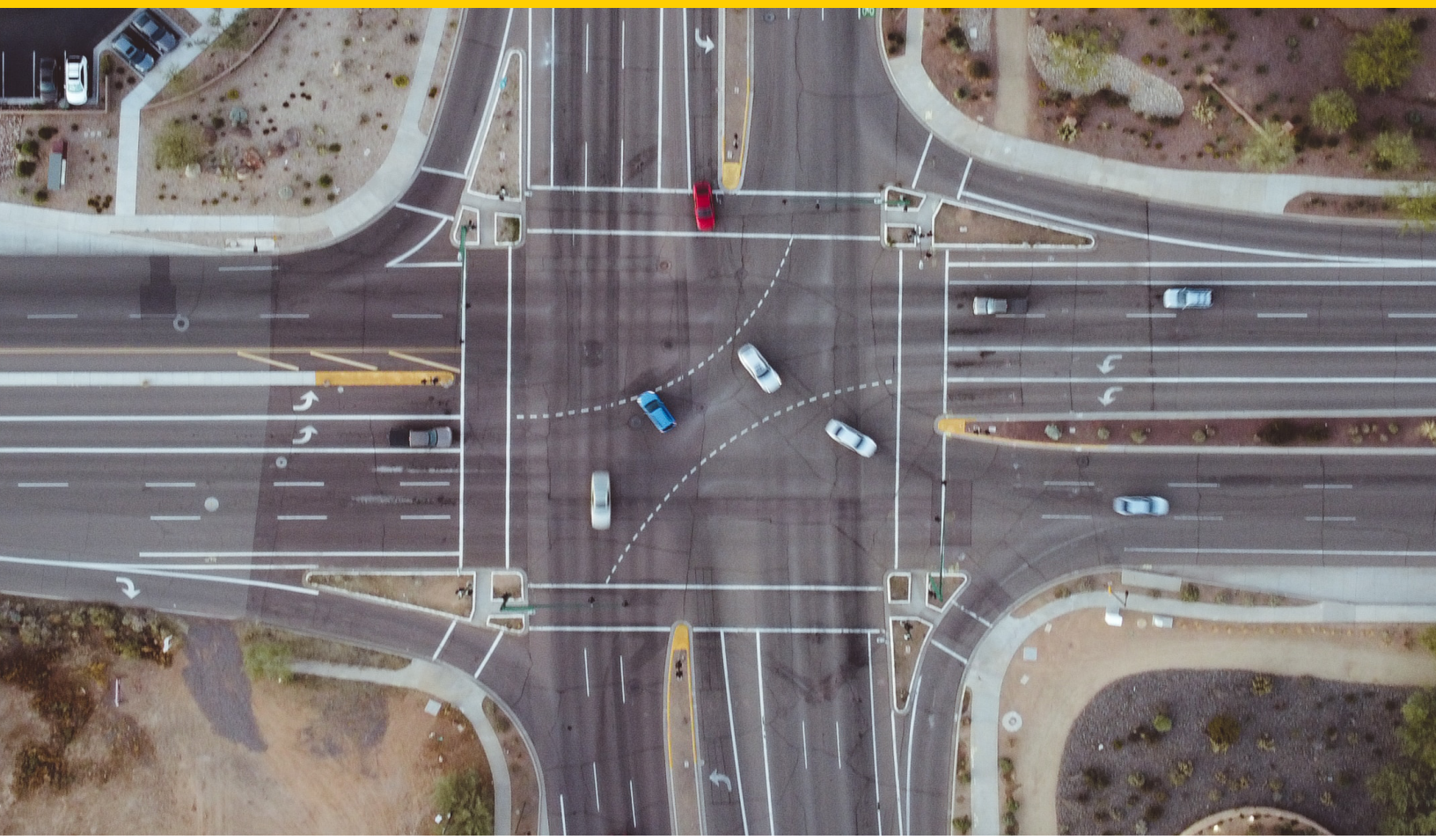
CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Final Report #56
September 2023



Leveraging Connected and Automated Vehicles for Participatory Traffic Control

Minghui Wu
Xingmin Wang
Yafeng Yin, PhD
Henry Liu, PhD

CIVIL & ENVIRONMENTAL ENGINEERING
UNIVERSITY OF MICHIGAN

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Report No. 56

August 2023

Project Start Date: 2/1/2021

Project End Date: 8/31/2023

Leveraging Connected and Automated Vehicles for Participatory Traffic Control

by
Minghui Wu
Xingmin Wang
Yafeng Yin (PI)
Henry Liu (PI)
University of Michigan

UMTRI | The University of Akron | Illinois Center for Transportation | Purdue University | Washtenaw Community College | Central State University

DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. 69A3551747105 of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested APA Format Citation: Wu, M., Wang, X., Yin, Y., Liu, H., Wang, B., Lynch, J. (2023). Leveraging Connected and Automated Vehicles for Participatory Traffic Control. Final Report.
DOI: 10.7302/8088

Contacts

For more information:

PI: Yafeng Yin
2120 GG Brown
Ann Arbor, Michigan
Phone: (734) 764-8249
Email: yafeng@umich.edu

PI: Henry X. Liu
2116 GG Brown
Ann Arbor, Michigan
Phone: (734) 764-4354
Email: henryliu@umich.edu

CCAT
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48152
uumtri-ccat@umich.edu
(734) 763-2498

Technical Report Documentation Page

1. Report No. CCAT Report No. 56	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Leveraging Connected and Automated Vehicles for Participatory Traffic Control DOI: 10.7302/8088		**5. Report Date** August 2023 **6. Performing Organization Code**
7. Author(s) Minghui Wu, https://orcid.org/0000-0002-1247-7688 Xingmin Wang, Ph. D., https://orcid.org/0000-0003-0435-2786 Yafeng Yin, Ph. D., https://orcid.org/0000-0003-3117-5463 Henry Liu, Ph. D., https://orcid.org/0000-0002-3685-9920 Ben Wang, https://orcid.org/0000-0003-0790-0995 Jerome P. Lynch, Ph. D., https://orcid.org/0000-0002-8793-0061		**8. Performing Organization Report No.**
9. Performing Organization Name and Address Center for Connected and Automated Transportation University of Michigan Transportation Research Institute 2901 Baxter Road Ann Arbor, MI 48109		**10. Work Unit No.** **11. Contract or Grant No.** Contract No. 69A3551747105
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE Washington, DC 20590		**13. Type of Report and Period Covered** Final report (February 2021–August 2023) **14. Sponsoring Agency Code** OST-R
15. Supplementary Notes Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's (OST-R) University Transportation Centers (UTC) program.		
16. Abstract This report lays the theoretical groundwork for participatory traffic control that integrates traditional infrastructure like traffic signals with connected and automated vehicles (CAVs) acting as mobile actuators. The study is divided into two main parts. First, we introduce a robust traffic state estimation method that leverages real-time data from CAVs to yield precise insights into the transportation system. Second, we present an analytical framework to guide the strategic control of CAVs, aiming to indirectly influence the behavior of human-driven vehicles and optimize traffic flow across various time periods and transportation facilities. Our research serves as a foundational step towards the practical deployment of participatory traffic control systems, contributing to the creation of smarter, more efficient transportation networks for the future.		
17. Key Words Connected and automated vehicles, traffic estimation, traffic control, mean-field control	**18. Distribution Statement** No restrictions.	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	40	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Abstract

In the future of transportation systems, traditional physical controllers, such as traffic signals, will be complemented by the active participation of connected and automated vehicles (CAVs) functioning as mobile actuators. This report is concerned with establishing a theoretical foundation for this innovative participatory traffic control scheme. In doing so, it is crucial to first gain accurate and ample information of the transportation system. Therefore, in the first part of the report, we propose a traffic state estimation method using the information from CAVs. In the second part of the report, we analytically examine how to control CAVs to indirectly influence the behaviors of human-driven vehicles, strategically redistributing traffic demand across various time periods and transportation facilities. This research paves the way for the practical implementation of participatory traffic control, contributing to the development of smarter and more efficient transportation networks in the future.

Table of Contents

List of Figures

List of Tables

1. Introduction

In the years to come, the traffic landscape will be a blend of traditional human-driven vehicles, connected vehicles, and various forms of connected automated vehicles (CAVs). Our focus is on leveraging CAVs to enhance the management and operation of road networks. Specifically, we propose a participatory traffic control model in which a community of CAVs is incentivized to contribute to traffic management. These CAVs may act as "travel demand distributors" to better allocate commuting needs over time and across facilities, or function as "traffic stream regulators" to improve the management of signalized intersections and prevent or delay bottlenecks. Our working hypothesis suggests that by influencing the behavior of a small, targeted percentage of CAVs (5-10% of total traffic), we can positively affect the decisions of a larger number of untargeted drivers, thereby improving overall system performance.

