for data assimilation (Wang et al., 2022c). Other than traffic state estimation, traffic flow models are also critical for model-based traffic control and optimization where the traffic state can be predicted under different traffic signal parameters (Lo, 1999; Aboudolas et al., 2009).

Traffic flow models have different scales from microscopic to macroscopic. These scales have different applications with the trade-off between model accuracy and computational efficiency. A macroscopic traffic flow model is usually sufficient for efficiency-related studies such as traffic signal optimization. First-order models like the Lighthill-Whitham-Richards (LWR) model (Light and Whitham, 1955; Richards, 1956) are the most commonly used traffic flow models for urban traffic networks which are characterized by the interrupted flow controlled by traffic signals. Different versions and formulations have been proposed based on the LWR model, including the cell and link transmission models (Daganzo, 1994; Yperman et al., 2005), variational formulation (Daganzo, 2005a,b), and Hamilton-Jacobi based formulations (Laval and Leclercq, 2013).

In addition to LWR-based models, there are other traffic flow models that can be used to model urban traffic networks with signalized intersections. Compared with LWR models, which are usually referred to as physical or spatial queue models, point-queue models have simpler traffic state representations and dynamics because they ignore vehicle lengths. For example, Aboudolas et al. (2009) proposed the store-and-forward model as well as different traffic signal optimization formulations based on it. Due to its simplicity, it is also used by most pressure-based control methods for the theoretical derivation of network stability (Varaiya, 2013; Wang et al., 2022b; Levin, 2023). Queueing models are another family of point-queue models which are more often used to study steady-state traffic performance (Van Woensel and Vandaele, 2007; Viti and Van Zuylen, 2010; Osorio and Bierlaire, 2009; Flötteröd and Osorio, 2017; Boon and van Leeuwaarden, 2018; Oblakova, 2019). Besides, Wu and Liu (2011) proposed a shockwave profile model by tracking the different shockwaves of each movement.

Most of these traffic flow models, except for the queueing models, are deterministic. However, in the real world, both traffic demand and driving behavior are stochastic. Compared with deterministic traffic flow models, stochastic traffic flow models are more realistic and can be easily used for stochastic traffic state estimation with incomplete or flawed observations. Therefore, researchers have spent much effort developing different stochastic traffic flow models. For example, Sumalee et al. (2011) and Flötteröd and Osorio (2017) proposed the stochastic version of the cell and link transmission model, respectively. Jabari and Liu (2012) proposed a stochastic traffic flow model in which the randomness originated from the drivers' gap choice. Jabari and Liu (2013) also derived the Gaussian approximation of the model and utilized it for traffic state estimation using loop detector data. While most of these models were established based on Eulerian coordinates, Zheng et al. (2018) proposed a stochastic traffic flow model based on Lagrangian coordinates.

Although different stochastic traffic flow models have been proposed, they can hardly be used for traffic state estimation based on vehicle trajectory data with a certain penetration rate. LWR models are built based on the Eulerian coordinates, which split the spatial-temporal space into grids and define the traffic state as the density in each grid. Trajectory data does not provide measurements in Eulerian coordinates and hence cannot be directly used to calibrate the traditional LWR models. Besides, LWR models like the cell transmission model already have a high dimension by splitting the roadway into cells. It becomes much more complicated when it is extended to a stochastic setting. Queueing models can either be directly used since they ignore the length of the vehicle and cannot model the spatial propagation or distribution of the vehicles.

1.3.3 Traffic state estimation using vehicle trajectories

Readers can refer to Guo et al. (2019) and Maripini et al. (2023) for a more complete review of traffic state estimation with vehicle trajectory data. Existing methods can be roughly divided into shockwave-based methods and statistical estimation methods. For shockwave-based methods, the basic idea is to detect the shockwave in the time-space diagram and use shockwave theory (Light and Whitham, 1955; Richards, 1956; Newell, 2002) to estimate the traffic state (Cheng et al., 2011; Ban et al., 2011; Hao et al., 2012). One of the typical works is from Cheng et al. (2011), which used a classification method to detect the featured points when the observed trajectories change their motions to construct the shockwave in the time-space diagram and used the shockwave to estimate the cycle-by-cycle queue length. Instead of directly constructing the shockwave by using the featured points from the observed trajectory, Ban et al. (2011); Hao et al. (2012) used travel time to construct the shockwaves and estimate the queue length and signal timing plan. However, these shockwave-based methods are deterministic estimation methods. They cannot utilize prior information in historical data and thereby cannot provide a reliable estimation result in a low penetration rate environment.

Most stochastic estimation methods are formulated based on the stop locations of connected vehicles: the observed stopped connected vehicles at certain snapshots are used as the input to estimate the unknown parameters or states (Comert and Cetin, 2009; Comert, 2016; Zheng and Liu, 2017; Wong et al., 2019; Zhao et al., 2019a,b, 2021). These methods are derived based on different assumptions such as the Poisson (Zheng and Liu, 2017) or general arrival (Zhao et al., 2019b) processes and independent (Zhao et al., 2019b) or correlated (Zhao et al., 2021) queue length distribution. The intuition of these studies is similar although different assumptions are adopted. Compared with deterministic estimation methods (Sun and Ban, 2013; Ramezani and Geroliminis, 2015; Ban et al., 2011), stochastic estimation methods can better utilize prior information and also provide stochastic estimation results including estimation uncertainty.

However, most of the existing stochastic estimation methods only look into the stop locations of collected vehicle trajectories at certain snapshots (Comert and Cetin, 2009, 2011; Zhao et al., 2019a,b, 2021). They did not perform the traffic state estimation based on a stochastic traffic flow model. Consequently, these statistical estimation methods exhibit certain limitations:

1. They only utilize stop locations at specific time slots, failing to fully leverage additional available information such as stop duration and the time at which vehicles join the queue.

2. Due to the absence of a dynamic model, these methods struggle to incorporate cycle-by-cycle correlations, particularly when the vehicle is not completely cleared within a single cycle.

3. Typically, these methods offer only specific estimated values, such as penetration rate and traffic volumes, without providing a complete spatial-temporal traffic state.

4. These methods cannot be directly used for traffic state prediction due to the lack of traffic dynamics.

As a result, these limitations hinder the effectiveness and completeness of the statistical estimation methods, highlighting the need for better approaches to address these shortcomings.

1.3.4 Traffic signal control and optimization

There are many survey papers on traffic signal control and optimization Guo et al. (2019); Wei et al. (2019); Li et al. (2023). Besides, readers can also refer to the traffic signal manual (Urbanik et al., 2015) for a more comprehensive introduction.

According to the responsiveness and complexity, traffic signal control systems can be divided into three categories: 1) fixed-time; 2) vehicle-actuated; and 3) adaptive control. Fixed-time control is usually used by intersections without any detection capability. For these intersections, traffic signal timing parameters are pre-determined and optimized by using offline historical data. Many offline tools, such as SYNCHRO (Husch and Albeck, 2004), TRANSYT-7F (Hale, 2005), and PASSERTM V (Chaudhary and Chu, 2002), can be used to generate the offline signal timing parameters. However, the main limitation of fixed-time control is that it cannot respond to the time-varying traffic demand. Vehicle-actuated control overcomes this problem by applying a more responsive rule-based strategy using the data from detectors (Urbanik et al., 2015). The vehicle-actuated control keeps the same phase if the headway between vehicles is less than a certain threshold, subjecting to the minimum and maximum green at the same time.

Compared with vehicle-actuated control which is a rule-based control with fixed parameters, adaptive traffic signal control is usually built based on certain traffic models and parameter selection (optimization) programs. Therefore, it is more flexible and complicated. The most commonly used adaptive signal control systems in the world include SCOOT (Hunt et al., 1981) and SCATS (Lowrie, 1990). Besides, there are also other adaptive traffic signal control systems such as OPAC and RHODES (Gartner, 1983; Mirchandani and Head, 2001). However, such adaptive signal control systems are not commonly deployed due to the computational complexity and hardware installation requirements Zhang and Wang (2010).

During the past decades, traffic signal control and optimization continue drawing attention from researchers. Different methods have been used for both fixed-time and real-time traffic signal optimization, including rule-based methods, model-based optimal control methods, and reinforcement learning (RL) methods, etc. The vehicle-actuated control, as aforementioned,

is a rule-based control method with given pre-determined parameters. Except for the rule-based methods, optimization-based methods are also frequently used to determine traffic signal parameters or states. Rolling-horizon optimization (i.e., receding-horizon optimization, model predictive control) has been widely used to formulate the real-time traffic signal control problem (Lo, 1999; Wada et al., 2017; Li and Ban, 2018), which minimizes the total delay of the system while subjecting to the traffic flow model and traffic signal constraints.

Reinforcement Learning (RL) has become a popular approach for traffic signal control, as evidenced by several studies (Arel et al., 2010; Khamis and Gomaa, 2014; Yau et al., 2017; Chu et al., 2019; Wei et al., 2019). RL can directly learn an end-to-end control policy from the observation by interacting with the simulation environment iteratively. Most of the existing literature using RL for traffic signal control focuses on the design of the input state space and reward (Wei et al., 2019) while utilizing different RL techniques such as the multi-agent algorithms (Chu et al., 2019). Despite the abundance of research utilizing RL for traffic signal optimization, there remains a significant gap before deploying these methods in the real world. One of the main concerns is the reliability of RL-based approaches. RL controllers trained offline in a simulation environment may not perform well in real-world scenarios due to the limited fidelity of the simulation. On the other hand, training RL controllers directly in the real world raises additional challenges, particularly in managing the risks associated with exploration during the learning process.

The development of connected and automated vehicles (CAV) brings new challenges and opportunities for traffic signal control and optimization. Both automated vehicles (AV) and connected vehicle (CV) could serve as mobile sensors, which provide data such as vehicle trajectories and their observations (for AVs with detection ability) that can be used to optimize the traffic signal operation (Feng et al., 2015). Moreover, AVs have the potential to serve as moving regulators to further improve the stability of the traffic flows (Feng et al., 2018; Yu et al., 2018; Stern et al., 2018). Readers can refer to survey papers written by (Guo et al., 2019) and Li et al. (2023) for a more comprehensive review. Although many studies have proposed different signal control methods with CAV and demonstrated promising results in the simulation environment, they can hardly be used in the field since the current market penetration rates for both CV and AV are much less than what is assumed in these research studies.

1.4 Bottlenecks of using trajectory data

Based on the characteristics of the trajectory data as well as the status of the existing literature, this section discusses the main bottlenecks of using vehicle trajectory data for traffic signal optimization.

1.4.1 Limited penetration rate

The main limitation of vehicle trajectory data is the sparse and incomplete observation caused by the current market penetration rate (usually below 10%). It is challenging to estimate or reconstruct the overall traffic state only with sparsely observed vehicle trajectory data. Although some studies have developed statistical methods to estimate certain traffic states or parameters such as queue length, penetration rate, etc. They are not built based on a traffic flow model and hence have certain limitations as aforementioned (Section 1.3.3).

1.4.2 Lack of a suitable traffic flow model

The utilization of stochastic traffic flow models enables the estimation and prediction of overall traffic states based on incomplete observations. However, the majority of existing traffic flow models mentioned in Section 1.3.2 are not well-suited for vehicle trajectory observations. These models primarily operate using Eulerian and Lagrangian coordinates, which are the two commonly used coordinate systems to describe traffic state.

Eulerian coordinates involve dividing the spatial-temporal space into grids and defining the traffic state as the density within each grid. However, trajectory data does not provide measurements in Eulerian coordinates, making it challenging to directly calibrate traditional models like the LWR model and its variations. Vehicle trajectory data, on the other hand, is represented in Lagrangian coordinates, which track the movement of individual vehicles. However, traffic flow models based on Lagrangian coordinates suffer from high dimensions and are not easily applicable to large-scale scenarios. Moreover, models utilizing both Eulerian and Lagrangian coordinates become much more complicated at higher dimensions when extended to stochastic settings.

Consequently, there is a need for a new stochastic traffic model specifically designed to accommodate vehicle trajectory data. This would allow for the closure of the right-hand-side loop illustrated in Figure 1.3, completing the integration of trajectory data into the overall traffic modeling framework.

1.4.3 Difficulties for real-world implementation

Other than the methodological challenges mentioned before, there are also other difficulties in utilizing vehicle trajectory data for real-world implementations. One of the main challenges is the preprocessing of trajectory data. Raw trajectory data cannot be directly used without being matched to traffic networks. Map data, signal phase and timing (SPaT) data, and trajectory data need to be integrated together to support different types of applications. There is no such toolkit

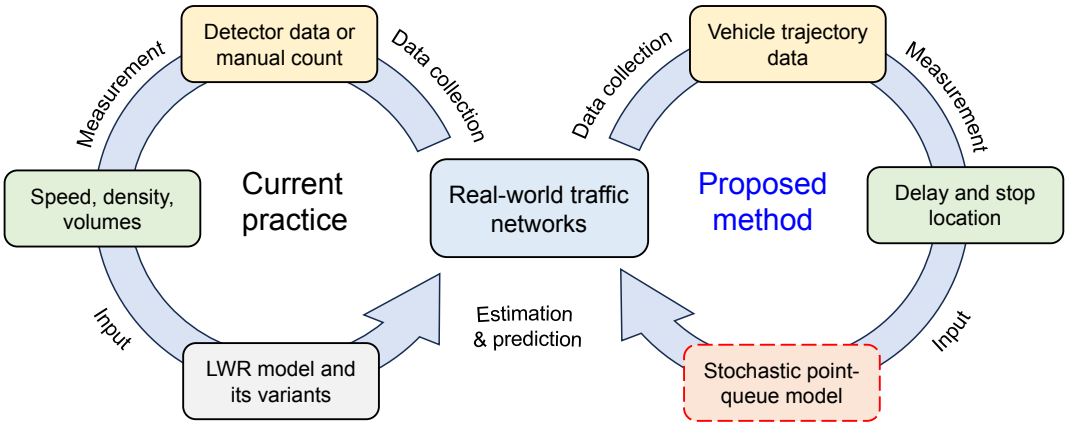


Figure 1.3: Current practice and the proposed method.

or software available that provides essential processing including matching the trajectory data to a large-scale movement-level traffic network.

In addition, real-world traffic networks present numerous corner cases, including intricate road topology or geometry, as well as the impact of side streets like on-street parking, pick-up and drop-off activities, and other factors. Furthermore, different traffic management agencies or stakeholders may have varying preferences or priorities. Therefore, it is essential for the system to be flexible and customizable to accommodate diverse needs and requirements.

1.5 Dissertation overview

1.5.1 Research scope

This dissertation aims at providing generic and systematic methods for traffic modeling, traffic state estimation, and traffic signal optimization with vehicle trajectory data at the current market penetration rate. We propose a novel stochastic traffic flow model, which is essentially a point-queue model under newly proposed "Newellian coordinates". This point-queue model can be transformed to obtain the spatial-temporal traffic state through the PTS diagram. Consequently, it is demonstrated that a simple point-queue model with lower dimensions can sufficiently capture the spatial-temporal traffic state. Besides, the proposed traffic flow model builds the connection between observed vehicle trajectory data with unknown traffic states and parameters, which enables us to apply different statistical estimation methods to estimate these unknown values. Based on the same calibrated traffic flow model with estimated traffic state and parameters, we also develop different optimization programs for the re-timing of fixed-time traffic signals and a rule-based QCC for real-time traffic signal control.

With the proposed methods, we develop a complete integrated traffic signal re-timing system

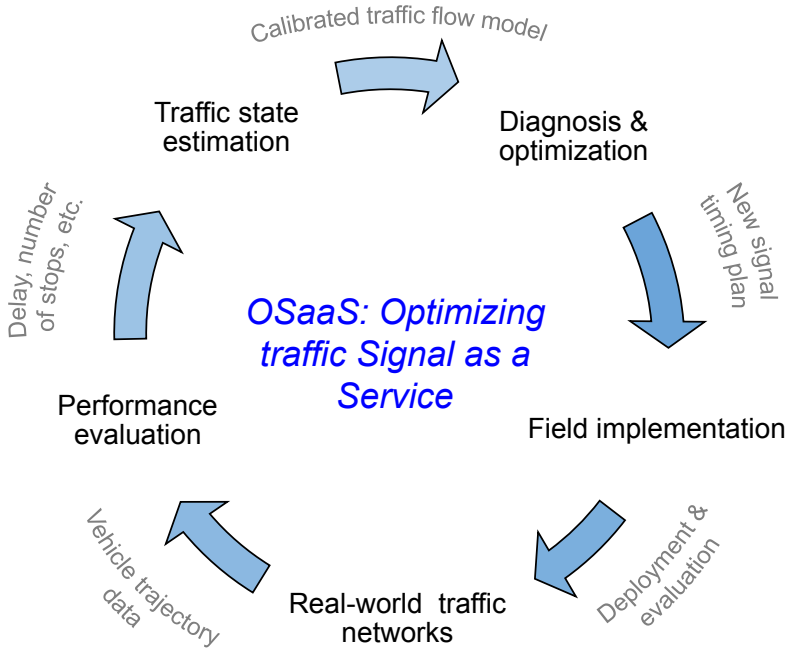


Figure 1.4: OSaaS (Optimizing traffic Signal as a Service) system framework.

called OSaaS. As shown by Figure 1.4, OSaaS is a closed-loop iterative system including performance evaluation, traffic state estimation, traffic signal diagnosis, and optimization. The system is able to diagnose specific congestion of causes at these intersections such as green split imbalances, offset issues, inefficient cycle lengths, and suboptimal TOD boundaries. The new signal timing plans are based on the existing signal timing plan while moving a certain step toward the direction guided by the diagnostic results. OSaaS significantly shortens each re-timing iteration, so a more responsive and strategic traffic signal re-timing is feasible. By not requiring the installation or maintenance of vehicle detectors, it provides a more scalable, sustainable, resilient, and efficient solution to traffic signal re-timing based on vehicle trajectory, which could be applied to every traffic signal in the world.

1.5.2 Contributions

The contributions of this dissertation are listed as the following:

1. We propose a novel stochastic traffic flow model (a point-queue model) under Newellian coordinates. Through the PTS diagram, the point-queue model with much lower dimensions can describe the complete spatial-temporal traffic state.

2. Utilizing the proposed traffic flow model, we apply different statistical estimation methods to

estimate both the traffic state and parameter using the low penetration rate vehicle trajectory data. We also quantify the uncertainty of all these estimated values.

3. We design different optimization programs to generate new signal timing plans for fixed-time traffic signals and a rule-based QCC for real-time traffic signal control.

4. We develop a complete system called OSaaS (Figure 1.4) and test it in the field, which showed improvement for both delay and number of stops for the implemented intersections.

In summary, this dissertation provides generic and comprehensive methods as well as an integrated system for traffic signal optimization with sparsely observed vehicle trajectory data. The right-hand-side loop in Figure 1.3 is closed with this dissertation.

1.5.3 Overview and organization of this dissertation

Other than this introduction Chapter 1 and the final summary Chapter 7. There are five chapters that cover content from performance evaluation, traffic flow model, traffic state estimation, and traffic signal optimization (Figure 1.5). These chapters together make the complete OSaaS system in Figure 1.4.

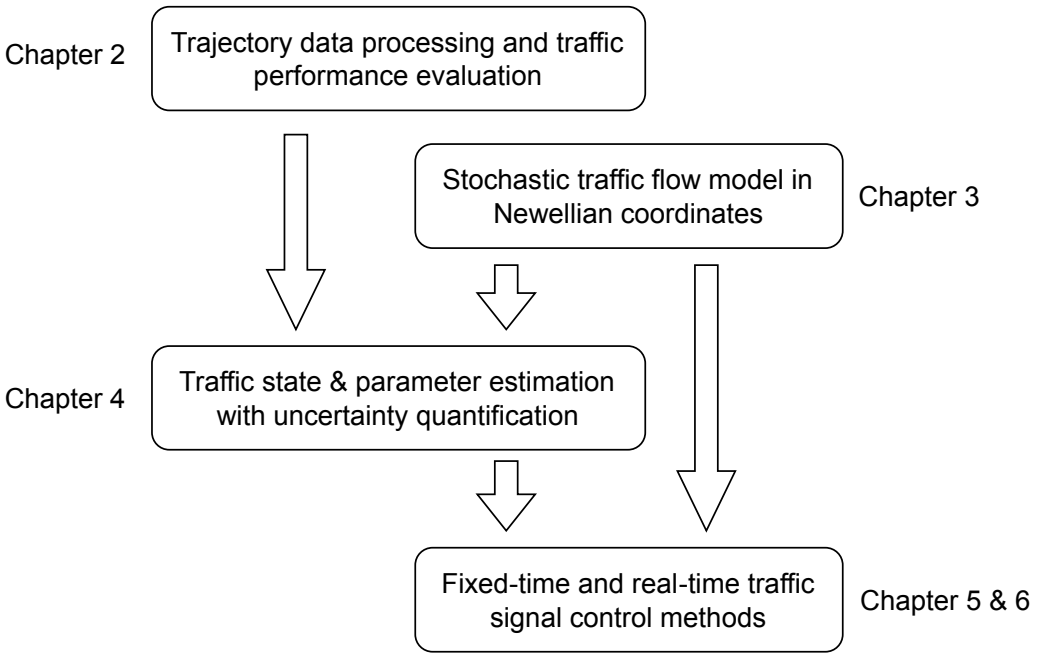


Figure 1.5: Dissertation overview.

Chapter 2 introduces the trajectory preprocessing and calculation platform we have built that helps to process the raw vehicle trajectory data to support different traffic operational applications

in large-scale traffic networks. The developed platform integrates map data, SPaT, and vehicle trajectory by matching both SPaT and vehicle trajectories to a well-defined network representation. Besides, we also develop robust algorithms to calculate traffic performance measurements such as control delay and number of stops from trajectories with noise and errors. This trajectory data processing platform is the foundation for all the remaining studies that use real-world trajectory data as the input.

Chapter 3 introduces the proposed stochastic traffic flow model. By assuming that all vehicles follow a deterministic Newell's car-following model, we establish the Newellian coordinates, which enables us to use a point-queue representation to describe the spatial-temporal traffic state. The PTS diagram is then derived to obtain the spatial-temporal trajectory distribution from the stochastic point-queue representation. This stochastic traffic flow model connects the sparsely observed vehicle trajectory data with unknown traffic states and parameters, making it possible to estimate all these unknown values with observed vehicle trajectories.

Chapter 4 shows how the traffic state and parameter can be estimated by utilizing statistical estimation methods with the proposed stochastic traffic flow model. For fixed-time traffic signals, by assuming stationary traffic parameters within a certain TOD period, we apply the method of moments (MM) estimator to estimate both penetration rate and arrival rate. It is demonstrated that, even at a low penetration rate, we can accumulate multi-day historical data to reconstruct the recurrent traffic state. Besides, we also utilize the Bayesian estimation methods to estimate both the stationary traffic parameters and cycle-by-cycle traffic state. The posterior distribution obtained by Bayesian methods can directly quantify the uncertainty of all the estimated values.

Chapter 5 and 6 focus on traffic signal control for fixed-time and real-time traffic signals, respectively. For fixed-time traffic signals, we develop essential programs to identify the optimality gap with different traffic signal parameters and generate new signal timing plans for those intersections with corresponding issues. We also did a field implementation for the re-timing of fixed-time traffic signals in the city of Birmingham, Michigan. Implementation of new signal timing plans resulted in significant reductions in control delay and the number of stops at both isolated intersections and corridors. For real-time traffic signals, we also design a rule-based controller called QCC. A simulation environment with an isolated intersection is used to demonstrate the effectiveness of the proposed controller.

CHAPTER 2

Trajectory Preprocessing and Traffic Performance Evaluation

2.1 Introduction

2.1.1 Background and related works

In recent years, vehicle trajectory data from different sources such as connected vehicles, ride-hailing services, and online navigation software has become readily available. Such data can be utilized for different applications including traffic state estimation and safety evaluations. However, it is difficult to use raw trajectory data without matching them to traffic networks. In addition, noise and errors in the real-world trajectory data need to be filtered and smoothed.

As aforementioned in Section 1.3.1, although there are many existing studies that focus on the preprocessing of the trajectory data and map data. There is no existing software or toolkit that can process the trajectory data to support movement-level traffic operational applications. It is not trivial to process the trajectory and map data to a usable format. The absence of such a toolkit causes a major bottleneck that prevents researchers use trajectory data for different applications in a large-scale traffic network.

Besides, traffic performance evaluation is one of the most important applications of trajectory data (Herrera et al., 2010; Lu et al., 2017; Saldivar-Carranza et al., 2021). It provides guidance for urban traffic control and helps traffic engineers quickly pinpoint critical locations for further analysis. For example, Saldivar-Carranza et al. (2021) utilized vehicle delay and a number of stops to generate different visualization plots. Herrera et al. (2010) conducted a comprehensive field experiment including trajectory data collection and travel speed estimation using the collected data.

2.1.2 Overview of the chapter

In this chapter, we develop a comprehensive trajectory data processing platform to serve different traffic operational applications in large-scale traffic networks. The trajectory processing pipeline includes several main steps: matching the raw trajectory data to a well-defined network representation, extracting distance information from the raw GNSS coordinates, and splitting each vehicle trip into different movements at each signalized intersection. Essential smoothing and filtering algorithms are also required which can reduce noise and errors in real-world data.

We also provide a robust calculation of standard traffic performance measurements (control delay, number of stops, etc.) from trajectory data with noise and errors. Both the trajectory data processing methods and the performance index calculation algorithms are designed with an emphasis on scalability and robustness. In this way, we can extract high-quality trajectory and performance indices from large-scale raw noisy trajectory data.

2.1.3 Contributions and organization of the chapter

The main contributions of this chapter are twofold:

1. A complete trajectory data processing pipeline that can support different traffic operational applications, especially for movement-level traffic signal performance evaluation.

2. Robust and scalable algorithms for calculating standard traffic performance indices including control/stop delay, number of stops, queue distance, and space-mean speed.

This chapter is organized as follows: Section 2.2 introduces the trajectory data and urban traffic network representation. Section 2.3 introduces the trajectory data processing procedure including trajectory data map matching, trajectory splitting, basic smoothing and filtering algorithms, etc. With the processed trajectory data, Section 2.4 introduces the algorithms that calculate different traffic performance metrics. Section 2.5 includes case studies that verify the proposed methods and algorithms. We conclude this chapter in Section 2.6.

2.2 Trajectory data in urban traffic networks

2.2.1 Vehicle trajectory data requirements

There is no specific requirement with regard to either the type or resource of trajectory data, it only needs to follow the requirements below:

1. **Attributes** Each trajectory point should have at minimum a unique trajectory ID, timestamp, and geometric coordinates (e.g., latitude and longitude). Additional attributes such as velocity and acceleration are preferred but not required.

2. **Resolution** The temporal resolution should be within a few seconds depending on the application of interest; otherwise, additional interpolation might be required and the algorithm's performance will suffer. For most traffic operational applications (instead of safety-related applications) in this dissertation, $1-5$ seconds should be sufficient.

3. **Location** This dissertation mainly deals with trajectories in urban traffic networks.

4. **Accuracy** The accuracy of the GNSS coordinates does not affect the overall processing methods. Although higher accuracy is always preferable, a standard derivation within $3-5$ meters is sufficient for traffic performance evaluation purposes.

2.2.2 Network representation

It is difficult to use the raw vehicle trajectory data if it is not matched to a traffic network. Network representation and intersection geometry are the basis for urban traffic network applications. In this dissertation, we design our own urban traffic network representation composed of different basic elements including links, segments, and junctions as shown in Figure 2.1. Generally, an urban traffic network is a directed graph composed of junctions and links $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$. The junction set $\mathcal{N}$ can be further categorized into end junctions $\mathcal{N}^e$ and intersection junctions $\mathcal{N}^i$, i.e., $\mathcal{N} = \mathcal{N}^e \cup \mathcal{N}^i$. Intersection junctions include signalized and unsignalized intersections, while end junctions are entry or exit points of the network. As illustrated by Figure 2.1, junction n is an example of an intersection junction while junction m is an example of an end junction. A link is defined as a directed road that connects junctions, shown by the blue lines in Figure 2.1.

As shown in Figure 2.1, a link that connects two junctions can be further divided into a list of segments. A segment is defined as a road segment that has homogeneous road parameters such as road class, number of lanes, speed limit, etc. For example, the link p in Figure 2.1 is split into two segments i and $i+1$ since there is an additional dedicated left-turn lane as it approaches the junction.

We also define a movement as a pair of links at an intersection: one upstream of the intersection and the other downstream. This definition of movement is consistent with the NEMA dual ring structure for signalized intersections (Koonce and Rodegerdts, 2008). In most cases, each movement at a signalized intersection corresponds to a traffic signal phase. A standard four-leg intersection has four through movements and four left-turn movements.

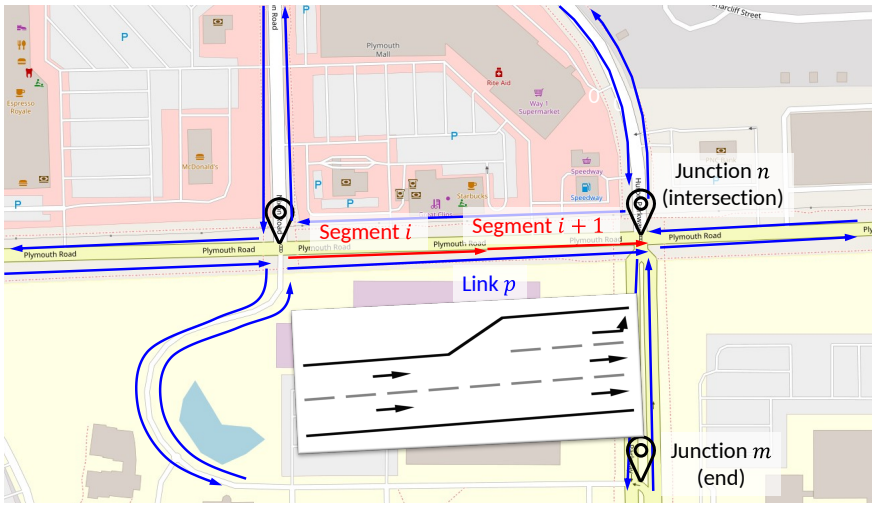


Figure 2.1: Urban traffic network representation

We do not match the trajectory data to the lane-level network but only movement level since it would already be sufficient for most offline traffic operational applications. Therefore, lane-level map data is not required. We use OpenStreetMap (2019) as the raw map data in this dissertation. In general, efficiency-related studies do not require a high-definition map compared with safety-related studies. In the following section, we will also see that, for high-resolution trajectory data, the map data is only used for trajectory map matching and providing the location of each intersection, the accuracy of the map geometry does not influence the calculation of the distance from the raw GNSS trajectory data.

2.3 Trajectory data processing

Figure 2.2 illustrates the overall pipeline of the trajectory data processing. The input is the raw trajectory data given by raw GNSS coordinates (latitude and longitude); the output is the matched trajectory points with map and distance information. There are three main trajectory processing procedures as shown in the pipeline including trajectory data map matching, splitting the trajectory into movements, and extracting distance information from the raw GNSS coordinates. There are also several data smoothing/filtering steps labeled over the arrows in the figure. In this section, we will first go through the three main procedures and then briefly discuss about data smoothing and filtering.

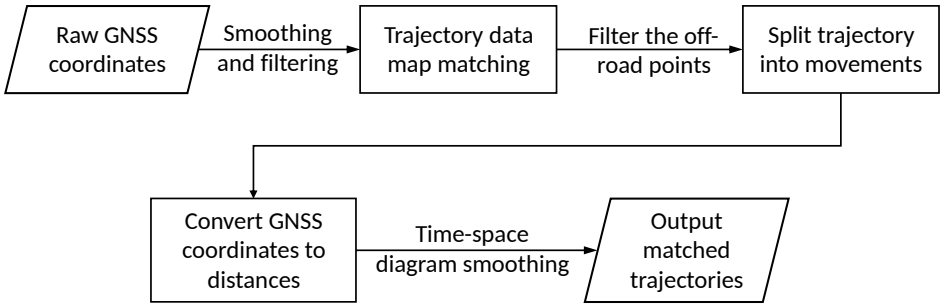


Figure 2.2: Overall procedure of the trajectory data processing

2.3.1 Trajectory data map matching

Trajectory data map matching matches raw trajectory data to the urban traffic network according to the GNSS coordinates. To match the trajectory data to the network representation introduced in the previous section, we only need to match each trajectory point to a segment. With the segment information, it is easy to add all the other network components such as link, segment, and upstream/downstream junctions to each trajectory point.

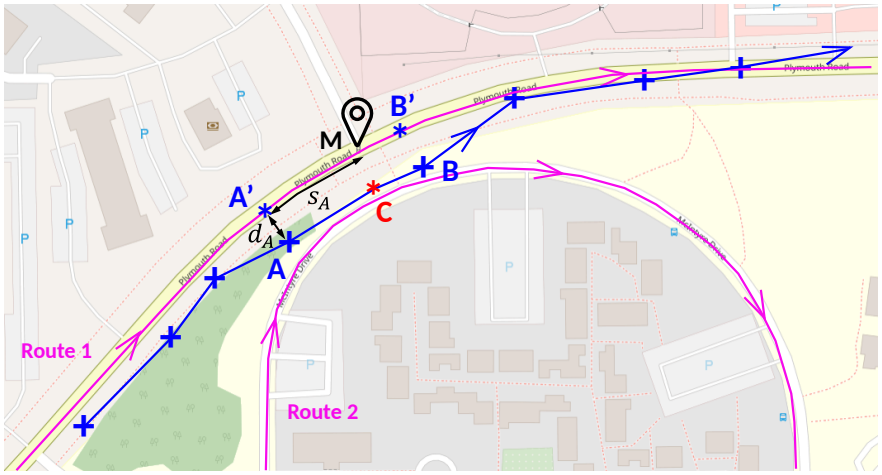


Figure 2.3: Hidden Markov trajectory map matching model.

The hidden Markov model is the most commonly used formulation for trajectory data map matching (Newson and Krumm, 2009). The goal is to find the most likely route considering both the distance of the raw trajectory points to the route and path feasibility (Figure 2.3). The blue line and the blue cross points are raw trajectory points while the two pink lines are candidate routes. If we only consider the distance between the trajectory and the candidate route, Point A and Point B will be assigned to Route 2. However, this violates path feasibility since clearly the trajectory belongs to Route 1 prior to Point A and it cannot jump to Route 2 directly. With the matched segment for each trajectory point, it is easy to find the matched points (A' and B' in Figure 2.3) of

the original trajectory (the closest points on the roadway to the original GNSS points A and B). The distance between the matched trajectory point A' and the raw GNSS point A is denoted as d_A. We can also calculate the distance between the matched trajectory point to the upstream/downstream junctions as shown by s_A in the figure.

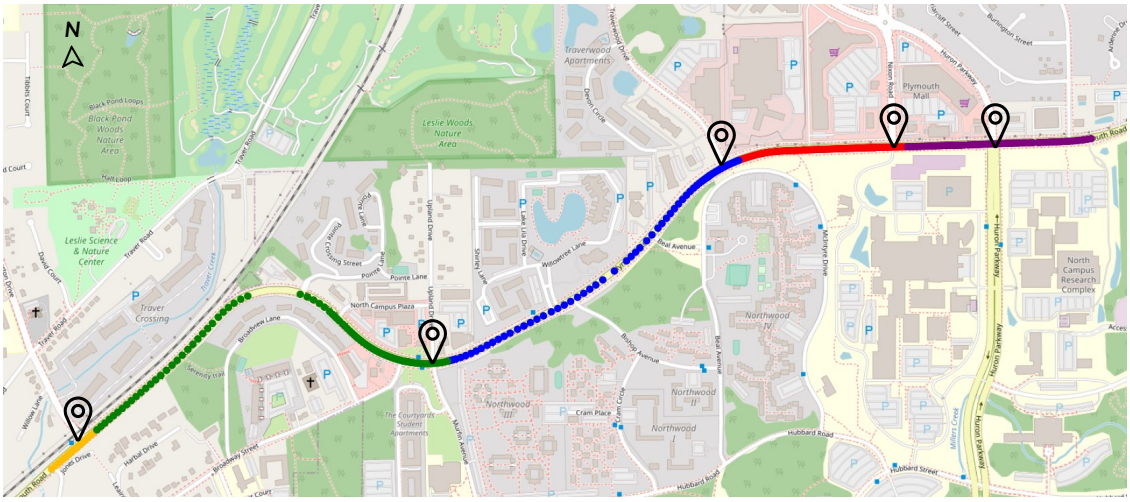
For more details on trajectory map matching, refer to Newson and Krumm (2009). If the trajectory data is collected within a certain range of a specific intersection, like BSM collected from a RSU through DSRC, a simpler algorithm proposed in Wang et al. (2020) can be directly used. In this dissertation, we implement the method proposed by Yang and Gidofalvi (2018). The underlying philosophies are the same for all mentioned map-matching algorithms. In summary, map matching can provide the complete network information of each trajectory point including its segment, link, upstream/downstream junctions, matched GNSS coordinates, distance to the matched point, and the distances to the upstream or downstream junctions. In this dissertation, we do not match the trajectory data to the lane level since the algorithms and methods proposed in this dissertation do not rely on lane information.

2.3.2 Split trajectories into movements

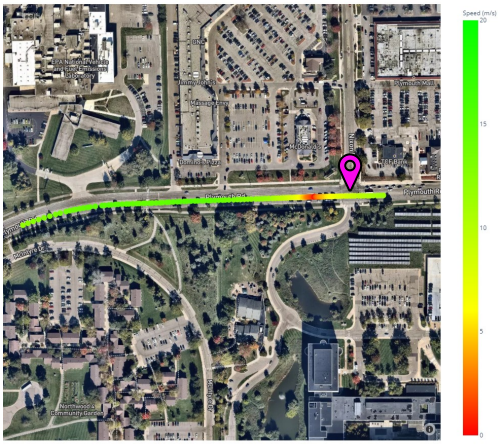
With all additional network information added to each trajectory point through the trajectory data map matching, we can split trajectories into different movements. A movement is defined as an upstream and downstream link pair connected by a junction. The method to split the trajectories into different movements is straightforward: whenever the trajectory traverses a junction over a certain distance, it will enter a new link from the upstream link; then we can truncate the trajectory and assign it to the corresponding movement.

Figure 2.4 (a) is an illustration of trajectory splitting. The example trajectory in the figure traveled across the whole arterial from the west to the east and markers in the map are locations of signalized intersections (junctions). The complete trajectory is split into different movements labeled as different colors; it is truncated whenever it passes the junction over 20 meters.

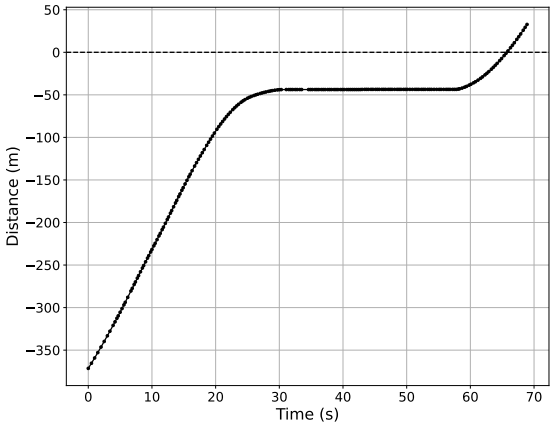
Splitting trajectories into different movements provides convenience for movement-level analysis of signalized intersections. As aforementioned, the movement defined in urban networks follows NEMA dual ring structure; generally, each movement will correspond to a certain traffic signal phase. By associating both trajectory data and SPaT data to the corresponding movement, we can get the controlled signal state for each given trajectory. We do not necessarily need to know the lane information to get the controlled signal state; the movement information inferred from the upstream/downstream links is sufficient.



(a) Split the trajectory into movements (trajectory traveled to the east)



(b) Trajectory points



(c) Time-space diagram

Figure 2.4: Convert GNSS coordinates to distance information after splitting trajectory into junctions.

2.3.3 GNSS coordinates to distance

After splitting trajectory data into different movements, we can convert original GNSS coordinates (latitude and longitude) to distance information. As shown before in Figure 2.1, trajectory data map matching can provide the matched trajectory point A' and distance s_A from A' to the downstream junction M. s_A can be roughly regarded as the distance to the downstream junction of Point A. However, this estimation method will have a large estimation error if the road geometry is not accurate enough. For example, the widely-used open-source map data (OpenStreetMap, 2019) usually does not have accurate geometry for different lanes but only the center of the roadway even for two-way roads.

Wang et al. (2020) provided another method to convert GNSS coordinates to distance which is suitable for high-resolution trajectory data. The intuition is to first calculate cumulative travel distance given GNSS coordinates and then set a certain point as zero distance point. For example, we can set the center of the intersection as the reference point and find the closest point in the trajectory as the zero distance point as shown by point C in Figure 2.3. In this way, the map data is only used for trajectory data map matching and providing the location of the intersection. The distance is calculated from the trajectory itself and it is not dependent on the map geometry. Therefore, the accuracy of the distance will not be influenced by the map data.

Figure 2.4 (b-c) shows an example of converting the original GNSS coordinates to distance using such method. The example trajectory traveled from the west to the east while the color shows the velocity of each trajectory point; the pink marker is the center of junction which is set as the zero distance point. Figure 2.4 (c) shows the time-space diagram, i.e., distance to the center of the intersection with respect to time; a negative distance corresponds to the upstream of the junction while a positive distance corresponds to the downstream. For traffic signal evaluation, we might need the distance to stopbar instead of the center of the intersection. It would be easy to either directly set the location of the stopbar as the zero distance reference point or shift the time-space diagram by the distance between the stopbar and the center of the intersection.

2.3.4 Data filtering and smoothing

Data smoothing and filtering are important for real-world data with noise and errors. Since the choice of the smoothing and filtering methods are largely dependent on different features of different data sources, we will not go to details but briefly introduce some essential data filtering and smoothing steps shown in Figure 2.2.

Raw trajectory point filtering and smoothing Given the raw trajectory points in GNSS coordinates, we can first remove outliers and then apply some basic smoothing algorithms. The outlier removal algorithm is usually developed based on specific causes of the error from the different data sources. For the data smoothing, there are many choices such as the median filter, Gaussian filter, local regression, etc. In this dissertation, the outlier points are detected as points with large spatial or temporal shifts and a simple Gaussian filter is applied to smooth the raw GNSS coordinates.

Remove off-road trajectories The trajectory map matching can provide the distance between the raw GNSS point to the matched point shown by d_A for point A in Figure 2.3. This distance can be used as a metric to evaluate the trajectory data map matching results. If network data is correct, a large distance d_A indicates that point A is away from the network roadways and could be removed.

Time-space diagram smoothing After converting the raw GNSS coordinates to distances, we can

apply another round of the data smoothing using the basic smoothing algorithms. More advanced algorithms such as the Kalman filter would perform better if the speed and acceleration are also available for each trajectory point.

2.4 Traffic performance evaluation

This section provides several algorithms to generate traffic performance metrics using the trajectory data processed before. These performance indices include vehicle delay, number of stops, and signal coordination-related measurements. We will also introduce space-mean speed estimation by using linear interpolation at last.

2.4.1 Trajectory state segmentation

Before we go to specific traffic performance index calculation, we will first introduce a trajectory state segmentation algorithm. The trajectory state segmentation algorithm splits the entire trajectory into different states including free-flow state, transition state, and stop state according to its speed profile. Free-flow state is defined as the state when a vehicle travels at a high speed while stop state is defined as the state when a vehicle stops; the transition state connects the free-flow state and stop state.

Figure 2.5 shows the details of the trajectory state segmentation algorithm. The overall algorithm includes two steps, the first step is to get the preliminary trajectory state split according to the speed profile; the second step is to filter the preliminary state split results. In the first step, a vehicle is considered to be in the stop state if the speed is less than a threshold v_s while in the free-flow state if the speed is larger than a threshold v_t; the state between the two thresholds is assigned to the transition state. The stop speed threshold v_s is chosen as $1\ m/s$ while the free-flow speed threshold v_t should be related to the speed limit of the roadway; in this dissertation, 80% of the speed limit is used.

After getting preliminary segmentation results according to the speed profile, the second step is vehicle state filtering. Vehicle state filtering is designed to improve the robustness of the overall vehicle state segmentation algorithm by removing some outliers such as short states caused by noise of the speed profile or abnormal driving behavior. Figure 2.5 shows details of the vehicle state filtering. Generally, there are two filtering steps for both stop state and free-flow state: state consolidation and short state removal. State consolidation is to combine two close states of the same category if the time and distance gap are both less than certain thresholds. As shown in Figure 2.5 point A, there are two stop states split by a short transition state; if the duration of the transition state and the distance traveled during this period are both less than predefined thresholds,

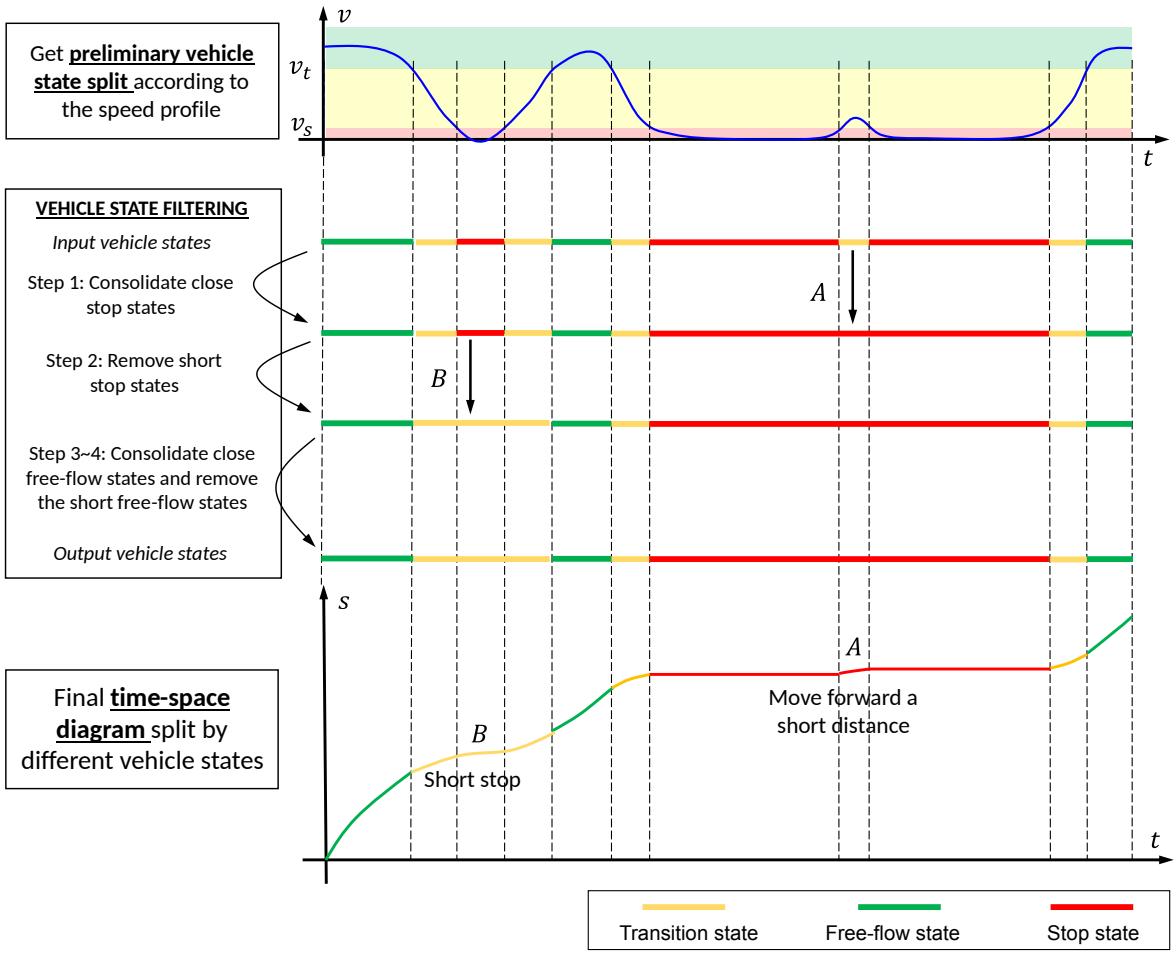


Figure 2.5: Illustration of trajectory state segmentation.

then the two stop states will be combined together as a longer stop. In this dissertation, the duration threshold is chosen as $3\ s$ while the distance gap threshold is chosen as $10\ m$. Short state removal is to convert short stop state to transition state as shown in Figure 2.5 point B. If the duration of the stop state is less than a certain threshold (which is chosen as $3\ s$), the stop state will be converted to a transition state. After the state consolidation and short state removal for stop state, we can do the same for free-free state. Then we will be able to get the final segmentation results shown by the time-space diagram at the bottom in Figure 2.5.

As aforementioned, vehicle state filtering is to improve the robustness of the trajectory segmentation algorithm. Stop state consolidation can help to avoid over-estimation of number of stop states. As shown in Figure 2.5, the stopped vehicle moved forward a short distance at point A; this could happen when the driver just slightly reduces the following distance with the leading vehicle. We should consider this situation as a long continuous stop instead of two stops. The short stop state removal is designed to reduce the influence of the noise; a valid stop state should at last

for a certain duration. More examples of vehicle state filtering will be discussed in the case study section.

2.4.2 Traffic performance index calculation

With the trajectory state segmentation introduced before, this subsection will introduce the traffic performance index calculation for each trajectory. Figure 2.6 is the illustration of the algorithm based on the trajectory segmentation results while Table 2.1 is the full list of the performance indices.

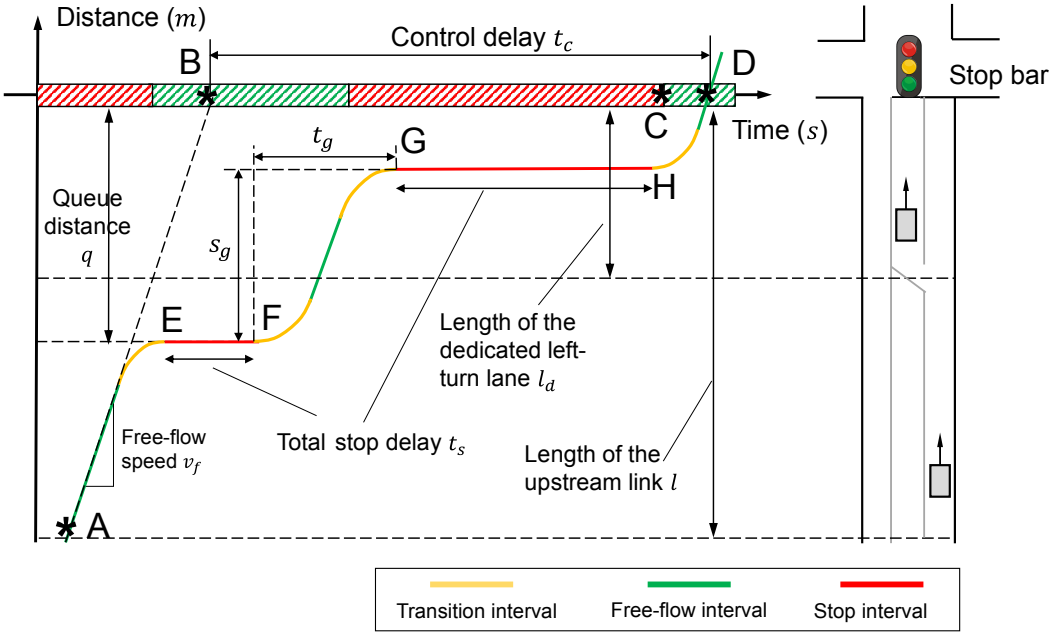


Figure 2.6: Traffic performance index calculation for single trajectory

Free-flow speed, free-flow arrival time Free-flow speed is defined as the desired speed of the vehicle when not blocked or influenced by the background traffic (Manual, 2010). Free-flow speed is an important parameter to estimate the control delay as well as whether a vehicle will arrive during the green time. Different drivers have different free-flow speeds due to various driving behaviors. With the trajectory segmentation results, free-flow speed can be estimated as the 80th percentile of the speed during all the free-flow states denoted by the green color in Figure 2.6. The 80th percentile is better than the average speed or the median value since the free-flow state might contain a small proportion of slowing-down or start-up period. With the free-flow speed, we can calculate the free-flow arrival time, which is defined as the estimated arrival time if the vehicle is not blocked by any background traffic or traffic signal. As shown in Figure 2.6, we can extend the

Name	Definition	Calculation (Figure 2.6)	Unit
Free-flow speed v_f	Desired speed of a vehicle not influenced by background traffic	80% percentile of the speed of free flow states	m/s
Free flow arrival time	Estimated arrival time traveled by free flow speed	t_B	s
Arrival on green	Whether a vehicle will arrive during green time	Signal state at point B	$\{0, 1\}$
Control delay d_c	Vehicle delay caused by traffic signal control	$t_D - t_B$	s
Level of service	Level of service from A to F according to HCM (Manual, 2010)	Determined by d_c	A-F
Stop delay d_s	Total stop duration before a vehicle passes intersection	Total duration of stop states	s
Number of stops n_s	Number of stops before a vehicle passes the intersection	Number of the stop states	1
Queue distance q	Maximum distance to the stop bar of all stop states	$\lvert s_E \rvert$	m
Split failure	Vehicle failed to pass the intersection within one cycle	Determined by n_s and d_c	$\{0, 1\}$
Spill-over warning	Queue distance is larger than 80% of upstream link or dedicated left-turn lane	Determined by q/l or q/l_d	$\{0, 1\}$

Table 2.1: Traffic performance index calculation based on trajectory segmentation

first trajectory point A by a dashed line with the slope as the free-flow speed to get the free-flow arrival time at point B. That is,

$$t_B = \frac{s_B - s_A}{v_f} = -\frac{s_A}{v_f} \tag{2.1}$$

where $s_B = 0$ if we set the location of the stopbar as the distance zero point.

Arrival on green A vehicle is referred to as *arrival on green* if it will arrive at the green time traveled by free-flow speed. It is determined by the traffic signal state at the free-flow arrival time t_B. Arrival on green is an important parameter to evaluate the coordination among intersections; good coordination will lead to a high proportion of vehicles that arrive at the green time.

Control delay, level of service Control delay is defined as the temporal difference between the actual travel time and free-flow travel time (Manual, 2010; Saldivar-Carranza et al., 2021). With the free-flow arrival time at point B, the control delay t_c can be easily calculated as:

$$t_c = t_D - t_B \tag{2.2}$$

where t_B is the actual time that the vehicle passes the stopbar. Based on control delay, highway capacity manual (HCM) (Manual, 2010) also provides the rating from A to F as level of service (LOS).

Stop delay, number of stops, and queue distance Vehicle stop is another important measurement for traffic performance. With the stop state obtained from the trajectory segmentation, total stop delay is calculated as the total duration of all stop states shown in Figure 2.6. The number of stops just equals the number of stop states while queue distance is defined as the distance to the stopbar of the first stop state.

Split failure Split failure occurs when green time is not enough for a certain movement. Split failure detection is important for traffic signal optimization since it usually leads to a large delay and might be improved significantly by optimizing the green split or the cycle length. The typical phenomenon of split failure is that a vehicle trajectory fails to pass the intersection within one cycle; as a result, the control delay of the vehicle will be larger than the red time of a cycle. Based on this observation, a trajectory is labeled as a split failure if both conditions are satisfied: 1) its control delay is larger than the red light duration; 2) number of stops is larger than 1.

Spill-over warning Spill-over refers to the situation when stopped vehicles occupy the whole roadway or dedicated left-turn lane. Spill-over could lead to gridlock of urban traffic networks, which would cause a significant capacity drop for the whole urban traffic network. Spill-over can be divided into left-turn spill-over and through movement spill-over. Left-turn spill-over occurs when stopped vehicles occupy the dedicated left-turn lane while through movement spill-over

occurs when stopped vehicles occupy the whole link. In this dissertation, we use queue length occupation rate to estimate the risk of left-turn spill-over and through movement spill-over. Queue length occupation rate is defined as the queue distance divided by length of dedicated left-turn lane or the entire link for the left-turn and through movement spill-over accordingly. If the queue length occupation rate for a certain trajectory is larger than 80%, we will post it as a spill-over warning.

2.4.3 Space-mean speed calculation

At last, we will introduce an algorithm to estimate space-mean speed; which is another important traffic performance measurement for urban traffic networks (Turner et al., 1998). As shown in Figure 2.7, the time-space diagram is split into different cells with a certain time interval Δt and distance interval Δs. Then we can apply linear interpolation to find the exact point that each trajectory traverses the boundaries of cells. After linear interpolation, we can calculate the total travel time t_c and the total travel distance s_c for all trajectories within each cell. With total travel time and total travel distance, the space-mean speed is calculated as:

$$v_c = \frac{s_c}{t_c}, \quad \forall c. \tag{2.3}$$

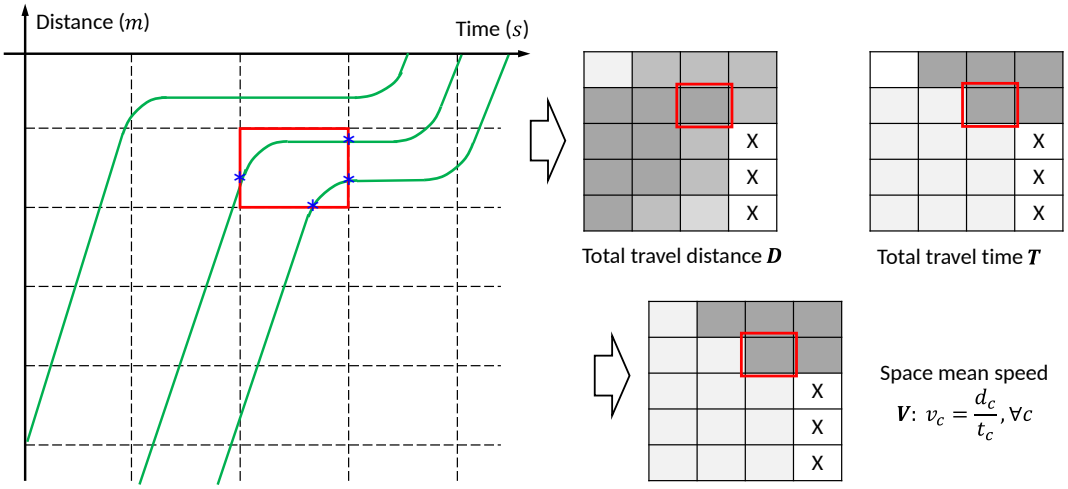


Figure 2.7: Trajectory grid interpolation and space mean speed calculation.

Based on this space-mean speed calculation, the speed heatmap can be plotted to visualize the overall traffic performance which will be introduced in the next section. Besides, the travel time of a certain route (e.g., a corridor) might also be estimated without requiring the trajectory to pass the complete route. We leave this for future study.

2.5 Results

The proposed data processing platform is tested for both AACVTE and GM vehicle trajectory data. Section 2.5.1 introduces typical cases for traffic performance index calculation. Section 2.5.2 and 2.5.3 show some plots for isolated movements and corridors accordingly. Results in Section 2.5.1 are based on AACVTE trajectory data while the other two are using GM trajectory data. An introduction to both data sets is available in Section 1.2.1.

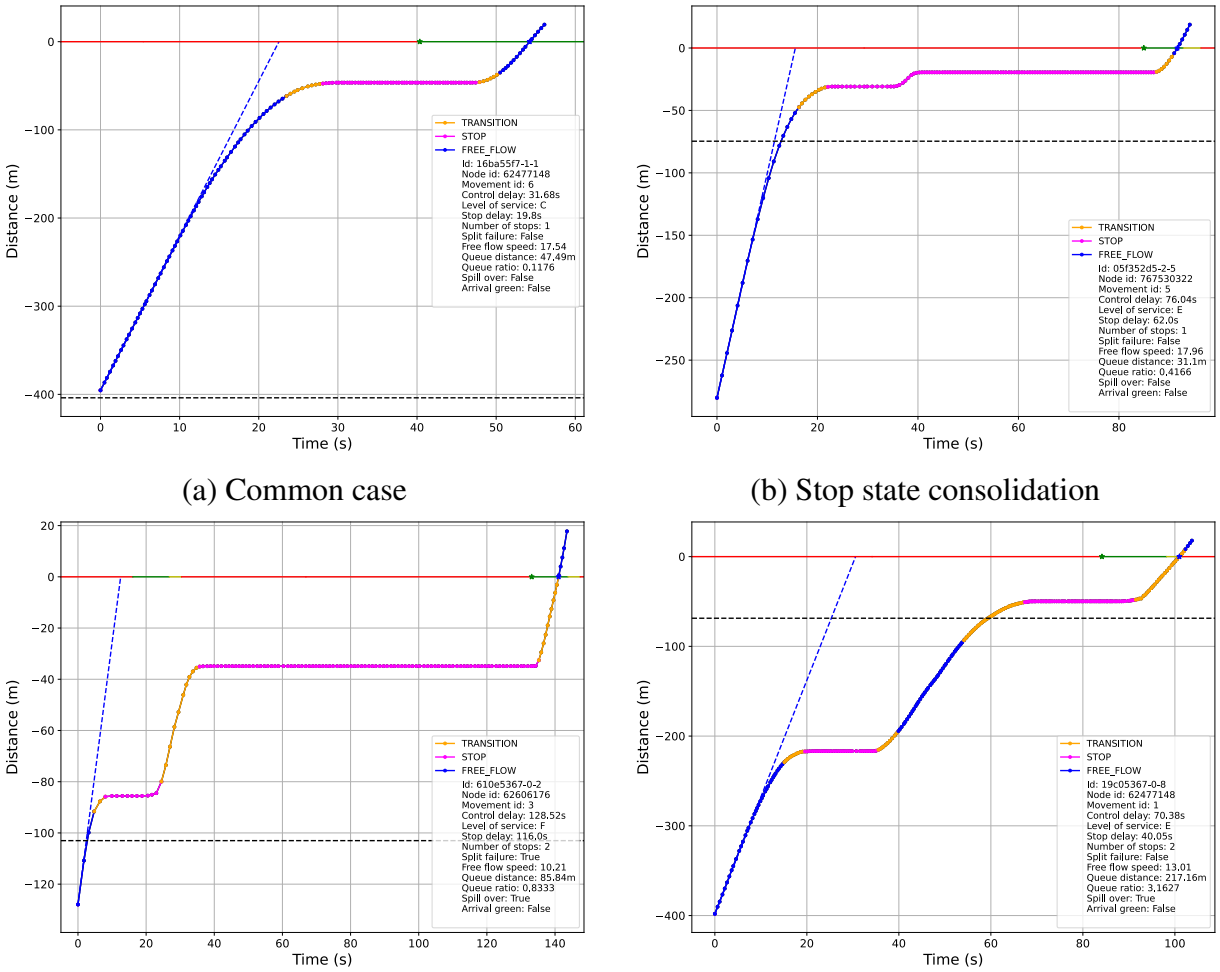
2.5.1 Typical cases of traffic performance index calculation

Figure 2.8 shows the time-space diagram of some typical cases for the traffic performance index calculation. Time-space diagrams are plotted by different colors representing different states including stop state, free-flow state, and transition state. For all four cases, distance 0 is the location of the stop bar while a negative distance indicates that the vehicle is from the upstream of the intersection. The dashed blue lines are free-flow arrival curves; dashed black horizontal lines show locations of the start of the dedicated left-turn lane or upstream link for left-turn and through movement accordingly. Traffic signal timing is also plotted as a horizontal line with different colors at the stop bar (distance=0). The legend in each figure shows some basic information and the calculated traffic performance indices. With the general introduction of Figure 2.8, then we will go to these four typical cases individually.

Common case Figure 2.8 (a) shows a common case of the trajectory time-space diagram; the vehicle stopped around 48 meters in front of the stop bar. There is clearly one stop labeled by pink color. There is no split failure or spill-over warning for this trajectory.

Stop state consolidation Figure 2.8 (b) is an example showing why the stop state consolidation introduced before is necessary, especially for an accurate estimation of the number of stops. The example trajectory experienced a long stop while moving forward a short distance in the middle. If not applying the stop state consolidation, the number of stops will probably be overestimated, which might lead to a false alarm of split failure.

Split failure Figure 2.8 (c) is an example of split failure. This vehicle is in a left-turn movement, it failed to pass the intersection for the first green time and waited for another red light. We can clearly see the two characteristics of the split failure trajectory: 1) control delay larger than the red time and 2) multiple stops. It is also a good comparison between Figure 2.8 (b) and (c) for the identification of the number of stops. With our trajectory segmentation algorithm introduced in before, two valid stop states should have a large distance and time gap; otherwise, they will be combined as one continuous stop.



(a) Common case

(b) Stop state consolidation

(c) Split failure

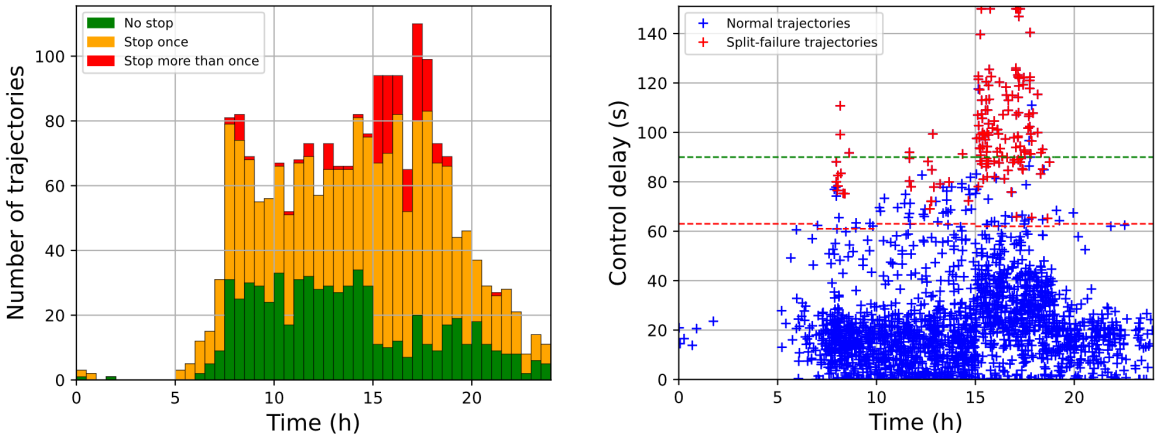
(d) Left-turn spillover

Figure 2.8: Typical cases for the traffic performance index calculation

Spillover warning Figure 2.8 (d) shows a trajectory posted as a spillover warning. This trajectory is in a left-turn movement and the black dashed line is the start of the dedicated left-turn lane. This trajectory is labeled as a spillover warning since the maximum queue distance is larger than the length of the dedicated left-turn lane. The first stop occurred before the start of the dedicated left-turn lane; this indicated that the vehicle was blocked by the residual queue before entering the dedicated left-turn lane.

2.5.2 Performance evaluation and trajectory aggregation

Although trajectories provide accurate delay measurements, observed trajectories are usually sparse due to low connected vehicle penetration rates. Because movements experience similar traffic demands for each day, historical trajectory data from multiple dates can be aggregated



(a) Number of trajectories

(b) Control delay scatters

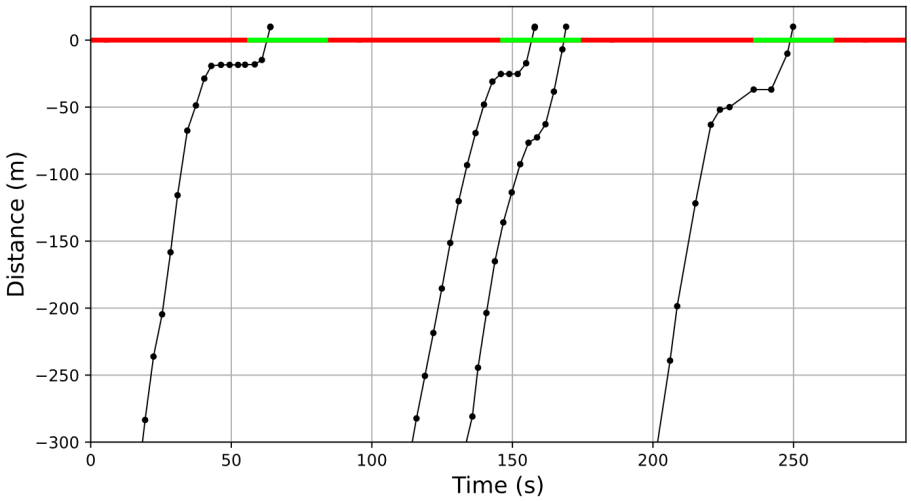
Figure 2.9: Performance evaluation figures for an example movement.

together to get more robust evaluations. Figure 2.9 are performance evaluation plots of an example movement from aggregated trajectory data of continuous five weekdays. Figure 2.9 (a) shows the number of observed trajectories across a whole day where different colors represent the numbers of stops experienced by those trajectories. More green means better performance while red indicates spill-over. The delay scatter plot given by Figure 2.9 (b) can show how control delay changes throughout the day. Each cross represents a trajectory's point when it passes the intersection (horizontal axis) and control delay (vertical axis). Blue and red crosses represent normal and split-failure trajectories, respectively.

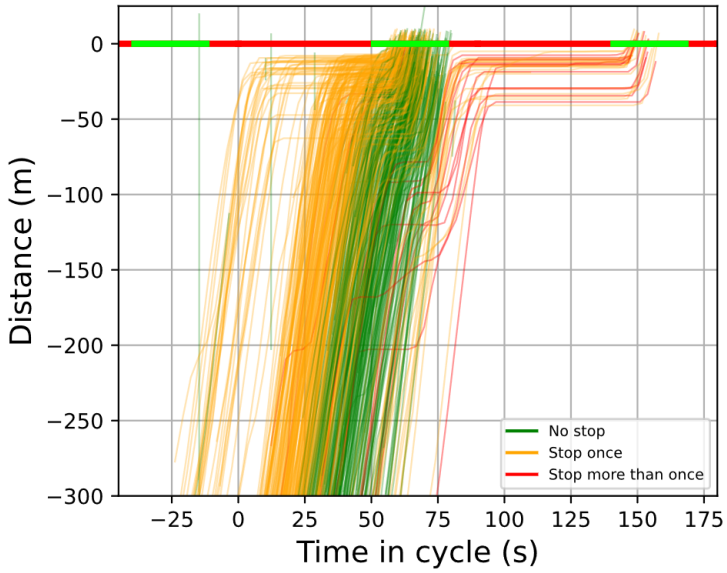
Since fixed-time traffic signal states are cyclic within a certain TOD, trajectories from different cycles can also be aggregated to one cycle to get the aggregated time-space diagram as shown in Figure 2.10; different colors also represent different numbers of stops. When aggregating the trajectory to one cycle, we shift each trajectory by an integer number of cycles to get their arrival times within the same cycle. Since the trajectories are aggregated according to their arrival times, the departure time might extend to the following cycle if some vehicles failed to pass the intersection within the cycle they arrived at. The aggregated time-space diagram shows the average traffic pattern of the movement at a certain TOD and demonstrates recurrent congestion issues such as bad coordination and split failure.

2.5.3 Space-mean speed of a corridor

A corridor, or any coordinated path, is composed of a series of movements traversing multiple intersections. Figure 2.11 shows the aggregated time-space diagram of an example corridor at a certain TOD, created by combining the movement time-space diagrams along the path. For visualization purposes, the aggregated time-space diagrams for each movement are repeated over



(a) Original time-space diagram



(b) Aggregated time-space diagram

Figure 2.10: Aggregated time-space diagram for an example movement.

several cycles. While Figure Figure 2.11 visualizes the northbound direction, a similar figure can be plotted for the opposite direction. The corridor aggregated-time space diagram clearly depicts how vehicles traverse the whole corridor and can be used to evaluate coordination performance. The corridor aggregated time-space diagram can be converted to the space-mean speed heatmap as shown by Figure 2.12. The spatial-temporal space is split into mesh grids by setting certain temporal and spatial intervals (e.g., 3 seconds and 20 meters); and the space-mean speed within each grid is the total travel distance divided by the total travel time of all the trajectories within the grid. Red indicates stopped vehicles, or queues, for each intersection and is where delay occurs.

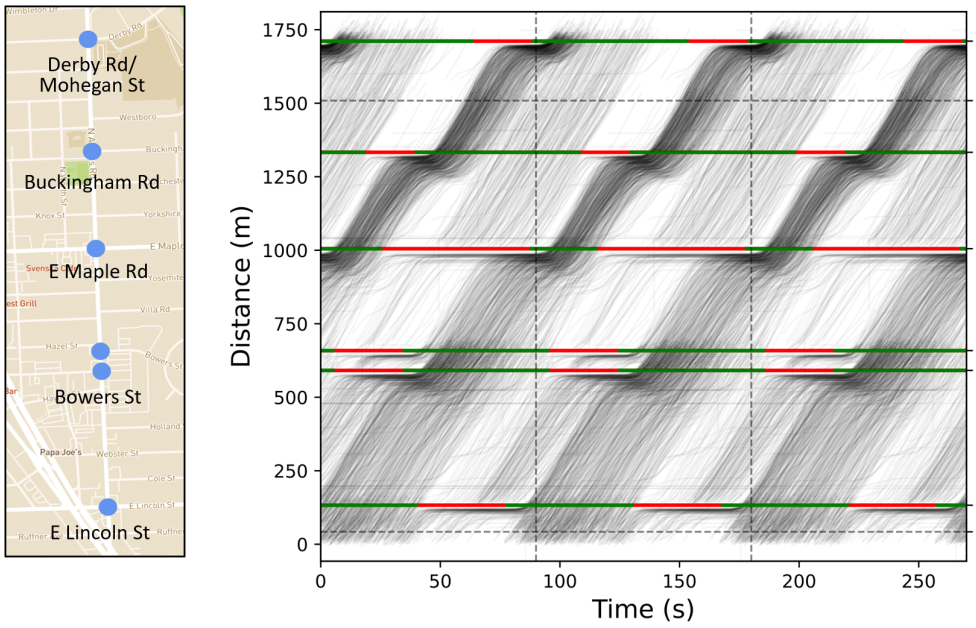


Figure 2.11: Aggregated time-space diagram of Adams Rd., Birmingham, MI.

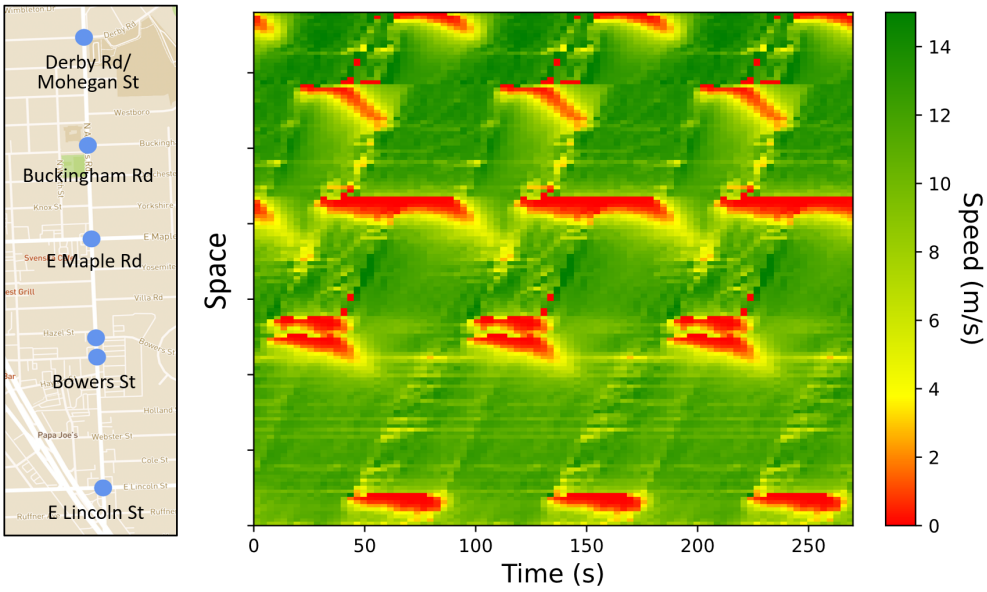


Figure 2.12: Space-mean speed of Adams Rd., Birmingham, MI.

2.6 Summary

This chapter proposes a trajectory data processing pipeline that serves different traffic operational applications in large-scale traffic networks. The trajectory data processing pipeline includes matching trajectory data to a well-defined network representation, splitting the trajectory data into different movements, and extracting distance information from raw GNSS coordinates. Smoothing

and filtering algorithms are also required to reduce the influence of noise and errors in real-world data. Based on the processed trajectory data, we also propose a series of efficient and robust algorithms for mobility evaluation of the signalized intersections including estimating the vehicle delay and number of stops, evaluating the coordination among intersections, etc.

Both AACVTE and GM trajectory data are used to test the proposed methods and algorithms. Different plots are generated to visualize the traffic performance of the studied corridor. The trajectory processing pipeline and mobility evaluation algorithms suit well for real-world implementation in a large-scale network with vehicle trajectory data, which can serve as a stepping stone for city-level traffic control and management. This chapter is the foundation for all remaining chapters that use real-world trajectory data as the input.

CHAPTER 3

Stochastic Traffic Flow Model in Newellian Coordinates

3.1 Introduction

3.1.1 Background and related works

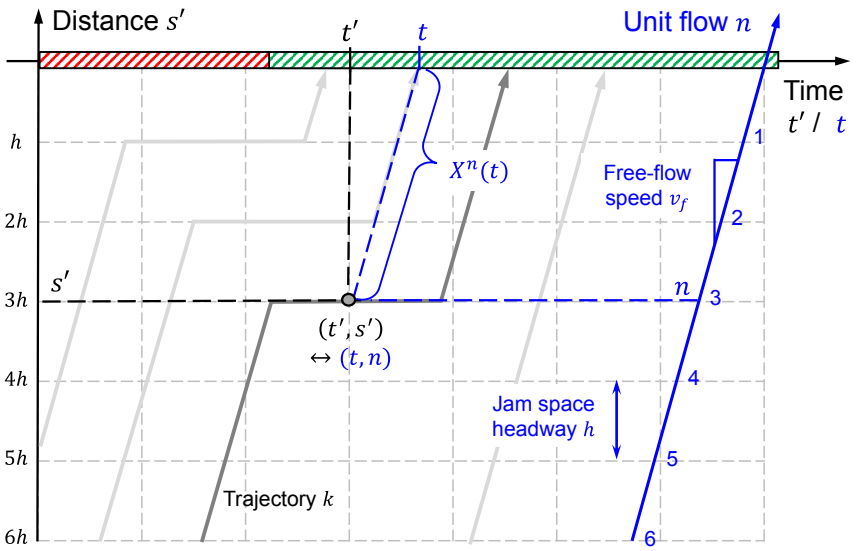
Stochastic traffic flow models can be used to estimate and predict the overall traffic state from incomplete observations. Please refer to Section 1.3.2 for a more comprehensive literature review of the traffic flow models.

Most existing traffic flow models do not fit with vehicle trajectory observations. Eulerian and Lagrangian coordinates are the two most used coordinate systems in existing models (Figure 3.1). Eulerian coordinates split the spatial-temporal space into grids and define the traffic state as the density in each grid. Trajectory data does not provide measurements in Eulerian coordinates and hence cannot be directly used to calibrate the traditional LWR model and its variants. Vehicle trajectory data is in the form of Lagrangian coordinates which keep track of each individual vehicle's movement, but traffic flow models under Lagrangian coordinates suffer from high dimensionality and are not applicable to large-scale applications.

In addition, models utilizing both Eulerian and Lagrangian coordinates become more complicated at higher dimensions when extended to stochastic settings (Jabari and Liu, 2012, 2013; Sumalee et al., 2011; Flötteröd and Osorio, 2017; Zheng et al., 2018). As a result, the lack of a suitable traffic flow model for vehicle trajectory data is one of the main bottlenecks of using such data for traffic signal optimization.

3.1.2 Overview of the chapter

To overcome this challenge, this chapter introduces a stochastic traffic flow model under newly proposed Newellian coordinates (Figure 3.1). By assuming that all vehicles follow a uniform



	Eulerian	Lagrangian	Newellian
Dim	Number of grids	Number of trajectories	1
State	Density $\bar{k}(t', s')$	Location $s'_k(t')$	Queue $X^n(t)$

Figure 3.1: Eulerian, Lagrangian, and the proposed Newellian coordinates and corresponding traffic state representations.

deterministic Newell's car-following model, the Newellian coordinates allow us to use a point-queue representation to describe the complete spatial-temporal traffic state. The probabilistic time-space (PTS) diagram is then utilized to derive the spatial-temporal vehicle trajectory distribution given the stochastic point-queue model.

The proposed stochastic traffic flow model builds the connection between the sparsely observed vehicle trajectory data and the unknown traffic states & parameters, which enables us to apply different statistical estimation algorithms to estimate these unknown values.

3.1.3 Contributions and organization of the chapter

The contributions of this chapter are summarized as follows:

1. We demonstrate that a point-queue model under the newly proposed Newellian coordinates can sufficiently describe the spatial-temporal traffic state.

2. We propose the PTS diagram, which can be used to derive the spatial-temporal distribution of vehicle trajectories with a point-queue representation.

3. The proposed model is compatible with the measurement provided by vehicle trajectory data.

It builds the connection between observed vehicle trajectory data and unknown traffic state and parameters.

This chapter is organized as follows: Section 3.2 introduces the Newellian coordinates, point-queue representation, and the PTS diagram. Section 3.3 shows the derivation of the stationary distribution of fixed-time traffic signals. Section 3.4 includes more details of the queueing model and PTS diagram when there is a residual queue at the end of a cycle. Section 3.5 is a case study that demonstrates how the proposed model is related to observed vehicle trajectory data. Section 3.6 shows some numerical examples and Section 3.7 is the summary of this chapter.

3.2 Newellian coordinates, point-queue representation, and probabilistic time-space diagram

3.2.1 Discrete approximation

The proposed stochastic traffic flow model is established based on a discrete approximation. For a certain movement, let q^m and z denote the saturation flow rate and the number of lanes, respectively. For each time interval Δt, the unit flow per time step Δu at saturation flow rate will be determined by:

$$\Delta u = q^m z \Delta t. \tag{3.1}$$

The discrete approximation assumes atomic traffic flow in units of Δu. If the time interval is chosen properly, each unit flow could represent exactly one vehicle. For example, if a movement has two lanes, $z = 2$, and saturation flow rate $q^m = 1800\ veh/(lane \cdot hour)$, then a unit traffic flow Δu will be one vehicle if $\Delta t = 1$ sec. Let h_o be the jam space headway with unit $meter/(veh \cdot lane)$, which is assumed to be a known constant. Then the jam space headway h per unit flow (unit: $meter/\Delta u$) is given by:

$$h = \frac{\Delta u \cdot h_0}{z} = q^m h_0 \Delta t. \tag{3.2}$$

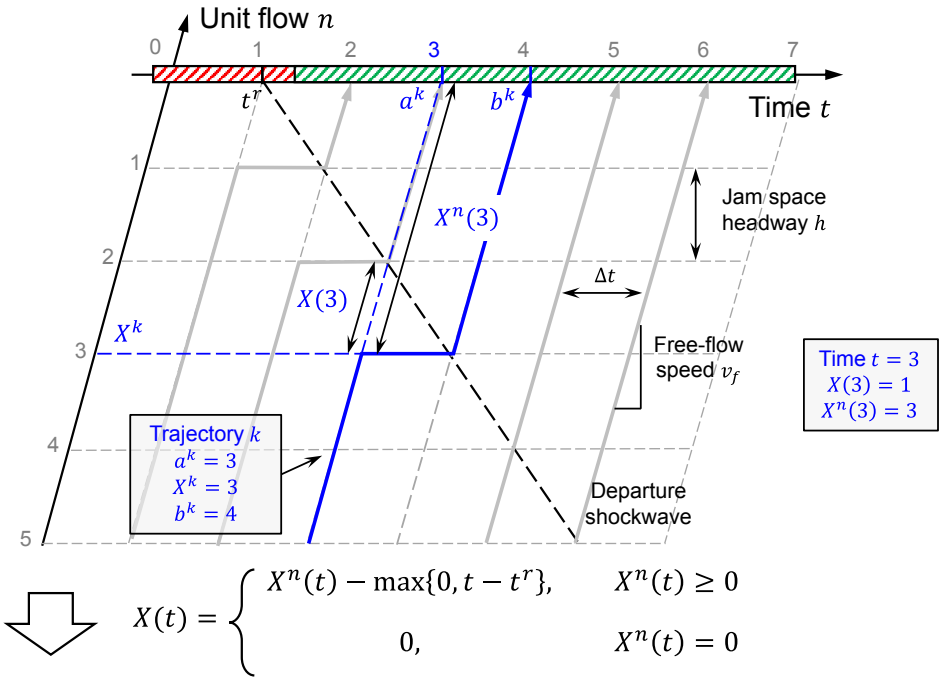
Without loss of generality, we will use $\Delta t = 1$ to simplify the notation in the rest of this dissertation, which means that time t directly represents the number of time steps. Besides, although Δu does not necessarily refer to one vehicle, for convenience, we directly use "one vehicle" to represent its complete rigorous description, i.e., "unit traffic flow Δu".

3.2.2 Newellian coordinates and point-queue model

Figure 3.2 is an illustration of the proposed Newellian coordinates, which are established on the assumption that all vehicles follow a homogeneous deterministic Newell's car-following model

(Newell, 2002). This assumption holds since stop-and-go is the dominant vehicle trajectory feature in urban areas with traffic signals. Most of the uncertainty arises from the stochasticity of traffic demand rather than stochastic driving behaviors, particularly at a low penetration rate. According to the previously introduced discrete approximation, for each time interval Δt, traffic flow comes atomically with either 0 or Δu (could refer to one or several vehicles). The Newellian coordinates are then defined as (t, n) where t represents the free-flow arrival time (in units of Δt) while n denotes the number of unit traffic flows (in units of Δu). The "distorted grid" in Figure 3.2 is an illustration of such coordinates, which are parameterized by the free-flow speed v_f, jam space headway h, and time interval Δt. As shown in Figure 3.1, the transformation between the time-space coordinates (t', s') and Newellian coordinates (t, n) is given by:

$$\begin{cases} t' = t - \frac{n \cdot h}{v_f} \\ s' = n \cdot h \end{cases} . \tag{3.3}$$



Time t	1	2	**3**	4	5	6	7
$A(t)$	1	1	**1**	0	1	1	0
$X^n(t)$	1	3	**3**	0	0	0	0
$X(t)$	1	1	**1**	0	0	0	0
$B(t)$	0	1	1	**1**	1	1	0

Figure 3.2: Illustration of Newellian coordinates and point-queue representation.

The major difference between these coordinate systems is that Newellian coordinates use the free-flow arrival time as time t. The free-flow arrival time can be interpreted as the time when a vehicle would have arrived at the intersection if it traveled at the free-flow speed and did not have to slow down or stop because of background traffic or the traffic signal. Based on the Newellian coordinates, one trajectory with one stop, taking trajectory k in Figure 3.2 as an example, can be encoded as (a^k, X^k, b^k) where a^k is the free-flow arrival time, X^k is the stop location, and b^k is the departure time when it leaves the intersection. The difference between the departure time and free-flow arrival time $|b^k - a^k|$ is the control delay (Manual, 2010; Wang et al., 2022a).

Newellian coordinates enable us to convert all vehicle trajectories to a point-queue representation. Let $X^n(t)$ represent the spatial queue length (in units of Δu) at time t, which is determined by the physical position of the last stopped vehicle. It can be further transformed to $X(t)$ through:

$$X(t) = \Psi_t(X^n(t)), \quad \text{where } \Psi_t(n) = \begin{cases} n - (t - t^r)^+ & n > 0 \\ 0 & n = 0 \end{cases} \tag{3.4}$$

where $\Psi_t(\cdot)$ denotes the mapping function at time t, $(t - t^r)^+ \equiv \max\{0, t - t^r\}$, and t^r is the end of the red light. $X(t)$ refers to the number of stopped vehicles at time t. The dynamic equation of $X(t)$ is given by:

$$X(t) = X(t-1) + A(t) - B(t) \tag{3.5}$$

where $A(t)$ and $B(t)$ is the arrival and departure, respectively. $X(t)$ behaves as a point queue since it does not have spatial information. In this way, we have converted the spatial-temporal traffic state under the Newellian coordinates to a point-queue representation.

Due to the uncertainty caused by the sparsely observed vehicle trajectory data, a stochastic model is required. The deterministic point-queue model can be easily converted to a stochastic version (i.e., a stochastic queueing model) by applying a stochastic arrival process. Although stochastic queueing models have been widely studied to model urban traffic networks (Viti and Van Zuylen, 2010; Boon and van Leeuwaarden, 2018; Osorio and Bierlaire, 2009; Osorio and Wang, 2017; Osorio and Yamani, 2017), few have established their connection with partially observed vehicle trajectory data (Maripini et al., 2023).

For the stochastic discrete queueing model, the arrival $A(t)$ is assumed to be binary which follows a Bernoulli distribution with arrival probability $a(t)$, that is, $A(t) \sim \text{Bernoulli}(a(t))$. For simplification, arrivals at different time steps are assumed to be independent. The queue length is updated by:

$$X(t) = X(t-1) + A(t) - B(t) = X'(t) - B(t) \tag{3.6}$$

where $X'(t)$ is the intermediate queue length after the new arrival at time t. In each time step,

the arrival happens before the departure since vehicles can directly pass the intersection without stopping. Otherwise, every vehicle in the model would need to wait at least one time step before passing the intersection. The departure $B(t)$ is also binary and controlled by the traffic signal state $S(t)$:

$$\mathbb{P}(B(t) = 1) \equiv b(t) = \mathbb{P}(X'(t) \leq 1) \cdot S(t). \tag{3.7}$$

where $S(t) = 0$ and $S(t) = 1$ correspond to red and green lights, respectively. Equation (3.7) means that the departure will happen whenever the queue is not empty, and the traffic signal state is green. Let $x(t, k)$ be the pmf (probability mass function) of the queue length, which is the probability that the queue length is k at time t. Given an input arrival profile $a(t)$, the queue length distribution and departure can be updated recursively according to the following equations:

$$x'(t, k+1) = x(t-1, k) \cdot a(t) + x(t-1, k+1) \cdot (1 - a(t)) \tag{3.8a}$$

$$x(t, k) = x'(t, k+1) \cdot S(t) + x'(t, k) \cdot (1 - S(t)),\ \forall k \leq 1 \tag{3.8b}$$

$$x(t, 0) = x'(t, 1) \cdot S(t) + x'(t, 0) \tag{3.8c}$$

$$b(t) = \sum_{k=1}^{\infty} x'(t, k) \cdot S(t) \tag{3.8d}$$

3.2.3 Probabilistic time-space (PTS) diagram

Even though the point-queue model uses a simple representation without considering spatial information, it can be projected back to the spatial-temporal space using the probabilistic time-space (PTS) diagram to capture complete vehicle movement and queue propagation (Figure 3.3). Here $\rho^n(t, n)$ and $\rho^t(t, n)$ denote the probability that there are vehicles traveling on the vertical and horizontal edges, corresponding to the free-flow and stop states, respectively. The probability at each edge can be calculated given the point-queue representations including arrival, queue length, and departure. Each edge is drawn using transparency to represent probability. Consequently, the PTS diagram directly shows the spatial-temporal distribution of vehicle trajectories.