For instance, CAVs can act as moving traffic controllers, influencing traffic flow at intersections by regulating their own speed. The presence of CAVs introduces a new spatial control dimension, revolutionizing traditional traffic control systems and creating new challenges for integrating CAVs and existing infrastructure.

CAVs are poised to play a significant role in next-generation traffic management, offering promising prospects for improving both mobility and fuel economy. This report represents our preliminary efforts to implement participatory traffic control. To develop effective control methods, it's crucial to first accurately estimate the state of transportation systems. Chapter 2 of this study, led by Xingmin Wang of the University of Michigan, focuses on utilizing CAVs for traffic monitoring and state estimation through a hidden Markov model. The effort is represented as the left-hand component in Figure 1.1.

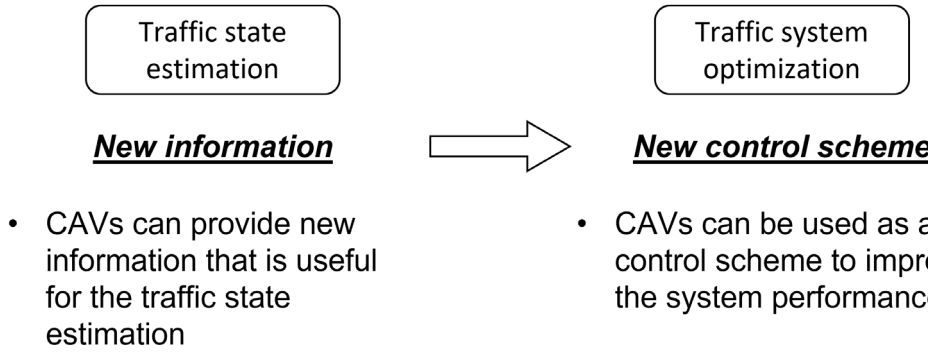


Figure 1.1: Participatory traffic control with CAVs: new input data and new control scheme

Once sufficient system information is obtained, CAVs can be used to influence the behavior of human-driven vehicles (HVs). See Figure 1.1. Chapter 3 explores a distributed, model-free approach to enhance system performance by controlling a fraction of CAVs. This is framed within the major-minor mean field control (MFC) framework. Reinforcement learning algorithms are applied to compute optimal control policies. This chapter documents collaborative findings with Minghui Wu and Ben Wang from the University of Michigan, as well as Jerome P. Lynch from Duke University.

2. Real-time urban traffic state estimation using connected and automated vehicle observation

2.1 Introduction

Compared with traditional traffic monitoring methods that are highly relied on the detectors, connected and automated vehicle data is more scalable, economical, and sustainable, which might become more prevalent in the future. Connected vehicle data is essentially in the form of vehicle trajectory data at a certain penetration rate. Many existing studies have applied statistical estimation methods to estimate the overall traffic state with low penetration rate vehicle trajectory data. Other than the vehicle trajectory, automated vehicles can also observe the surrounding traffic, which could contain much more useful information, particularly for the traffic from the opposing direction. In this case, automated vehicles act as moving observers in traffic networks. Researchers have noticed this potential and performed certain explorations during past years.

In this chapter, we utilize data from both connected and automated vehicles for the real-time estimation of urban traffic state. The overall estimation problem is formulated through a hidden Markov model. The hidden state is the overall traffic state while the observable state is the observed data. An existing stochastic traffic flow model is used to model the transition of the hidden state while new observation models are developed to connect the hidden state and observable state. A simulation environment built on SUMO is used to test the proposed method. This chapter is organized as follows: Section 2.2 introduces the problem state and Section 2.3 is the main methodology. Section 2.4 shows the numerical experiments based on SUMO simulation environment.

2.2 Problem statement

Figure 2.1 is an illustration of a road segment with both directions with the time-space diagram of northbound direction (from the bottom to the top). In the left figure, the blue color denotes automated vehicles while the grey color represents ordinary vehicles. The light blue area is an illustration of the detection range. In the corresponding time-space diagram on the right, solid blue lines represent vehicle trajectories of automated vehicles while dashed lines represent others. In this case, there are two automated vehicles moving in the opposite direction. The light blue color in the time-space diagram shows the observation of automated vehicles.