As shown in Figure 3.3, edges in the grid can be divided into three categories including the arrival, departure, and stop states. For the stop state, the probability of each edge can be calculated by:

$$\rho^t(t, \Psi_t^{-1}(n)) = \mathbb{P}(X(t) \leq n) = \sum_{k=n}^{\infty} x(t, k) \tag{3.9}$$

where $\Psi_t^{-1}(\cdot)$ is the inverse function of $\Psi_t(\cdot)$ in Equation (3.4). $\Psi_t^{-1}(\cdot)$ is given by:

$$\Psi_t^{-1}(n) = \begin{cases} n + (t - t^r)^+ & n > 0 \\ 0 & n = 0 \end{cases} \tag{3.10}$$

Time	Arrival probability	Queue length distribution					Departure probability
		0	1	2	3	…	
1	$a(1)$	$x(1,0)$	$x(1,1)$	$x(1,2)$	$x(1,3)$	…	$b(1)$
2	$a(2)$	$x(1,0)$	$x(2,1)$	$x(2,2)$	$x(2,3)$	…	$b(2)$

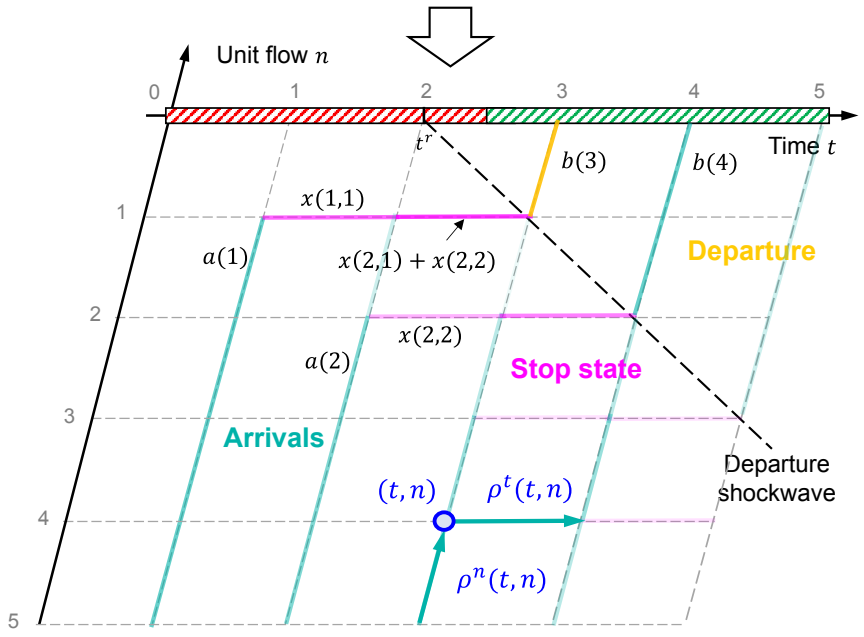


Figure 3.3: Probabilistic time-space (PTS) diagram.

which projects the point queue $X(t) = n$ to the spatial queue $X^n(t) = \Psi_t^{-1}(n)$. In this way, $\rho^t(t, \Psi_t^{-1}(n))$ is the probability that there is a vehicle waiting from time t to $t+1$ at point queue $X(t) = n$. Equation (3.9) can be interpreted as the probability that there is a vehicle stopped at point queue $X(t) = n$, which is equal to the total probability that $X(t) \leq n$.

For the departure edges as shown in Figure 3.3, the probability is calculated by:

$$\rho^n(t, 0 : \Psi_t^{-1}(-1)) = \mathbb{P}(B(t) = 1) = b(t) \tag{3.11}$$

where $\rho^n(t, 0 : \Psi_t^{-1}(-1))$ represents all the departure edges at time t starting from the departure shockwave until leaving the intersection as shown in Figure 3.3.

For the arrival edges, the probability is calculated by:

$$\rho^n(t, \Psi_t^{-1}(n)) = \mathbb{P}(A(t) = 1) \cdot \mathbb{P}(X(t) < n) = a(t) \cdot \sum_{k=0}^{n-1} x(t,k), \ n \leq 1. \tag{3.12}$$

$\rho^n(t, \Psi_t^{-1}(n))$ represents the probability a vehicle travelling from $X(t) = n+1$ to $X(t) = n$. This

event happens when there is a new arrival A(t)=1 and the queue length $X(t)$ is less than n.

Equation (3.9-3.12) shows how the probability of a vehicle trajectory traveling on each edge in Figure 3.3 is calculated from the discrete queueing model given by Equation (3.6-3.8). The probability of each edge is used as the edge's transparency in the diagram. In this way, the discrete queueing model is mapped to the probabilistic time-space (PTS) diagram and directly shows the spatial-temporal distribution of the vehicle trajectories.

Algorithm 1: Calculation of the stationary queue length distribution

Input 1) Arrival profile $\boldsymbol{a} = [a(1), a(2), \ldots, a(T)]$. 2) traffic signal state $\boldsymbol{s} = [s(1), s(2), \ldots, s(T)]$. and 3) stopping criteria $\epsilon = 1e-6$.

Initiation: 1) Queue length distribution $\boldsymbol{X}^0 \in \mathbb{R}^{T\times N}$ of iteration 0 where $x^0(t,k)$ is the probability that queue length at time step t is k. To start with, the queue is empty at $t=0$: $x^0(0,0)=1, x^o(0,k)=0, k\geq 1$; and 2) Departure profile $\boldsymbol{b}^0 = [b^0(1), \ldots, b^0(T)]$ at iteration 0 and $b(t)^0 = 0, \forall t \in \{1, \ldots, T\}$.

while *for iteration* $i = 0, 1, 2, ...$ **do**

Initiate the queue length distribution at the start of the cycle:

$$x^{i+1}(0,k) = x^i(T,k), \quad \forall n \tag{3.13}$$

for *time in cycle* $t = 1, 2, ..., T$ **do**

Update the queue length distribution after new arrival:

$$x^{i+1}(t,k+1)' = x^{i+1}(t-1,k)\cdot a(t) + x^{i+1}(t-1,k+1)\cdot(1-a(t)), \quad \forall k \tag{3.14}$$

Update the queue length distribution after new departure:

$$x^{i+1}(t,k) = x^{i+1}(t,k+1)'\cdot s(t) + x^{i+1}(t,k)'\cdot(1-s(t)), \quad \forall k \geq 1 \tag{3.15a}$$

$$x^{i+1}(t,0) = x^{i+1}(t,1)'\cdot s(t) + x^{i+1}(t,0)' \tag{3.15b}$$

Get the departure probability $b(t)$:

$$b^{i+1}(t) = \left(\sum_{k=1}^{N} x^{i+1}(t,k)'\right)\cdot s_i(t) \tag{3.16}$$

if $\|\boldsymbol{X}^{i+1} - \boldsymbol{X}^i\|_F \geq \epsilon$ **then**

Set $\boldsymbol{b} = \boldsymbol{b}^{i+1}$, $\boldsymbol{X} = \boldsymbol{X}^{i+1}$ and terminate the iteration.

Return: departure probability $\boldsymbol{b}$ and queue length distribution $\boldsymbol{X}$.

3.3 Stationary cycle for fixed-time traffic signals

For fixed-time traffic signals, since both the traffic signal state and input arrival rate are cyclic with cycle T, that is, $S(t+kT) = S(t)$ and $a(t+kT) = a(t)$ for any cycle k, both the resulting departures and queue lengths will converge to a stationary traffic cycle if the average traffic demand is within the traffic signal capacity:

$$\lim_{k\to\infty} X(t+kT) \to \bar{X}(t), \quad \lim_{k\to\infty} B(t+kT) \to \bar{B}(t),\ \forall t \in \{1,2,\ldots,T\} \tag{3.17}$$

where $\bar{X}(1:T)$ and $\bar{B}(1:T)$ represent the stationary queue length and departure in a traffic cycle which can be calculated iteratively over cycles according to Equation (3.8) (Algorithm 1). Equation (3.17) also requires that the movement is under-saturated on average: $\sum_{t=1}^{T} a(t) < \sum_{t=1}^{T} S(t)$. This assumption holds in the real world since the queue length of each movement is restricted by the length of the roadway, and the arrival rate will always be less than the capacity in the long term.

With the stationary arrival $\bar{A}(t)$ and queue length $\bar{X}(t)$, the PTS diagram for the stationary traffic cycle can be drawn according to the same Equations (3.9-3.12). and the average delay can be calculated according to Little's law (Little and Graves, 2008):

$$\bar{d} = \frac{\sum_{t=1}^{T} \mathbb{E}[\bar{X}(t)]}{\sum_{t=1}^{T} \mathbb{E}[\bar{A}(t)]}. \tag{3.18}$$

3.4 Traffic flow model with residual queue

3.4.1 Discrete queueing model with residual queue

Section 3.2 has demonstrated how the discrete queueing model can be mapped to the probabilistic time-space (PTS) diagram without considering any residual queue or over-saturation (see Remark 3.1). To make the model more generic so that it can also deal with the over-saturation case, we need to keep track of the queue length of each individual cycle.

Remark 3.1. *For clarification, in the rest of the dissertation, the "over-saturation" means that there is an unignorable probability that some vehicles cannot pass the intersection within the first cycle or there is a residual queue at the end of the green end time. As aforementioned in Section 3.3, all movements are under-saturated by average if a constant arrival rate is assumed: the average arrival rate needs to be strictly less than the capacity, otherwise, the queue will increase to infinity. Although the arrival rate is less than the capacity by average, there will still be a certain probability that the vehicle is not completely cleared due to the stochastic arrival process. Therefore, when it comes to "over-saturation" afterward, it does not mean the movement is over-*

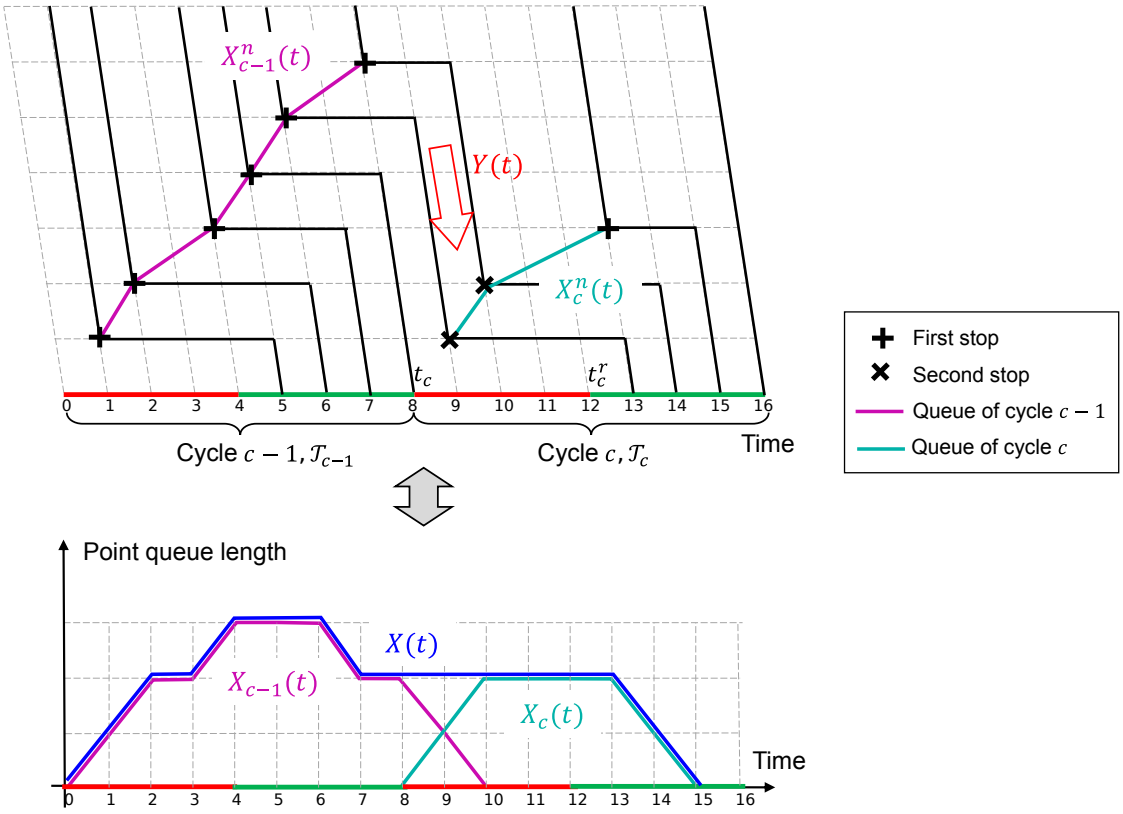


Figure 3.4: Residual queue.

saturated by average but with a high traffic volume (more rigorously, a large volume-to-capacity ratio) so that there is likely a residual queue at the end of the cycle.

Figure 3.4 illustrates how to decompose the total queue into queues for each individual cycle. A cycle c starts with a red light at time t_c and the queue length starts to increase. Let $X_c(t)$ and $X^n_c(t)$ denote the point and spatial queue lengths of cycle c, respectively. For each cycle, the spatial and point queue lengths have the following mapping relationship:

$$X^n_c(t) = \Psi^{-1}_{c,t}(X_c(t)), \quad \text{where } \Psi^{-1}_{c,t}(n) = \begin{cases} n + (t - t^r_c)^+, & n > 0 \\ 0 & n = 0 \end{cases}. \tag{3.19}$$

where $\Psi^{-1}_{c,t}(\cdot)$ is the mapping function that projects the point queue $X_c(t)$ to the spatial queue $X^n_c(t)$. The mapping function $\Psi_{c,t}(\cdot)/\Psi^{-1}_{c,t}(\cdot)$ is similar to Equation (3.4) and Equation (3.10). The only difference is that we further specify the cycle c since different cycles have different red light end times t^r_c. Let $X(t)$ represent the total point queue at time t. We have:

$$X(t) = X_c(t) + X_{c-1}(t). \tag{3.20}$$

By using $X(t)$ as the total queue length, it is easy to verify that the discrete queueing model given by Equation (3.6-3.8) still holds. However, we are no longer able to get the spatial queue information from the total queue length because queues of different cycles are mixed together. The total queue $X(t)$ needs to be decomposed into the different cycles so that the spatial queue can be derived according to Equation (3.19). As shown in Figure 3.4, the queue length $X_c(t)$ is essentially the downstream of the residual queue $X_{c-1}(t)$ from the previous cycle and the internal flow $Y(t)$ denotes vehicles that depart the residual queue and join the new queue.

Let $\mathcal{T}_c$ be the set of time steps of cycle c, then for each cycle c and $t \in \mathcal{T}_c$, the discrete queueing model can be written as:

$$X_{c-1}(t) = X_{c-1}(t-1) + A(t) - Y(t) = X'_{c-1}(t) - Y(t) \tag{3.21}$$

$$X_c(t) = X_c(t-1) + Y(t) - B(t) = X'_c(t) - B(t) \tag{3.22}$$

where $Y(t)$ and $B(t)$ are determined by:

$$\mathbb{P}(B(t) = 1) = b(t) = \mathbb{P}(X'_c(t) \geq 1) \cdot S(t); \tag{3.23}$$

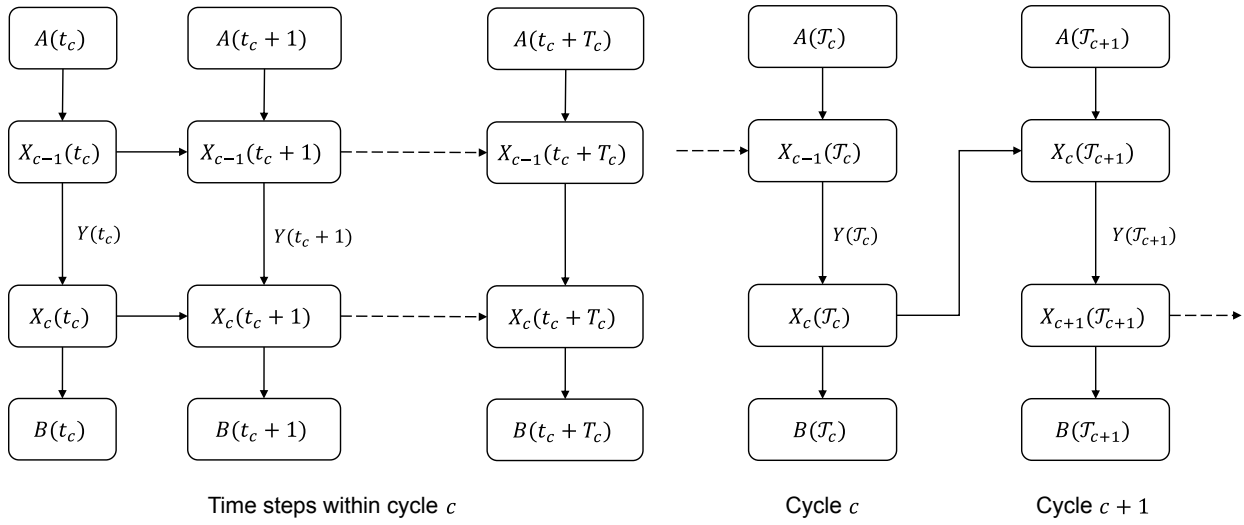
$$\mathbb{P}(Y(t) = 1) \equiv y(t) = \mathbb{P}(X'_{c-1}(t) \geq 1) \cdot 1. \tag{3.24}$$

The internal flow given by Equation (3.24) can be considered to be controlled by a constant green light since the vehicles in the residual queue $X_{c-1}(t)$ are not blocked and will join the new queue $X_c(t)$ continuously. Figure 3.5 shows the probabilistic graphical model by decomposing the queue length into different cycles (residual queue and queue of the current cycle). The left-hand-side figure shows the time steps within the cycle c while the right-hand-side figure shows the transition between different cycles. Note that we need to assume that the queue length will only extend to the following cycle. See Assumption 3.1 for more details.

Assumption 3.1. *The queue length of a cycle does not extend to the cycle after the following cycle. This assumption holds when the traffic volume is slightly larger than capacity for some of the cycles, which is true in most real-world cases. One simple counterexample of this assumption is a highly congested movement where some vehicles need to wait for more than* 2 *cycles to pass the intersection. The same method proposed in this section can be used but it will lead to a more complicated formulation, and hence we do not spend effort repeating the same procedure.*

By adding Equation (3.21) and Equation (3.22), we have:

$$\underbrace{X_c(t) + X_{c-1}(t)}_{X(t)} = \underbrace{X_c(t-1) + X_{c-1}(t-1)}_{X(t-1)} + A(t) - B(t). \tag{3.25}$$



(a) Graphical model within cycle c (b) Graphical model between cycles

Figure 3.5: Probabilistic graphical models with the residual queue.

This is the same as Equation (3.6) in Section 3.2, which means that the total queue length $X(t)$ has the same dynamics but it is decomposed into different cycles.

Let $x(t, k_r, k)$ represent the joint distribution of the residual queue $X_{c-1} = k_r$ and queue of the current cycle $X_c = k$. The transition of Equation (3.21-3.24) can be written as:

$$x'(t, k_r + 1, k) = x(t-1, k_r, k) \cdot a(t) + x(t-1, k_r + 1, k) \cdot (1 - a(t)) \tag{3.26a}$$

$$x''(t, k_r - 1, k) = x'(t, k_r, k+1),\ k_r \geq 1 \tag{3.26b}$$

$$x(t, k_r, k) = x''(t, k_r, k+1) \cdot S(t) + x''(t, k_r, k) \cdot (1 - S(t)),\ k \geq 1 \tag{3.26c}$$

$$x(t, k_r, 0) = x''(t, k_r, 1) \cdot S(t) + x''(t, k_r, 0) \tag{3.26d}$$

We also have the internal flow $y(t)$ and departure $b(t)$ determined by:

$$y(t) = \sum_{k_r=1}^{\infty} \sum_{k=0}^{\infty} x'(t, k_r, k) \tag{3.27}$$

$$b(t) = \sum_{k_r=0}^{\infty} \sum_{k=1}^{\infty} x''(t, k_r, k) \cdot S(t) \tag{3.28}$$

3.4.2 PTS diagram with residual queue

This subsection shows how to project the discrete queueing model to the corresponding PTS diagram when considering residual queues. As shown in Figure 3.6, for each cycle c and time

step $t \in \mathcal{T}_c = \{t_c, t_c + 1, \ldots, t_c + T_c\}$, there are five different parts: 1) arrivals to the residual queue, 2) residual queue stop state, 3) internal flows from the residual queues to the new queues, 4) stop state of the new queue, and 5) departures. The probability of each part is given below.

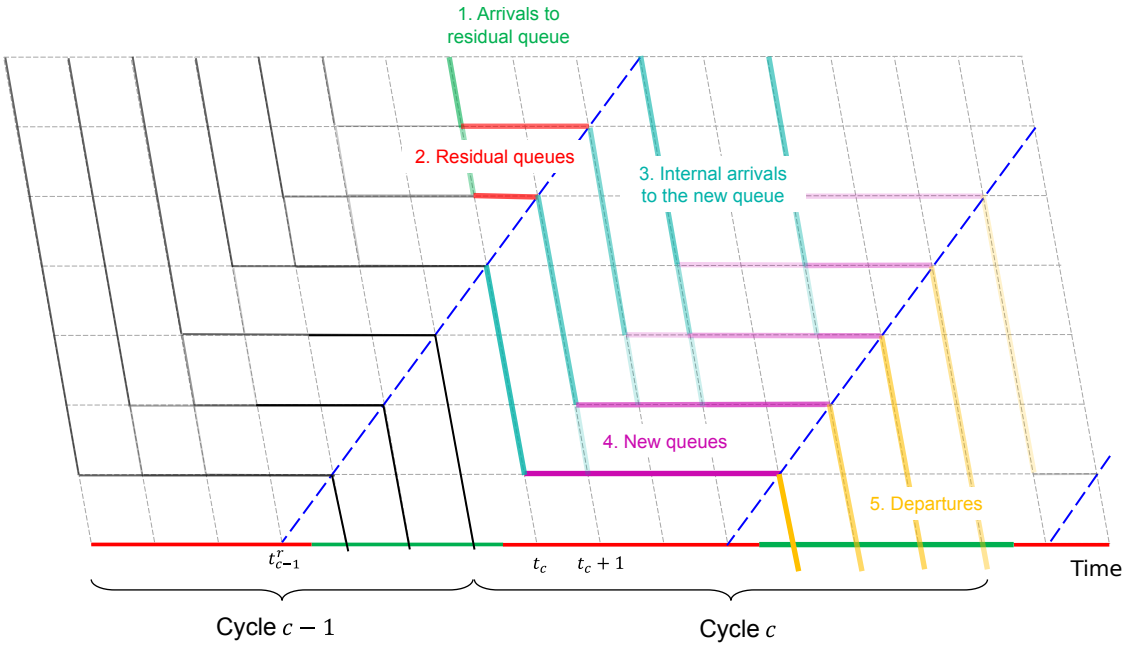


Figure 3.6: Probabilistic time-space diagram with a residual queue.

The arrival to the residual queue is similar to Equation (3.12) in Section 3.2. $\forall t \in \mathcal{T}_c$, we have:

$$\rho^n\left(t, \Psi^{-1}_{c-1,t}(n)\right) = \mathbb{P}(A(t) = 1) \cdot \mathbb{P}\left(X_{c-1}(t) < n\right) = a(t) \cdot \sum_{k_r=0}^{n-1} \sum_{k=0}^{\infty} x(t, k_r, k). \tag{3.29}$$

The residual queue stop state is similar to Equation (3.9) in Section 3.2. $\forall t \in \mathcal{T}_c$, we have:

$$\rho^t\left(t, \Psi^{-1}_{c-1,t}(n)\right) = \mathbb{P}(X_{c-1}(t) \geq n) = \sum_{k_r=n}^{\infty} \sum_{k=0}^{\infty} x(t, k_r, k). \tag{3.30}$$

The internal arrival to the new queue is given by ($\forall t \in \mathcal{T}_c$):

$$\rho^n\left(t, \Psi^{-1}_{c,t}(n)\right) = \mathbb{P}\left(X'_{c-1}(t) \geq 1 \;\&\; X_c(t) < n\right) = \sum_{k=1}^{\infty} \sum_{k=0}^{n-1} x'(t, k_r, k). \tag{3.31}$$

Equation (3.31) shows the probability that an internal flow departs from the residual queue and arrives at the new queue at location $X_c(t) = n$. It happens whenever the residual queue $X'_{c-1}(t)$ is not empty and the new queue $X_c(t)$ is less than n at the same time.

New queues:

$$\rho^t\left(t, \Psi_{c,t}^{-1}(n)\right) = \mathbb{P}(X_c(t) \geq n) = \sum_{k_r=0}^{\infty}\sum_{k=n}^{\infty} x(t, k_r, k). \tag{3.32}$$

Final departures:

$$\rho^n\left(t, 0 : \Psi_{c,t}^{-1}(-1)\right) = \mathbb{P}(B(t) = 1) = b(t). \tag{3.33}$$

3.5 Case study: proposed model and vehicle trajectory data

Previous sections have shown how we convert the spatial-temporal traffic state to a point-queue representation under the newly proposed Newellian coordinates and how the PTS diagram can project a stochastic point-queue model back to the whole spatial-temporal space. This means that a point-queue model can sufficiently capture the spatial-temporal traffic state with much less dimensionality and can be easily converted to a stochastic model.

This section will demonstrate another major advantage of the proposed model with a case study: it suits the sparsely observed vehicle trajectory data well. Assuming that at time 2, the queue length distribution is given by Table 3.1, the penetration rate $\phi = 20\%$, and the arrival rate at time 2 is $a = 0.4\ veh/sec$.

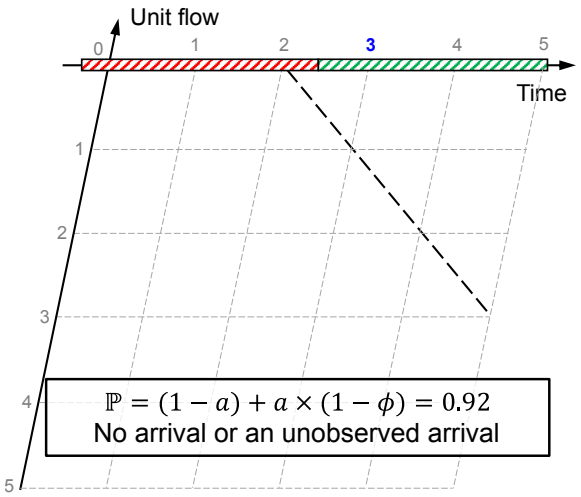
$X(2)$	0	1	2	$\geq$3
Probability	0.2	0.5	0.3	0

Table 3.1: Assumed queue length distribution at time 2.

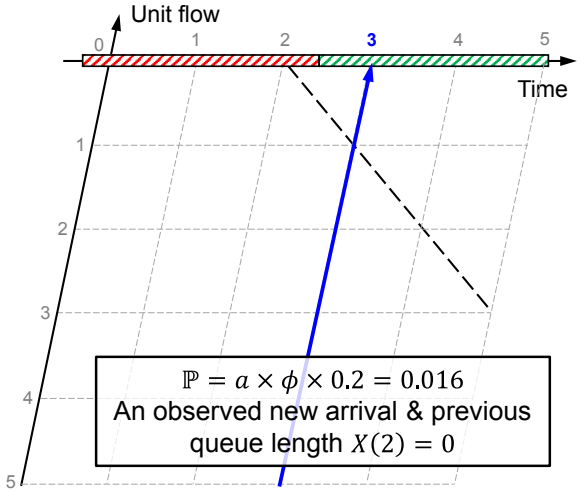
Based on these given conditions, Figure 3.5 shows all possible observed vehicle trajectories at time 3 and the corresponding probabilities:

- Case 1: no observed trajectory, which will happen when there is no arrival or an unobserved arrival. The calculation of the probability is given in the figure.
- Case 2: there is one observed new arrival at time 3 which directly passes the intersection without a stop. This event will happen when there is an observed arrival and at the same time the existing queue length is 0.
- Case 3-4: there is one observed new arrival that stops at locations 2 and 3, respectively.

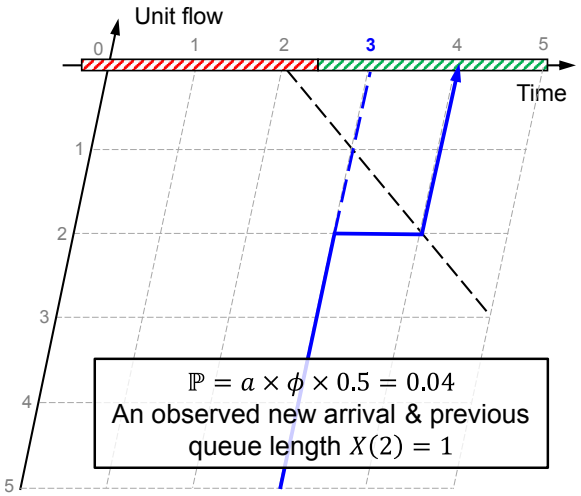
This example illustrates how the proposed model is related to the observed vehicle trajectory data. In fact, the probability of each case is the likelihood given different observed vehicle trajectories as well as the traffic parameters. The next chapter will utilize this likelihood function to estimate both unknown traffic states and parameters.



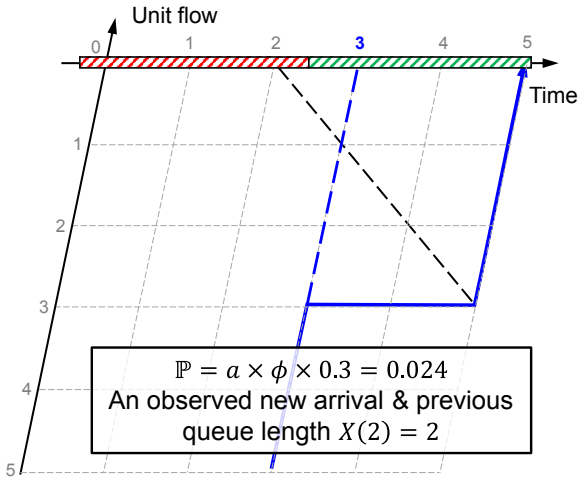
Case 1: No observed trajectory



Case 2: Observed trajectory directly passed the intersection



Case 3: Observed trajectory stopped at location 2



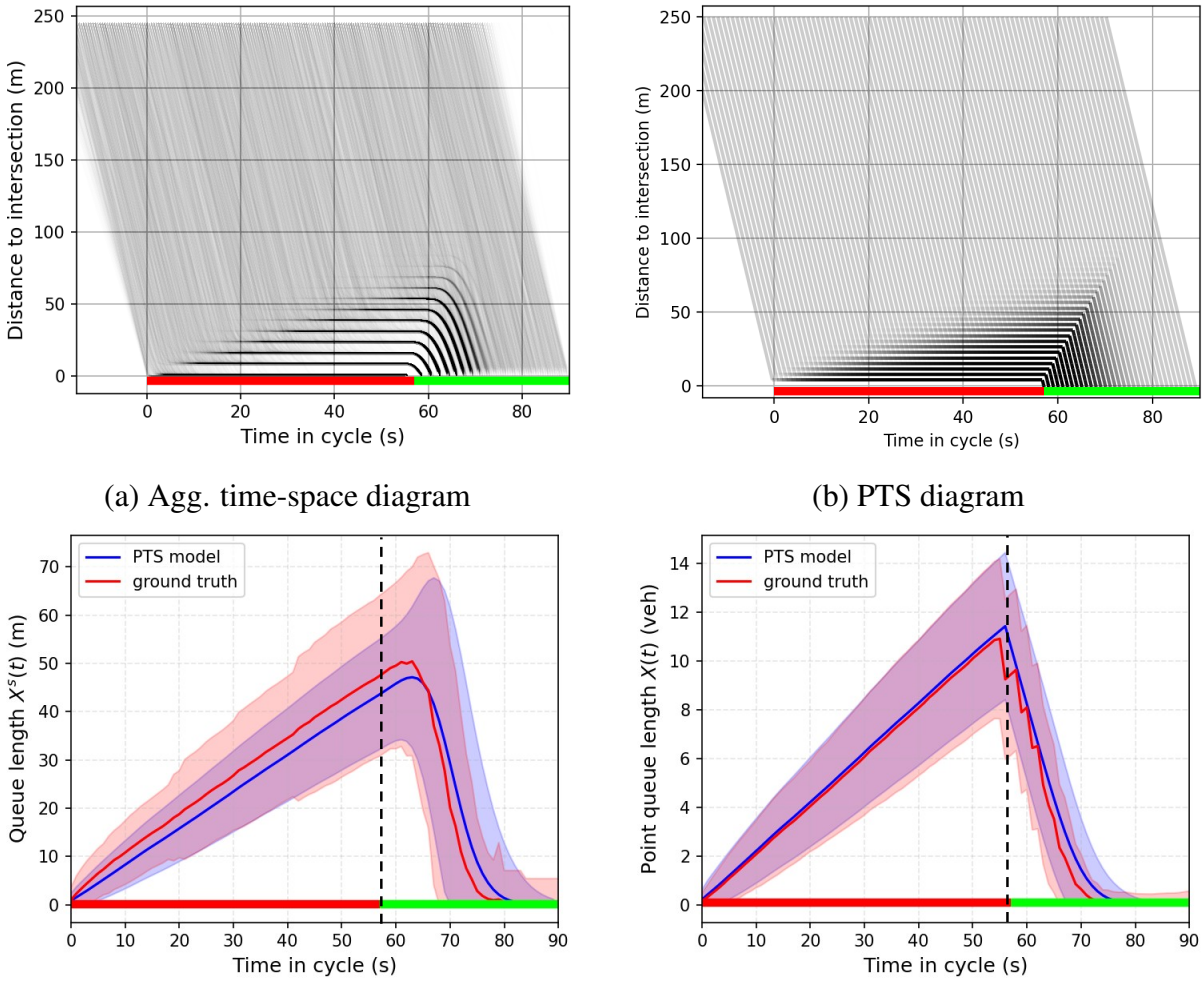
Case 4: Observed trajectory stopped at location 3

Figure 3.7: Possible observed trajectories given initial conditions.

3.6 Numerical examples

3.6.1 PTS diagram without the residual queue

This section will show some numerical examples of the PTS diagram. A single isolated movement controlled by a fixed-time traffic signal is built in the SUMO simulation environment. This movement has two lanes with a length of $250\ m$. The traffic signal is fixed-time with a cycle of $90\ s$; the green duration for this movement is $35\ s$. The average free-flow speed of all vehicles is $30\ mph$ and the jam space headway is a deterministic constant with a value of $7.5\ m/(veh{\cdot}lane)$. The vehicle arrival follows a Poisson process with an average traffic arrival rate of $720\ vph$. The volume-to-capacity (v/c) ratio is approximately 0.52.



(a) Agg. time-space diagram

(b) PTS diagram

(c) Spatial queue profile $X^s(t)$

(d) Point queue profile $X(t)$

Figure 3.8: PTS diagram and point/spatial queue profiles (without the residual queue).

Figure 3.8 (a) is the aggregated time-space diagram of the test movement. Vehicle trajectories

from 800 cycles are aggregated into the same cycle according to the aggregation method as aforementioned in Section 2.5.2. This aggregated time-space diagram shows the recurrent (average) pattern of the test movement in a cycle. Given the same parameters including the traffic volume as well as the traffic signal timing plan, Figure 3.8 shows the resulting PTS diagram, which represents the spatial-temporal distribution of the vehicle trajectories. Figure 3.8 (b) matches Figure 3.8 (a) well, demonstrating the effectiveness of the PTS diagram.

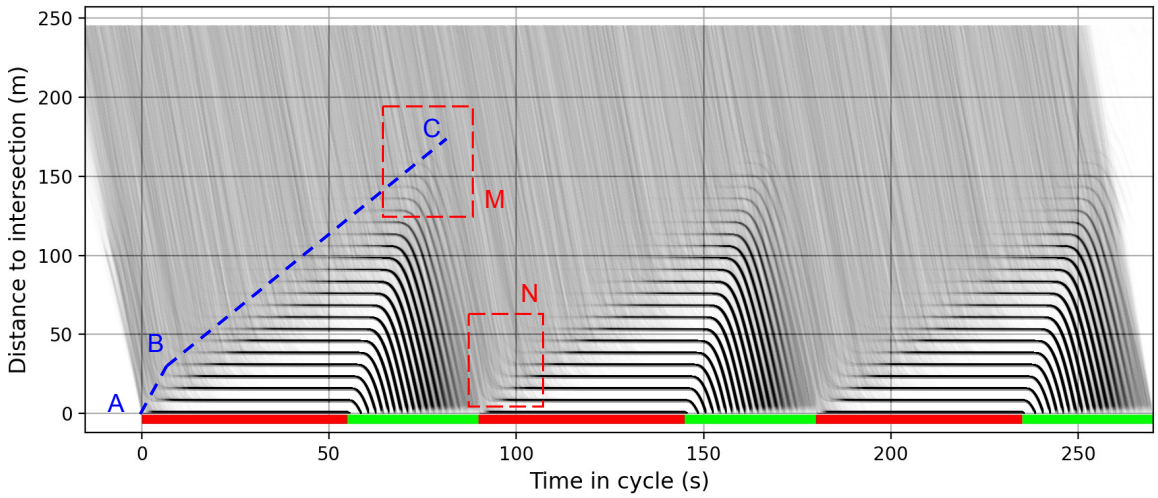
Figure 3.8 (c) and (d) show the corresponding spatial and point queue profiles. The spatial queue $X^s(t)$ refers to the physical location of the last stopped vehicle while the point queue $X(t)$ refers to the number of stopped vehicles. Notice that there are two different notations for the spatial queue which have different units: $X^s(t)$ is in units of meters (measured by distance) while $X^n(t)$ is in units of Δu (measured by unit traffic flow). They can be converted to each other through $X^s(t) = h \cdot X(t)$, where the space headway h is a constant. Since the figure shows the effective green time directly, the point queue $X(t)$ starts to decrease once the traffic signal turns green. However, the spatial queue $X^s(t)$ starts to decrease until the last stopped vehicle starts to move, which is later than the green start time. Both spatial queue $X^s(t)$ and point queue $X(t)$ are based on the proposed Newellian coordinates: the queue profiles show the queue length at the free-flow arrival time t of the Newellian coordinates instead of normal time t'. Please refer to Section 3.2.2 and Equation (3.3) for a recap of their difference and mapping relationship.

As shown in Figure 3.8 (c) and (d), the blue color denotes the queue profile extracted from the PTS diagram while the red color denotes the aggregated time-space diagram, which is considered to be the ground truth. The solid lines are mean values while shadow areas show the standard derivation (std). In general, the PTS model matched the ground truth very well for both spatial and temporal queues, as well as mean and std. Compared with the point queue in Figure 3.8 (d), the spatial queue of PTS diagram in Figure 3.8 (c) performs slightly worse since the PTS diagram uses a Newell's car-following model without considering vehicle slow-down and speed-up behaviors.

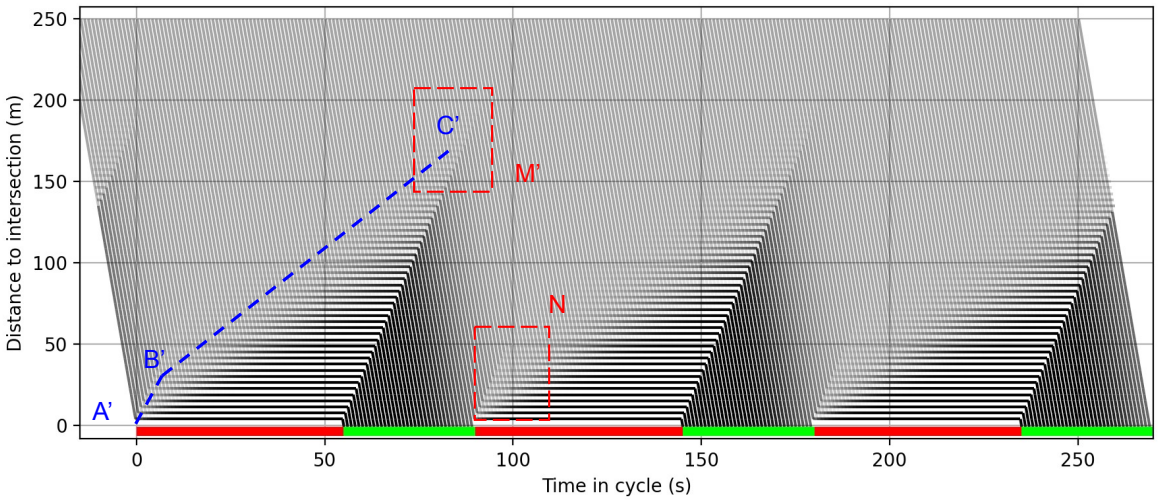
3.6.2 PTS diagram with residual queue

Figure 3.9 and Figure 3.10 show the aggregated time-space diagram, PTS diagram, and the spatial/point queue profiles of the same movement but with a higher traffic volume and, consequently, a higher probability that the queue is not cleared within one cycle. For this scenario, the average traffic volume is 1296 vph and the corresponding v/c ratio is approximately 0.92. Although this is still a strictly under-saturated case by average, the number of arrivals could be larger than the capacity for some cycles since the arrival is stochastic which follows a Poisson distribution (see Remark 3.1).

Both the aggregated time-space diagram and PTS diagram in Figure 3.9 are drawn with three



(a) Agg. time-space diagram

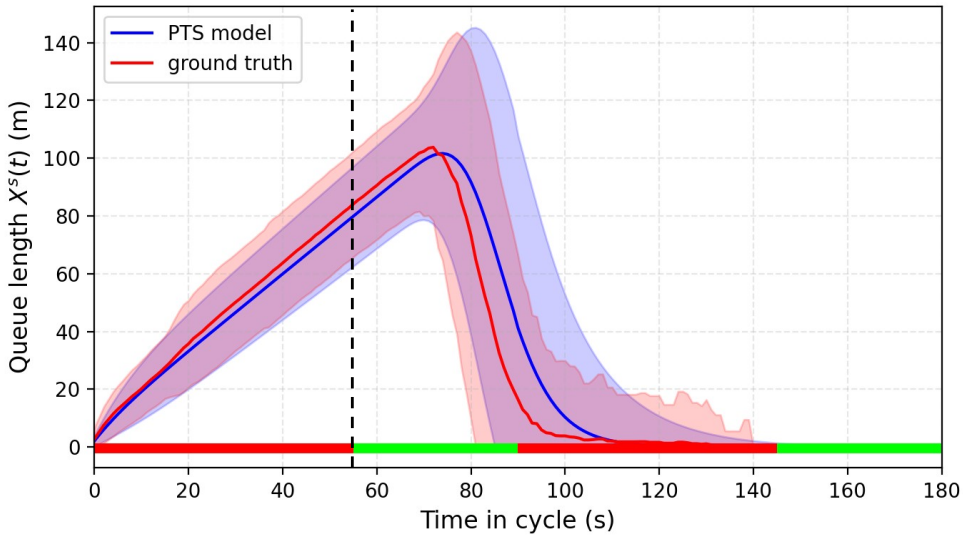


(b) PTS diagram

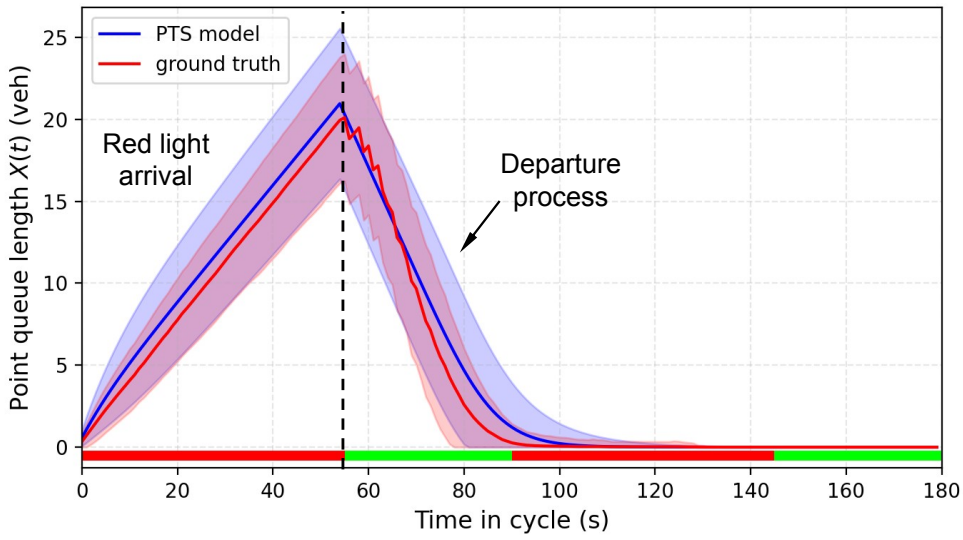
Figure 3.9: PTS diagram and point/spatial queue profiles (with the residual queue).

repeated cycles for visualization purposes. Three cycles are the same while some annotations are put to the first two cycles. As shown in both figures, a small proportion of vehicles fail to pass the intersection within the cycle they arrived at. These split-failure trajectories stop once in region M (M') and proceed to experience another stop in region N (N'). The residual queue will also lead to a different "arrival shockwave", which is defined as the boundary between the upstream moving vehicles and downstream stopped vehicles. For the scenario in the previous section without the residual queue, this arrival shockwave has a constant slope due to the constant Poisson arrival rate. In this case, as shown in Figure 3.9, the arrival shockwave has two segments: segment A-B (A'-B') has a slightly larger slope compared with segment B-C (B'-C') since there are also vehicles from

the residual queue arriving during A-B (A'-B') other than the new arrival. This is consistent with the shockwave theory (Light and Whitham, 1955; Richards, 1956). Compared with the traditional shockwave or kinematic wave theory which has a deterministic shockwave, the shockwave in the PTS diagram is not a deterministic line but stochastic.



(a) Spatial queue profile $X^s(t)$



(b) Point queue profile $X(t)$

Figure 3.10: Spatial and point queue profiles (with the residual queue).

Figure 3.10 shows the spatial and point queue profiles extracted from both the aggregated time-space diagram and PTS diagram in Figure 3.9. All queue profiles are the queue length of an average traffic cycle $X_c(t)$ instead of the total queue length $X(t)$. Please refer to Section 3.4, particularly Figure 3.4, for a recap of their difference and relationship. Therefore, all queue profiles start from

0 at the beginning of the cycle and extend to the following cycle. As shown in Figure 3.10 (b), the point queue profile of the PTS diagram is slightly different from the ground truth. For the departure process, the PTS diagram initially over-estimates the departure rate and ends up with an underestimation. This is because the PTS diagram only allows for a constant saturation flow rate (i.e., maximum departure rate) while it is monotonically increasing in the real world since the headway between vehicles decreases (Urbanik et al., 2015). As a result, the PTS diagram slightly over-estimate the residual queue at the end of the cycle. This is also the reason why the blue curve is above the red curve during the red light arrival part since the PTS model has more residual queue from the previous cycle. Compared with the point queue profile, the spatial queue profile as shown in Figure 3.9 (a) has a larger difference as well as variance since the start-up and slow-down behaviors will exaggerate their differences. Nevertheless, the PTS diagram and the resulting queue profiles match the ground truth very well, which could be further improved by adjusting some of the parameters.

3.7 Summary and discussions

3.7.1 Summary

This chapter introduces a stochastic traffic flow model which is established based on the newly proposed Newellian coordinates. It is assumed that all vehicles follow a uniform deterministic car-following model. This assumption ignores the stochastic driving behavior but works well when the uncertainty caused by stochastic traffic demand outweighs the former. Under the Newellian coordinates, we demonstrate that a point-queue model can sufficiently capture the spatial-temporal traffic state through the PTS diagram. At last, we also show how the proposed traffic flow model builds the connection between observed vehicle trajectory data and unknown traffic state/parameters, which will be utilized in the following chapter for statistical traffic state estimation.

In summary, the proposed stochastic traffic flow model has two major advantages:

1. By utilizing a point-queue representation, the proposed stochastic traffic flow has much lower dimensions and can capture the entire spatial-temporal traffic state.

2. It is compatible with the measurements provided by vehicle trajectory data, which means that it can be easily calibrated by directly taking vehicle trajectories as the input.

3.7.2 Discussions

Here are some brief discussions of the limitations of the current model and possible solutions.

Heterogeneous stochastic driving behavior Apparently, one major limitation of the proposed model is that it requires a deterministic first-order car-following behavior with constant free-flow speed and jam density. As aforementioned, this simplification does not undermine the model's accuracy if the major uncertainty comes from the stochastic traffic demand (upstream arriving traffic volumes). This is true when the penetration rate is low but becomes less valid with a higher penetration rate.

Platoon dispersion A uniform deterministic car-following model with a constant free-flow speed means that the current model cannot capture the platoon dispersion from the upstream intersection to the downstream intersection. However, this could be included if the proposed model is only utilized to model the traffic state nearby the signalized intersection (the queueing area where vehicle stop happens) while applying another platoon dispersion model to model the traffic connecting queueing areas of different signalized intersections that are far away from each other.

Deterministic departure, permissive movements The current model utilizes a deterministic departure process when the vehicle is able to depart the intersection whenever the signal state is green. However, this is not true for many cases such as the permissive movements, start-up period, etc. Some of the cases are discussed in Appendix A.3, note that Appendix A.3 is also relied on in the next chapter. Therefore, we recommend readers refer to it after finishing reading the next chapter.

Shared lane of different movements Another implicit assumption of the proposed model is that queues of different movements are separate and do not influence each other. However, this is not the case when vehicles of different movements share the same lane. For example, it is very common that the right-turn movement and through movement share the same lane. In this case, we would still need to assume that queues of different movements are separate and do not interfere with each other. One possible approximation is to use the "equivalent lane number". For example, if there is a case that the through movement and the right-turn movement share a single lane and the traffic volume ratio is $3:1$ (through : right-turn), then we can assign 0.75 lanes to the through movement and 0.25 lanes to the right-turn movements. However, this is just an approximation since different queues are still separate and the blocking between them cannot be modeled.

CHAPTER 4

Traffic State and Parameter Estimation with Uncertainty Quantification

4.1 Introduction

4.1.1 Background and related works

One of the primary limitations of the existing vehicle trajectory data, as discussed in Section 1.4, is the sparsity and incompleteness resulting from a limited penetration rate. It is crucial to estimate the overall traffic state, which serves as an essential input for traffic signal control and optimization.

Please refer to Section 1.3.3 for a more detailed literature review for traffic state estimation with vehicle trajectory data. Most related works to this chapter include Comert and Cetin (2011); Comert (2013, 2016); Zheng and Liu (2017); Zhao et al. (2019a,b); Wong et al. (2019), which utilize the stop locations of connected vehicles to estimate the traffic states or parameters such as queue length (distribution), penetration rate, and arrival rate. However, none of these studies utilize a stochastic traffic flow model and hence they have certain limitations: 1) they only utilize the stop locations of connected vehicles at certain time slots, which do not fully utilize the connected vehicle information which also contains the time when the vehicle joins the queue and the stop duration; 2) they can only estimate certain parameters such as Poisson arrival rate or penetration rate; they cannot provide the complete spatial-temporal traffic state; 3) they almost do not have the prediction ability due to the lack of a model or dynamics; 4) few of them provide the reliability or uncertainty quantification of the estimated values.

4.1.2 Overview of the chapter

Based on the proposed stochastic traffic flow model in Chapter 3, this chapter applies different statistical estimation methods to estimate both traffic states and parameters. The overall estimation problem can be decomposed into two sub-problems: 1) traffic parameter estimation and 2) traffic

state estimation. Traffic parameters (arrival rate, penetration rate, etc.) can be regarded as prior or hyper-parameters to the real-time traffic state (queue length).

For fixed-time traffic signals, we will show how the methods of moments (MM) can be used to estimate the unknown traffic parameters by assuming they are stationary within a certain TOD. The stationary traffic state can then be directly derived given the estimated traffic parameters. It is demonstrated that, by aggregating sufficient historical data, the recurrent traffic pattern can be accurately reconstructed by the proposed method.

We also apply the Bayesian estimation techniques for traffic state and parameter estimation. The Bayesian methods not only provide the point estimation of estimated values but uncertainty quantification.

4.1.3 Contributions and organization of the chapter

The contributions of this chapter are listed below:

1. We apply different statistical estimation methods (MM estimation and Bayesian estimation) to estimate both traffic state and parameter based on the previously proposed stochastic traffic flow model.

2. The Bayesian estimation also quantifies the uncertainty of the estimated values.

This chapter is organized as follows: Section 4.2 introduces the probabilistic model which is used for traffic state estimation. Section 4.3 shows how the MM can be used to estimate traffic parameters for fixed-time traffic signals. Section 4.4 and Section 4.5 are about applying Bayesian methods for estimation and uncertainty quantification. Section 4.6 is a summary of this chapter.

4.2 Probabilistic model and estimation problem formulation

4.2.1 Probabilistic graphical model

Based on the stochastic point-queue model and PTS diagram, the overall probabilistic graphical model (a Bayesian network) is given by Figure 4.1. There are three main parts: 1) Parameters Θ include the penetration rate and arrival rate. It could also contain other pre-determined and calibrated parameters such as free-flow speed, jam density, and turning ratios (Appendix B). These parameters are assumed to be stationary within a certain TOD. 2) Traffic state $\mathcal{X}$ including arrivals, departures, and queue lengths. 3) Observation $\mathcal{O}$ comes from the vehicle trajectory data. This probabilistic model enables us to use different statistical estimation methods to estimate both unknown traffic states and parameters from sparsely observed vehicle trajectories.

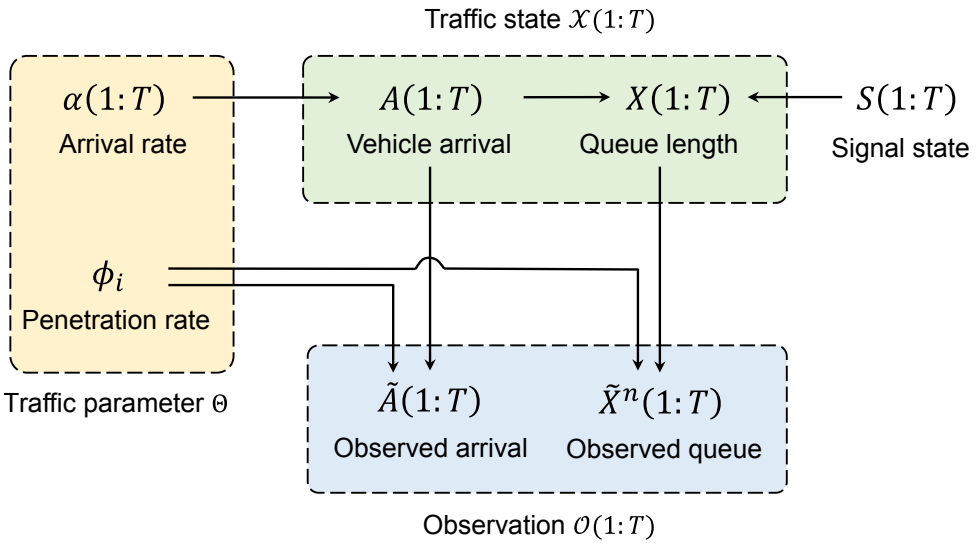


Figure 4.1: Probabilistic graphical model (Bayesian network) with observed vehicle trajectory and unknown traffic state & parameters.

4.2.2 Estimation problem decomposition

Based on the probabilistic model given in Figure 4.1, the traffic estimation problem can be decomposed into two problems: 1) stationary parameter estimation and 2) traffic state estimation. Traffic parameters are estimated first since they provide prior information for the traffic state.

Let us take maximum likelihood estimation (MLE) as an example. The parameter estimation can be formulated as:

$$\hat{\Theta}_{\text{MLE}} = \arg\max_{\Theta} p(\mathcal{O}|\Theta) = \arg\max_{\Theta} \int p(\mathcal{O}, \mathcal{X}|\Theta) d\mathcal{X}. \tag{4.1}$$

As shown by this equation, the estimation of Θ is also dependent on the latent variable $\mathcal{X}$. This is why expectation-maximization (EM) algorithm is frequently used in the related literature (Zheng and Liu, 2017; Zhao et al., 2019a,b). With estimated parameters $\hat{\Theta}$, the hidden traffic state can be estimated by finding its posterior:

$$\hat{\mathcal{X}} \sim p(\mathcal{X}|\mathcal{O}, \hat{\Theta}). \tag{4.2}$$

These two equations show how we can decompose the estimation problems into two sub-problems. They are closely related and can also be formulated as a joint problem. Instead of using MLE, this chapter will first introduce an easier method by using the methods of moments (MM) to estimate the unknown parameters and stationary traffic state for fixed-time traffic signals. Bayesian estimation is applied afterward for both traffic state and parameter estimation with uncertainty quantification.

4.3 Method of the moments (MM) estimation for fixed-time traffic signals

4.3.1 Methodology

By assuming that the penetration rate and arrival rate are stationary within a specific TOD, different frequentist methods can be used to estimate these parameters by aggregating historical data. This section uses the methods of moments (MM) estimator. The intuition is to find the parameters such that the observed average delay and the model-estimated delay are equivalent:

$$\mathbb{E}\left[\hat{d}(\hat{\Theta}_{\text{MM}})\right] = \tilde{d} \tag{4.3}$$

where $\hat{d}(\Theta)$ is the estimated average delay given input parameter Θ while $\tilde{d}$ is the average control delay directly measured from the observed trajectories. Throughout the entire dissertation, we use the superscript tilde ($\sim$) to indicate that this variable is obtained from the observed vehicle trajectory.

Historical data from multiple cycles is needed for the method of moments (MM) estimator. Let $\tilde{a}(t)$ represent the total number of observed arrivals in a cycle (arrival histogram in Figure 4.3) by aggregating trajectories from N_c cycles. Given the penetration ϕ, the arrival rate of each time in the cycle can be estimated as:

$$\hat{a}(t) = \frac{\tilde{a}(t)}{N_c \Delta u \phi}, \quad \forall t \in \{1, \ldots, T\}. \tag{4.4}$$

Utilizing this estimated arrival profile as the input, the average delay will be a function of penetration rate ϕ and can be written as $\hat{d}(\phi)$. $\hat{d}(\phi)$ is the model-estimated average control delay which is calculated according to Section 3.3. Then the penetration rate can be estimated according to the following equation:

$$\phi^* = \arg\min_{\phi} \left[\hat{d}(\phi) - \tilde{d}\right]^2. \tag{4.5}$$

We also apply this method to estimate the penetration rates of multiple movements in a network of signalized intersections. For a movement with upstream arrival, the arrival from the upstream movement is estimated by an affine transformation of the upstream departure through a shift and scaling down (Appendix A.3). The shift duration is determined by the free-flow travel time and the relative offset, while the scaling coefficient is the turning ratio which can be directly calculated from the observed vehicle trajectory data. Since the penetration rates of different movements are close but different, the following centralized formulation is used to estimate the penetration rates

of multiple movements in a network ($\mathcal{M}$ is the set of movements):

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \sum_{i \in \mathcal{M}} \tilde{n}_i \left[\hat{d}_i(\phi_i) - \tilde{d}_i \right]^2 + \beta \mathbb{V}(\boldsymbol{\phi}) \tag{4.6}$$

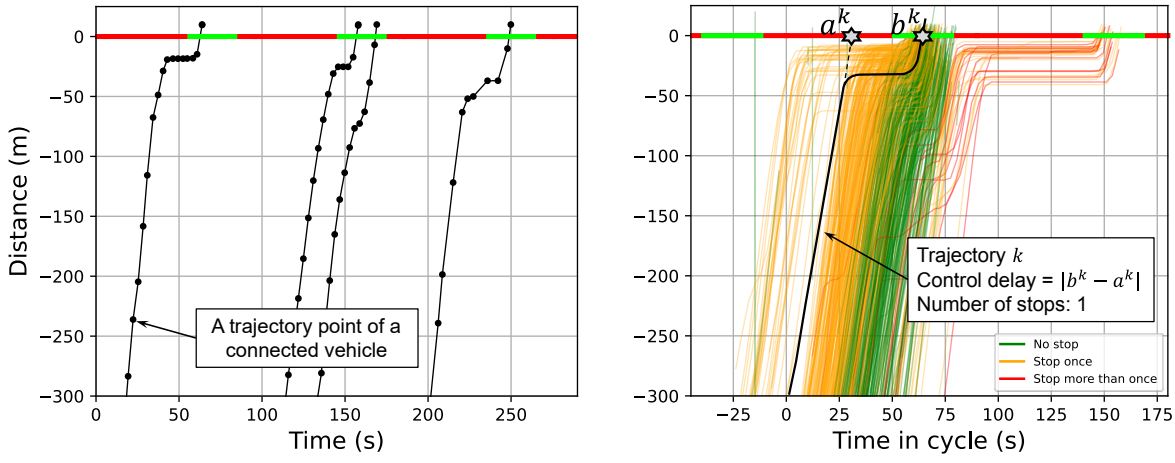
where $\boldsymbol{\phi}$ is a column vector consisting of penetration rates of all the movements, $\tilde{n}_i$ is the total number of observed trajectories of movement i, and $\mathbb{V}(\boldsymbol{\phi})$ is the variance of the penetration rates weighted by total delay $\tilde{n}_i \tilde{d}_i$:

$$\mathbb{V}(\boldsymbol{\phi}) = \frac{1}{\sum_{i \in \mathcal{M}} \tilde{n}_i \tilde{d}_i} \sum_{i \in \mathcal{M}} \tilde{n}_i \tilde{d}_i \cdot (\phi_i - \bar{\phi})^2 \tag{4.7}$$

where

$$\bar{\phi} = \frac{1}{\sum_{i \in \mathcal{M}} \tilde{n}_i \tilde{d}_i} \sum_{i \in \mathcal{M}} \tilde{n}_i \tilde{d}_i \cdot \phi_i. \tag{4.8}$$

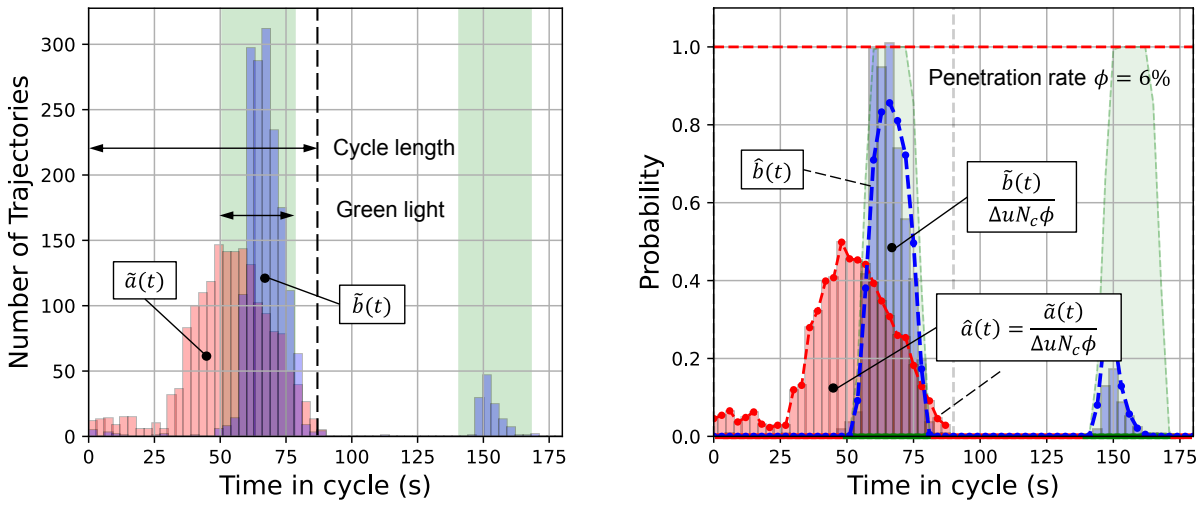
The first term of Equation (4.6) is the summation of the delay difference between the traffic model and the observed trajectories weighted by the number of vehicles $\tilde{n}_i$. The second term is a regularization through the dispersion of penetration rates. β is the coefficient of the regularization term. A larger β will lead to more densely distributed penetration rates. If β is sufficiently large, each movement will have the same penetration rate. Based on this centralized formulation, more congested movements with more delay will have a larger influence on the overall estimation program and will improve the estimation accuracy of the less congested movements.



(a) TS diagram

(b) Aggregated TS diagram

Figure 4.2: Aggregated time-space diagram of the example movement.



(a) Arrival/departure histograms

(b) Scaled histogram

Figure 4.3: Point-queue arrival and departure profiles.

4.3.2 Case studies with real-world trajectories

Figure 4.2-4.5 is an illustration of parameter estimation for a specific movement (i.e., direction through an intersection). Figure 4.2 (a) shows a short period (3 cycles) of the time-space (TS) diagram where the observed trajectories are sparse due to the low penetration rate. Since fixed-time traffic signal states are cyclic and traffic demand is assumed to be stationary within a certain TOD (also periodic within the same cycle), the stochastic point-queue model of the movement will converge to a stationary traffic cycle. Correspondingly, as shown in Figure 4.2 (b), trajectories can be aggregated to one cycle to get the aggregated TS diagram. By assuming that the observable connected vehicles are randomly distributed among all vehicles, the aggregated TS diagram shows the average and recurrent traffic state which directly corresponds to the stationary cycle of this movement. Figure 4.3 (a) shows arrival and departure time histograms of all the trajectories in Figure 4.2 (b). Note that since vehicle trajectories are aggregated according to their free-flow arrival times, some vehicles might depart in the following cycle if they fail to pass the intersection within the cycle in which they arrived.

Given sufficient vehicle trajectory data, the arrival and departure probability profiles can be estimated by scaling down the histograms ($\tilde{a}(t)$ and $\tilde{b}(t)$ in Figure 4.3 (a)) according to Equation (4.4). The red and blue bars in Figure 4.3 (b) show the scaled arrival and departure probability profiles. Using the scaled arrival probability as the cyclic input arrival profile $\hat{a}(t)$ (red dashed line in Figure 4.3 (b)), the blue dashed line is the resulting departure probability profile $\hat{b}(t)$ estimated from the stochastic queueing model. The average delay per vehicle $\hat{d}(\phi)$ can also be calculated. Figure 4.4 shows how the model-estimated average delay $\hat{d}(\phi)$ changes with different penetration

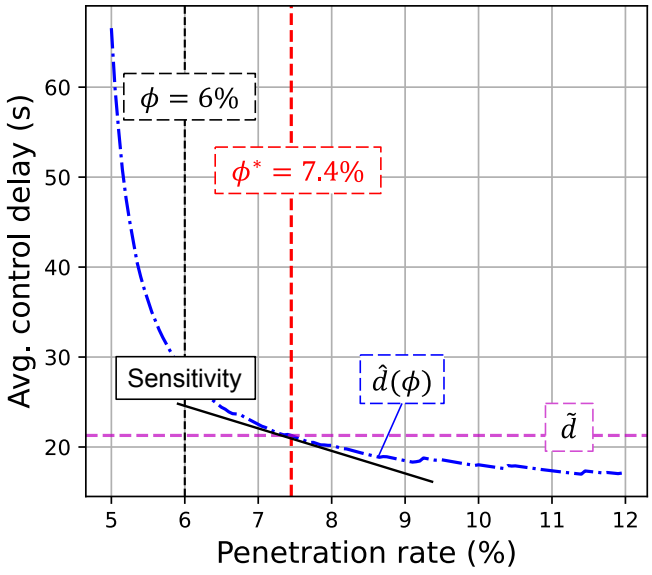
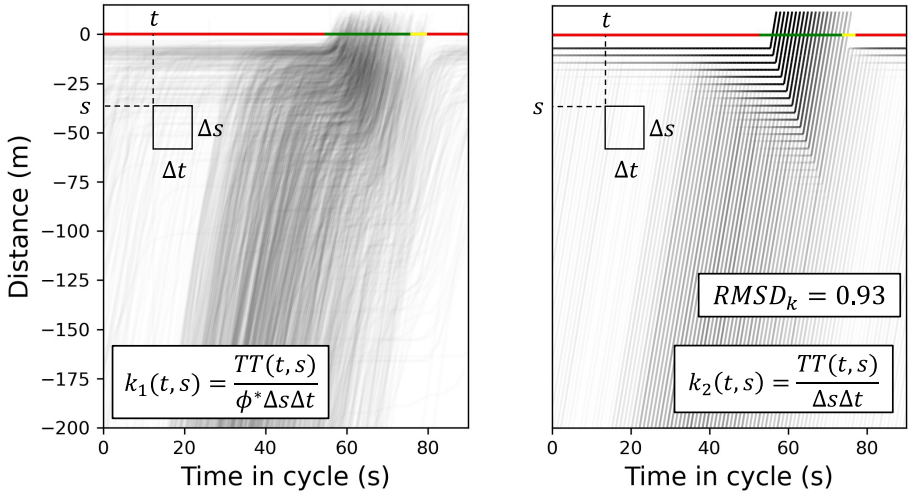


Figure 4.4: Penetration rate estimation of an isolated movement.

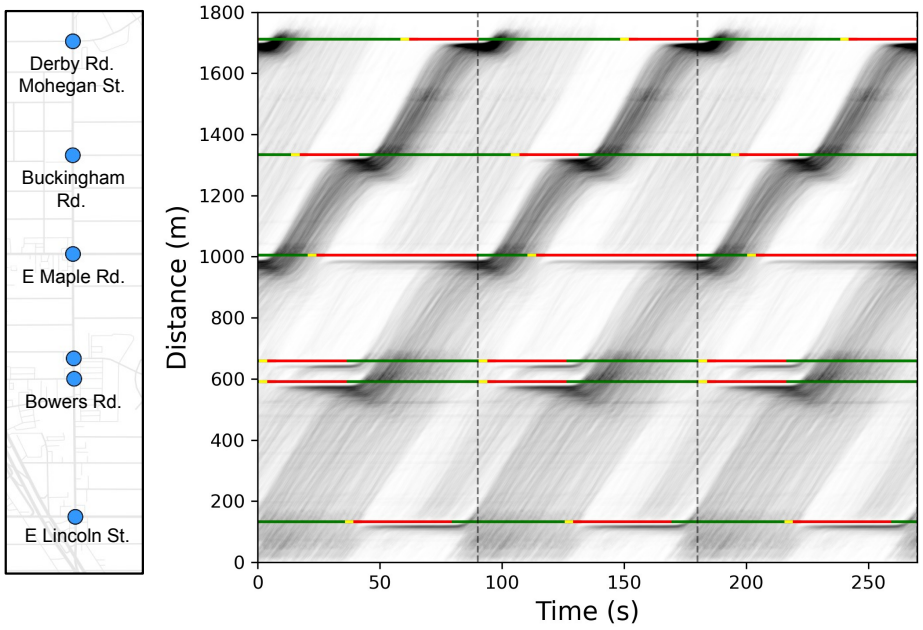
rates ϕ. Therefore, the optimal penetration rate ϕ^* can be determined under which the model-estimated average delay $\hat{d}(\phi)$ matches the measurement $\tilde{d}$ from the observed trajectories.



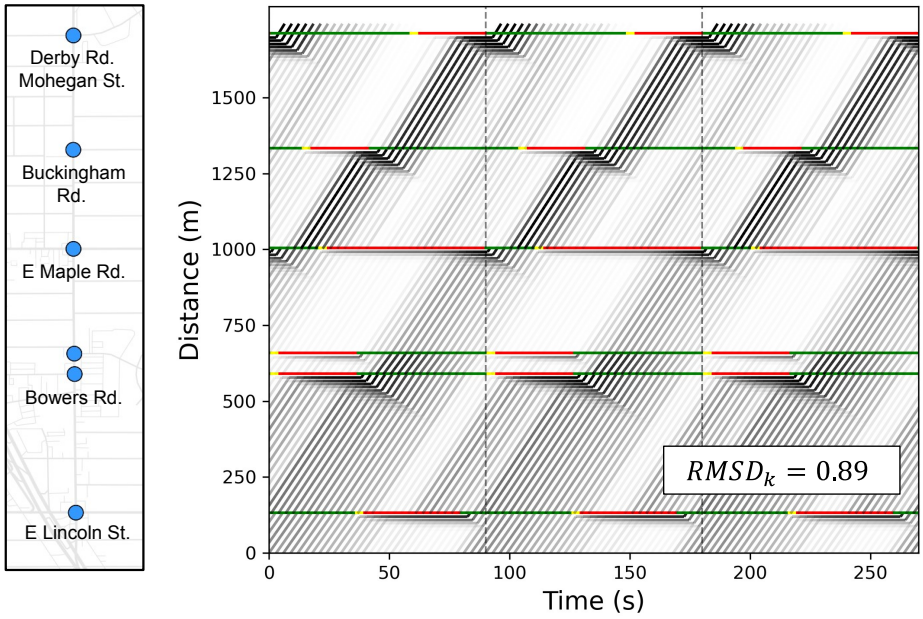
(a) Agg. TS diagram (b) Reconstructed PTS diagram

Figure 4.5: Original aggregated TS diagram and reconstructed PTS diagram.

With the estimated penetration rate, Figure 4.5 shows the comparison between the PTS diagram and the aggregated TS diagram. To evaluate and validate the constructed PTS diagram, both diagrams in Figure 4.5 are split into small grids given certain spatial and temporal intervals and the traffic densities of each cell are calculated. The root-mean-square derivation (RMSD) of the traffic



(a) Aggregated TS diagram



(b) Reconstructed PTS diagram

Figure 4.6: Original aggregated TS diagram and reconstructed PTS diagram (corridor).

densities (unit: $veh/100m$) quantifies their difference. A similar estimation method can also be applied to a corridor consisting of multiple movements. Figure 4.6 shows the results of the traffic state estimation of a corridor (northbound direction as an example, similar plots can be generated for the southbound). Figure 4.6 (a) is the corridor aggregated TS diagram, which is generated

by combining the aggregated TS diagrams of all movements along the path. For visualization purposes, the aggregated TS diagrams for each movement are repeated over several cycles so that trajectories can traverse the whole corridor. Figure 4.6 (b) shows the corresponding PTS diagram, which matches the aggregated corridor TS diagram well and demonstrates the effectiveness of the method for traffic state and parameter estimation.

4.4 Bayesian estimation

4.4.1 Motivation

The previous section introduces a simple and intuitive method to estimate the penetration rate and arrival rate by matching the control delay between the model-estimated value and the observed value. Although it is a simple, effective, and practical algorithm, there are some limitations: 1) it only estimates the stationary traffic state and is more suitable for fixed-time traffic signals; 2) it cannot quantify the uncertainty of estimated values.

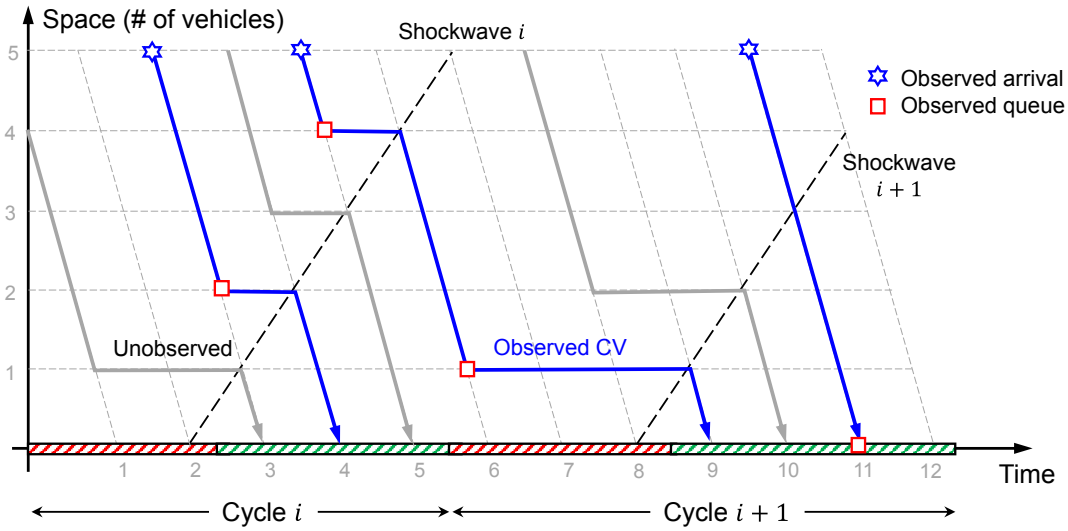
In this section, we will introduce a more formal method using Bayesian estimation that can 1) estimate both stationary parameters and real-time traffic states; 2) provide the distribution of the estimated value instead of a single value. Besides, we will see in the discussion (Section 4.6) that Bayesian models are more flexible and can be easily extended to fit more complex settings and assumptions.

4.4.2 Hidden Markov model

Based on the proposed stochastic traffic flow model, the overall system with observed vehicle trajectory data under a certain penetration rate is essentially a hidden Markov model. The dynamics of the hidden state $X(t)$ have been introduced in Chapter 3. It is easy to verify that, $X(t)$ is a discrete Markov chain according to the transition given by Equation (3.6) or Equation (3.21-3.22). This subsection will focus on the observation model.

Figure 4.7 is an illustration of the observation model and the encoding of observed vehicle trajectory data. There are three observed trajectories that arrive at time $t = 3$, $t = 5$, and $t = 11$. Therefore, we have the observed arrival $\tilde{A}(t) = 1$ for $t = 3, 5, 11$ and $\tilde{A}(t) = 0$ for the rest of the time slots. These three observed trajectories represent different cases:

1. The first trajectory arrives at time 3 is a typical trajectory that stops once before passing the intersection. Other than the observed arrival $\tilde{A}(3) = 1$, we also have the observed queue length $\tilde{X}_i^n(3) = 2$ since the trajectory stops at location 2. The superscript s indicates that this is the spatial queue while the subscript i is the index of the cycle.



Time t	1	2	3	4	5	6	7	8	9	10	11	12
Arrival $\tilde{A}(t)$	0	0	**1**	0	**1**	0	0	0	0	0	**1**	0
Queue $\tilde{X}_c^n(t)$	/	/	**2**	/	**4**	**1**	/	/	/	/	**0**	/
Cycle c	/	/	***i***	/	***i***	***i*+1**	/	/	/	/	***i*+1**	/

Figure 4.7: Observation model and encoding of observed vehicle trajectories.

2. The second trajectory (arrives at time 5) is an over-saturated case that stops twice before passing the intersection. For the first stop, we have the observed queue length $\tilde{X}_i^n(5) = 4$. Besides, we also have the second stop $\tilde{X}_{i+1}^n(6) = 1$. Note that the subscripts are different since these two stops belong to different cycles.

3. The third trajectory (arrives at time 11) directly passes the intersection without a stop. Therefore, we have the observed queue length $\tilde{X}_{i+1}^n(11) = 0$.

It turns out the over-saturated case (with the residual queue) is similar to the under-saturation case but is more complicated and tedious. Therefore, we will only focus on a simple scenario when there is no over-saturation or residual queue.

Figure 4.7 shows the overall hidden Markov model (without the residual queue) based on the stochastic traffic flow model and the observation model. The hidden layer consists of the queue length $X(t)$ and arrival $A(t)$ of all vehicles (both observable and unobservable). The transition of the hidden state is given by the stochastic traffic flow model introduced in Chapter 3. For each newly arrived vehicle trajectory, we have probability ϕ (penetration rate) to observe it. Whenever we observe a new arrival, that is, $\tilde{A}(t) = 1$, we can also observe the corresponding stop location $\tilde{X}^n(t)$.

Here we will provide the mathematical formulation of the hidden Markov model. The overall

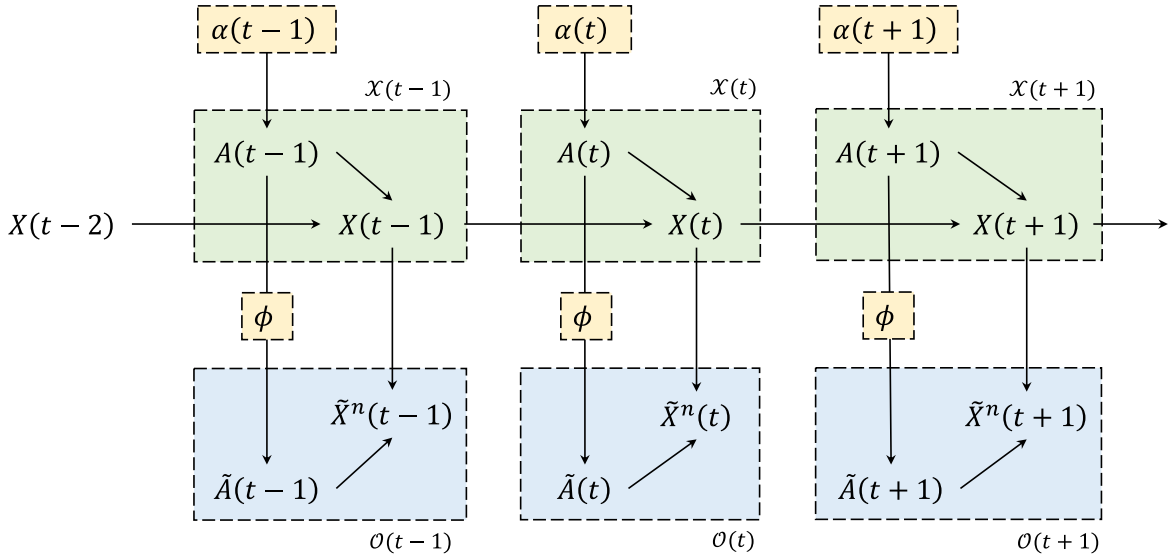


Figure 4.8: Hidden Markov model (without the residual queue).

probabilistic model can be written as:

$$p(\mathcal{O}(1:T), \mathcal{X}(1:T), \Theta) = p(\Theta)p(\mathcal{X}(1:T)|\Theta)p(\mathcal{O}(1:T)|\mathcal{X}(1:T), \Theta) \tag{4.9}$$

where $p(\Theta)$ is the prior of the traffic parameters, which can be set as a uniform distribution, i.e., $p(\Theta) = 1$, if there is no prior information. The hidden layer $p(\mathcal{X}(1:T)|\Theta)$ is the stochastic traffic flow model which can be decomposed as:

$$p(\mathcal{X}(1:T)|\Theta) = p(\mathcal{X}(1)) \prod_{t=1}^{T} p(\mathcal{X}(t+1)|\mathcal{X}(t), \Theta), \tag{4.10}$$

$$p(\mathcal{X}(t+1)|\mathcal{X}(t), \Theta) = p(A(t+1)|\alpha(t+1))p(X(t+1)|X(t), A(t+1), S(t+1)). \tag{4.11}$$

where $p(X(t+1)|X(t), A(t+1), S(t+1))$ is determined by Equation (3.6-3.7) in Section 3.2.2.

The observation model can be written as:

$$p(\mathcal{O}(1:T)|\mathcal{X}(1:T), \Theta) = \prod_{t=1}^{T} p(\mathcal{O}(t)|\mathcal{X}(t), \phi), \tag{4.12}$$

$$p(\mathcal{O}(t)|\mathcal{X}(t), \phi) = p(\tilde{A}(t)|A(t), \phi)p(\tilde{X}^n(t)|X(t), \tilde{A}(t)). \tag{4.13}$$

As mentioned before, the observation model can be decomposed into 1) whether a new arrival is observed $p(\tilde{A}(t)|A(t), \phi)$ and 2) what is the observed queue length $p(\tilde{X}(t)|X^n(t), \tilde{A}(t))$.

The observed arrival $p(\tilde{A}(t)|A(t),\phi)$ is determined by:

$$p(\tilde{A}(t)=1|A(t)=1,\phi)=\phi \tag{4.14a}$$

$$p(\tilde{A}(t)=0|A(t)=1,\phi)=1-\phi \tag{4.14b}$$

$$p(\tilde{A}(t)=0|A(t)=0,\phi)=1 \tag{4.14c}$$

Equation (4.14a-4.14b) means that we have probability ϕ and $1-\phi$ to observe or miss a new arrival, respectively. Equation (4.14c) means that we cannot observe an arrival if there is no new arrival at all. Note that to make sure Equation (4.14) holds, each vehicle trajectory needs to exactly correspond to one unit traffic flow. This can be easily achieved by choosing a proper time interval Δt such that $\Delta u = 1$ in Equation (3.1) (refer to the discrete approximation in Section 3.2.1). Otherwise, one new observed arrival at time t will not be consistent with $A(t)=1$.

The observed queue length $p(\tilde{X}^n(t)|X(t),\tilde{A}(t))$ is determined by:

$$\tilde{X}^n(t)=\begin{cases}\Psi_t^{-1}(X(t)) & \tilde{A}(t)=1\\ \emptyset & \tilde{A}(t)=0\end{cases}. \tag{4.15}$$

which means that whenever we observe a new arrival $\tilde{A}(t)=1$. We will also see the true queue length $\Psi_t^{-1}(X(t))$. As a simplification, we assume no noise or errors in the observed queue length (Assumption 4.1). Although this assumption might lead to a slight underestimation of the system uncertainty, it is much less compared with the uncertainty caused by incomplete observation, especially at a low penetration rate. However, it will undermine the estimation performance when the penetration rate is high. This assumption can be addressed by applying an additional noise model. See discussions in Section 4.6.2 for more details.

Assumption 4.1. *It is assumed that there are no errors or noise in the observed queue length. This means that we assume that 1) there is no noise or errors in the collected GNSS coordinates; 2) queue lengths of multiple lanes are equivalent; and 3) jam space headway is a deterministic constant. These assumptions can be relaxed by using a more accurate noise model; refer to discussions in Section 4.6.2 for more details.*

4.4.3 Filtering and marginal likelihood calculation of the hidden Markov model

The previous section has provided the complete mathematical formulation of the hidden Markov model with hidden layer $\{A(t),X(t)\}$ and the observation layer $\{\tilde{A}(t),\tilde{X}^n(t)\}$. Other than the unknown hidden layer, traffic parameters Θ including the arrival rate and the penetration rate are

also unknown. Therefore, there are also two estimation problems: 1) traffic parameter estimation and 2) traffic state (hidden state) estimation. This subsection will briefly introduce the recursive algorithm that is used by both estimation problems. Readers can refer to Doucet et al. (2009) for more details on the derivation.

The Bayesian estimation of traffic parameter Θ is to find the posterior distribution:

$$p(\Theta|\mathcal{O}(1:T)) = p(\Theta)p(\mathcal{O}(1:T)|\Theta) \tag{4.16}$$

where $p(\Theta)$ is the prior and $p(\mathcal{O}(1:T)|\Theta)$ is the so-called marginal likelihood function:

$$p(\mathcal{O}(1:T)|\Theta) = \int p(\mathcal{O}(1:T), \mathcal{X}(1:T)|\Theta) d\mathcal{X}(1:T). \tag{4.17}$$

Traffic state estimation is to find the posterior distribution:

$$p(\mathcal{X}(1:T)|\mathcal{O}(1:T)) = \int p(\mathcal{X}(1:T)|\mathcal{O}(1:T), \Theta) p(\Theta|\mathcal{O}(1:T)) d\Theta. \tag{4.18}$$

In practice, instead of finding the posterior according to Equation (4.18), we usually only perform the following real-time estimation:

$$p(\mathcal{X}(t)|\mathcal{O}(1:t)) = \int p(\mathcal{X}(t)|\mathcal{O}(1:t), \Theta) \underbrace{p(\Theta|\mathcal{O}(1:T))}_{\text{Eq. (4.16)}} d\Theta, \quad \forall t. \tag{4.19}$$

In the literature, Equation (4.18) is called smoothing while Equation (4.19) is called filtering. The former one finds the posterior of the hidden state at time t based on all available observations from $1:T$ while the latter one finds the posterior at time t only based on the current and previous observations from $1:t$. The filtering problems only require a forward recursive calculation while the smoothing problems need both forward and backward calculations.

Here we will show how we can calculate Equation (4.17) and Equation (4.19) through a recursive method:

$$p(\mathcal{X}(t+1)|\mathcal{O}(1:t+1), \Theta) = \frac{p(\mathcal{O}(t+1)|\mathcal{X}(t+1), \Theta) p(\mathcal{X}(t+1)|\mathcal{O}(1:t), \Theta)}{p(\mathcal{O}(t+1)|\mathcal{O}(1:t), \Theta)} \tag{4.20}$$

where:

$$p(\mathcal{X}(t+1)|\mathcal{O}(1:t), \Theta) = \int p(\mathcal{X}(t+1)|\mathcal{X}(t), \Theta) p(\mathcal{X}(t)|\mathcal{O}(1:t)) d\mathcal{X}(t), \tag{4.21}$$

$$p(\mathcal{O}(t+1)|\mathcal{O}(1:t),\Theta) = \int p(\mathcal{X}(t)|\mathcal{O}(1:t),\Theta)p(\mathcal{X}(t+1)|\mathcal{X}(t),\Theta) \cdot p(\mathcal{O}(t+1)|\mathcal{X}(t+1),\Theta)d\mathcal{X}(t:t+1). \tag{4.22}$$

Equation (4.21) is called the prediction step, which can be derived by moving one time step further based on the previous estimation $p(\mathcal{X}(t)|\mathcal{O}(1:t))$. Equation (4.20-4.22) is a recursive process since the input is the previous estimation $p(\mathcal{X}(t)|\mathcal{O}(1:t))$ while the output is the estimation of the next time $p(\mathcal{X}(t+1)|\mathcal{O}(1:t+1))$.

Equation (4.22) can also be used to calculate the marginal likelihood function in Equation (4.17) through the following factorization:

$$p(\mathcal{O}(1:T)|\Theta) = p(\mathcal{O}(1)|\Theta)\prod_{t=1}^{T-1} p(\mathcal{O}(t+1)|\mathcal{O}(1:t),\Theta). \tag{4.23}$$

In practice, we calculate the log-likelihood function defined below:

$$L(\mathcal{O}(1:T),\Theta) = \log(p(\mathcal{O}(1:T)|\Theta)) = \log(p(\mathcal{O}(1)|\Theta)) + \sum_{t=1}^{T}\log(p(\mathcal{O}(t+1)|\mathcal{O}(1:t),\Theta)). \tag{4.24}$$

This subsection only provides a high-level mathematical formulation while the next subsection will introduce the actual detailed algorithms that are used to estimate both unknown traffic parameters and states.

4.4.4 Estimation algorithms

Algorithm 2 is an implementation of the recursive calculation introduced in the previous subsection. Given input traffic parameters Θ including arrival rate and penetration rate, Algorithm 2 outputs 1) the real-time estimation results of the hidden traffic state (filtering): $p(\mathcal{X}(t)|\mathcal{O}(1:t),\Theta)$; and 2) the overall marginal log-likelihood given the current input parameters: $L(\mathcal{O}(1:T),\Theta) = \log(p(\mathcal{O}(1:T)|\Theta))$. All the following estimation algorithms are based on Algorithm 2.

Remark 4.1. *Selection of the initial queue length distribution in Algorithm 2. We can either run multiple cycles as a warm-up period to get the initial queue length distribution or use the stationary queue length distribution as described in Section 3.3.*

Based on the marginal log-likelihood function calculated through Algorithm 2, as a frequentist method, the traffic parameter can be estimated through MLE:

$$\hat{\Theta}_{\text{MLE}} = \arg\max_{\Theta} L(\mathcal{O}(1:T),\Theta), \tag{4.35}$$

Algorithm 2: Recursive calculation of the hidden Markov models

Input: Arrival rate $\boldsymbol{a} = [a(1), \ldots, a(T)]$ and penetration rate ϕ; observed arrival $[\tilde{A}(1), \ldots, \tilde{A}(T)]$ and observed queue lengths $[\tilde{X}^n(1), \ldots, \tilde{X}^n(T)]$ (Figure 4.7); Traffic signal state $\boldsymbol{S} = [S(1), \ldots, S(T)]$; maximum queue N.