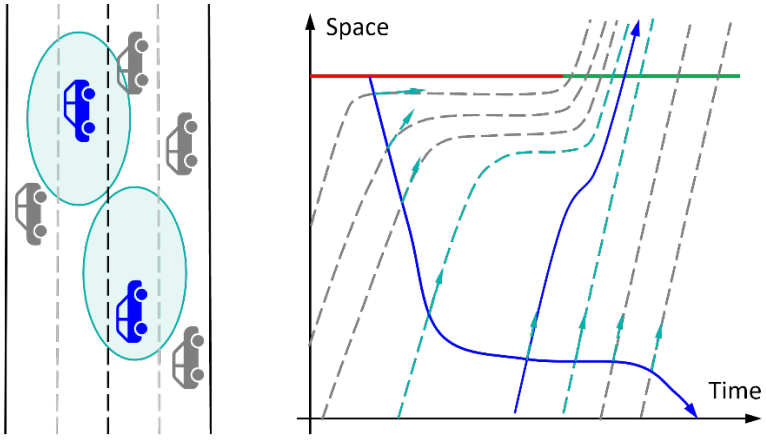


Figure 2.1: Observation of the automated vehicle

As illustrated in the figure, the observation coming from automated vehicles can be roughly divided into two categories: 1) background vehicles of the same direction and 2) background vehicles of the opposing direction. For a specific automated vehicle, the background vehicles of the same direction remain relatively unchanged unless the automated vehicle passes or is passed by other background vehicles. However, an automated vehicle can observe more vehicles from the opposing direction. Ideally, it could see all vehicles passing through from the other direction if its sight is not blocked. Therefore, the observation from the opposing information could contain more useful information intuitively. This is also the major difference between automated vehicles and connected vehicles. Their difference will be marginal if only observation of the same direction is available.

In this chapter, the objective is to estimate the overall traffic state utilizing data from connected and automated vehicles. For connected vehicles, only the trajectory is available while we have additional surrounding observation for automated vehicles.

2.3 Methodology

2.3.1 Hidden Markov Model

We formulate the traffic state estimation problem with connected and automated vehicle data as a hidden Markov model as illustrated by Figure 2.2. The hidden state represents the overall traffic state we try to estimate while the observable state is the observation from connected and automated vehicles.

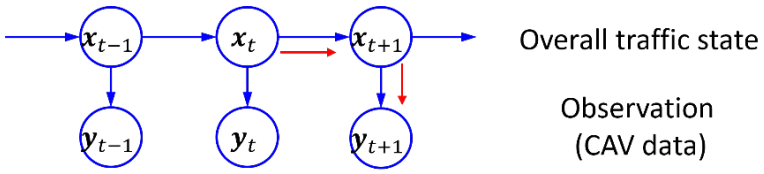


Figure 2.2: Hidden Markov model

To get a complete formulation for this hidden Markov model, we need to further specify the transition between the hidden state, i.e., a stochastic traffic flow model, and the transition between the hidden state and observable state, i.e., an observation model. Given the complete observation of such a hidden Markov model, finding the posterior distribution of the hidden state given all observations will be a recursive Bayesian estimation problem. Depending on the traffic flow model as well as the observation model, different filtering algorithms can be applied. For example, if both models are linear Gaussian, the Kalman filter can be utilized [2]. For a more complicated nonlinear model otherwise, we might only be able to use sampling-based method, which is usually more computational costly [3]. The following two subsections will introduce more details on the stochastic traffic flow model as well as the observation model.

2.3.2 Stochastic traffic flow model

We use the stochastic traffic flow model proposed by Jabari and Liu [4]. This stochastic traffic flow model is built based on Eulerian coordinates by splitting the roadway into cells. Figure 2.3 is an illustration of the stochastic traffic flow model. A road segment will be split into cells.

For each cell i, the length is l_i and the traffic density is $\rho(i,t)$ at time t. In this case, the overall traffic state of the road segment is denoted by the column vector $\boldsymbol{\rho}(t)$ which contains traffic densities of all the cells. The vector $y(x,t)$ is defined as a tuple including the traffic densities of two adjacent cells

$$y(x,t) = \begin{bmatrix} \rho(x,t) & \rho(x+1,t) \end{bmatrix}^T$$

and $\lambda(\cdot)$ is the boundary flow between cells.