Initiation: Initial queue length distribution $x(1, k)$ at time 0 (Remark 4.1), initial marginal log-likelihood function $L(0) = 0$ where $L(t) = L(\mathcal{O}(1:t)|\Theta)$.

for $t = 1, 2, ..., T$ **do**

Given observed arrival $\tilde{A}(t) \in \{0, 1\}$, the estimated arrival probability at time t:

$$\hat{a}(t) = \begin{cases} 1 & \tilde{A}(t) = 1 \\ a(t) \cdot (1 - \phi) & \tilde{A}(t) = 0. \end{cases} \tag{4.25}$$

Prediction step of the queue length distribution:

for $k = 0, 1, ..., N$ **do**

$$\bar{x}'(t, k) = \hat{x}(t-1, k-1) \cdot \hat{a}(t) + \hat{x}(t-1, k) \cdot (1 - \hat{a}(t)) \tag{4.26}$$

if $S(t) = 1$ **then**

$$\bar{x}(t, 0) = \bar{x}'(t, 0) + \bar{x}'(t, 1) \tag{4.27}$$

$$\bar{x}(t, k) = \bar{x}'(t, k+1), \quad \forall k = 1, \ldots, N-1 \tag{4.28}$$

$$\bar{x}(t, N) = 1 - \sum_{i=0}^{N-1} \bar{x}(t, i) \tag{4.29}$$

else

$$\bar{x}(t, k) = \bar{x}'(t, k), \forall k \tag{4.30}$$

Update the log-likelihood calculation:

if $\hat{a}(t) = 1$ **then**

There is an observed queue $\tilde{X}^n(t)$, point queue $X(t) = \Psi_t(X^n(t))$

$$L(t) = L(t-1) + \log(a(t) \cdot \phi) + \log(\bar{x}(t, X(t))) \tag{4.31}$$

else

$$L(t) = L(t-1) + \log((1 - a(t)) + a(t) * (1 - \phi)) \tag{4.32}$$

Update step of the queue length distribution:

if $\hat{a}(t) = 1$ **then**

$$\hat{x}(t, k) = \begin{cases} 1 & k = X(t) \\ 0 & otherwise. \end{cases} \tag{4.33}$$

else

$$\hat{x}(t, k) = \bar{x}(t, k), \forall k \tag{4.34}$$

Return: Marginal likelihood function $L(T) = L(\mathcal{O}(1:T), \Theta)$ and estimated queue length distribution $\hat{x}(t, k),\ \forall t \in \{1, \ldots, T\}$ given parameter Θ.

which will provide a point estimation of the traffic parameters. However, this point estimation cannot quantify the uncertainty of the estimated values. Instead of using the frequentist method, the Bayesian method will be used to get the posterior distribution of the unknown parameters $p(\Theta|\mathcal{O}(1:T))$ so that we will directly have a distribution of the estimated values.

The Bayesian estimation of the traffic parameters is based on the following Bayes' theorem:

$$p(\Theta|\mathcal{O}(1:T)) = \frac{p(\mathcal{O}(1:T)|\Theta)p(\Theta)}{p(\mathcal{O}(1:T))} \rightarrow p(\Theta|\mathcal{O}(1:T)) \propto p(\mathcal{O}(1:T)|\Theta) \cdot p(\Theta) \tag{4.36}$$

where $p(\Theta)$ is the prior of the traffic parameters.

There are different numerical methods to estimate the posterior distribution given by Equation (4.36). For example, if we assume that the arrival process is a homogeneous Poisson process with a single parameter, there are only two parameters (arrival rate α and penetration rate ϕ) to be estimated. In this low-dimension case, grid sampling can be used to approximate the posterior distribution: with given ranges and resolutions of both parameters, the parameter space can be split into a mesh grid and the likelihood of each point in the mesh grid can be calculated. Then the probability of each point can be estimated by normalizing the total probability.

However, grid sampling is a low-efficient algorithm. There are many other more advanced and efficient sampling methods such as the importance sampling and Markov chain Monte Carlo (MCMC) (Gelman et al., 2013). Here we will introduce one specific estimation procedure based on importance sampling. There are four main steps:

1. Finding the maximum a posterior (MAP). The MAP is given by:

$$\hat{\Theta}_{\text{MAP}} = \arg\max_{\Theta} = p(\Theta|\mathcal{O}(1:T)) = \arg\max_{\Theta} p(\Theta)p(\mathcal{O}(1:T)|\Theta) \tag{4.37}$$

   which will have the same results as the MLE when selecting a uniform prior $p(\Theta) = 1$.

2. Applying Laplace's approximation (see MacKay (2003) for more details). The observed Fisher information is given by:

$$\mathcal{J}(\mathcal{O}(1:T), \hat{\Theta}_{\text{MAP}}) = -\nabla_{\Theta}\nabla_{\Theta}^{T} L(\mathcal{O}(1:T), \Theta)|_{\hat{\Theta}_{\text{MAP}}} \tag{4.38}$$

   Laplace's approximation will give a Gaussian approximation of the posterior distribution:

$$\hat{\Theta}_{\text{Laplace}} \sim \mathcal{N}(\hat{\Theta}_{\text{MAP}}, \mathcal{J}^{-1}(\mathcal{O}(1:T), \hat{\Theta}_{\text{MAP}})) \tag{4.39}$$

   where the covariance matrix of the Gaussian distribution is the inverse of the observed Fisher information.

3. Importance sampling of the traffic parameters. Since it turns out that Laplace's approximation provides a good approximation of the posterior distribution, we can use importance sampling with the approximated Gaussian distribution as the proposal distribution. We will not provide the details of the importance sampling since it is a simple and standard algorithm (Tokdar and Kass, 2010). The output of the importance sampling will be a set of sampled points with the corresponding weights: $\{(\Theta_i, w_i), \forall i\}$.

4. Real-time queue length estimation. Based on the results of the importance sampling $\{(\Theta_i, w_i), \forall i\}$, the real-time traffic state estimation is determined by:

$$p(\mathcal{X}(t)|\mathcal{O}(1:t)) = \frac{1}{\sum_i w_i} \sum_i w_i \cdot p(\mathcal{X}(t)|\mathcal{O}(1:t), \Theta_i) \tag{4.40}$$

which is a weighted sum of all the sampled points according to the associated weights.

This procedure is much more efficient since it requires much less calculation of Algorithm 2. In practice, this procedure can be further simplified if an accurate posterior distribution is not required: Laplace's approximation in step 2 can be directly used as the Bayesian estimation of the traffic parameter and the real-time traffic state can be directly derived based on the single MAP value.

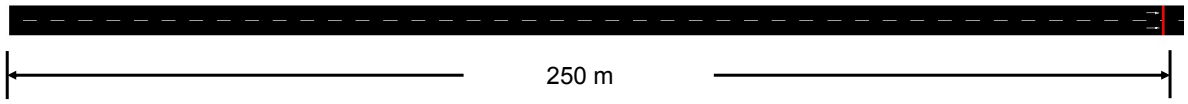
4.5 Simulation studies of Bayesian traffic state estimation

4.5.1 Simulation configuration

We use a simulation environment to test the proposed methods introduced in the previous section. Figure 4.9 is an illustration of the setup of the testing scenario. There is only one movement with two lanes; different parameters are included in the figure. It is also assumed that the vehicle arrival is a Poisson process with a constant arrival rate and the penetration rate is also a constant. The next two subsections will show the estimation results for traffic parameters and real-time queue length, respectively.

4.5.2 Parameter estimation

Figure 4.10 shows the log-likelihood function and the corresponding posterior distribution by using the grid sampling estimation. The arrival rate ranges from 540 vph to 900 vph with an interval of 9 vph (vehicle per hour) while the penetration rate ranges from 5% to 15% with an interval of 0.5%. Figure 4.10 (b) is the final result of the posterior distribution by using a uniform prior. As shown in



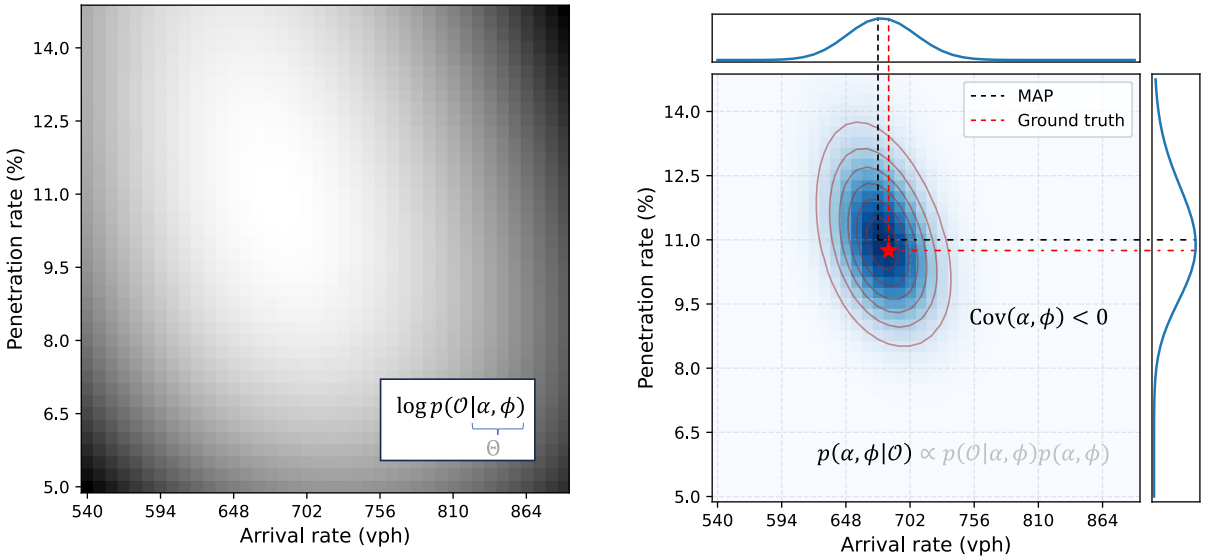
SUMO setup

- Single movement
- Number of lanes: 2
- Departure lane: random
- Simulation $\Delta t = 0.5s$
- Arrival: uniform
- SPaT: $C = 90s, G = 35s$
- Jam density: $7.5m/veh$
- Total simulation time: $20hrs$
- Stop bar distance: $1m$

Other parameters

- Stop speed threshold: $1m/s$
- Reaction time $2s$
- Warm-up period: 50 cycles

Figure 4.9: Simulation setup for Bayesian traffic state estimation.



(a) Log-likelihood function

(b) Posterior distribution

Figure 4.10: Log-likelihood function and posterior distribution.

the figure, the estimated result is given by a distribution instead of a single value. The peak value of the distribution (the intersection of two black dashed lines) can also be used as a point estimation, which is essentially the MLE or MAP since a uniform prior is used. The red dot is the ground truth. Another observation in this figure is that the estimated penetration rate and the penetration rate are negatively correlated, that is, $\mathrm{cov}(\alpha, \phi) < 0$. This is because the production of these two values is approximately the number of observed vehicles, which is a given constant. This means

that an overestimation of one of these two parameters might lead to the underestimation of another one.

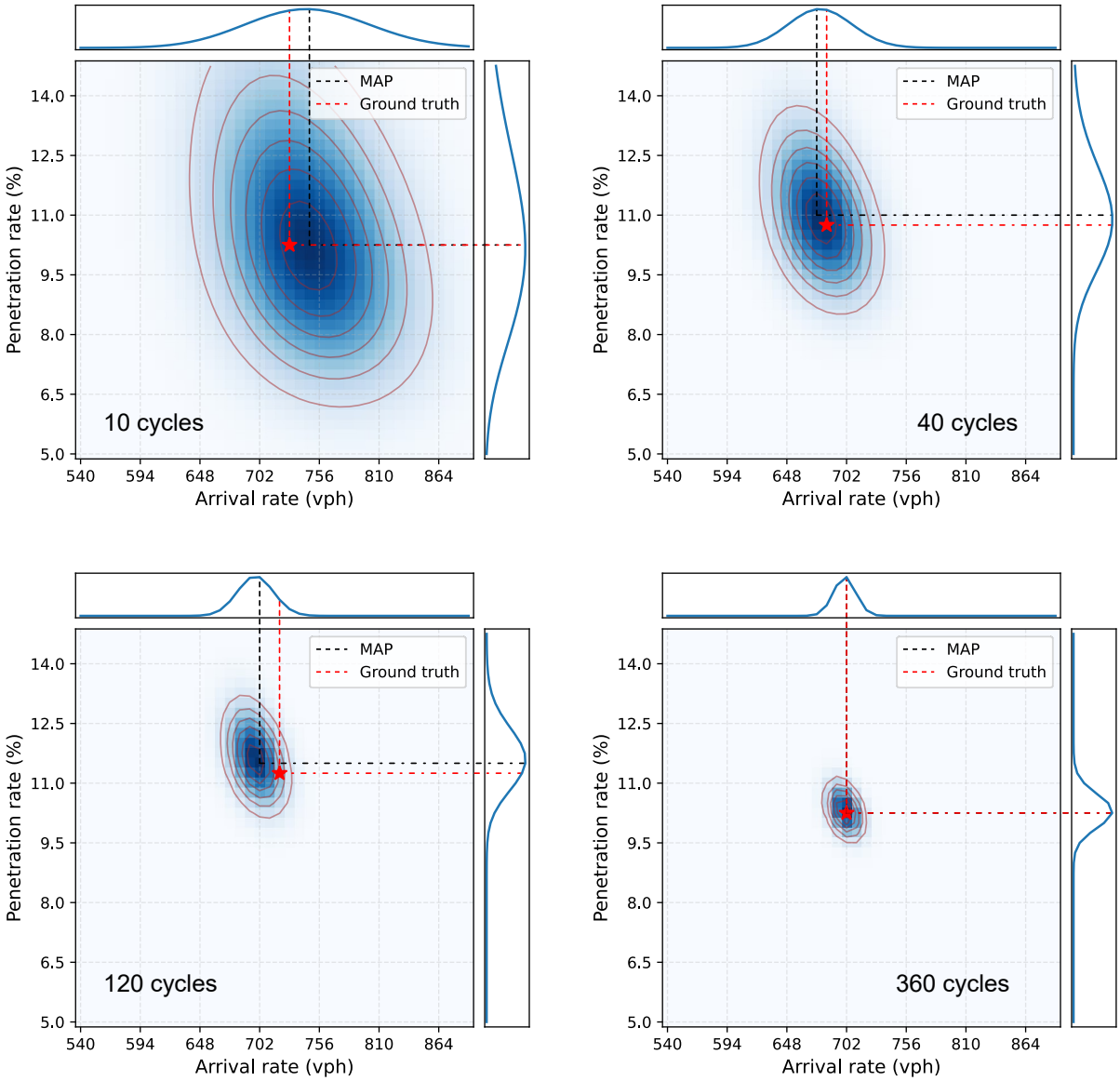
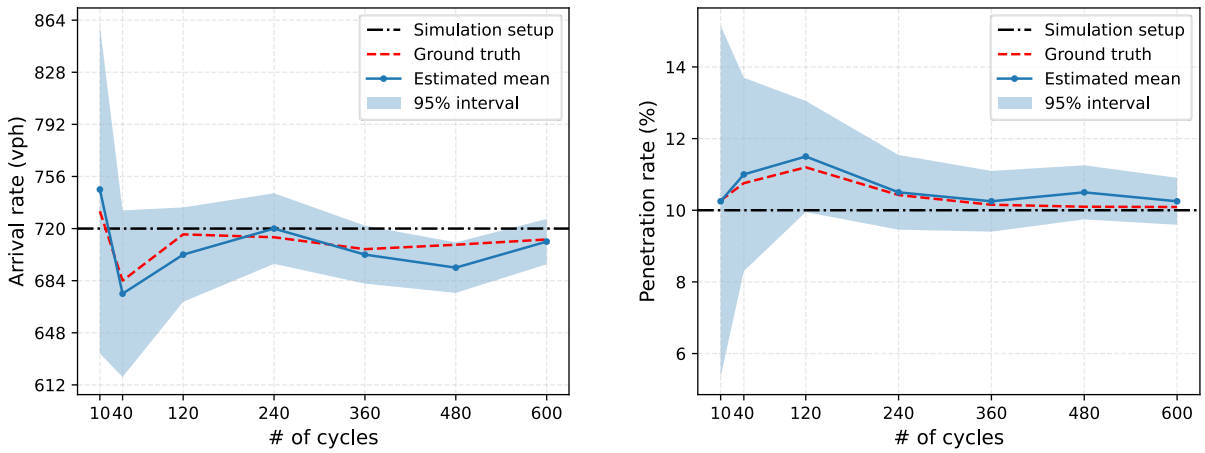


Figure 4.11: Parameter estimation with different data sizes.

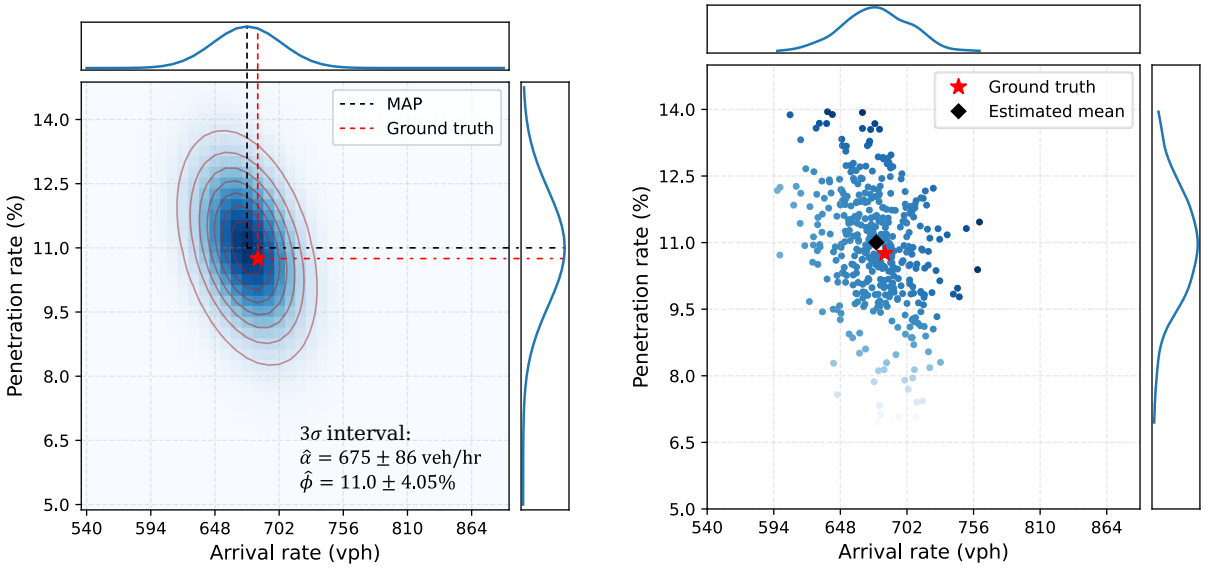
Figure 4.11 shows the estimated posterior distribution by using the same grid sampling methods but different numbers of cycles' data. As expected, when the number of cycles increases, the posterior distribution becomes more concentrated and gradually converges to the ground truth. Figure 4.12 shows the estimation results for each individual traffic parameter and how it changes with different sizes of input data. The horizontal axis is the duration of the input data while the vertical axis is the corresponding traffic parameter. The red dashed line is the ground truth; the blue



(a) Arrival rate estimation (b) Penetration rate estimation

Figure 4.12: Parameter estimation with uncertainty quantification.

line is the mean value of the posterior distribution while the blue area denotes the 95% confidence interval.



(a) Laplace's approximation (b) Importance sampling

Figure 4.13: Laplace's approximation and importance sampling.

However, as aforementioned, the grid sampling is low-efficient. The likelihood function needs to be calculated according to Algorithm 2 for each point in the mesh grid to obtain the entire heatmap. We propose another procedure based on Laplace's approximation and importance sampling in Section 4.4.4. Figure 4.13 shows the results of Laplace's approximation and the

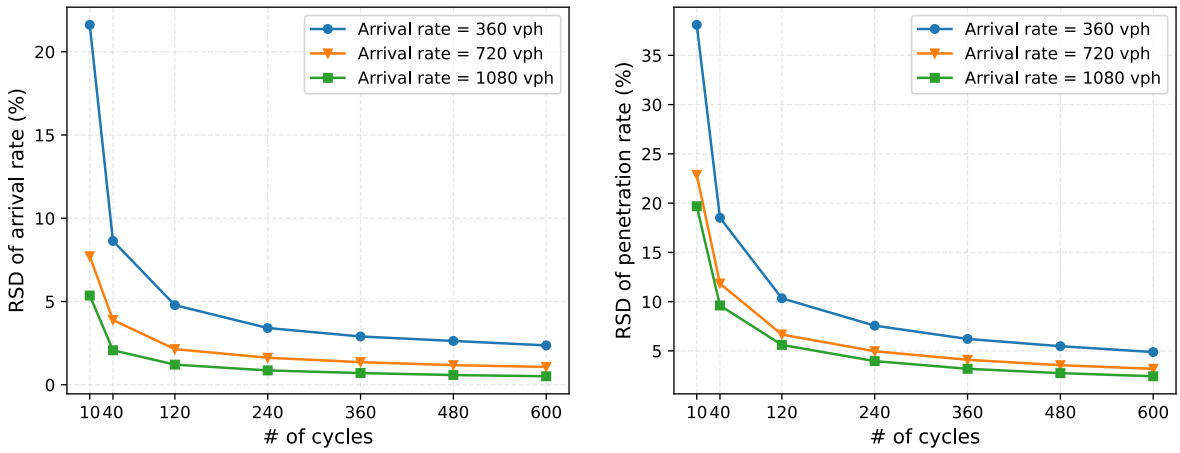
importance sampling by using the former one as the proposal distribution. Laplace's approximation finds the observed Fisher information, which is the second-order derivative at the peak value of the log-likelihood given by Figure 4.10 (a). A Gaussian distribution is then used to approximate the posterior distribution, of which the center is the peak value and the variance is the inverse of the observed Fisher information.

Laplace's approximation in Figure 4.13 (a) provides a Gaussian distribution to approximate the posterior in Figure 4.10 (b). Notice that performing Laplace's approximation is much more cost-efficient compared with grid sampling: the former only requires getting the peak value (by using optimization-based methods) and estimating the second-order derivative nearby while the latter needs to calculate the likelihood of each grid in the heatmap. Therefore, Laplace's approximation provides a much cheaper way to get both the estimated value and reasonable metrics (variance) to quantify its uncertainty.

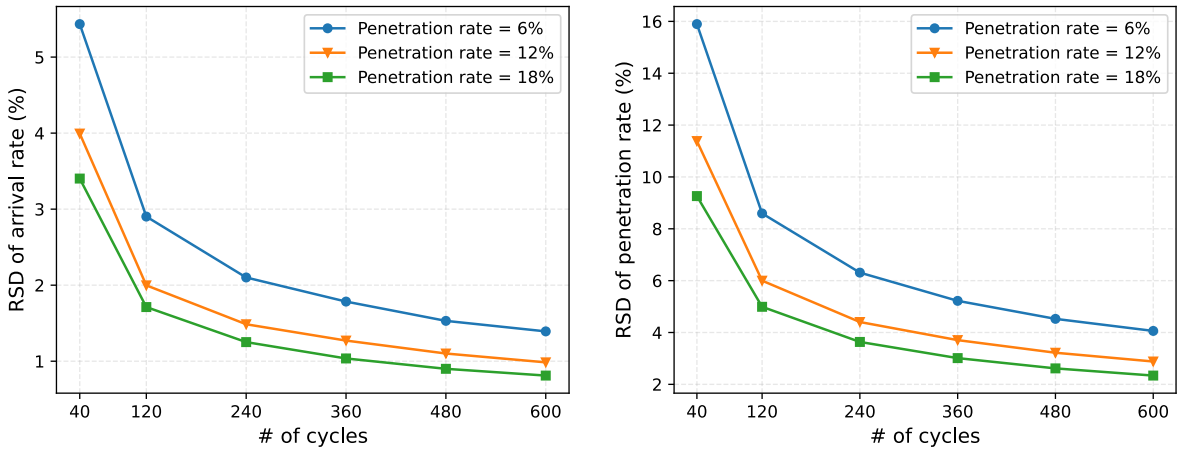
Given that Laplace's approximation provides a fairly good approximation of the posterior distribution, the importance sampling can be further utilized by using it as the proposal distribution to obtain a more accurate result. Figure 4.13 (b) shows the result of the importance sampling: each point is the sampled point and the transparency is the associated importance weight.

One of the major advantages of the Bayesian estimation method is that it can not only provide the estimated value but also the distribution as well as the associated uncertainty. Figure 4.14 shows the estimation uncertainty of the traffic parameters under different conditions. The relative standard derivation (RSD) is used to quantify the uncertainty of the estimated value, which is determined by the standard derivation divided by the mean value (unit: %). For each of the figures, the horizontal axis is the number of input cycles while the vertical axis is the RSD of the estimated parameters. Generally, the estimation uncertainty decreases with the increase of the input data. The left figures show how different parameters influence the arrival rate estimation while the right figures are about the penetration rate estimation.

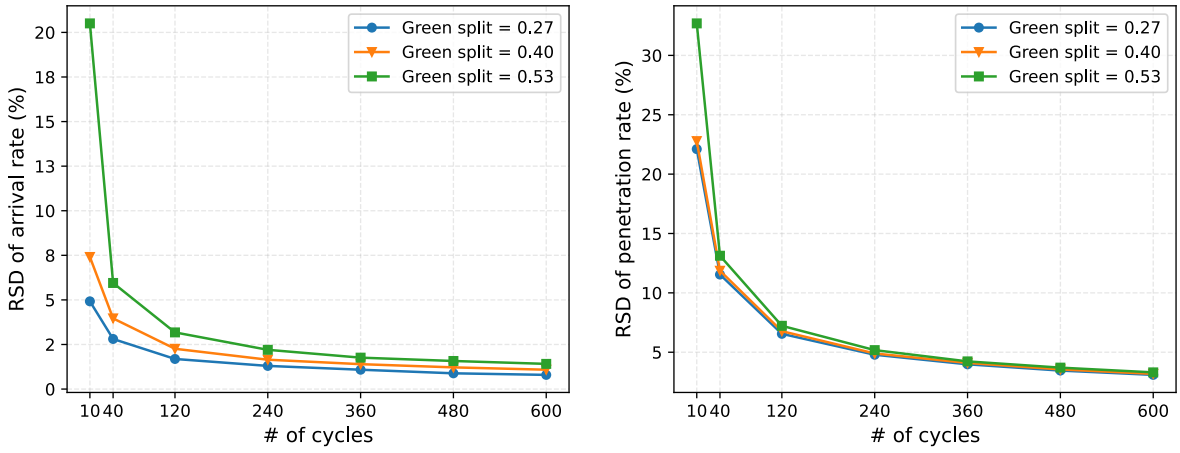
Figure 4.14 (a) shows the estimation uncertainty under different arrival rates (traffic volumes). For both the arrival rate and penetration rate, the estimation uncertainty quantified by RSD decreases with the increases in traffic volumes. When the traffic volume is larger, more trajectories can be observed, which leads to a more accurate estimation for both parameters. Figure 4.14 (b) shows the results under different penetration rates. As expected, when the penetration rate increases, more trajectories can be observed and the resulting posterior distributions are denser with smaller RSD. Figure 4.14 (c) shows the results under different green splits. When the green split is smaller, there are more vehicle stops and the estimated arrival rate distribution is denser. However, the penetration rate estimation seems not influenced a lot by the green split.



(a) Estimation uncertainty under different arrival rates (traffic volumes)

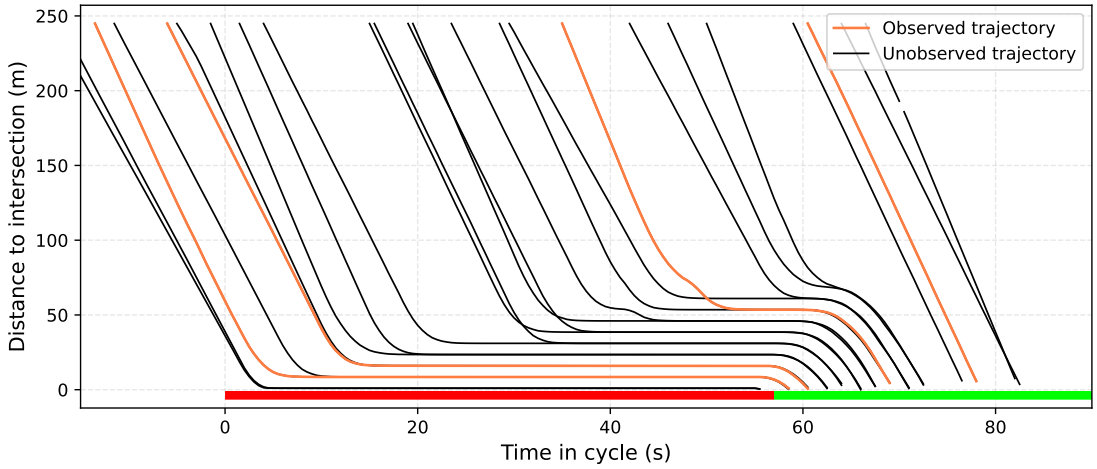


(b) Estimation uncertainty under different penetration rates

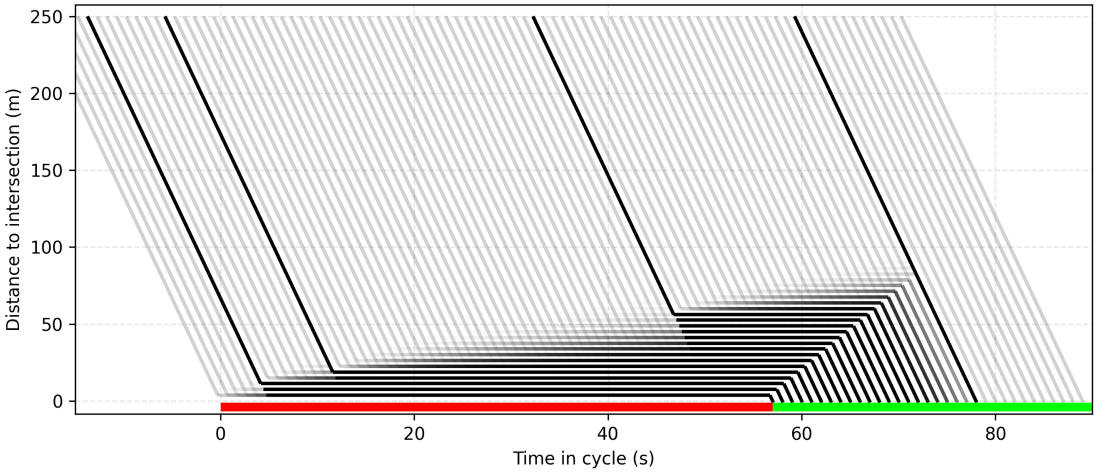


(c) Estimation uncertainty under different green splits

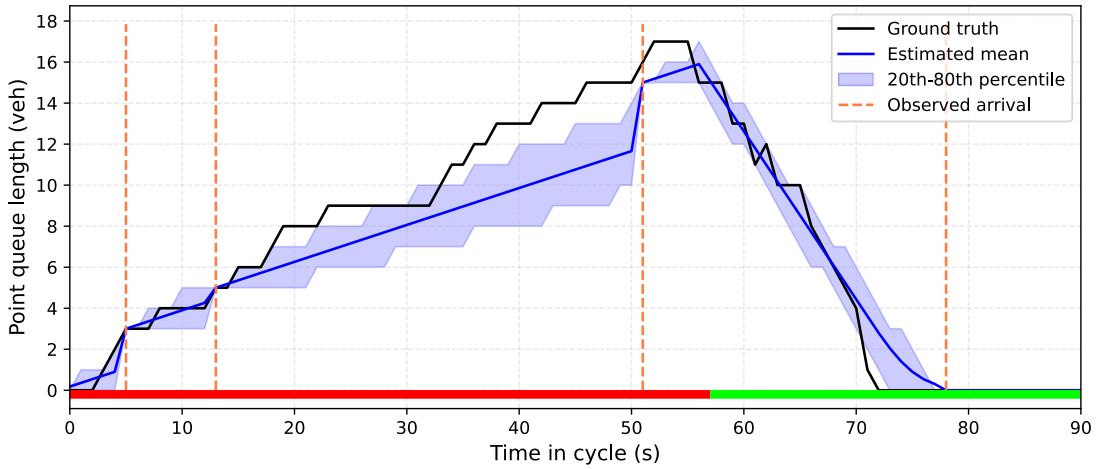
Figure 4.14: Uncertainty of parameter estimation (left: arrival rate estimation, right: penetration rate estimation.)



(a) Observed vehicle trajectories



(b) Estimated PTS diagram



(c) Real-time queue length estimation

Figure 4.15: Example of real-time queue length estimation.

4.5.3 Real-time traffic state estimation

This subsection introduces the results of the real-time traffic state (queue length under Newellian coordinates) estimation. It is assumed that the penetration rate and arrival rate are known or already accurately estimated. It is slightly different from the previously introduced method in Section 4.4.4, which finds:

$$p(\mathcal{X}(t)|\mathcal{O}(1:t)) = \sum_{\Theta} p(\mathcal{X}(t), \Theta|\mathcal{O}(1:t)) = \sum_{\Theta} p(\mathcal{X}(t)|\Theta, \mathcal{O}(1:t))p(\Theta), \tag{4.41}$$

in Step 4, where $p(\Theta)$ represents the distribution of the traffic parameters which is estimated in the previous subsection. Equation (4.41) requires performing the real-time estimation for every traffic parameter Θ, which is time-consuming and turns out to be unnecessary. Instead, we calculate $p(\mathcal{X}(t)|\mathcal{O}(1:t), \Theta)$, which means that we only calculate the real-time estimation for a single traffic parameter Θ.

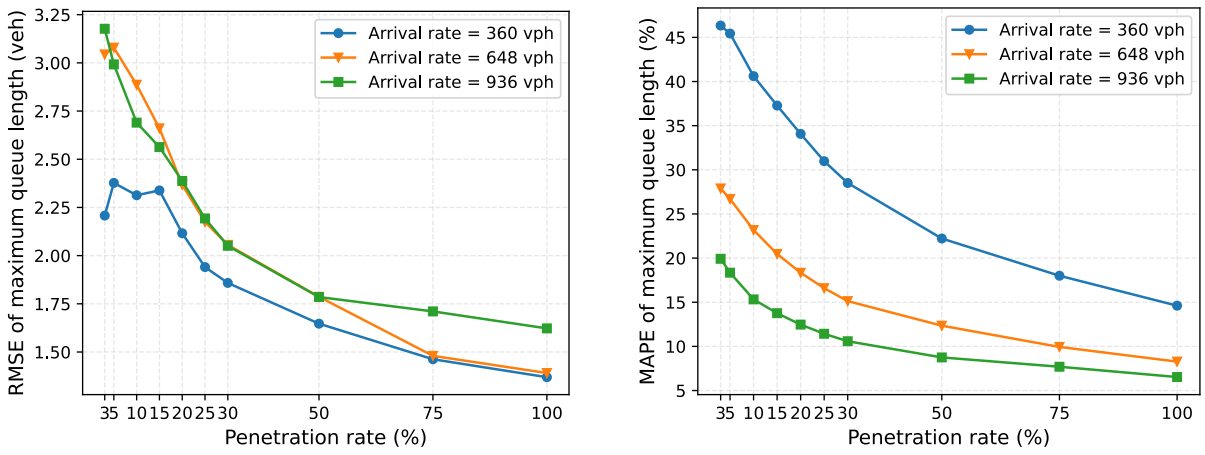
Figure 4.15 is an example of the real-time queue length estimation. Figure 4.15 (a) shows the time-space diagram including both observed and unobserved trajectories, denoted by orange and black lines, respectively. Figure 4.15 (b) is the estimated PTS diagram while Figure 4.15 (c) directly shows the estimated queue length profile. As shown in Figure 4.15 (c), whenever a new vehicle is observed, the queue length is updated. Since we assume that the observed queue length is accurate (Assumption 4.1), it is directly set as the observed queue length, and uncertainty is 0. There could be a discontinuous "jump" of the estimated queue length when a new observation is used to update it. This discontinuity is caused by the filtering algorithm, which only finds the posterior of the queue length given all previous observations, i.e., $p(X(t)|\mathcal{O}(1:t))$. This can be improved if a smoothing algorithm is applied instead.

Figure 4.16 shows the estimation error of the maximum queue in a cycle under different arrival rates and penetration rates. The maximum queue length in a cycle is used since it is the most useful and representative metric for real-time traffic signal control. Let $X_i^{\max}$ and $\hat{X}_i^{\max}$ denote the ground truth and estimated maximum queue of cycle i. Two different metrics are used to evaluate the maximum queue length estimation: root-mean-square error (RMSE) and mean absolute percentage error (MAPE), which are calculated according to:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(X_i^{\max} - \hat{X}_i^{\max}\right)^2}{N}} \tag{4.42}$$

and

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{X_i^{\max} - \hat{X}_i^{\max}}{X_i^{\max}}\right|. \tag{4.43}$$



(a) RMSE

(b) MAPE

Figure 4.16: Estimation error of maximum queue length under different penetration rates and arrival rates.

Each point in Figure 4.16 is generated by using an 8-hour simulation (320 cycles). Both RMSE and MAPE decrease with the increase in the penetration rate. However, they have different trends with regard to the arrival rate (traffic volumes): MAPE monotonically decreases with the increase in the arrival rate while the RMSE does not show the same trend. When the traffic volume becomes larger, more vehicle trajectories can be observed, which can improve the accuracy of the queue length estimation. This is why MAPE decreases with the increase in traffic volumes. On the other side, however, a larger traffic volume will lead to a larger variance in the maximum queue length, making it harder to be estimated. RMSE is influenced by both factors and hence does not show a clear trend with the change in traffic volumes.

Although both RMSE and MAPE have a decreasing trend in Figure 4.16, they do not decrease to 0 when the penetration rate is 100%. Ideally, the estimation error should be completely eliminated with a 100% penetration rate since all vehicles are observable in this case. This demonstrates the limitation of the proposed estimation methods based on the proposed stochastic traffic flow model: they are designed for low-penetration scenarios. Many of the assumptions introduced before work well under a low penetration rate but become less effective when the penetration rate gets higher. More discussion is available in Section 4.6.2, which also provides possible solutions to improve some of the assumptions.

Figure 4.17-4.18 show how the number of observed vehicle trajectories as well as the stop locations of the observed trajectories influence the estimation performance. Figure 4.17 is the RMSE with different numbers of observed trajectories in a cycle. The blue line shows the results by utilizing the accurate traffic parameters (prior) as the input while the orange line uses a biased prior with a 25% over-estimation of the arrival rate. As expected, with more observed trajectories,

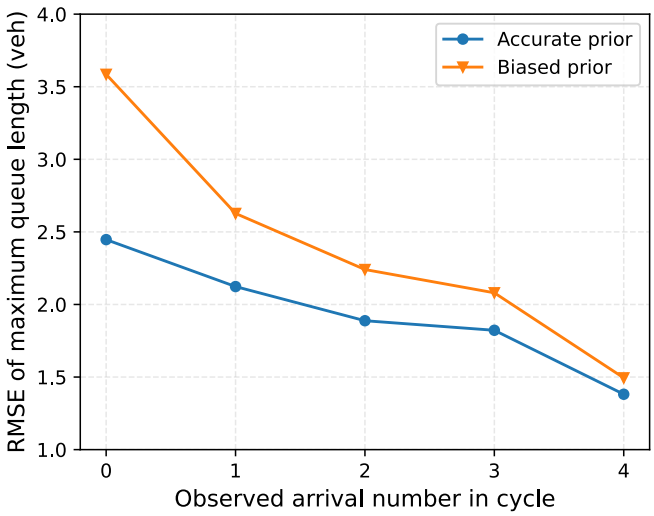


Figure 4.17: RMSE of maximum queue estimation with different numbers of observed trajectories.

RMSE decreases for both accurate and biased input prior. This decrease is more significant when the input prior is not accurate. Figure 4.18 shows how the stop locations of the observed trajectories influence the estimation performance. For all those cycles of which only one trajectory is observed, RMSE decreases with the increase of the observed queue length. This means that the observed vehicle trajectory with a large stop distance to the stop bar (a larger observed queue length) contains more useful information.

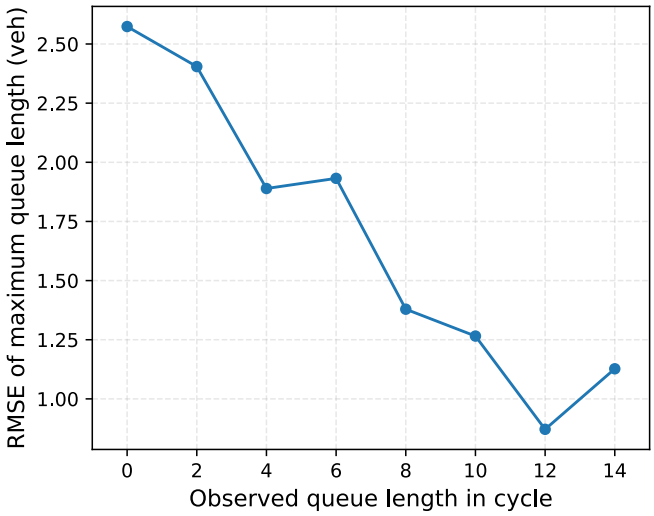


Figure 4.18: RMSE of the maximum queue length estimation with different observed queue lengths (for all those cycles with only one observed trajectory).

4.6 Summary and discussions

4.6.1 Summary

This chapter focuses on the estimation of both traffic state and parameter based on the previously introduced stochastic traffic flow model. Two different estimation methods are used. For the estimation and reconstruction of the traffic state of fixed-time traffic signals, we propose to utilize the MM estimator to match the average control delay between the model-estimated value and observation. This is a simple and efficient algorithm that can be easily applied to practice. We also use the real-world trajectory to validate the proposed method, which demonstrates that, even at a low penetration rate, the recurrent average traffic state can still be reconstructed by aggregating more historical data.

Other than the point estimation, the reliability of the estimation is also important. We apply formal Bayesian techniques to obtain the complete posterior distribution of the estimated values, which can be directly used to quantify the uncertainties of these values. Both traffic state and parameter can be estimated within the same recursive estimation program based on a well-formulated hidden Markov model. Simulated data is used to validate the proposed method. We also design simulation experiments to study how different factors influence the estimation uncertainties of different values.

4.6.2 Discussions

To make the main content in this chapter more logical and clean, we made several assumptions and simplifications to avoid those complicated and tedious considerations. Many of these assumptions and simplifications work well under a low penetration rate but become less valid when the penetration rate is high. Firstly, the stochastic traffic flow model and Newellian coordinates ignore the stochastic driving behaviors, which will play an important role under a high penetration rate. Secondly, Assumption 4.1 ignores the noise of the observed queue length, which will also significantly undermine the accuracy of the observation model for multiple-lane scenarios under a high penetration rate.

The rest of this subsection will briefly discuss how some of these issues could be modeled more accurately. At last, we will also introduce the hierarchical Bayesian model, which can be utilized to further facilitate estimation accuracy, especially in the real world with more uncertainty and randomness.

Multiple-lane scenario To simplify the estimation algorithm, it was assumed that all the queue lengths are the same for different lanes of the same movement. However, they are usually different

but similar. Additional assumptions about lane choice need to be made if one wants to get a more accurate estimation result. Two commonly used assumptions include: 1) drivers of the same movement will always choose the lane with a shorter queue; 2) drivers randomly choose the lane, which means that the queue lengths of different lanes are i.i.d.. After introducing either lane choice assumption, the observation model given by Equation (4.15) can be modified accordingly. The current observed queue length model is a deterministic model given by Equation (4.15) since we assume no error and all the queue lengths are homogeneous. It will become a stochastic model based on the newly introduced lane choice assumption. A simple change to the observation model given by Equation (4.15) will not change much for both the structure of the probabilistic model and the estimation algorithms. It will still be a hidden Markov model and the same recursive program can be used.

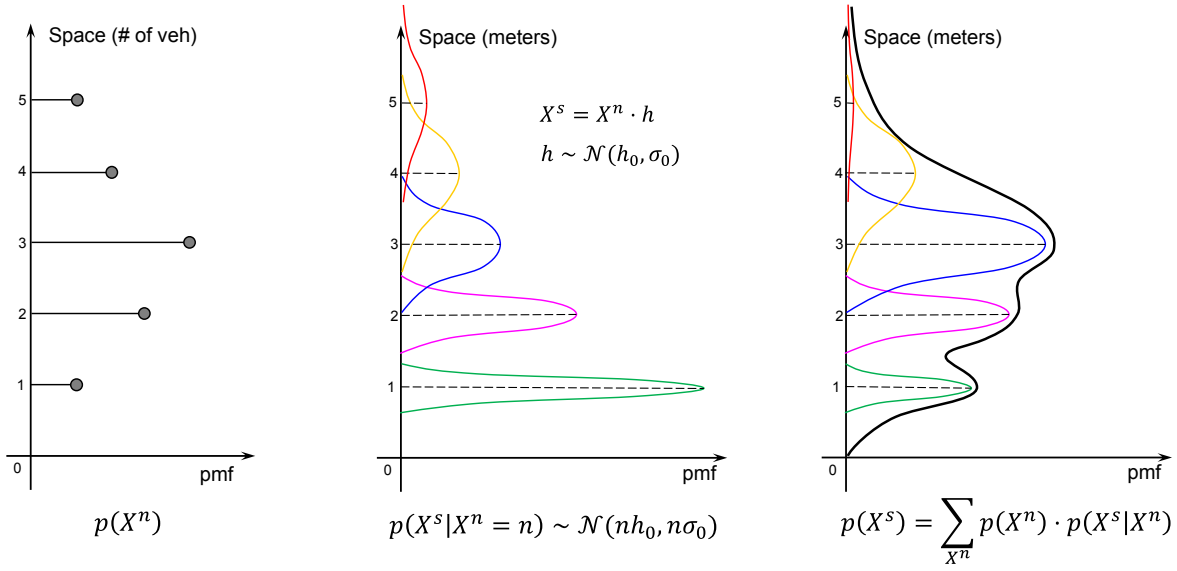
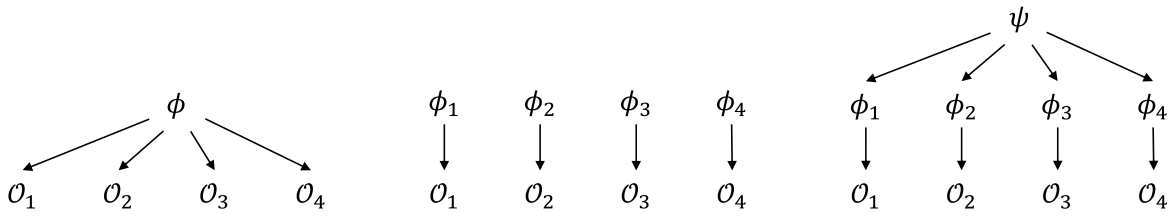


Figure 4.19: Stochastic jam space headway.

Stochastic jam space headway It was also assumed in Assumption 4.1 that the jam space headway is a constant. This is usually not a major concern but a more accurate model can certainly be used to consider the stochastic jam space headway. Figure 4.19 is an illustration of how it can be considered. Let X^n be the queue length in units of the number of vehicles while X^s is the queue length with actual observed distance in units of meters. The mapping between X^s and X^n is deterministic if a constant jam space headway is used. However, it will become a stochastic model if the jam space headway is stochastic. For example, one reasonable assumption is that it follows a Gaussian distribution with a certain mean and variance. This will also change the observed queue length model given by Equation (4.15), which can be reformulated according to the illustration in Figure 4.19.



(a) Joint (centric) model (b) Separate model (c) Hierarchical model

Figure 4.20: Different structures of probabilistic models.

Hierarchical Bayesian model The Bayesian estimation in this chapter only considers one single movement with a constant arrival rate and penetration rate. Both parameters are also assumed to be deterministic and all the uncertainty of the estimation comes from the incomplete observation. Real-world cases could be much more complicated: the arrival rate changes over time and the penetration rate might also be different in different locations. In this context, we will briefly mention how the hierarchical Bayesian model can be used to improve estimation accuracy. Figure 4.20 is an illustration of different structures of the probabilistic models. ϕ is the parameter to be estimated and $\mathcal{O}_i$ is the observation that can be used to infer ϕ. For example, ϕ and $\mathcal{O}_i$ can be regarded as the unknown penetration rate and observed vehicle trajectories, accordingly. The subscript i denotes the index of movement, which means that we are now dealing with the estimation for multiple movements. There are three different designs as illustrated by Figure 4.20. The joint model assumes a uniform penetration rate for different movements, which might ignore the possible difference at different locations. The separate model assumes each penetration rate to be different and independent from each other; this could lead to different penetration rates but we might not get sufficient data for certain movements. A better design might be the hierarchical model given by Figure 4.20 (c), instead of a joint or a separate model, penetration rates of different movements are assumed to follow the same distribution parameterized by a hyper-parameter ψ. In this way, we essentially transfer our knowledge of the penetration rate of different movements to that movements with few observations can have a more accurate estimation result. This is similar to the idea in Equation (4.6) but in a more formal method. Other than the penetration rate, we can apply the same technique to the arrival rate. For example, instead of assuming a constant arrival rate within the same TOD, we can assume it follows a Gaussian process.

CHAPTER 5

Fixed-Time Traffic Signal Optimization and Field Implementation

5.1 Introduction

5.1.1 Background and related works

Traffic signal control systems can be divided into 1) fixed-time control, 2) vehicle-actuated control, and 3) adaptive traffic signal control. Limited by the availability of loop detectors, there is still a large proportion of traffic signals that do not have effective monitoring ability and still use fixed-time traffic control. Traffic signal timing plans for these intersections only get re-timed every $3-5$ years. As a result, these intersections easily get outdated when the traffic demand and conditions change over time.

To fill the gap, this chapter will focus on the re-timing of these fixed-time traffic signals utilizing vehicle trajectory data. The calibrated traffic flow model in the previous section can be directly used for the optimization of the fixed-time traffic signal parameters, which is also the main advantage of the proposed method compared with most existing studies (Zheng et al., 2018; Liu and Zheng, 2019; Ma et al., 2020).

5.1.2 Overview of the chapter

In this chapter, we aim to develop effective and practical traffic re-timing strategies for fixed-time traffic signals based on the calibrated traffic flow model introduced in Chapter 3-4. Instead of a one-shot optimization, we propose an iterative diagnosis and optimization framework since the vehicle trajectory can be collected continuously over time. For each re-timing iteration, the traffic diagnosis module finds the optimality gap with respect to different traffic signal timing parameters, and the newly suggested signal timing parameters move towards the optimal direction

for a certain step size. As a result, the signal re-timing process can be regarded as a gradient descent optimization in the long run.

This chapter also introduces the field implementation in Birmingham, Michigan, including citywide offline monitoring, diagnosis, and optimization of all 34 signalized intersections using one month of trajectory vehicle data as the only input. Implementation of the new timing plans resulted in significant reductions in control delay and the number of stops at both isolated intersections and corridors. For isolated intersections, changes in green split allocations reduced these measures by up to 8% and 12%, respectively. A TOD split change reduced the number of stops by 21%. At corridors, offset adjustments reduced control delay and the number of stops along the entire corridor by up to 22% and 28% during the morning peak hours.

5.1.3 Contributions and organization of the chapter

The contributions of this chapter are summarized below:

1. We develop an iterative traffic signal diagnosis and optimization method for isolated intersections based on the calibrated traffic flow model.

2. We propose a pairwise traffic signal coordination diagnosis method to detect the suboptimal offsets of coordinated intersections. A coordinate-descent program is utilized to generate the new offsets.

3. A field implementation at the City of Birmingham demonstrated the effectiveness of the overall traffic signal optimization system.

The remainder of this chapter is organized as follows: Section 5.2 introduces the proposed diagnosis and optimization algorithms. Section 5.3 introduces case studies of traffic signal diagnosis using GM data including both a corridor and an isolated intersection. Section 5.4 shows the result of the field implementation. Section 5.5 is a summary of this chapter.

5.2 Diagnosis & optimization algorithms

5.2.1 Traffic signal timing parameters and optimization

Before going to the details of the traffic signal diagnosis and optimization, this subsection introduces the traffic signal timing parameters and the formulation of the traffic signal optimization. Figure 5.1 shows the main parameters of the fixed-time traffic signals, including the TOD plans,

cycle, splits, and offsets of each TOD. Let $\boldsymbol{s}$ represent the traffic signal parameters:

$$\boldsymbol{s} = \left\{ \boldsymbol{\tau} = [\tau^1, \tau^2, \ldots, \tau^{K-1}], \{ s^k = (C^k, \boldsymbol{g}^k, \boldsymbol{o}^k) \} \right\}, \tag{5.1}$$

where $\boldsymbol{\tau}$ is the TOD splits and τ^k is the boundary between TOD k and $k+1$, K is the number of TODs; s^k refers to the signal timing plan of TOD k, including the common cycle C^k, green splits for each movement $\boldsymbol{g}^k$, and offset for each intersection $\boldsymbol{o}^k$. Then the all-day performance index (PI) is composed of all TOD intervals:

$$I(\boldsymbol{s}) = \sum_{k=1}^{K} I^k(\boldsymbol{s}) = \sum_{k=1}^{K} \left[D^k(s^k) + \alpha L^k(s^k) \right] \tag{5.2}$$

where the PI of each TOD k is a weighted sum (with parameter α) of total estimated delay $D^k(\cdot)$ and total estimated number of stops $L^k(\cdot)$. Given the calibrated traffic demand, both the total estimated delay and stops are determined by the traffic signal parameters s^k. The traffic signal optimization problem can be formulated as:

$$\boldsymbol{s}^* = \arg\min_{\boldsymbol{s}} I(\boldsymbol{s}) \tag{5.3}$$

which finds the optimal traffic signal parameters that minimize the overall delay and stops.

5.2.2 General idea of traffic signal diagnosis and optimization

The proposed traffic signal diagnosis module finds optimality gaps with respect to different signal timing parameters as aforementioned. Since the calibrated traffic flow model explicitly takes traffic signal parameters as an input, it can be directly used to predict network performance under different signal parameters by assuming unchanged traffic demand. The optimality gap can then be easily identified through either gradient-based or line search methods.

Figure 5.2 shows the flowchart of traffic signal diagnosis. The output diagnostic results can be categorized into different specific issues such as green split imbalances, insufficient cycle length, etc. These diagnostic results are directly used for generating new signal timing plans which move a certain step size in the gradient direction. Instead of a one-shot optimization, we propose an iterative diagnosis and optimization framework: for each re-timing iteration which can be $2-3$ weeks depending on whenever sufficient data is collected, the traffic flow model is calibrated with newly collected data and used for the generation of the new signal timing plan. The overall continuous iterative traffic signal diagnosis and optimization framework can be regarded as a gradient descent solution algorithm of Equation (5.3).

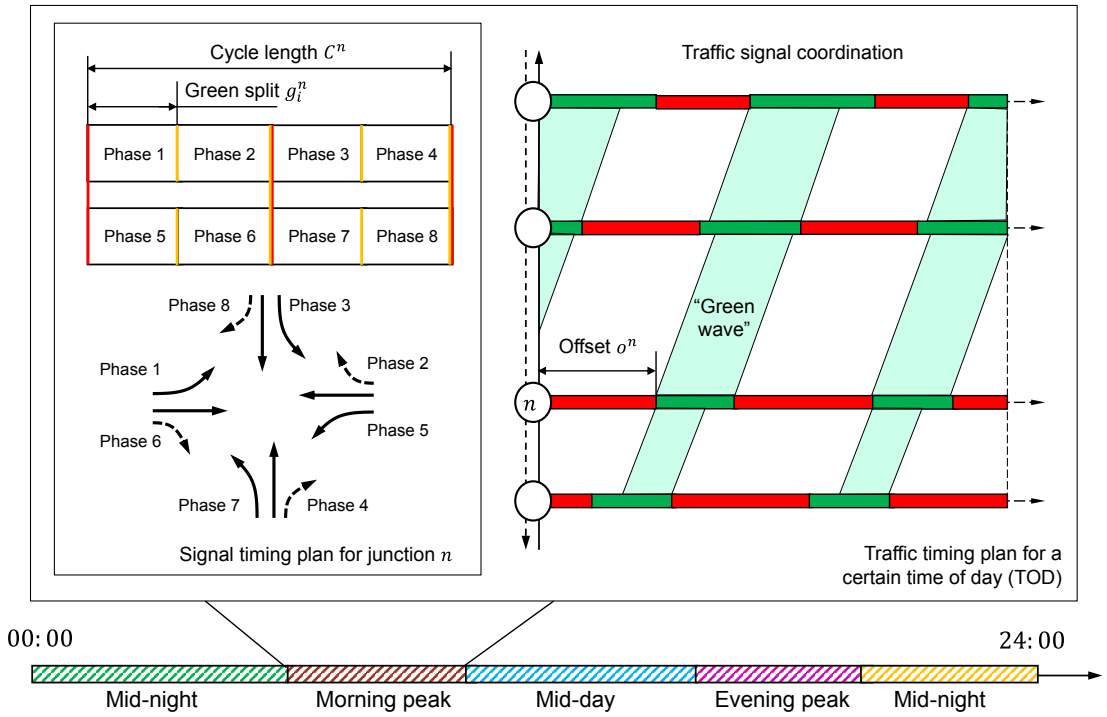


Figure 5.1: Traffic signal timing parameters (fixed-time).

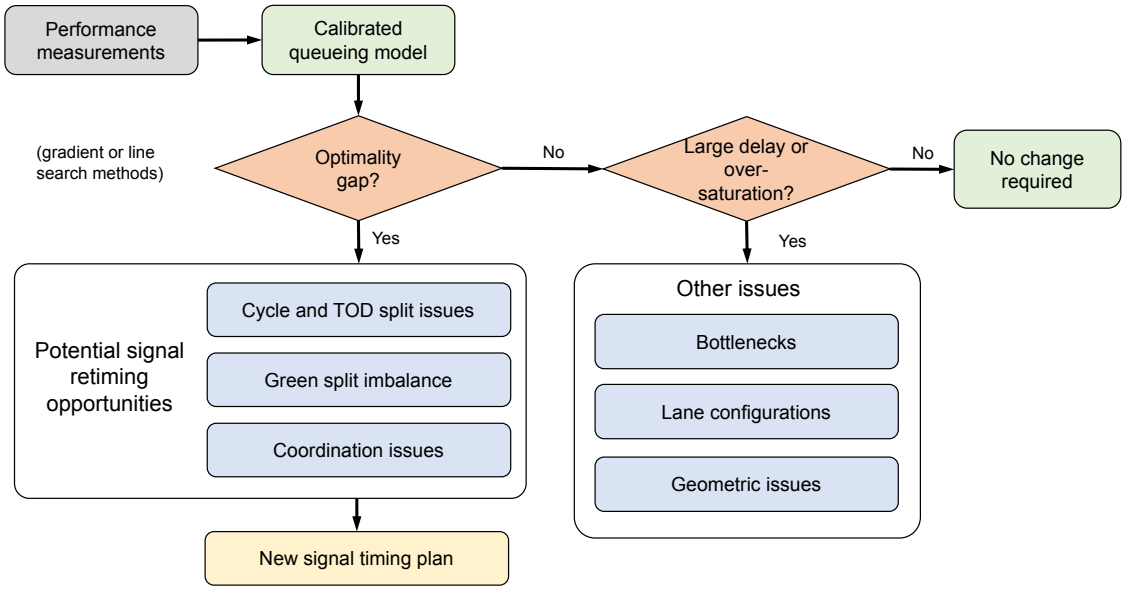


Figure 5.2: Traffic signal diagnosis.

5.2.3 Isolated intersections

For the signal timing parameters of isolated intersections such as cycle lengths and green splits, gradient-based methods are used since they usually do not require major changes. The sign of the gradient indicates the direction that could improve the system performance while the magnitude

of the gradient quantifies the potential benefits. The gradient of the performance of an isolated intersection can be written as:

$$\nabla I(\boldsymbol{s}) = \sum_{k=1}^{K-1} \frac{\partial I(\boldsymbol{s})}{\partial \tau^k} d\tau^k + \sum_{k=1}^{K} \left[\frac{\partial I^k(S^k)}{\partial \boldsymbol{g}^k} d\boldsymbol{g}^k + \frac{\partial I^k(S^k)}{\partial C^k} dC^k + \frac{\partial I^k(S^k)}{\partial \boldsymbol{o}^k} d\boldsymbol{o}^k \right] \tag{5.4}$$

Without having the closed analytical form, the partial derivative is estimated by numerical method. Taking the cycle length C^k of the kth TOD as an example, the partial derivative is given by:

$$\frac{I^k(\cdot)}{\partial C^k} = \frac{I^k(\cdot, C + \Delta C) - I^k(\cdot, C)}{\Delta C} \tag{5.5}$$

Equation (5.5) has a clear physical meaning which quantifies how the system PI changes by adding unit cycle length. Such gradient information can be used as an indication to the traffic signal diagnosis. The sign of the gradient indicates the direction that could improve the system performance while the magnitude of the gradient quantifies the potential benefits. As shown in Equation (5.4-5.5), the total derivative of the system performance can be decomposed into different terms with regard to different traffic signal parameters where their gradients are estimated separately. These gradients with clear physical meanings lead to different well-tagged diagnostic results including green time imbalances, suboptimal cycle lengths, and inaccurate TOD splits as shown in Figure 5.2.

Based on the diagnostic results given by these gradients, the traffic signal optimization is essentially a gradient-descent algorithm in the long run. For each iteration, roughly $2 - 3$ weeks, new data is collected, and new gradients are estimated from the calibrated traffic flow model. The new signal timing plan will be based on the original timing plan and moves along the derivative direction for a certain step size. Note that it is not necessary to update all the traffic signal parameters each time, only those with large gradients. This is a simple yet practical and effective algorithm, especially for the update of the green split, cycle length, and TOD splits. They do not require major changes in most cases. The traffic patterns might also change over time and hence it is probably better to make minor adjustments each iteration while keeping the overall update process continuously.

5.2.4 Coordinated intersections

We also propose a pair-wise coordination diagnosis method that efficiently detects better coordination opportunities for coordinated intersections. Figure 5.3 demonstrates some basic traffic coordination concepts including green band, offsets, and relative offsets. The main objective of traffic coordination is to optimize the offsets of each intersection such that vehicles stop less

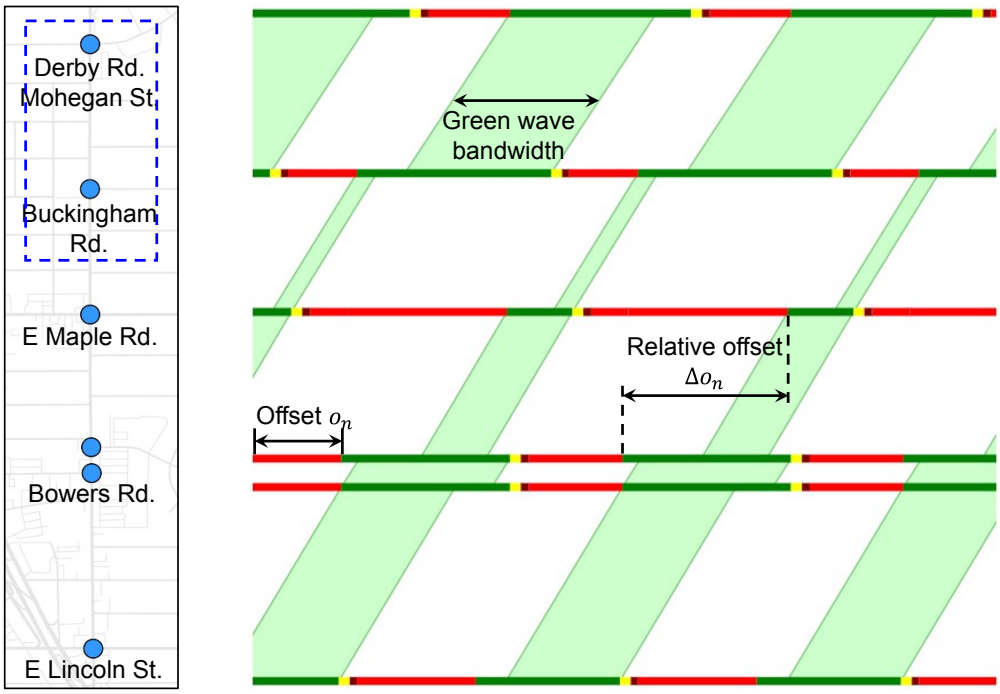


Figure 5.3: Traffic signal diagnosis.

when they traverse multiple intersections. For pair-wise coordination diagnosis, each pair of adjacent intersections are extracted as a sub-network. We then conduct a line search on the relative offset between them to identify potential opportunities for better coordination.

Offsets of intersections do not have much influence on the intersection capacity but could lead to better coordination among intersections. Therefore, unlike other traffic parameters which usually do not need to change much, major changes can be applied to the offsets. If we look into a specific TOD, the offset optimization problem can be formulated as:

$$\Delta \boldsymbol{o}^* = \underset{\Delta \boldsymbol{o}=[\Delta o_1,\ldots,\Delta o_{N-1}]}{\arg\min} I(\Delta o_1, \Delta o_2, \ldots, \Delta o_{N-1}) \tag{5.6}$$

where $I(\cdot)$ is the PI of the calibrated traffic flow model which is determined by the relative offset vector $\Delta \boldsymbol{o} = [\Delta o_1, \ldots, \Delta o_{N-1}]$. Δo_j is the relative offset between intersection j and $j+1$ as shown in Figure 5.3. Given the relative offsets $\Delta \boldsymbol{o}$, the offset o_j of intersection j can be determined as:

$$o_j = \left(\sum_{i=1}^{j-1} \Delta o_i \right) \mod T \tag{5.7}$$

where T is the common cycle length. The optimization problem given by Equation (5.6) can be solved by a coordinate-descent algorithm. For each iteration i, relative offsets are optimized

sequentially according to:

$$\Delta o_j^i = \arg\min_{\Delta o_j} I(\Delta o_1^i, \ldots, \Delta o_{j-1}^i, \Delta o_j, \Delta o_{j+1}^{i-1}, \ldots, \Delta o_{N-1}^{i-1}), \quad \forall j = \{1, 2, \ldots, N-1\} \tag{5.8}$$

which can be solved through a line search program. This iterative program will stop when the improvement in the last iteration is less than a certain threshold.

This proposed offset optimization program outperforms traditional green-band-based method (Little et al., 1981; Gartner et al., 1991; Yan et al., 2019) in two aspects: 1) it explicitly considers the vehicle distribution through the stochastic traffic flow model calibrated from vehicle trajectories; 2) it directly takes the total delay and number of stops as the objective function instead of the green band which does not always indicate good coordination.

5.3 Case studies of traffic signal diagnosis

5.3.1 Isolated intersection diagnosis

For isolated intersections, we can diagnose three specific issues in the existing signal timing plan: green split imbalances, suboptimal cycle lengths, and inefficient TOD splits. Each issue is analyzed individually while keeping the other parameters constant. For instance, the impacts of different green split allocations are explored while keeping the cycle length constant.

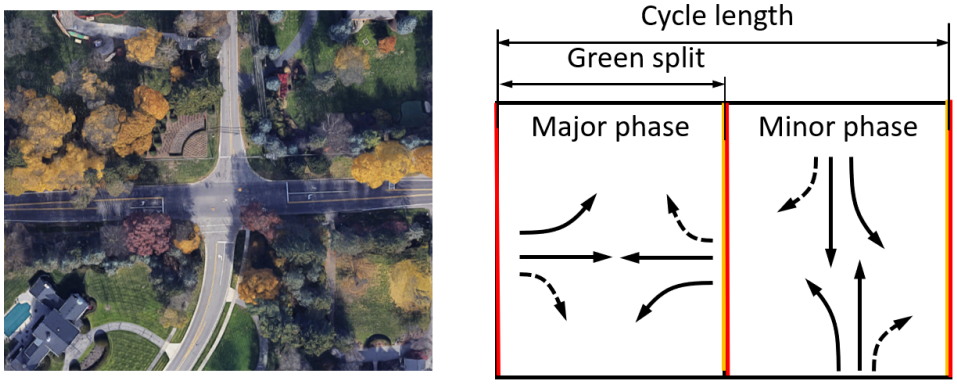
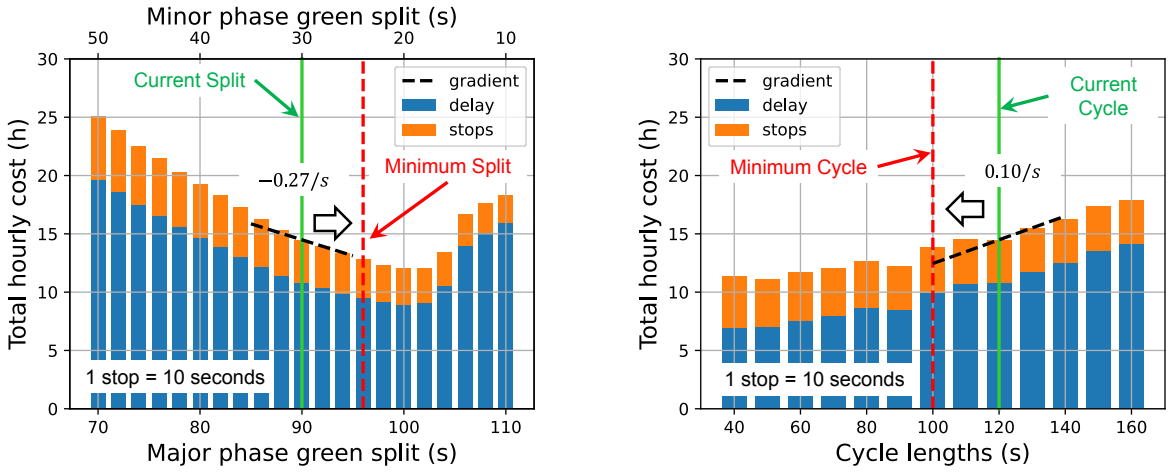


Figure 5.4: Case study of traffic signal diagnosis for isolated intersections.

We use a PI that calculates the sum of the total delay and the weighted number of stops (1 stop = 10 seconds), as this is what is commonly used in signal optimization software such as Synchro, TRANSYT, VISSIM, and Vistro. The PI is hence measured in "equivalent seconds" or "equivalent hours". With this PI, the partial derivative of a parameter is calculated by evaluating the change in the PI after an increase in the respective parameter.

An example intersection diagnostic is demonstrated in Figure 5.4. This isolated intersection utilizes a two-phase signal operation, where the major phase controls the major street, and the minor phase controls the minor street. The diagnosis uses aggregated data from 3 work weeks (M-F from March 7th - 25th, 2022).



(a) Green split diagnosis

(b) Cycle length diagnosis

Figure 5.5: Green split and cycle length diagnosis.

Figure 5.5 (a) is an illustration of the green split diagnosis of the morning peak hours. Orange and blue bars represent predicted number of stops and control delay under different green split. The green line indicates the current split, and the dashed red line represents the minimum minor phase green split (24 seconds) calculated from the pedestrian crossing time. Overall, the negative gradient indicates that increasing the major green split (see the black arrow in the figure) by one second will decrease the hourly PI cost by 0.27 equivalent hours. In this case, the current signal timing plan would be diagnosed with a green split imbalance and can be adjusted by increasing time for the major green split.

The cycle length during the morning peak hours can be evaluated in a similar way. Figure 5.5 (b) illustrates how the hourly PI cost changes at different cycle lengths. The red dashed line indicates the minimum allowable cycle length at the current green split ratio (from the minimum green split). The traffic flow model predicts that reducing the cycle length by one second will reduce the hourly PI costs by 0.10 equivalent hours. Therefore, we can identify the current signal timing plan as having an inefficient cycle length and can be adjusted in the retiming.

5.3.2 Pairwise coordination diagnosis

Figure 5.6 is an illustration of the pairwise traffic signal coordination diagnosis, which can be applied to detect better coordination opportunities. Each pair of adjacent intersections are

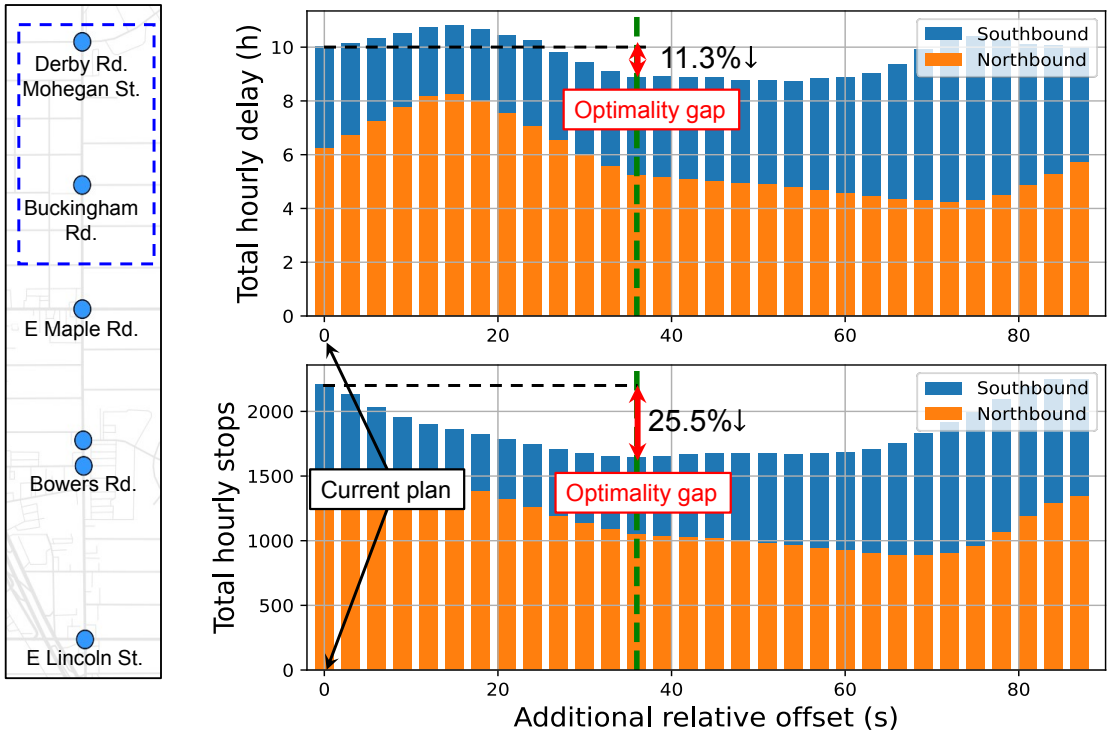


Figure 5.6: Pair-wise traffic signal coordination diagnosis.

extracted as a sub-network and an additional relative offset is added to compare with the existing performance. Taking the first two intersections as an example, Figure 5.6 shows the predicted total delay and the number of stops under different additional offsets. According to these curves, by adding an additional 36-second relative offset, the total delay and number of stops of these two intersections would decrease by about 11% and 25%, respectively.

5.4 Field implementation

5.4.1 Overview of the test bed

The proposed system was tested in the city of Birmingham, Michigan, United States as shown in Figure 5.7. Birmingham has a total of 34 signalized intersections including three main corridors and some other isolated intersections. More than three quarters of these intersections had not been retimed in more than 2 years. One-month offline data was used for performance evaluation, diagnosis, and optimization. Two isolated intersections were detected with cycle/split issues and two of the three corridors were identified with coordination improvement opportunities. New signal timing plans of these intersections were also generated and implemented in late March 2022. Three weeks' data both before and after the implementation was used to evaluate the new

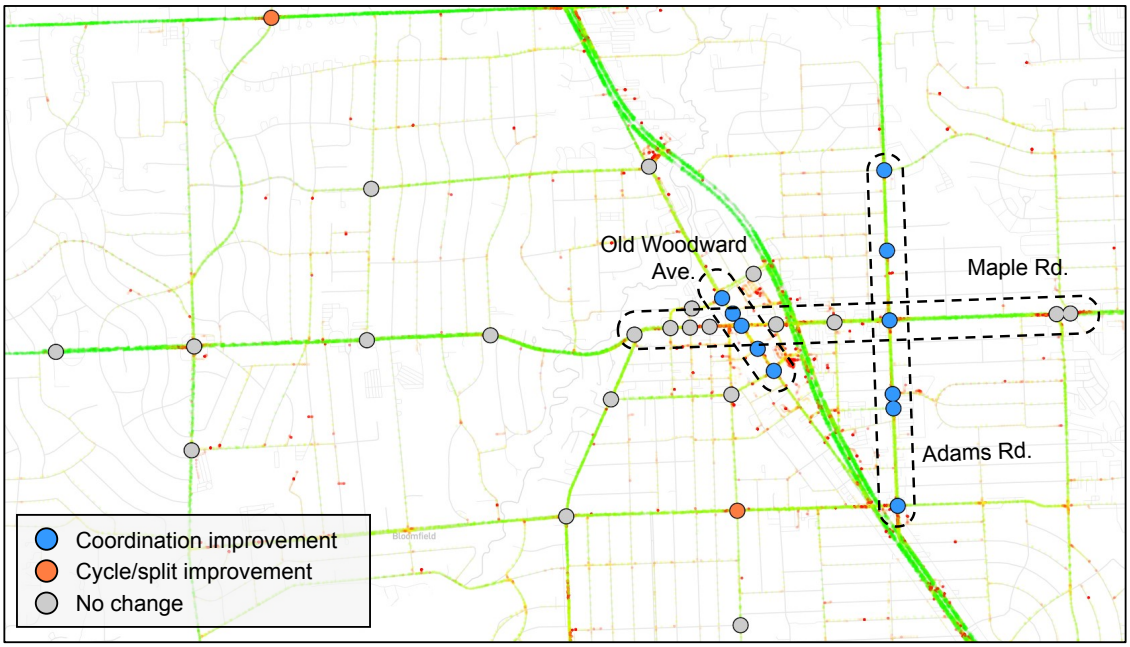


Figure 5.7: Signalized intersections in the City of Birmingham

signal timing plans.

5.4.2 Corridor coordination optimization

For both corridors including Adams Rd. and Old Woodward Ave., offsets of three different TOD intervals were changed including the morning peak hours (AM, 07:00-10:00), mid-day (MD, 10:00-15:00), and the evening peak hours (PM, 15:00-19:00). Table 5.1 and 5.2 shows the original offsets, new offsets, and the relative changes. New offsets of each TOD were generated from the offset diagnosis and optimization program introduced before.

Table 5.1: Adams Rd. offset adjustment

Side street	Time of day	Original offset	New offset	Change (s)
Buckingham Ave.	7:00 - 10:00 (AM)	40	20	-20
	10:00 - 15:00 (MD)	40	20	-20
	15:00 - 19:00 (PM)	40	30	-10
Bowers St.	7:00 - 10:00 (AM)	35	13	-22
	10:00 - 15:00 (MD)	35	13	-22
	15:00 - 19:00 (PM)	25	23	-2
Derby Rd.	7:00 - 10:00 (AM)	89	20	-69
	10:00 - 15:00 (MD)	89	21	-68
	15:00 - 15:15 (PMa)	89	31	-58
	15:15 - 15:40 (PMb)	89	31	-58

Table 5.2: Old Woodward Ave. offset adjustment

Side street	Time of day	Original offset	New offset	Change (s)
Merrill St.	7:00 - 10:00 (AM)	69	14	-55
	10:00 - 15:00 (MD)	52	22	-30
	15:00 - 19:00 (PM)	53	22	-31
Willits St.	7:00 - 10:00 (AM)	58	32	-26
	15:00 - 19:00 (PM)	77	39	-38
Brown St.	7:00 - 10:00 (AM)	39	22	-17
	10:00 - 15:00 (MD)	10	30	20
	15:00 - 19:00 (PM)	15	30	15
Oakland Ave.	7:00 - 10:00 (AM)	69	50	-19

Different metrics such as the average control delay and number of stops were used to evaluate the performance of these two corridors. The average control delay and average number of stops of the corridor are calculated by the total control delay and number of stops divided by the total number of "traversed trajectories"; which is counted by one vehicle passing one signalized intersection. The space-mean speed of the corridor is calculated as the total travel distance along the through movements divided by the total travel distance. All these three metrics are used to evaluate the travel efficiency of the corridor. The average number of stops is also closely related to energy consumption and emissions; since it would take more energy consumption as well as emissions for a vehicle to come to a complete stop and then accelerate back to normal speed. Since only the offsets were changed and the green splits stayed the same, side street traffic is not influenced and hence it is not included in the performance evaluation.

Table 5.3 shows the comparison of these three metrics before and after the offset optimization. For overall three optimized TODs from 07:00 to 19:00, the average control delay of Adams Rd. was decreased by around 12% while the average number of stops was decreased by over 18%. All three TODs performed better than before for both the average control delay and average number of stops. Less improvements were observed in the Old Woodward Ave. through all three TODs; however, certain TODs have much better performance: the average delay was decreased by over 15% during the morning peak hours (AM) while the average number of stops was decreased by over 10% during the evening peak hours. Some TOD intervals such as the mid-day period of the Old Woodward Ave. did not improve much since the original offsets worked well and there was not a large optimality gap.

Figure 5.8 shows more details on how the new offsets led to better traffic signal coordination along the corridors. Figure 5.8 (a-b) shows the aggregated time-space diagram of the Adams Rd. before and after the offset optimization. All the figures are generated using three consecutive weeks' data collected at the mid-day (10:00-17:00) during the weekdays. As shown in the figure,

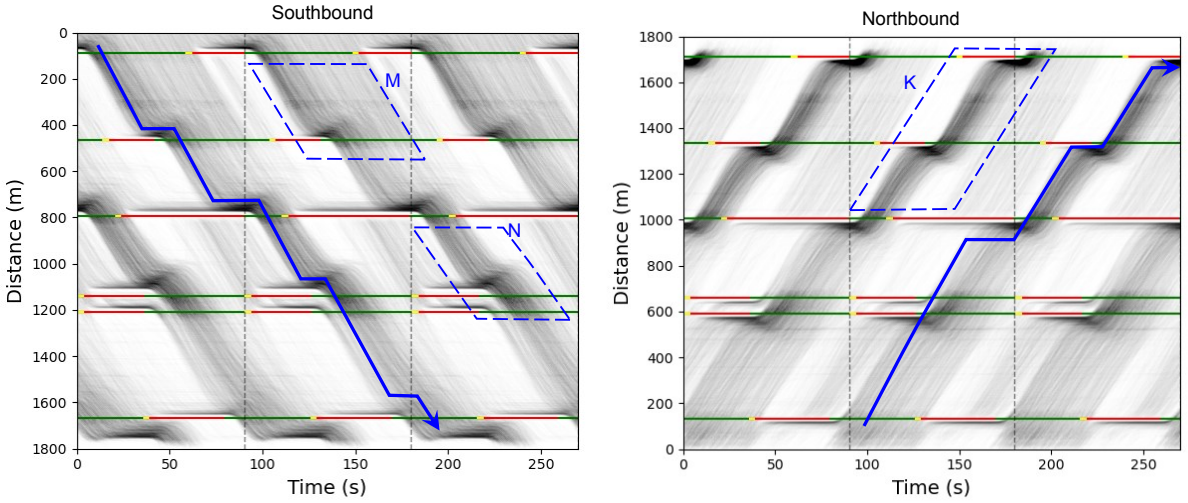
Table 5.3: Corridors performance table

Measurements		Adams Rd. AM	Adams Rd. MD	Adams Rd. PM	Adams Rd. All	Old Woodward Ave. AM	Old Woodward Ave. MD	Old Woodward Ave. PM	Old Woodward Ave. All
Average control delay (second)	Before	13.92	11.88	16.03	13.89	19.03	17.72	18.58	18.29
	After	10.85	10.55	14.57	12.19	16.05	17.88	18.11	17.60
	Change	-22.05%	-11.27%	-9.09%	-12.23%	-15.66%	0.91%	-2.54%	-3.78%
Average number of stops	Before	0.46	0.41	0.45	0.44	0.44	0.48	0.53	0.49
	After	0.33	0.33	0.41	0.36	0.42	0.44	0.45	0.44
	Change	-28.69%	-21.13%	-10.55%	-18.51%	-6.09%	-8.50%	-14.63%	-10.77%
Space-mean speed (mph)	Before	36.42	38.94	34.30	36.51	17.97	17.76	17.34	17.64
	After	41.68	42.02	36.14	39.43	19.48	17.62	17.34	17.85
	Change	14.44%	7.92%	5.34%	7.98%	8.43%	-0.81%	-0.04%	1.19%

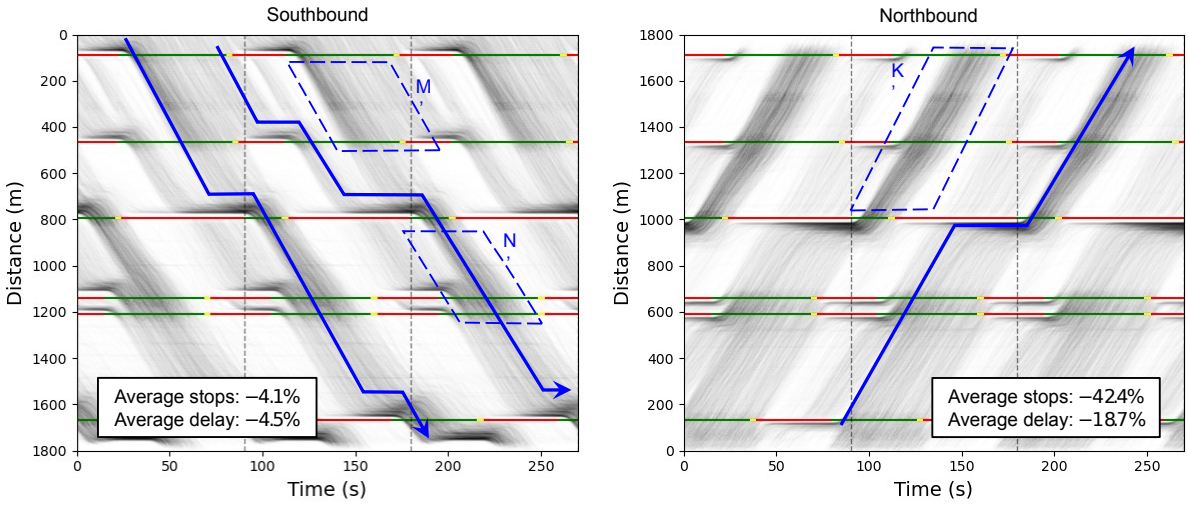
Table 5.4: Intersections performance table

Intersection	Analysis Period	Avg Delay (Before)	Avg Delay (After)	Delay % Change	Avg Stops (Before)	Avg Stops (After)	Stops % Change	PI % Change
Quarton Rd. & Cranbrook Rd.	06:00-09:00 (AM)	13.19	12.13	-8.03%	0.39	0.35	-9.51%	-9.13%
	14:00-15:00 (PM*)	12.65	12.24	-3.17%	0.45	0.36	-21.45%	-8.69%
	19:00-24:00 (EVE)	8.80	8.51	-3.29%	0.32	0.31	-3.09%	-3.34%
Lincoln Rd. & Pierce St.	10:00-14:00 (MD)	11.56	11.54	-0.18%	0.41	0.40	-4.46%	-1.32%
	14:00-15:00 (PM*)	12.43	11.48	-7.65%	0.46	0.41	-12.44%	-9.83%
	15:00-19:00 (PM)	12.71	12.57	-1.15%	0.45	0.45	-0.33%	-0.77%

the average delay and number of stops of the northbound through traffic were decreased by around 20% and 40%; the southbound also outperformed the previous with a slight decrease of 4% for both the average delay and number of stops.



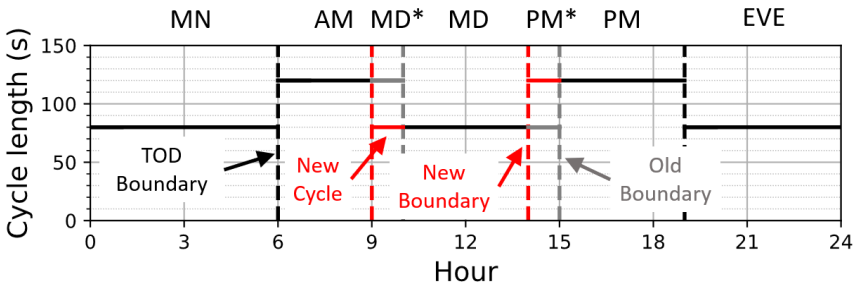
(a) Before optimization



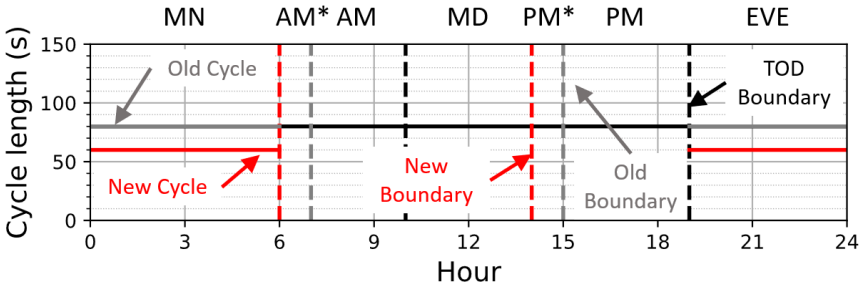
(b) After optimization

Figure 5.8: Aggregated time-space diagram before and after offset optimization.

Rectangular areas M, N, K in Figure 5.8 (a) and the associated areas M', N', K' in Figure 5.8 (b) illustrate where the coordination became better. Before the offset optimization, trajectories that departed from the upstream queue in rectangular areas M, N, and K arrived at downstream intersections during the red time and most of them stopped at least once before passing the downstream intersections. On the contrary, most of these trajectories from the upstream queue directly passed the downstream intersections without any stops as shown in M', N', and K'. By explicitly considering the trajectory arrival and departure distributions within each cycle, the



(a) Quarton Rd. & Cranbrook Rd.



(b) Lincoln Rd. & Pierce St.

Figure 5.9: TOD change of isolated intersections.

proposed offset optimization program will assign more green bands to the green time with more trajectories passing by. As shown by the rectangular area W in Figure 5.8 (b), although there was also a clear wide green band from the upstream to the downstream before the offset optimization; few trajectories traveled within the green band. This coordination failure can be easily identified by the proposed method.

5.4.3 Isolated intersections

New signal timing plans were implemented at two intersections. The TOD boundary changes and their respective cycle lengths are shown in Figure 5.9. All the parameter changes are reported in Table 5.5 and Table 5.6. The results for selected analysis periods are reported in Table 5.4. At Quarton Rd. & Cranbrook Rd, a 4 second increase in the major split resulted in a 9.13% decrease in the PI during the AM TOD. The intersection also benefited from changing the PM TOD start time from 3:00 PM to 2:00 PM (results shown by the PM* analysis period). The increased cycle length during this hour resulted in a 21.45% reduction in the number of stops. The EVE TOD also experienced reductions in delay and the number of stops.

Lincoln Rd. & Pierce St. also experienced improvements in the PI for certain analysis periods. The boundary start change for the PM TOD resulted in a 9.83% reduction in the PI. However, a close look at Table 1a shows that the only parameters that changed during this time period were the green splits. The large improvement for this specific hour may indicate that new TODs may

need to be formed as this hour improved much more compared to the MD and PM TODs that experienced the same change.