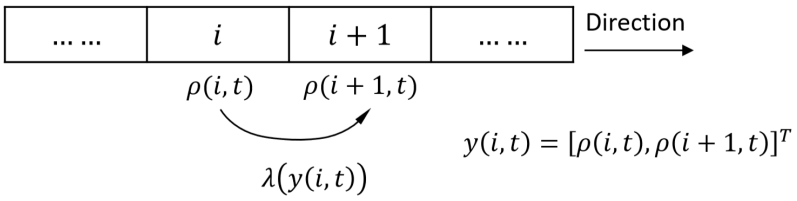


Figure 2.3: Stochastic traffic flow model

The stochastic traffic flow model is a Gaussian approximation model and thereby includes two parts: 1) mean dynamics and 2) covariance dynamics. As a Gaussian approximation, the traffic state at each time follows a Gaussian distribution with mean value $\bar{\boldsymbol{\rho}}(t)$ and covariance matrix $\boldsymbol{\Sigma}(t)$. The mean dynamics is given by the following equation:

$$\bar{\boldsymbol{\rho}}(t) = \bar{\boldsymbol{\rho}}(0) + \int_0^t \mathbf{B}\lambda(u)\,du$$

where the matrix **B** is determined by:

$$\mathbf{B} = \begin{bmatrix} \frac{1}{l_1} & -\frac{1}{l_1} & 0 & 0 & 0 \\ 0 & \frac{1}{l_2} & -\frac{1}{l_2} & 0 & 0 \\ 0 & 0 & \frac{1}{l_3} & -\frac{1}{l_3} & 0 \\ 0 & 0 & 0 & \dots & \dots \end{bmatrix} \in |\mathscr{C}| \times |\mathscr{C}+1|$$

$\lambda(\cdot)$ is the boundary flow function given the following equation:

$$\lambda(y(x,t)) = \min\left\{ S_e(\rho(x,t)),\ R_e(\rho(x+1),\ t) \right\}$$

where $S_e(\cdot)$ and $R_e(\cdot)$ denote the maximum sending and receiving functions accordingly:

$$S_e\left(\bar{\rho}\right)=\begin{cases} v_f\cdot\bar{\rho}, & \bar{\rho}<\rho_c \\ q_{max}, & \bar{\rho}\geq\rho_c \end{cases}$$

$$R_e\left(\bar{\rho}\right)=\begin{cases} q_{max}, & \bar{\rho}<\rho_c \\ w\cdot(\bar{\rho}-\rho), & \bar{\rho}\geq\rho_c \end{cases}$$

Figure 2.4 is an illustration of the fundamental diagram as well as the corresponding maximum receiving and sending functions. ρ_c is the critical traffic density. In this study, we use a triangular fundamental diagram with free-flow speed v_f and shockwave speed w.

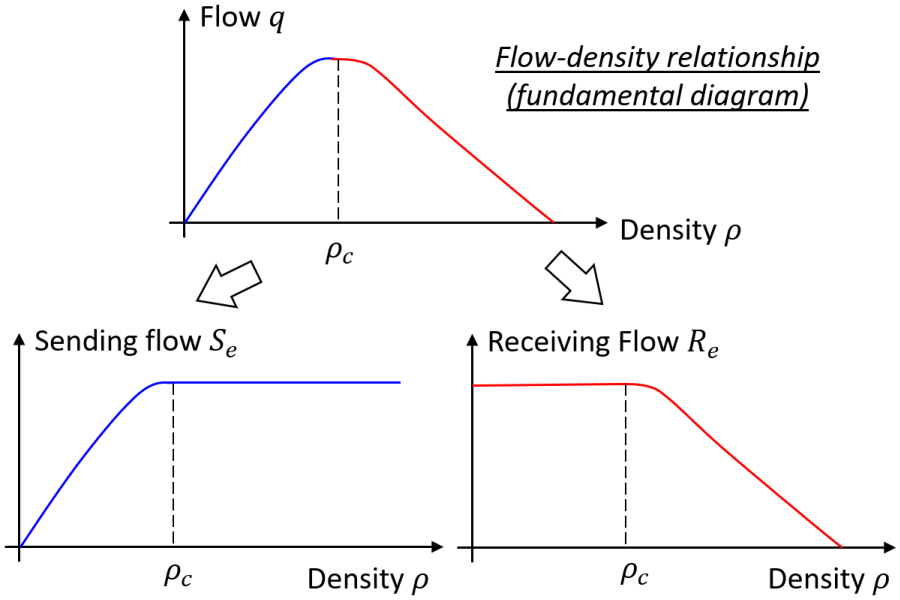


Figure 2.4: Fundamental diagram, maximum sending and receiving functions

The covariance dynamics is given by the following equation:

$$\frac{d\boldsymbol{\Psi}(t)}{dt}=\mathbf{D}(t)\,\boldsymbol{\Psi}(t)+\boldsymbol{\Psi}(t)\,\mathbf{D}(t)^{T}+\mathbf{B}\boldsymbol{\Gamma}(t)\,\boldsymbol{\Gamma}(t)^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}$$

where matrix **D** is determined by:

$$\mathrm{D}_i=\frac{1}{l_i}\left[\cdots\ \frac{\partial\lambda\left(\bar{y}(i-1,t)\right)}{\partial\bar{\rho}(i-1,t)}\left(\frac{\partial\lambda\left(\bar{y}(i-1,t)\right)}{\partial\bar{\rho}(i,t)}-\frac{\partial\lambda\left(\bar{y}(i,t)\right)}{\partial\bar{\rho}(i,t)}\right)-\frac{\partial\lambda\left(\bar{y}(i,t)\right)}{\partial\bar{\rho}(i+1,t)}\ \cdots\right]$$

and the partial derivation of $\lambda(\,y(\,a,b)\,)$ within this matrix **D** is given by the following:

$$\frac{\partial \lambda}{\partial a} = \begin{cases} 0, & S_e(a) \geq R_e(b) \\ v, & S_e(a) < R_e(b) \end{cases}$$

$$\frac{\partial \lambda}{\partial b} = \begin{cases} w & S_e(a) \geq R_e(b) \\ 0 & S_e(a) < R_e(b) \end{cases}$$

The matrix Γ in the covariance dynamics is determined by:

$$\Gamma(t) = \begin{bmatrix} \sqrt{\lambda\left(\bar{y}(0,t)\right)} & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \sqrt{\lambda\left(\bar{y}(|\mathscr{C}|,t)\right)} \end{bmatrix} \in |\mathscr{C}+1| \times |\mathscr{C}+1|$$

Although this stochastic traffic flow model seems complicated, the general idea is simple. It uses the traffic density of each cell $\boldsymbol{\rho}(t)$ as the overall traffic model shows us how its mean value and covariance change over time.

2.3.3 Observation model

Since the stochastic traffic flow model is a Gaussian approximation. We will be able to use the more efficient Kalman filter or its variant if the observation is also Gaussian. The mathematical formulation of a Gaussian observation model can be written as:

$$\boldsymbol{y}_t^i | \, \boldsymbol{\rho}_t \sim \mathcal{N}\left(\mathbf{H}_t^i \boldsymbol{\rho}_t, \boldsymbol{\Sigma}_t^i\right), \quad \forall i$$

where the superscript i is the index of the observation. There will be multiple observations that come from different resources. $\boldsymbol{y}_t^i$ denotes the number of observed vehicles in each cell. $\boldsymbol{\rho}_t$ represents the traffic density of each cell. $\mathbf{H}_t^i$ is called the observation matrix and $\boldsymbol{\Sigma}_t^i$ is the covariance matrix quantifying the uncertainty of this observation.

Figure 2.5 is an illustration of the observation model. In this case, it is assumed that we have a road-side detector that can observe all vehicles within cell 3 and cell 4. However, instead of having the accurate location of each vehicle, the location is given by a uniform distribution. As shown in the illustrated case, we have the entire vehicle on the right in cell 4 while the other vehicle is at the boundary between cell 3 and cell 4. For the vehicle at the boundary, we have probability ¾ that it is in cell 3 while ¼ it is in cell 4. Given this observation, Figure 2.5 also provides the mathematical formulation. In this case, the observation matrix H is determined by:

$$\mathbf{H} = l \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in R^{2\times 4}$$