Table 5.5: Quarton Rd. & Cranbrook Rd. Parameter Changes

TOD	Original Cycle	New Cycle	Original Major Split	New Major Split	Original Minor Cycle	New Minor Split
MN	80	80	54	56	26	24
AM	120	120	90	94	30	26
MD	80	80	54	56	26	24
PM	120	120	90	94	30	26
EVE	80	80	54	56	26	24

Table 5.6: Lincoln Rd. & Pierce St. Parameter Changes

TOD	Original Cycle	New Cycle	Original Major Split	New Major Split	Original Minor Cycle	New Minor Split
MN	80	60	55	38	25	22
AM	80	80	55	55	25	25
MD	80	80	55	58	25	22
PM	80	80	55	58	25	22
EVE	80	60	55	58	25	22

5.5 Summary and discussions

5.5.1 Summary

This chapter introduces the iterative traffic signal diagnosis and optimization framework that is used for the traffic signal re-timing of fixed-time traffic signals. We also demonstrate the effectiveness of the whole signal re-timing system through a field test. The field test was conducted in Birmingham, Michigan included monitoring, diagnoses, and optimization of 34 coordinated and isolated signalized intersections. The system was able to diagnose specific congestion of causes at these intersections such as green split imbalances, offset issues, inefficient cycle lengths, and suboptimal TOD boundaries. The new signal timing plans are based on the existing signal timing plan while moving a certain step toward the direction guided by the diagnostic results. These new plans resulted in decreases in both the delay and number of stops by up to 20% and 30%, respectively. The field test shows the potential of the proposed system to improve signal timing plans from only vehicle trajectory data.

5.5.2 Discussions

Here we briefly discuss some issues that need to be further improved or explored in the future for the re-timing of fixed-time traffic signals.

Practical considerations and corner cases There are many detailed practical considerations and corner cases which are not fully covered by the proposed methods. For example, pick-up and drop-off zones, bus stations, street parking, and side street traffic would require further consideration for the traffic signal re-timing. Besides, some intersections that have weird geometry also need a customized design of the signal timing plan.

Traffic signal optimization considering vehicle re-routing The current method assumes the traffic demand does not change with a different input traffic signal timing. This is not true in many cases since drivers might adjust their routes in reaction to the change in traffic signal timing. Although this is well aware by researchers, few of them consider it for traffic signal optimization, which will lead to a bi-level formulation that is hard to solve. The influence of vehicle re-routing is dependent on the network topology as well as the pattern of the OD demand. For example, it can be ignored if we only look into one corridor but might play an important role in a more complicated network with many alternative routes for a single OD. Considering route choice for traffic signal optimization is not an easy problem. It not only needs an efficient algorithm to solve the bi-level formulation but also requires a new network model with OD demand instead of simple link flow.

Geometry optimization Geometry design of an intersection is also important in intersection management, which includes lane assignment, storage lane design, etc. This is another interesting problem that can be investigated using vehicle trajectory data.

CHAPTER 6

Real-Time Traffic Signal Control

6.1 Introduction

6.1.1 Background and related works

The previous chapter focuses on the re-timing of fixed-time traffic signals. However, the fixed-time control cannot react to time-varying traffic conditions. By taking the real-time traffic state as the input, a real-time traffic signal control could outperform a fixed signal timing plan by dynamically adapting to new traffic conditions. Therefore, this chapter focuses on the development of a real-time signal controller with vehicle trajectory data.

Many existing works have explored traffic signal control with vehicle trajectory data. Li and Ban (2018) utilized the trajectory of each vehicle to minimize both travel time and energy consumption, however, a 100% penetration rate is assumed which is not available currently. Feng et al. (2015) developed an adaptive traffic signal control system with trajectory data, the traffic state is estimated by identifying the boundary between the free-flow region, slow-down region, and queueing region. A large penetration rate of no less than 25% is also required. Although different methods have been proposed for traffic signal control with vehicle trajectory data (Lee et al., 2013; Liang et al., 2020; Yao et al., 2020), the real bottleneck under the current market penetration rate ($\leq 10\%$) is to get an accurate estimation of the overall traffic state with limited observations. Most existing works are limited by the lack of a suitable stochastic traffic flow model and a reliable real-time traffic state estimation method.

6.1.2 Overview of the chapter

Leveraging the real-time traffic state estimation method proposed in Chapter 4, this chapter aims at developing a real-time traffic signal controller with vehicle trajectory data. Following the logic of the vehicle-actuated control with loop detectors, we propose a simple rule-based Queue Clearance Control (QCC). By taking the real-time estimated queue length in Chapter 4 as the input, QCC

terminates the currently active phase and switches to the following phase whenever the queue length is cleared with a certain confidence level.

A simulation with an isolated intersection is built to test the proposed QCC. Both the fixed-time control and vehicle-actuated control are used to compare with the proposed controller. We also study how the different parameters and lag time will influence the controller's performance.

6.1.3 Contributions and organization of the chapter

The contributions of this chapter are summarized below:

1. A rule-based QCC is proposed to utilize the vehicle trajectory data for real-time traffic signal control.

2. A simulation environment with an isolated intersection is used to demonstrate the effectiveness of the proposed controller. Different insights are provided in terms of the parameter selection and how the lag time could influence the controller's performance.

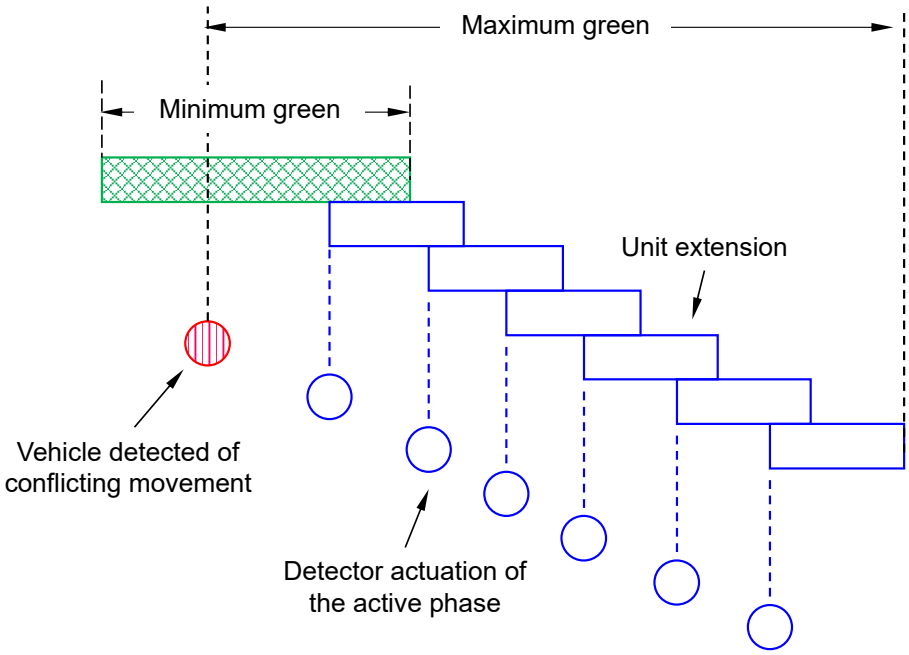
This chapter is organized as follows: Section 6.2 introduces the rule-based QCC and Section 6.3 shows the simulation results. Section 6.4 includes both a summary and some discussions of this chapter.

6.2 Rule-based queue clearance control (QCC)

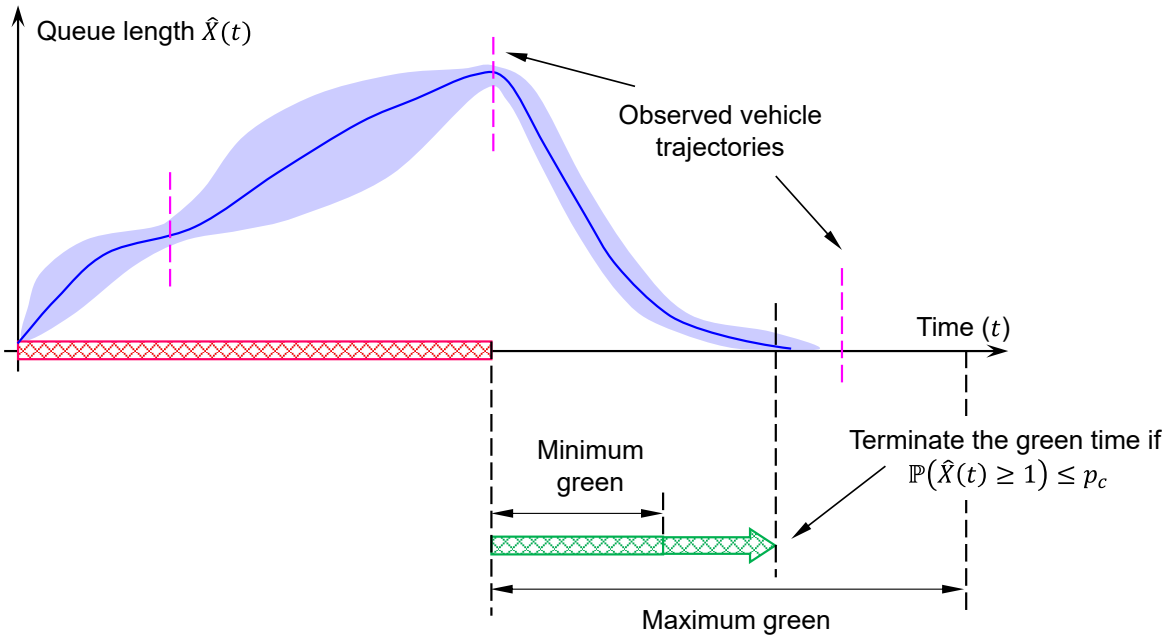
6.2.1 Control logic

This section will focus on the design of a rule-based real-time traffic signal controller with vehicle trajectory data. Without solving any complicated optimization program, rule-based control methods are usually more robust and easier to be implemented in the real world. If designed properly, a rule-based controller might have comparable or even better performance with optimal control methods.

For the current practice, vehicle-actuated control is one of the most commonly used real-time control methods with installed loop detectors. Figure 6.1 (a) is an illustration of the vehicle-actuated control (Urbanik et al., 2015). The key of the actuated control is the mechanism to terminate the currently active phase, including gap out and max out. The gap out means that the current phase will be terminated if the headway between two vehicles is larger than a certain threshold called minimum gap. The max out refers to the termination of a phase whenever it reaches the maximum green time (or maximum extension). With these two phase-termination



(a) Vehicle-actuated control



(b) Queue clearance control (QCC)

Figure 6.1: Vehicle-actuated control and the proposed QCC.

mechanisms, the vehicle-actuated control operates in the currently active phase as long as vehicles keep going through until it reaches the maximum green time.

Inspired by the vehicle-actuated control, we design a rule-based controller called QCC illustrated by Figure 6.1 (b). Chapter 4 has introduced the real-time estimation method which

can be used to estimate the real-time queue length with observed vehicle trajectories. The proposed QCC directly uses this estimated queue length as the input. As shown in Figure 6.1 (b), the estimated queue length distribution will be more dense whenever a vehicle trajectory is observed, denoted by the pink dashed line. Like the vehicle-actuated control, QCC also has the minimum/maximum green time and very similar mechanisms to determine the phase switching. For the traffic signal control system with sparsely observed vehicle trajectory data, the headway between vehicles cannot be directly measured like detector data but the estimated queue length distribution is available. Therefore, instead of using the gap out like vehicle-acuated control, QCC terminates the current phase whenever the queue is cleared. Since the queue length estimation result also has the distribution information, the following condition is used to determine whether the queue is cleared:

$$\mathbb{P}(\hat{X}_i(t) \geq 1) \leq p_c \tag{6.1}$$

where p_c is the pre-determined confidence level. For example, a p_c of 85% means that we choose to terminate the current phase if we are 85% sure that the queue of this phase has been fully discharged.

The max out remains the same for QCC. As a result, QCC has a similar control logic with the vehicle-actuated control. It has three parameters for each phase: minimum green, maximum green, and the confidence level p_c that the queue is fully discharged.

Although QCC has a similar intuition with the vehicle-actuated control, they have several differences. The major difference is the "gap out" mechanism. With installed detectors at the stop bar that can detect every vehicle passing through, the vehicle-actuated control determines whether to maintain the current phase through the time headway between two adjacent vehicles. On the contrary, the proposed QCC determines the phase time according to the queue length distribution, which can be estimated by combining the prior arrival rate and sparsely observed vehicle trajectories (Chapter 4). Theoretically, the vehicle-actuated control will lead to a larger green time for each phase compared with QCC, since it will not terminate the active phase immediately when the queue is cleared but until the time headway is less than the minimum gap.

6.2.2 QCC with lag time

In practice, a timely real-time estimation of the queue length is not always available due to the existence of lag time. Figure 6.2 shows how the real-time queue length distribution can be estimated with a lag time t_l. Assuming the current time is t, the lag time t_l is the temporal difference between the current time t and the time when the most recent estimation is available. This latency could come from a variety of origins including the trajectory data collection delay, data processing time, and the time cost by the estimation algorithms, etc. The trajectory data collection alone

might be up to minutes due to the communication delay. Not all connected vehicle trajectories are directly collected by RSUs, some data might go through a long routing process before it is readily used.

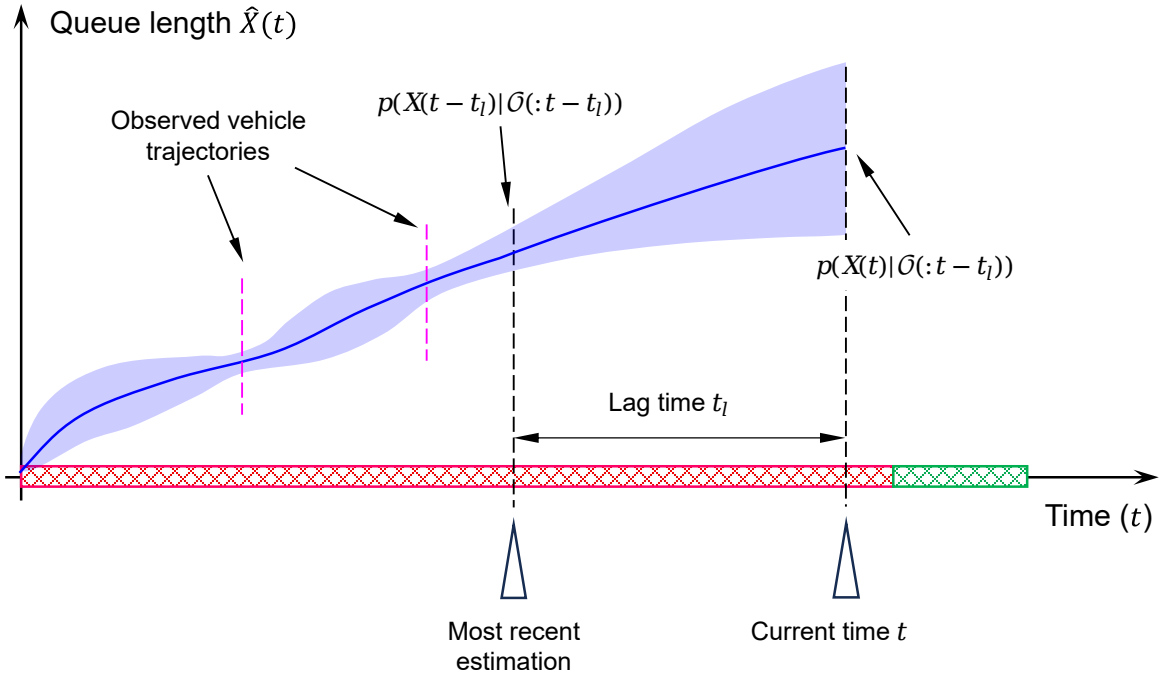


Figure 6.2: Estimated queue length distribution with the lag time.

Due to the latency t_l, at each time t, only the observation earlier than $t - t_l$ can be utilized, which is denoted by $\mathcal{O}(: t - t_l)$. As shown in Figure 6.2, the most recent estimation $\hat{X}(t - t_l)$ is given by the posterior distribution $p(\hat{X}(t - t_l)|\mathcal{O}(: t - t_l))$. To get queue length $X(t)$ at time t, the best approach is to perform the prediction with the traffic flow model based on the most estimated queue length $\hat{X}(t - t_l)$. In this way, the estimated queue length $\hat{X}(t)$ is determined by the posterior $p(X(t)|\mathcal{O}(: t - t_l))$, of which only the observation earlier than $t - t_l$ is used. Without the observation between time $t - t_l$ and t, the queue length estimation will be certainly undermined and has a larger uncertainty.

6.3 Simulation studies

6.3.1 Simulation setup

Figure 6.3 shows the setup of the simulation environment (based on SUMO) that is used to test the proposed real-time traffic controller. The simulation environment has an isolated intersection with

four approaches and 8 movements. Each movement has a single lane with the same length $250\ m$. All left-turn movements are protected with a dedicated left-turn lane.

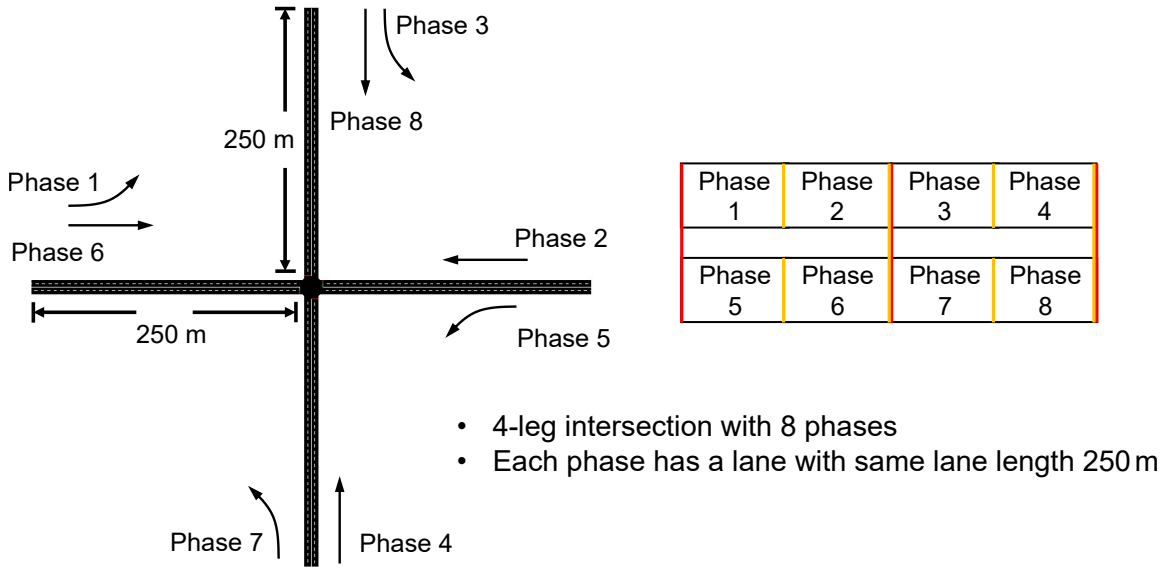


Figure 6.3: Simulation setup for real-time traffic signal control.

Table 6.1 shows the traffic volumes of all movements (phases). All movements have a Poisson arrival process with a uniform arrival rate. There are two different demand levels: peak hours with a v/c (volume to capacity) ratio of 0.8 and off-peak hours with a v/c ratio of 0.6.

	v/c ratio	Traffic volumes (vph)			
		Phase 1 & 5	Phase 2 & 6	Phase 3 & 7	Phase 4 & 8
Peak	0.8	216	540	288	396
Off-peak	0.6	144	432	216	288

Table 6.1: Traffic volume configuration for the simulation environment.

Table 6.2 reports all the traffic signal parameters that are used by different traffic controllers. Both fixed-time and actuated control are used as benchmark controllers. For the fixed-time traffic signal timing plan, the cycle length is determined by Webster's equation while the green time of each movement is proportional to the corresponding traffic volumes. For the actuated control, the maximum green is obtained by scaling up the green time of the fixed-time plan while the minimum green time is set as $5\ s$ for each phase. Maximum and minimum green times are used by both the proposed QCC and vehicle-acuated control.

Two different metrics are used to evaluate each traffic signal control method: 1) average control delay and 2) split failure ratio. The average control delay quantifies the overall average performance of the intersection and is used to determine the LOS (Manual, 2010). Split failure

Phase index	1 & 5	2 & 6	3 & 7	4 & 8	Total (cycle)
			Peak		
Green time (fixed-time)	14	31	17	23	85
Maximum green	16	41	21	29	107
Minimum green	5	5	5	5	-
			Off-peak		
Green time (fixed-time)	7	21	10	14	52
Maximum green	10	30	15	20	75
Minimum green	5	5	5	5	-

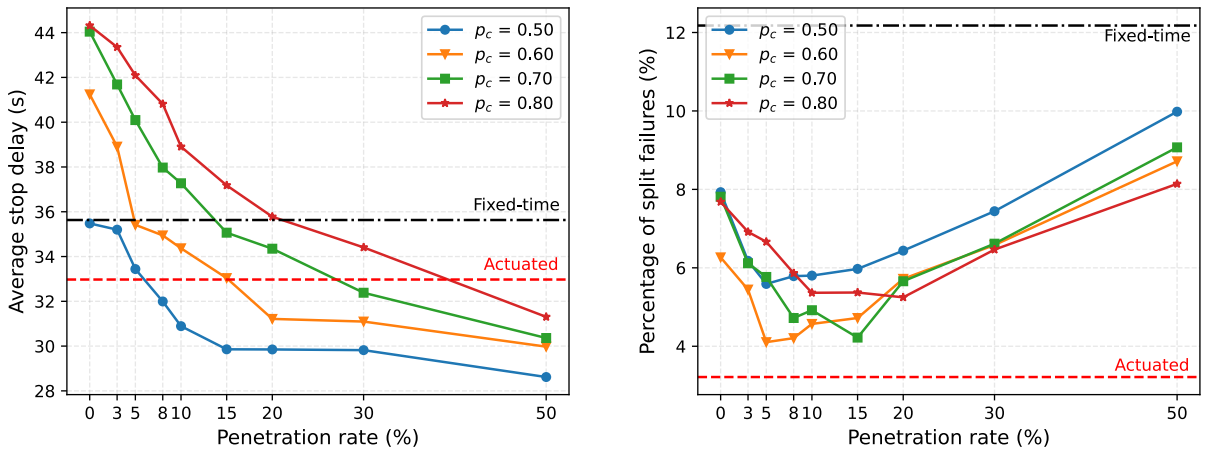
Table 6.2: Fixed-time parameters and max/min green of each phase.

occurs when a vehicle is unable to clear the intersection within the cycle it arrives at, resulting in a delay that exceeds the duration of a single cycle. Therefore, split failures are often a cause of driver dissatisfaction and draw additional attention from traffic engineers. While the average control delay provides a measurement of the average system performance, the split failure can be regarded as a measurement of the worst-case performance.

6.3.2 Main results

Figure 6.4 shows the average delay and split failure ratio of QCC with different p_c values under different penetration rates (using peak-hour traffic volumes in Table 6.1). Both fixed-time and actuated control are also labeled in the figure. Each controller is evaluated by a 5-hour simulation test. As shown in Figure 6.4 (a), for each QCC controller, the average control delay decreases with the increase of the penetration rate. When the penetration rate goes higher, the benefits become marginal. There are two potential reasons: 1) the proposed estimation methods fit the low penetration rate case better; it is not designed for the high penetration rate case (see discussion in Section 4.6.2); 2) the improvement brought by an accurate traffic state estimation is marginal. Due to the first reason particularly, we only show the results when the penetration rate is less than 50%.

Another observation from Figure 6.4 (a) is that the average control delay monotonically increases with the increase of the p_c value. That is, the controller performs worse with a large required clearance confidence p_c. This is because a large p_c will lead to an overly conservative control strategy: the green time is larger than usually what is needed for each phase. Although we might expect that a large p_c will have better split-failure performance on the other side, this is not always consistent with the results given by Figure 6.4 (b): $p_c = 0.6$ is always better than $p_c = 0.5$ but increasing p_c to 0.7 and 0.8 makes it even worse. This might be because when p_c gets higher, the average performance is too bad according to Figure 6.4 (a), which also limits the worst-case performance of the system given by Figure 6.4 (b). When p_c is not too large (less than



(a) Avg. stop delay

(b) Split failure ratio

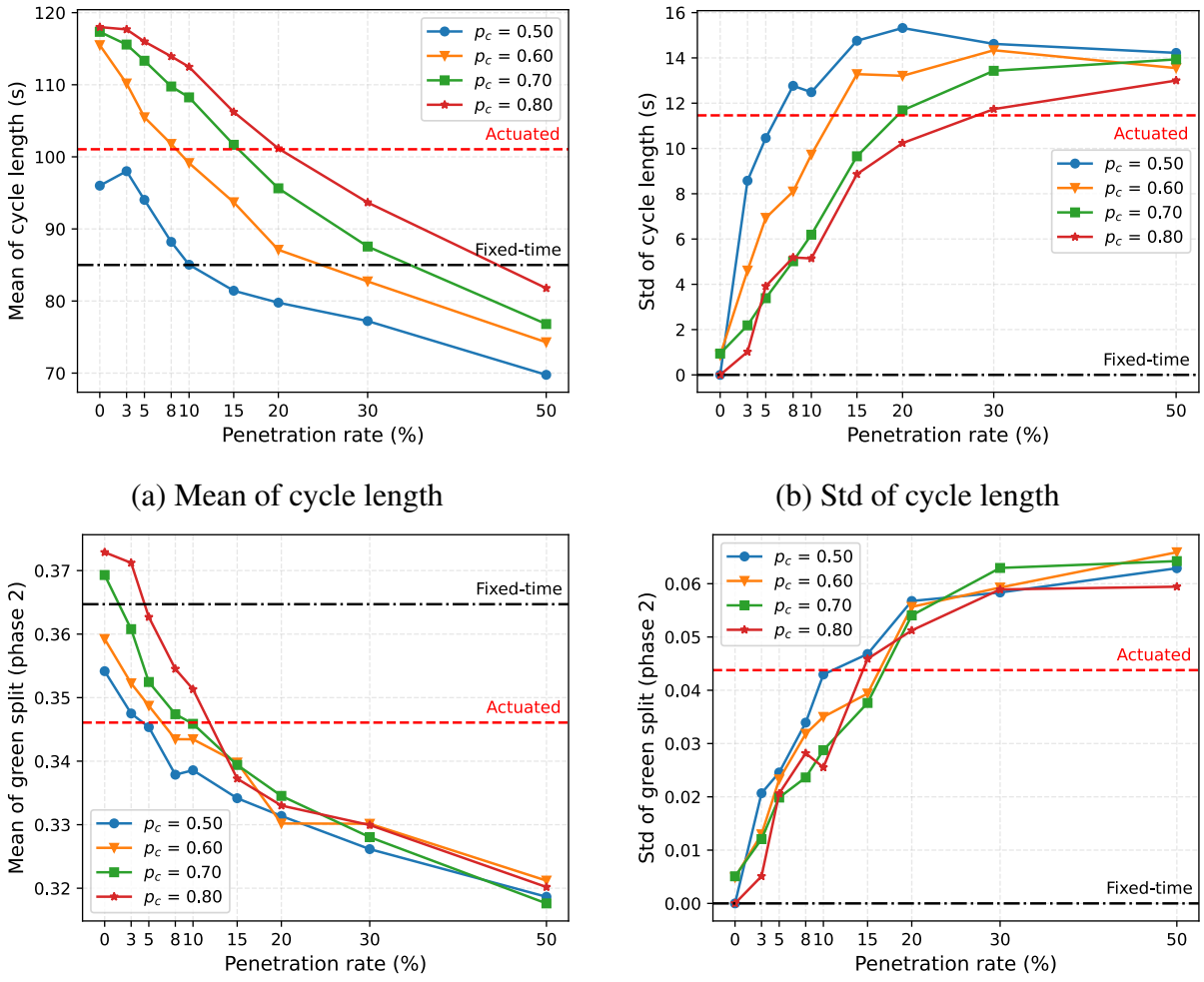
Figure 6.4: Avg. stop delay and split failure ratio of QCC control (peak hours).

0.7), we can still observe the trade-off between the average and worst-case performance of the system, quantified by average control delay and split failure ratio, respectively.

Nevertheless, Figure 6.4 clearly demonstrates the effectiveness of the proposed QCC method. When selecting p_c as 0.5, the average delay as shown by the blue line in Figure 6.4 (a) outperforms both fixed-time (black dashed line) and actuated control (red dashed line) when the penetration rate is larger than 8%. When the penetration rate is 15%, particularly, the QCC with $p_c = 0.5$ has approximately 20% and 10% less delay, compared with the fixed-time and actuated control, respectively. For the split failure ratio as shown in Figure 6.4 (b), all QCC controllers with different p_c values perform between the fixed-time and actuated control.

However, this result does not intend to show that the proposed QCC with vehicle trajectory data outperforms the vehicle-actuated control with detector data. QCC and vehicle-actuated control use similar control strategies. Vehicle-actuated control should have better performance since detectors can provide complete traffic information while vehicle trajectory data with a low penetration rate cannot. The current vehicle-actuated control has a split-failure ratio less than 4%, which is less than all QCC controllers as shown in Figure 6.4 (a), indicating that it is more conservative. This is the reason why vehicle-actuated control performs worse than the QCC with $p_c = 0.5$ in Figure 6.4 (a) when the penetration rate is larger than only 8%.

Figure 6.5 shows the statistics of the resulting cycle length and green split (Phase 2) of different traffic signal controllers, which also verifies some of the previous explanations. Figure 6.5 (a) shows the mean of the cycle lengths. For QCC, the cycle length monotonically decreases with the increase of the penetration rate. As illustrated by Figure 6.6, this is because when the penetration rate gets higher, the estimated (posterior) distribution of the queue length becomes



(a) Mean of cycle length

(b) Std of cycle length

(c) Mean of green split

(d) Std of green split

Figure 6.5: Resulting statistics of cycle length and green split (Phase 2) of QCC control (peak hours).

more concentrated with more observed vehicle trajectories. Consequently, less green time is needed for each movement to ensure the same queue clearance probability (p_c value). This might be the same reason that the green split of Phase 2 monotonically decreases as shown in Figure 6.5 (c). Phase 2 has a larger traffic volume and thereby more vehicle trajectories, which means that it might need less green time compared with other phases.

We can also see based on Figure 6.5 (a), even with the same min/max green time, the vehicle-actuated control has a larger cycle length compared with QCC since it will not terminate the current phase immediately when the queue is cleared. It will keep the current phase unchanged as long as the time headway is less than the minimum gap until the maximum green time is reached.

Figure 6.5 (b) and (d) show the standard derivation (std) of cycle length and green split. Both parameters increase with a larger penetration rate. This is because when the penetration rate goes

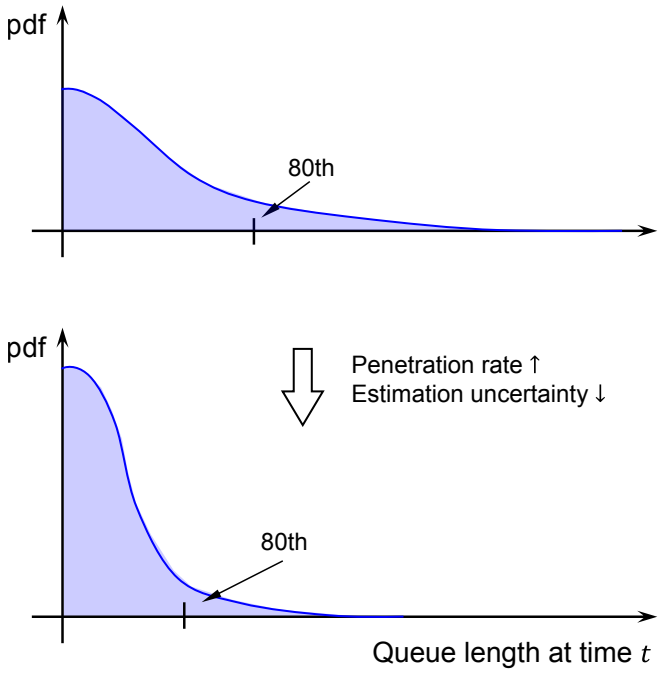


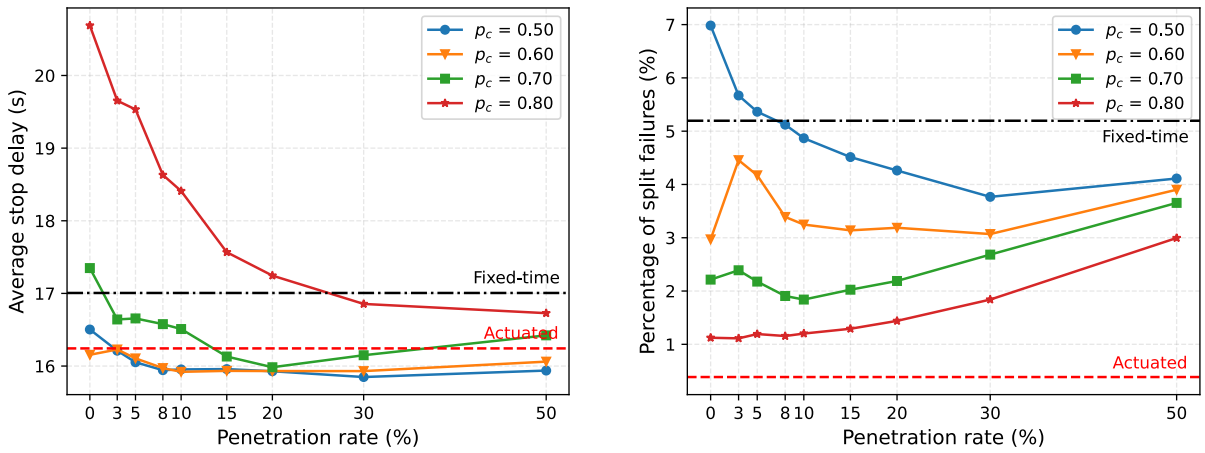
Figure 6.6: Change of the queue length percentile under different penetration rates.

higher, more observation is available and enables a more responsive control.

6.3.3 More insights

This subsection will show more results that can provide us with more insights. Figure 6.7 shows the system performance of different traffic controllers under the off-peak demand given by Table 6.1. In this light traffic scenario, a larger queue clearance confidence interval p_c can be used. Unlike the previous results under heavy traffic (Figure 6.4) in which the average system performs significantly worse with a larger p_c, QCC under light traffic performs well until $p_c = 0.70$. As shown in 6.7 (b), the split failure ratio now monotonically decreases with the increase of p_c value. Based on this observation, a less p_c value should be used for a larger traffic demand level. Besides, much less improvement is observed under light traffic compared with the previous heavy traffic scenario. This is simply because much fewer trajectories can be observed when the traffic volume is low, which is consistent with the sensitivity analysis of the traffic state estimation in Section 4.5.3.

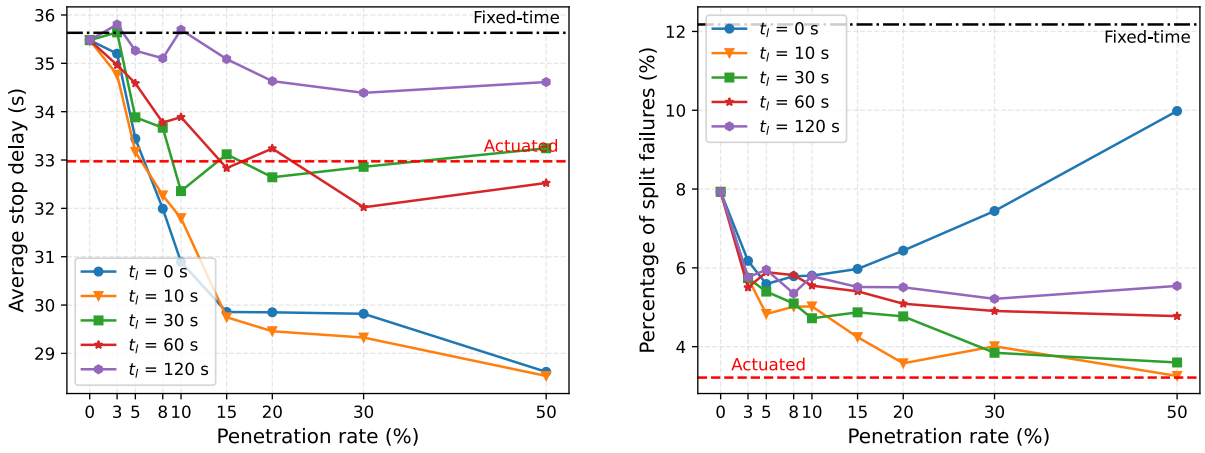
Figure 6.8 shows how the time lag t_l influences the performance of QCC (peak hours, $p_c = 0.5$ as an example). As shown in Figure 6.8 (a), the QCC performs worse with an increase of the average stop delay when the lag time t_l increases. When $t_l = 120\ s$, which is larger than one cycle, the QCC behaves like a fixed-time control. With a lag time of $30 - 60$ seconds, QCC at



(a) Avg. stop delay (b) Split failure ratio

Figure 6.7: Avg. stop delay and split failure ratio of QCC control (off-peak hours).

a penetration rate of 15% can only have similar performance with QCC without latency at a 5% penetration rate. A large lag t_l means that the most recently available observed vehicle trajectory and the resulting estimated queue length become outdated, making it less useful for real-time traffic signal control.



(a) Avg. stop delay (b) Split failure ratio

Figure 6.8: Influence of the lag time t_l ($p_c = 0.5$, peak hours).

6.4 Summary and discussions

6.4.1 Summary

This chapter proposes a rule-based real-time traffic signal control strategy called QCC. Inspired by the vehicle-acuated control, QCC switches the phase whenever the estimated queue length is cleared with a certain confidence interval p_c, subjecting to the min/max green constraint. A simulation environment with an isolated intersection is built to test the proposed QCC and compare it with both fixed-time and vehicle-actuated control. Different traffic conditions and parameters are tested for QCC, including light/heavy traffic, queue clearance confidence interval p_c, and the time lag t_l.

Unlike the recurrent traffic state that can still be accurately estimated even at a low penetration rate by aggregating sufficient historical data, real-time traffic state estimation faces inherent fundamental limits caused by sparse and incomplete observation, which will also hinder the performance of real-time traffic signal control. The latency of data collection also has a significant influence on the controller's performance.

Nevertheless, this chapter demonstrates the benefits of utilizing vehicle trajectory data for real-time traffic signal control, particularly for those intersections with large traffic volumes.

6.4.2 Discussions

The proposed QCC in this chapter directly takes the real-time estimated queue length under the Newellian coordinates as the input. Figure 6.9 illustrates one potential limitation of such a method. Recap that the Newellian coordinates system uses a different time t. Instead of using the actual time t', it uses the free-flow arrival time t', and their mapping is given by Equation (3.3) in Chapter 3. As shown in Figure 6.9, the black dashed line shows the normal time t' while the pink line shows the Newellian time t. The real-time queue length estimation methods introduced in Chapter 4 find the posterior of the current traffic state given all previous observations under the Newellian coordinates, i.e., $p(X(t)|\mathcal{O}(1:t))$. This posterior distribution is used as the input for QCC in this chapter. The observable region under Newellian coordinates from the initial time to time t is labeled by the green color in the figure. However, at time t', the overall observable region is given by the red region ($\mathcal{O}'(1:t')$). Their difference is the blue triangle denoted by $\tilde{\mathcal{O}}$.

By taking the real-time estimated queue length under the Newellian coordinates as the input, the proposed QCC only utilizes the observation within the green region. If there are any connected vehicle trajectories in the blue region, they will be ignored although can be observed at the current time. This is a fundamental limitation of using the Newellian coordinates for real-time traffic signal control, which is not only for the proposed QCC but for all real-time traffic signal controllers that

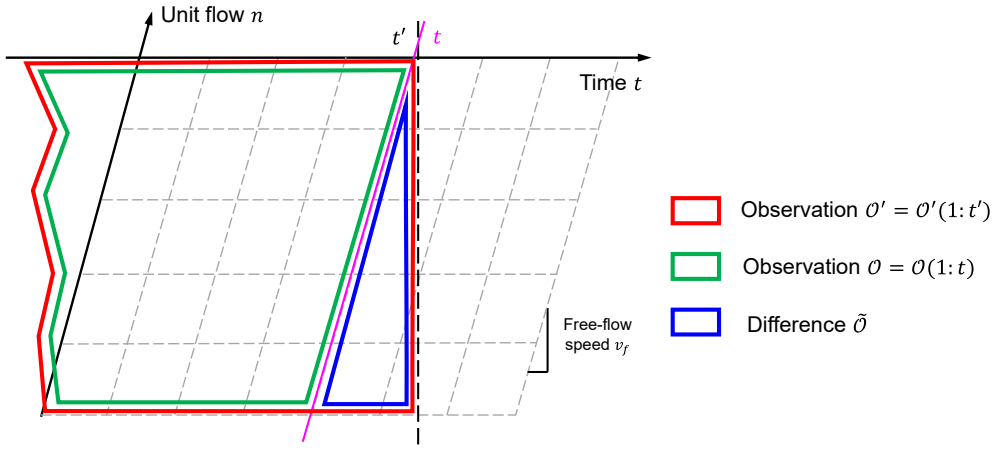


Figure 6.9: Limitation of the real-time traffic state estimation with Newellian coordinates.

use the same input. It has a similar effect with the time lag as shown in Figure 6.2. In practice, this is not a severe issue since the blue region is much less than that caused by the time lag.

For future works, there are multiple directions that can be further explored. For example, the QCC proposed in this chapter is only a simple rule-based method, it will be interesting to investigate other control methods such as the optimal control methods, the max pressure control, and some other model-free data-driven methods. Besides, this chapter only shows the real-time control of an isolated intersection, it needs to be extended to a larger scale such as a corridor or even a general traffic network. Last but certainly not least, we look forward to testing these real-time control methods in the field.

CHAPTER 7

Summary and Future Directions

7.1 Summary of the dissertation

This dissertation focuses on traffic signal optimization with vehicle trajectory data at a low penetration rate. While most existing traffic signal control systems are established based on loop detectors, vehicle trajectory data provides an alternative that is more scalable, accessible, and cost-efficient. Instead of traffic counts and speed at certain locations installed with detectors, vehicle trajectory data provides different and more enriched information including vehicle delay, stop, and path, etc. However, the major limitations of utilizing such data for traffic signal optimization include sparse observation caused by the low penetration rate and the lack of a suitable traffic flow model. The incorporation of a stochastic traffic flow model is important for an accurate traffic state estimation when the observation is incomplete.

To overcome these challenges, this dissertation presents a systematic approach and framework built upon a novel stochastic traffic flow model introduced in Chapter 3. This traffic flow model is established based on a newly proposed Newellian coordinates system. By assuming all vehicles follow a uniform deterministic Newell's car-following model, vehicle trajectories can be projected to a point queue process under the Newellian coordinates. A point-queue representation has much lower dimensions and can be easily extended to a stochastic setting. We also propose the PTS diagram that can project the stochastic point queue process back to the spatial-temporal space. In this way, a complete mapping between the point-queue representation and the spatial-temporal traffic state is established so that the simple point-queue model under the Newellian coordinates can sufficiently capture the spatial-temporal traffic state.

Other than being a stochastic traffic flow model with much lower dimensions, another major advantage of the proposed model is that it can be directly calibrated by taking vehicle trajectory data as input. This model enables us to build a probabilistic graphical model (a Bayesian network) to connect unknown traffic states & parameters with observed vehicle trajectories. Based on the same probabilistic model, Chapter 4 proposes two different methods for the traffic state and

parameter estimation. The first method is the MM estimation, which is a frequentistic method and can be used to estimate stationary traffic state and parameters by matching the model-estimated delay with observed delay from vehicle trajectories. The second method is based on the Bayesian estimation, which not only provides the estimation of both traffic state and parameters but the uncertainty of these values in the form of the posterior distribution.

Estimated traffic state and parameter given by Chapter 4 are directly used for the traffic signal optimization methods in Chapter 5 and Chapter 6. Chapter 5 develops traffic signal retiming methods for fixed-time traffic signals. A gradient-based method is used for isolated intersections while a coordinate-descent method is proposed to improve the coordination of coordinated intersections like a corridor or any coordinated critical path. Chapter 6 proposes a rule-based method named QCC. Either simulation or field implementation is used to test the proposed traffic signal optimization methods in these two chapters.

The content of this thesis encompasses a comprehensive integrated traffic signal control system called OSaaS, which includes data preprocessing (Chapter 2), traffic modeling (Chapter 3), traffic state estimation (Chapter 4), and traffic signal optimization (Chapter 5-6). A citywide field test of OSaaS was conducted in Birmingham, Michigan included monitoring, diagnosis, and optimization of 34 coordinated and isolated signalized intersections. Two corridors and two isolated intersections were detected with a relatively large optimality gap and new signal timing plans were generated and implemented. These new plans resulted in decreases in both the delay and number of stops by up to 20% and 30%, respectively. As a closed-loop iterative system, OSaaS significantly shortens each re-timing iteration, so a more responsive and strategic traffic signal retiming is feasible. By not requiring installation or maintenance of vehicle detectors, OSaaS provides a more scalable, sustainable, resilient, and efficient solution to traffic signal re-timing based on vehicle trajectory, which could be applied to every fixed-time traffic signal in the world.

7.2 Future directions

At last, this dissertation will provide some future directions including both research problems and practical issues in real-world implementation.

Real-world implementation Most of the methods proposed in this dissertation are designed for real-world implementation. Some methods have been tested in the field using real-world trajectory data. However, some of them are not, particularly for real-time traffic signal control. We are looking forward to also implementing those methods in the real world. Besides, this dissertation aims at proposing a generic methodology and framework for traffic signal control with vehicle trajectory data. Many assumptions are used for simplification purposes so that more clean and

logical content can be presented. However, real-world traffic conditions could be much more complicated. There are many corner cases and details need to be taken into account.

Data-driven methods Most methods proposed in this dissertation are model-based methods. As just mentioned above, one significant drawback of model-based approaches is their limited ability to handle various corner cases that are beyond the scope of the proposed universal method. In recent decades, data-driven methods have drawn tremendous attention and achieved remarkable success, particularly in some fields like computer vision and natural language processing. Although these methods cannot be directly or easily used for sparsely observed vehicle trajectory data, it is worthwhile to invest more effort in exploring this direction. Another technique path is to combine both data-driven and model-based methods; many researchers have already made promising explorations in this regard (Cai et al., 2021; Li et al., 2022; Di et al., 2023).

Network-level traffic signal control with re-routing The traffic signal optimization approach in this dissertation is limited to an isolated intersection or a corridor (a coordinated path). Although a network is usually decomposed into corridors and isolated intersections for traffic signal management, network-level traffic signal control requires additional considerations. One assumption used in this dissertation is that the traffic demand does not change much over time. However, the traffic demand in the real world is elastic, which can be induced by a better service level (Lee Jr et al., 1999). Drivers might also change their route in response to a different traffic signal timing. Therefore, network-level traffic signal optimization should proactively consider the change in the network traffic demand pattern, particularly in the long run. This is not an easy research direction. As aforementioned in Section 5.5, proactively considering drivers' route choice will lead to a bi-level formulation which is hard to solve. Besides, it needs the OD traffic demand as the input, which is much harder to be estimated with vehicle trajectory data considering the potential bias of the data collection process.

Other applications with vehicle trajectory data While this dissertation focuses on traffic signal optimization, there are many other potential applications utilizing vehicle trajectory data. For example, it can be used to study the network-level traffic demand including the OD estimation (Liu et al., 2023). It can also be used for parking-cruising detection (Weinberger et al., 2020), map generation (Shi et al., 2009), and safety-related applications.

APPENDIX A

Additional Details of the Traffic Flow Model

A.1 Effective green time

Due to the perception-reaction time (PRT) and vehicle acceleration after the green light starts, effective green time is known to be slightly different from the display green time. Similarly, after the green time ends, there is still a certain probability that some vehicles clear the intersection during the yellow time. Figure A.1 shows how the effective green time can be derived based on the raw SPaT information. For this specific movement, the green time and yellow time are G and Y, respectively. Let μ_g be the average PRT while $\mu_y = G + Y/2$ is the time of half of the yellow time. Here we will show two different methods to get effective green time under different uses.

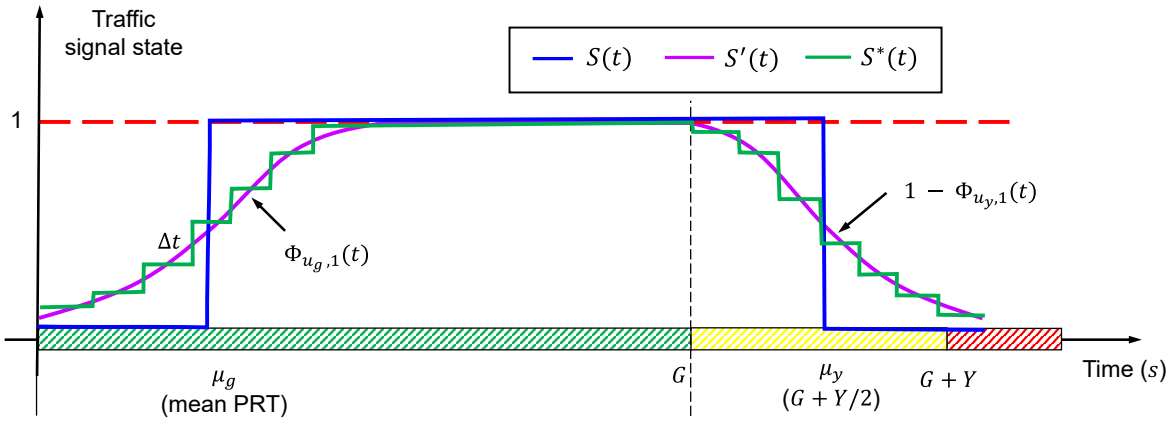


Figure A.1: Effective green time.

The generation of the PTS diagram requires a deterministic binary traffic signal state. Therefore, a rectangular effective green time $S(t)$ will be used in this case which is determined by:

$$S(t) = \begin{cases} 1 & t \in [\mu_g, \mu_y] \\ 0 & \text{otherwise} \end{cases} . \tag{A.1}$$

If we only care about the point-queue representation (arrival/departure profile, delay, etc.) of the movement without requiring the PTS diagram, the traffic signal state can also be a decimal number. We can use a cumulative Gaussian to model the green start-up time and the yellow light interval as shown by the purple curve $S'(t)$ as shown in Figure A.1. Let $\Phi_{\mu,\sigma^2}(x)$ be the cumulative density function (cdf) of a Gaussian distribution with mean μ and variance σ^2, $S'(t)$ is determined by:

$$S'(t) = \begin{cases} \Phi_{\mu_g,\sigma^2}(t) & t \leq G \\ 1 - \Phi_{\mu_y,\sigma^2}(t) & t > G \end{cases}. \tag{A.2}$$

The variance σ^2 is chosen as 1. Since the time is discrete with interval Δt, let $S^*(t)$ be the discrete approximation of $S'(t)$:

$$S^*(t) = \frac{1}{\Delta t} \int_{\tau=t}^{t+\Delta t} S'(t) d\tau \tag{A.3}$$

Figure A.2 demonstrates some real-world examples of $S(t)$ and $S^*(t)$ and how the predicted departures for both methods compare to the observations. The x-axis plots the signal light indication given by the traffic signal while the dashed green lines indicate the calculated signal states for each method. The model is able to match the observed departures fairly well with both methods, but $S^*(t)$ is a little more detailed, particularly during the green start-up time.

A.2 Permissive movements

This subsection will introduce how we approximate the effective green time or traffic states of permissive movements that must yield to other protected movements. Figure A.3 shows two examples. In the first case, the left-turn and through movement from the opposing direction share the same green duration and the left-turn movement i needs to yield the opposing protected through movement p. For the second case, the right-turn movement i can turn right during the red time while yielding to the protected through movement p. We will only show the details of the first case. The intuition is to use a gap acceptance model to get the left-over capacity for the permissive movements after subtracting the through movement utilization.

As shown in Figure A.3a, let $S_p(t)$ and $B_p(t)$ represent the traffic signal states and the departure profile of the protected movement accordingly. The departure profile essentially represents the utilization of the traffic signal state of the movement. Define the $B_p^c(t)$ as the left-over capacity, we have:

$$B_p^c(t) = \begin{cases} S_p(t) - B_p(t) & t \leq G \\ 1 - B_p(t) & G < t \leq G + Y \end{cases}. \tag{A.4}$$

We use $1 - B_p(t)$ to calculate the left-over capacity for the yellow time since left-turn vehicles can usually clear the intersection during the entire yellow time. Instead of directly using the left-over

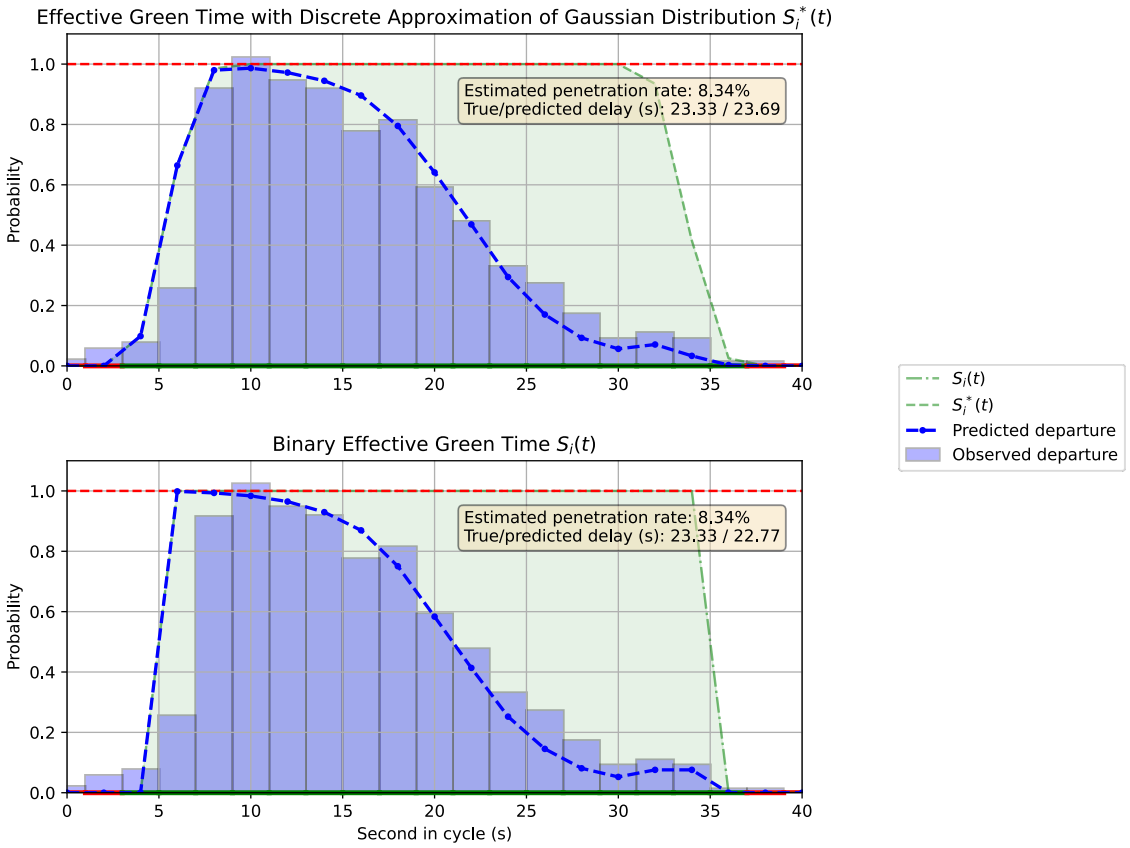


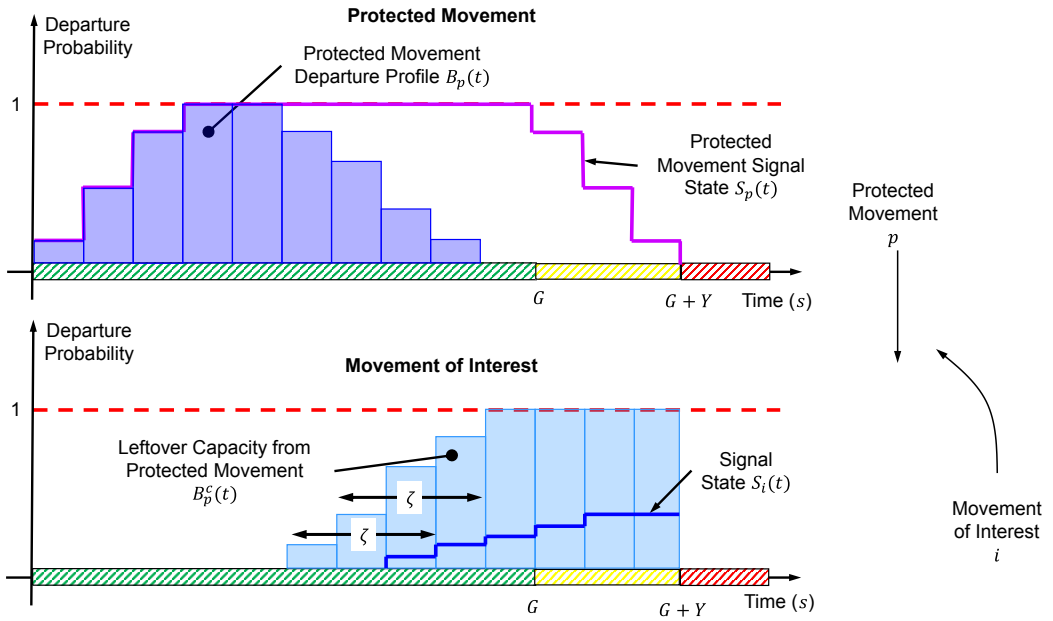
Figure A.2: Predicted vs. observed departures using $S^*(t)$ and $S(t)$

capacity as the traffic state for the permissive movement, a gap acceptance model is further applied since vehicles in the permissive movement might require the protected movement to be empty for a few consecutive time steps. Let ζ be the number of time steps of the gap acceptance model. The traffic signal state of the permissive movement is eventually determined by:

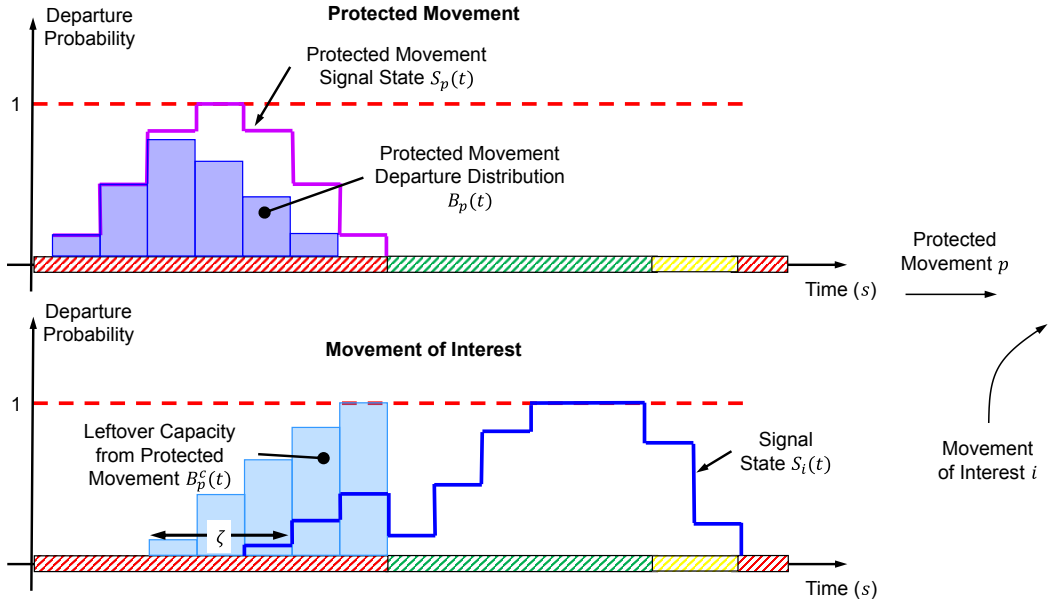
$$S_i(t) = \prod_{\tau=t+1-\zeta}^{t} B_p^c(\tau). \tag{A.5}$$

We can apply the gap acceptance model to get the traffic signal states for right turn on reds in Figure A.3.

The real-world example in Figure A.4 illustrates how this method can accurately capture the observed departure profiles and the measured delays. The intersection analyzed in this figure is controlled by two phases, one for each street. As a result, the left-turn movement and the oncoming protected through movement share the same SPaT information and left turning vehicles must wait for a reasonable gap in the protected movement departures before proceeding through the intersection. When considering the protected movement, the predicted departure profile



(a) Permissive Left Turn Movement



(b) Right Turn on Red

Figure A.3: Illustration of permissive movements.

resembles the observed departure profile because it doesn't immediately allow vehicles to clear the intersection. One limitation of this model is that it will usually predict zero departures in the early stages of the green time because the model will predict maximum protected departures when the queue is first released (there is no leftover capacity when the light first turns green). Left turn departures could happen earlier in the green time during some random cycles where the conflicting movement's queue is small, but this is a rare occurrence and does not impact the model's ability to capture the average traffic state of the movement.

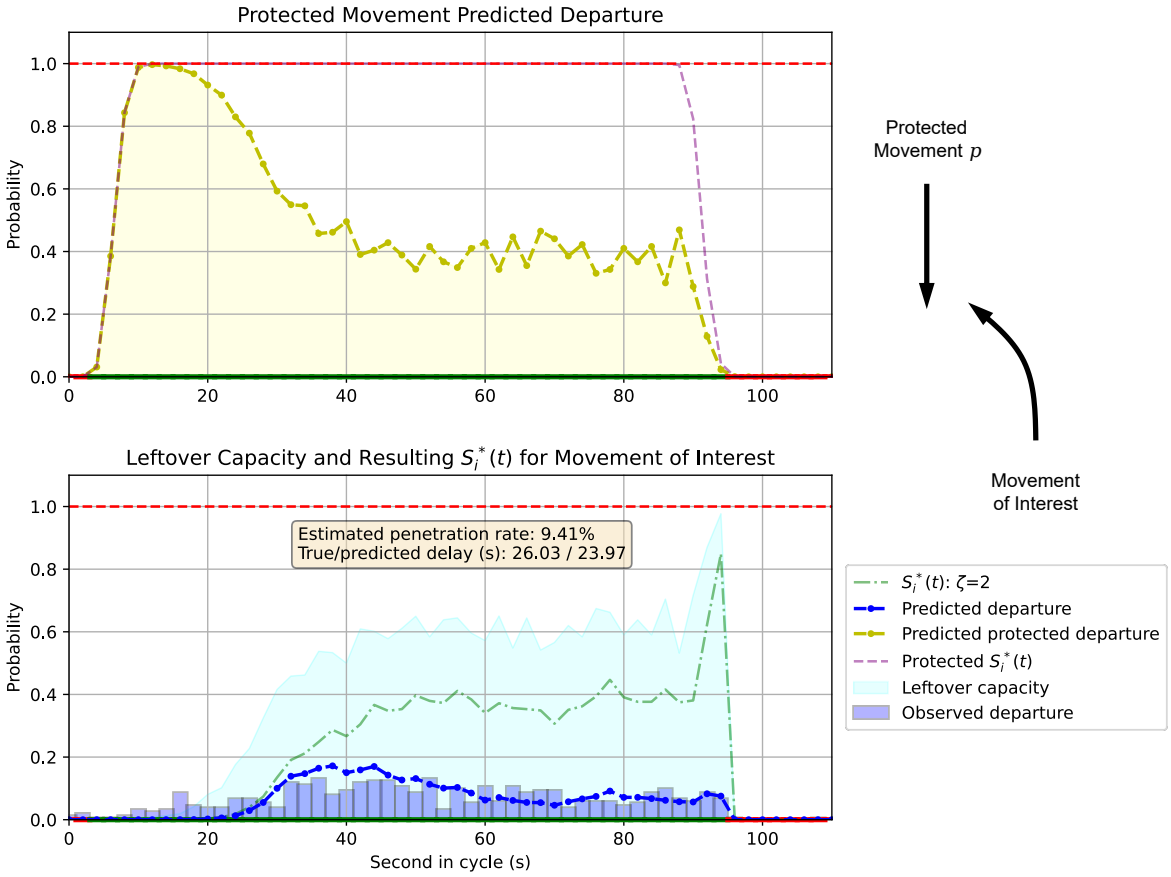


Figure A.4: Permissive Left Turn Movement Example: Quarton Road and Cranbrook RD WBL Movement - PM TOD

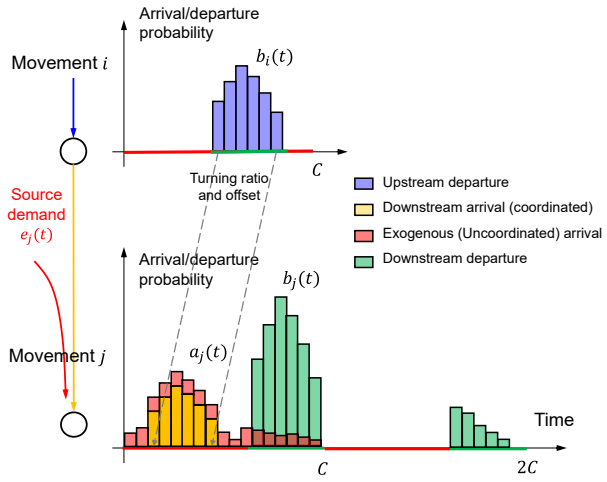
A.3 Approximation of a network of movements

We also use single-queue decomposition approximation to model a general network consisting of multiple movements. For a movement within a traffic network as shown in Figure A.5, the arrival can be decomposed into the external arrival coming from external demand and the internal arrival

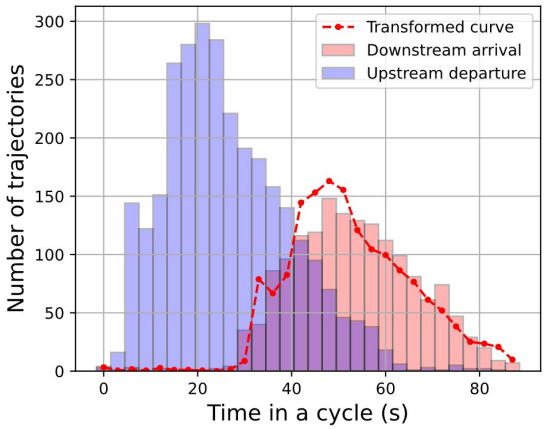
from upstream movements. The arrival is determined by:

$$\mathbb{P}\left(A_i(t)=1\right)=\sum_{k\in\mathcal{M}_i^u}\left(\mathbb{P}(B_k(t-T_{ki})=1)\cdot r_{ki}\frac{\Delta u_k}{\Delta u_i}\right)+\mathbb{P}(E_i(t)=1) \tag{A.6}$$

where $\mathcal{M}_i^u$ is the set of upstream movements of movement i, r_{ki} is the turning ratio, T_{ki} is the free-flow travel time from movement k to movement i; $E_i(t)$ is the exogenous arrival. Δu_i is the unit flow of movement i which is defined in the main paper as the saturation flow within the time interval Δt. The arrival given by Equation (A.6) is also assumed to follow a Bernoulli distribution and arrivals at different time steps are also independent. Based on this assumption, the whole network is then decomposed into a set of movements; the stationary distribution of each movement queue length can be calculated according to the network topology from upstream to downstream.



(a) Coordinated movement arrival



(b) Internal arrival

Figure A.5: Arrival of the coordinate movement.

Note that the arrival coming from the upstream is actually correlated and will be dependent on all the previous states (Osorio and Wang, 2017; Boon and van Leeuwaarden, 2018). Therefore, the actual stationary distribution of a network is hard to obtain, and the proposed method is only a single-queue decomposition approximation. Besides, the upstream platoon might also disperse when vehicles travel along the link (Robertson and Bretherton, 1991) and the downstream vehicles might also block the upstream vehicles (Osorio and Bierlaire, 2009). In this paper, we use the simplest approximation without considering these complicated scenarios.

APPENDIX B

Pre-Determined and Calibrated Parameters

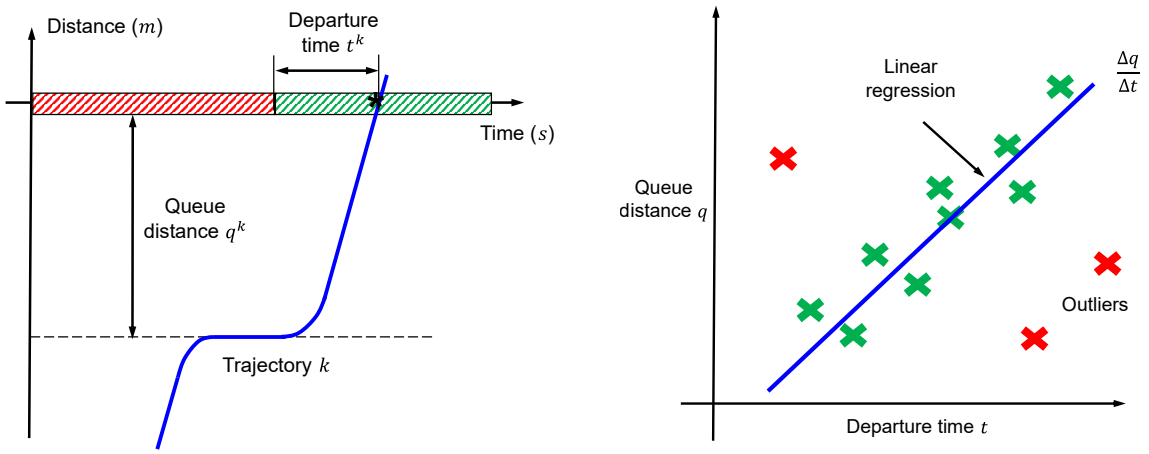
B.1 Saturation flow rate estimation

The saturation flow rate q_i^m for each movement $i \in \mathcal{M}$ in the traffic network can be estimated from the vehicle trajectory data. For each trajectory k, the departure time after the green start t^k and queue distance q^k are illustrated by Figure B.1 (a). By assuming that the jam space headway is h_0, there will be q^k/h_0 vehicles in the queue, and this means that it takes time t^k to allow q^k/h_0 vehicles to clear the intersection. The saturation flow rate can then be estimated according to the following equation:

$$q_i^m = \frac{\Delta n}{\Delta t} = \frac{\frac{\Delta q}{h_0}}{\Delta t} = \frac{\Delta q}{\Delta t} \cdot \frac{1}{h_0} \tag{B.1}$$

where Δn is the number of vehicles clearing the intersection within the time interval Δt. The first equality is the definition of the saturation flow rate. $\Delta q/\Delta t$ is the slope of the $q-t$ scatter as shown in Figure B.1 (b). This means that the saturation flow rate can be estimated through a linear regression over the $q-t$ scatters for all the collected trajectories. For a set of observations $\mathcal{O}_i = \left\{t_i^k, q_i^k, \forall k\right\}$ of movement i, we first eliminate trajectories that departed during the "start-up loss time". The RANSAC regression (Derpanis, 2010) is then used to further remove the outliers and get an accurate $q-t$ slope.

Figure B.2 illustrates an example of the saturation flow rate estimation where the estimated queue discharge rate is 3.54 m/s. Inlier points that are included in the final regression estimate are shown in green, while the outliers are shown in yellow. Around 92% of the total observed points were considered in the estimation. The R^2 value of 0.93 is a measure of linear regression accuracy. According to Equation (B.1) and using an assumed jam space headway of 7 meters per vehicle, the estimated saturation flow rate of this movement will be $1,820$ vphpl (vehicle per hour per lane).



(a) Queue distance and departure time (b) q-k scatters

Figure B.1: Saturation Flow Rate Estimation

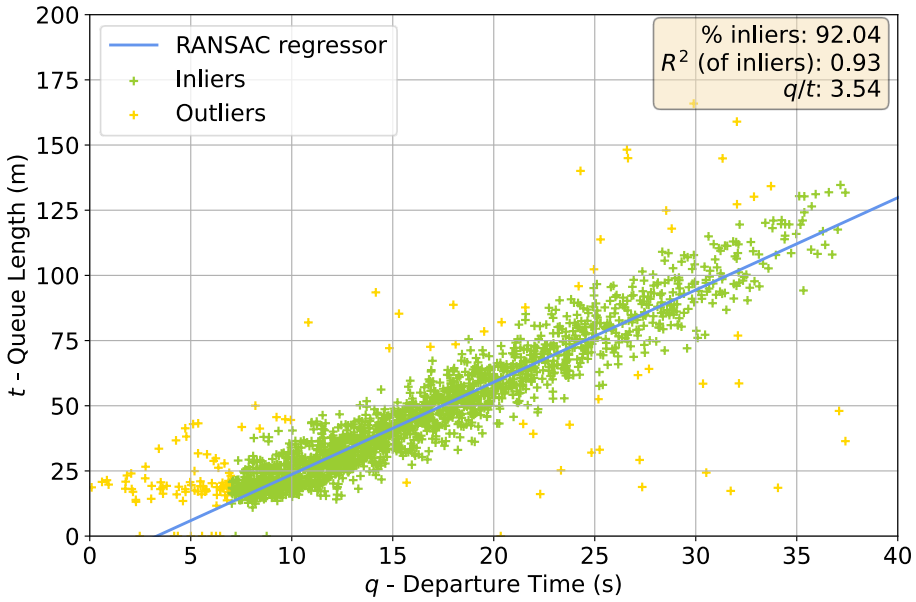


Figure B.2: Saturation Flow Rate Estimation Example: Maple Road and Adams Road WB Movement - PM TOD

B.2 Other parameters

Jam space headway Jam space headway refers to the space headway when vehicles stop at the signalized intersections. It is usually a constant determined by the vehicle length and the space between stopped vehicles (front bumper to front bumper). This paper assumes a constant jam space headway of 7 meters per vehicle.

Free-flow speed The free-flow speed of each movement is determined by the average free-flow speed extracted from each individual vehicle trajectory of the movement (Wang et al., 2022a).

Turning ratios Since the trajectory data includes path information, the turning ratio can be estimated from the observed trajectories. For example, let movement j be the downstream of movement i, $\tilde{n}_i$ is the total number of observed trajectories of movement i while $\tilde{n}_{ij}$ is the total number of observed trajectories that travels to movement j from movement i, then the turning ratio r_{ij} is estimated as:

$$\hat{r}_{ij} = \frac{\tilde{n}_{ij}}{\tilde{n}_i}. \tag{B.2}$$

Lane numbers There are cases when some movements share the same lane, for example, right-turn movement and through movement share one lane. Since this paper assumes that the queue lengths of different movements are separate, the equivalent lane number for each movement is proportional to the number of observed trajectories under this case.

BIBLIOGRAPHY

Aboudolas, K., Papageorgiou, M., and Kosmatopoulos, E. (2009). Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17(2):163–174.

Arel, I., Liu, C., Urbanik, T., and Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 4(2):128–135.

Avila, A. and Mezić, I. (2020). Data-driven analysis and forecasting of highway traffic dynamics. *Nature communications*, 11(1):1–16.

Ban, X. J., Hao, P., and Sun, Z. (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.

Barmpounakis, E. and Geroliminis, N. (2020). On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment. *Transportation research part C: emerging technologies*, 111:50–71.

Bezzina, D. and Sayer, J. (2014). Safety pilot model deployment: Test conductor team report. *Report No. DOT HS*, 812:171.

Boeing, G. (2017). Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.

Boon, M. A. and van Leeuwaarden, J. S. (2018). Networks of fixed-cycle intersections. *Transportation Research Part B: Methodological*, 117:254–271.

Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E. (2021). Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738.

Chaudhary, N. and Chu, C. (2002). Passer v—software for timing signalized arterials. *College Station, TX: Texas Transportation Institute*.

Chen, C., Ding, Y., Xie, X., Zhang, S., Wang, Z., and Feng, L. (2019). Trajcompressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):2012–2028.

Cheng, Y., Qin, X., Jin, J., Ran, B., and Anderson, J. (2011). Cycle-by-cycle queue length estimation for signalized intersections using sampled trajectory data. *Transportation research record*, 2257(1):87–94.

Chien, S. I., Kim, K., and Daniel, J. (2006). Cost and benefit analysis for optimized signal timing-case study: New jersey route 23. *Institute of Transportation Engineers. ITE Journal*, 76(10):37.

Chu, T., Wang, J., Codecà, L., and Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*.

Comert, G. (2013). Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals. *Transportation Research Part B: Methodological*, 55:59–74.

Comert, G. (2016). Queue length estimation from probe vehicles at isolated intersections: Estimators for primary parameters. *European Journal of Operational Research*, 252(2):502–521.

Comert, G. and Cetin, M. (2009). Queue length estimation from probe vehicle location and the impacts of sample size. *European Journal of Operational Research*, 197(1):196–202.

Comert, G. and Cetin, M. (2011). Analytical evaluation of the error in queue length estimation at traffic signals from probe vehicle data. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):563–573.

Cui, Z., Ke, R., Pu, Z., and Wang, Y. (2020). Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, 118:102674.

Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287.

Daganzo, C. F. (2005a). A variational formulation of kinematic waves: basic theory and complex boundary conditions. *Transportation Research Part B: Methodological*, 39(2):187–196.

Daganzo, C. F. (2005b). A variational formulation of kinematic waves: Solution methods. *Transportation Research Part B: Methodological*, 39(10):934–950.

Department, B. T. and Howard/Stein-Hudson Associates, I. (2010). The benefits of retiming/ rephasing traffic signals in the back bay, benefit cost evaluation of signal improvements. Technical report.

Derpanis, K. G. (2010). Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3.

Di, X., Shi, R., Mo, Z., and Fu, Y. (2023). Physics-informed deep learning for traffic state estimation: A survey and the outlook. *Algorithms*, 16(6):305.

Dobrota, N., Stevanovic, A., and Mitrovic, N. (2020). Development of assessment tool and overview of adaptive traffic control deployments in the us. *Transportation research record*, 2674(12):464–480.

Dobrota, N., Stevanovic, A., and Mitrovic, N. (2022). Modifying signal retiming procedures and policies by utilizing high-fidelity modeling with medium-resolution traffic data. *Transportation research record*, 2676(3):660–684.

Doucet, A., Johansen, A. M., et al. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.

Fazzinga, B., Flesca, S., Furfaro, F., and Parisi, F. (2014). Cleaning trajectory data of rfid-monitored objects through conditioning under integrity constraints. In *EDBT*, pages 379–390.

Feng, Y., Head, K. L., Khoshmagham, S., and Zamanipour, M. (2015). A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:460–473.

Feng, Y., Yu, C., and Liu, H. X. (2018). Spatiotemporal intersection control in a connected and automated vehicle environment. *Transportation Research Part C: Emerging Technologies*, 89:364–383.

Flötteröd, G. and Osorio, C. (2017). Stochastic network link transmission model. *Transportation Research Part B: Methodological*, 102:180–209.

Gartner, N. H. (1983). *OPAC: A demand-responsive strategy for traffic signal control*. Number 906.

Gartner, N. H., Assman, S. F., Lasaga, F., and Hou, D. L. (1991). A multi-band approach to arterial traffic signal optimization. *Transportation Research Part B: Methodological*, 25(1):55–74.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Guo, Q., Li, L., and Ban, X. J. (2019). Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101:313–334.

Hale, D. (2005). Traffic network study tool–transyt-7f, united states version. *Mc-Trans Center in the University of Florida*.

Hao, P., Ban, X., Bennett, K. P., Ji, Q., and Sun, Z. (2012). Signal timing estimation using sample intersection travel times. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):792–804.

Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., and Bayen, A. M. (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.

Hunt, P., Robertson, D., Bretherton, R., and Winton, R. (1981). Scoot-a traffic responsive method of coordinating signals. Technical report.

Husch, D. and Albeck, J. (2004). Trafficware synchro 6 user guide. *TrafficWare, Albany, California*, 11.

Jabari, S. E. and Liu, H. X. (2012). A stochastic model of traffic flow: Theoretical foundations. *Transportation Research Part B: Methodological*, 46(1):156–174.

Jabari, S. E. and Liu, H. X. (2013). A stochastic model of traffic flow: Gaussian approximation and estimation. *Transportation Research Part B: Methodological*, 47:15–41.

Khamis, M. A. and Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, 29:134–151.

Koonce, P. and Rodegerdts, L. (2008). Traffic signal timing manual. Technical report, United States. Federal Highway Administration.

Kovvali, V. G., Alexiadis, V., and Zhang PE, L. (2007). Video-based vehicle trajectory data collection. Technical report.

Laval, J. A. and Leclercq, L. (2013). The hamilton–jacobi partial differential equation and the three representations of traffic flow. *Transportation Research Part B: Methodological*, 52:17–30.

Lee, J., Park, B., and Yun, I. (2013). Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. *Journal of Transportation Engineering*, 139(10):1020–1029.

Lee Jr, D. B., Klein, L. A., and Camus, G. (1999). Induced traffic and induced demand. *Transportation Research Record*, 1659(1):68–75.

Levin, M. W. (2023). Max-pressure traffic signal timing: A summary of methodological and experimental results. *Journal of Transportation Engineering, Part A: Systems*, 149(4):03123001.

Li, J., Yu, C., Shen, Z., Su, Z., and Ma, W. (2023). A survey on urban traffic control under mixed traffic environment with connected automated vehicles. *Transportation Research Part C: Emerging Technologies*, 154:104258.

Li, L., Chen, X., Liu, Q., and Bao, Z. (2020). A data-driven approach for gps trajectory data cleaning. In *International Conference on Database Systems for Advanced Applications*, pages 3–19. Springer.

Li, W. and Ban, X. (2018). Connected vehicles based traffic signal timing optimization. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4354–4366.

Li, W., Yang, C., and Jabari, S. E. (2022). Nonlinear traffic prediction as a matrix completion problem with ensemble learning. *Transportation science*, 56(1):52–78.

Liang, X. J., Guler, S. I., and Gayah, V. V. (2020). An equitable traffic signal control scheme at isolated signalized intersections using connected vehicle technology. *Transportation Research Part C: Emerging Technologies*, 110:81–97.

Light, M. and Whitham, B. (1955). On kinematic waves. i: Flow movement in long rivers; ii: A theory of traffic flow on long crowed roads [c]. *Proceedings of Royal Society A*, (229):281–345.

Little, J. D. and Graves, S. C. (2008). Little's law. In *Building intuition*, pages 81–100. Springer.

Little, J. D., Kelson, M. D., and Gartner, N. H. (1981). Maxband: A versatile program for setting signals on arteries and triangular networks.

Liu, X. and Zheng, J. (2019). Traffic signal control using vehicle trajectory data. US Patent 10,497,259.

Liu, Z., Yin, Y., Bai, F., and Grimm, D. K. (2023). End-to-end learning of user equilibrium with implicit neural networks. *Transportation Research Part C: Emerging Technologies*, 150:104085.

Lo, H. K. (1999). A novel traffic signal control formulation. *Transportation Research Part A: Policy and Practice*, 33(6):433–448.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE.

Lowrie, P. (1990). Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic.

Lu, J. and Zhou, X. (2022). Modeling partially schedulable connected and automated mobility systems on layered virtual-track networks: modeling framework and open-source tools.

Lu, Y. C., Krambeck, H., and Tang, L. (2017). Use of big data to evaluate and improve performance of traffic signal systems in resource-constrained countries: evidence from cebu city, philippines. *Transportation Research Record*, 2620(1):20–30.

Ma, W., Wan, L., Yu, C., Zou, L., and Zheng, J. (2020). Multi-objective optimization of traffic signals based on vehicle trajectory data at isolated intersections. *Transportation research part C: emerging technologies*, 120:102821.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Manual, H. C. (2010). Hcm2010. *Transportation Research Board, National Research Council, Washington, DC*, 1207.

Maripini, H., Khadhir, A., and Vanajakshi, L. (2023). Traffic state estimation near signalized intersections. *Journal of Transportation Engineering, Part A: Systems*, 149(5):03123002.

Mirchandani, P. and Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432.

NACTO (2015). Fixed vs. actuated signalization. *National Association of City Transportation Officials*.

Newell, G. F. (2002). A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, 36(3):195–205.

Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343.

Oblakova, A. I. (2019). Queueing models for urban traffic networks.

OpenStreetMap (2019). Openstreetmap. Technical report, https://www.openstreetmap.org.

Osorio, C. and Bierlaire, M. (2009). An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. *European Journal of Operational Research*, 196(3):996–1007.

Osorio, C. and Wang, C. (2017). On the analytical approximation of joint aggregate queue-length distributions for traffic networks: A stationary finite capacity markovian network approach. *Transportation Research Part B: Methodological*, 95:305–339.

Osorio, C. and Yamani, J. (2017). Analytical and scalable analysis of transient tandem markovian finite capacity queueing networks. *Transportation Science*, 51(3):823–840.

Quddus, M. and Washington, S. (2015). Shortest path and vehicle trajectory aided map-matching for low frequency gps data. *Transportation Research Part C: Emerging Technologies*, 55:328–339.

Ramezani, M. and Geroliminis, N. (2015). Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civil and Infrastructure Engineering*, 30(6):414–432.

Richards, P. I. (1956). Shock waves on the highway. *Operations research*, 4(1):42–51.

Robertson, D. I. and Bretherton, R. D. (1991). Optimizing networks of traffic signals in real time-the scoot method. *IEEE Transactions on vehicular technology*, 40(1):11–15.

Saldivar-Carranza, E., Li, H., Mathew, J., Hunter, M., Sturdevant, J., and Bullock, D. M. (2021). Deriving operational traffic signal performance measures from vehicle trajectory data. *Transportation Research Record*, page 03611981211006725.

Shi, W., Shen, S., and Liu, Y. (2009). Automatic generation of road network map from massive gps, vehicle trajectories. In *2009 12th international IEEE conference on intelligent transportation systems*, pages 1–6. IEEE.

Son, P. (2019). Traffic signal benchmarking and state of the practice report. Technical report, National Operation Center of Excellence.

Stern, R. E., Cui, S., Delle Monache, M. L., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Pohlmann, H., Wu, F., Piccoli, B., et al. (2018). Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, 89:205–221.

Sumalee, A., Zhong, R., Pan, T., and Szeto, W. (2011). Stochastic cell transmission model (sctm): A stochastic dynamic traffic model for traffic state surveillance and assignment. *Transportation Research Part B: Methodological*, 45(3):507–533.

Sun, Z. and Ban, X. J. (2013). Vehicle trajectory reconstruction for signalized intersections using mobile traffic sensors. *Transportation Research Part C: Emerging Technologies*, 36:268–283.

Sunkari, S. (2004). The benefits of retiming traffic signals. *Institute of Transportation Engineers. ITE Journal*, 74(4):26.

Tokdar, S. T. and Kass, R. E. (2010). Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60.

Turner, S. M., Eisele, W. L., Benz, R. J., and Holdener, D. J. (1998). Travel time data collection handbook. Technical report, United States. Federal Highway Administration.

Urbanik, T., Tanaka, A., Lozner, B., Lindstrom, E., Lee, K., Quayle, S., Beaird, S., Tsoi, S., Ryus, P., Gettman, D., et al. (2015). *Signal timing manual*, volume 1. Transportation Research Board Washington, DC.

Van Woensel, T. and Vandaele, N. (2007). Modeling traffic flows with queueing models: a review. *Asia-Pacific Journal of Operational Research*, 24(04):435–461.

Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195.

Viti, F. and Van Zuylen, H. J. (2010). Probabilistic models for queues at fixed control signals. *Transportation Research Part B: Methodological*, 44(1):120–135.

Wada, K., Usui, K., Takigawa, T., and Kuwahara, M. (2017). An optimization modeling of coordinated traffic signal control based on the variational theory and its stochastic extension. *Transportation research procedia*, 23:624–644.

Wang, S., Bao, Z., Culpepper, J. S., and Cong, G. (2021). A survey on trajectory data management, analytics, and learning. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Wang, X., Jerome, Z., Zhang, C., Shen, S., Kumar, V. V., and Liu, H. X. (2022a). Trajectory data processing and mobility performance evaluation for urban traffic networks. *Transportation Research Record*, page 03611981221115088.

Wang, X., Shen, S., Bezzina, D., Sayer, J. R., Liu, H. X., and Feng, Y. (2020). Data infrastructure for connected vehicle applications. *Transportation Research Record*, 2674(5):85–96.

Wang, X., Yin, Y., Feng, Y., and Liu, H. X. (2022b). Learning the max pressure control for urban traffic networks considering the phase switching loss. *Transportation Research Part C: Emerging Technologies*, 140:103670.

Wang, Y., Zhao, M., Yu, X., Hu, Y., Zheng, P., Hua, W., Zhang, L., Hu, S., and Guo, J. (2022c). Real-time joint traffic state and model parameter estimation on freeways with fixed sensors and connected vehicles: State-of-the-art overview, methods, and case studies. *Transportation Research Part C: Emerging Technologies*, 134:103444.

Wang, Z., Lu, M., Yuan, X., Zhang, J., and Van De Wetering, H. (2013). Visual traffic jam analysis based on trajectory data. *IEEE transactions on visualization and computer graphics*, 19(12):2159–2168.

Wei, H., Zheng, G., Gayah, V., and Li, Z. (2019). A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117*.

Weinberger, R. R., Millard-Ball, A., and Hampshire, R. C. (2020). Parking search caused congestion: Where's all the fuss? *Transportation Research Part C: Emerging Technologies*, 120:102781.

Wong, W., Shen, S., Zhao, Y., and Liu, H. X. (2019). On the estimation of connected vehicle penetration rate based on single-source connected vehicle data. *Transportation Research Part B: Methodological*, 126:169–191.

Wu, X. and Liu, H. X. (2011). A shockwave profile model for traffic flow on congested urban arterials. *Transportation Research Part B: Methodological*, 45(10):1768–1786.

Yan, H., He, F., Lin, X., Yu, J., Li, M., and Wang, Y. (2019). Network-level multiband signal coordination scheme based on vehicle trajectory data. *Transportation Research Part C: Emerging Technologies*, 107:266–286.

Yang, C. and Gidofalvi, G. (2018). Fast map matching, an algorithm integrating hidden markov model with precomputation. *International Journal of Geographical Information Science*, 32(3):547–570.

Yao, Z., Jiang, Y., Zhao, B., Luo, X., and Peng, B. (2020). A dynamic optimization method for adaptive signal control in a connected vehicle environment. *Journal of Intelligent Transportation Systems*, 24(2):184–200.

Yau, K.-L. A., Qadir, J., Khoo, H. L., Ling, M. H., and Komisarczuk, P. (2017). A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys (CSUR)*, 50(3):1–38.

Yperman, I., Logghe, S., and Immers, B. (2005). The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks. In *Proceedings of the 10th EWGT Meeting*, pages 122–127. Poznan Poland.

Yu, C., Feng, Y., Liu, H. X., Ma, W., and Yang, X. (2018). Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation research part B: methodological*, 112:89–112.

Zhang, G. and Wang, Y. (2010). Optimizing minimum and maximum green time settings for traffic actuated control at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):164–173.

Zhao, Y., Shen, S., and Liu, H. X. (2021). A hidden markov model for the estimation of correlated queues in probe vehicle environments. *Transportation Research Part C: Emerging Technologies*, 128:103128.

Zhao, Y. and Tian, Z. (2012). An overview of the usage of adaptive signal control system in the united states of america. *Applied Mechanics and Materials*, 178:2591–2598.

Zhao, Y., Zheng, J., Wong, W., Wang, X., Meng, Y., and Liu, H. X. (2019a). Estimation of queue lengths, probe vehicle penetration rates, and traffic volumes at signalized intersections using probe vehicle trajectories. *Transportation Research Record*, page 0361198119856340.

Zhao, Y., Zheng, J., Wong, W., Wang, X., Meng, Y., and Liu, H. X. (2019b). Various methods for queue length and traffic volume estimation using probe vehicle trajectories. *Transportation Research Part C: Emerging Technologies*, 107:70–91.

Zheng, J. and Liu, H. X. (2017). Estimating traffic volumes for signalized intersections using connected vehicle data. *Transportation Research Part C: Emerging Technologies*, 79:347–362.

Zheng, J., Sun, W., Huang, S., Shen, S., Yu, C., Zhu, J., Liu, B., and Liu, H. X. (2018). Traffic signal optimization using crowdsourced vehicle trajectory data. Technical report.

Zheng, Y. (2015). Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41.