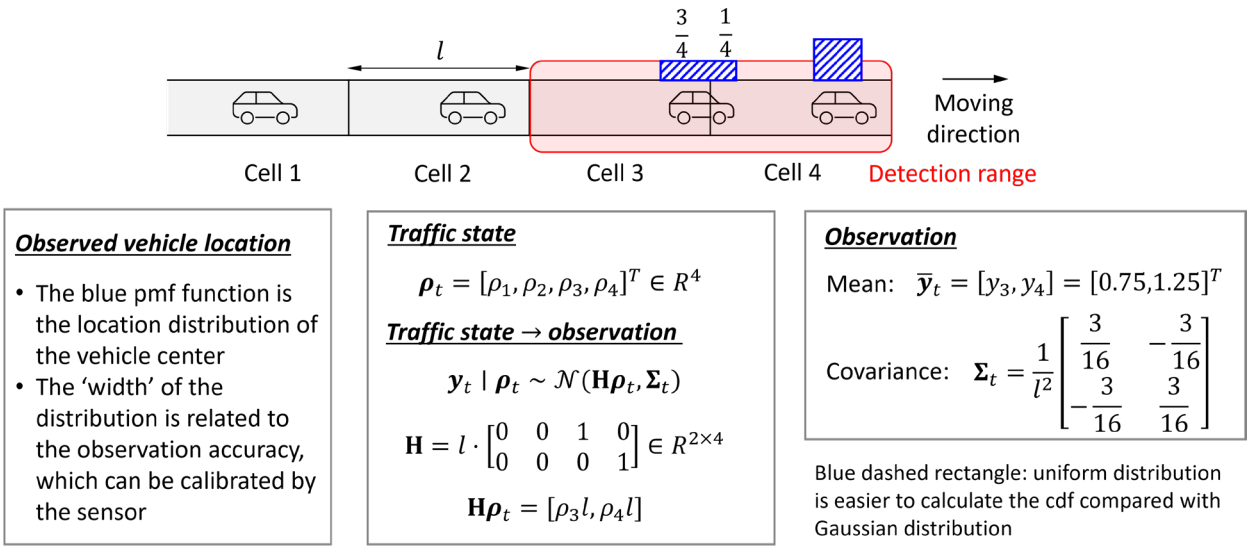


Figure 2.5: Example of the observation model

The reason we call it observation matrix is that it shows us which part of the overall traffic state can be observed. For the observed mean and covariance, we have:

$$\bar{y}_t = \left[y_3, y_4\right] = \left[0.75,\ 1.25\right]^T$$

$$\Sigma_t = \frac{1}{l^2}\begin{bmatrix} \frac{3}{16} & -\frac{3}{16} \\ -\frac{3}{16} & \frac{3}{16} \end{bmatrix}$$

The diagonal of the covariance matrix is variance of the number of vehicles in each cell while other entries denote the covariance. In this case, the covariance is negative since the number of vehicles in cell 3 and cell 4 are negatively correlated with each other: the vehicle at the boundary is in either cell 3 or cell 4 such that their summation is a constant.

Figure 2.5 shows an example of a fixed-location observation, which is referred to as stationary observer. In this study, we focus on the observation coming from the automated vehicle, which is essentially a moving observer. Figure 2.6 is an illustration of the observation from the automated vehicles. Compared with the stationary observer in Figure 2.5, the observable area will change while the automated vehicle moving in the roadway. Nevertheless, the mathematical formulation of the observation model is similar: instead of a stationary observation matrix **H** for stationary observer, the observation matrix **H** for automated vehicle will need to change over time and cover the area within the detection range.

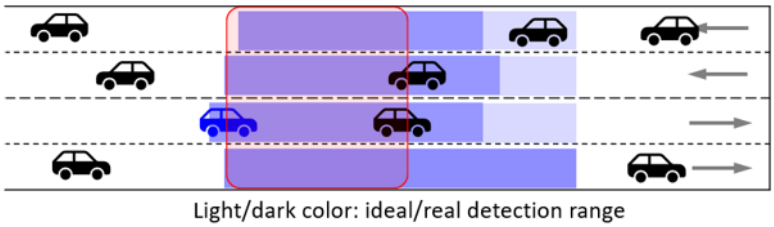


Figure 2.6: Observation from the automated vehicles

To establish the observation model for the automated vehicle, the time-varying observation matrix should indicate the location within the detection range. One potential issue that will lead to an inaccurate estimation is the block of the sight as illustrated by Figure 2.6. The light blue color in the figure shows the ideal detection range if it is not blocked by the background traffic while the dark blue color, a subset of the light color, shows the actual detection range excluding the blocked area. It could lead to an underestimation of overall traffic if the detection range is not chosen properly, and a vehicle blocked by the background traffic is ignored. In practice, it will be troublesome to model the effective detection range in real time. Here we come with a method to avoid bothering by this issue by ***using a truncated observation region*** illustrated by the red block in Figure 2.6. Instead of using the full detection range which could be larger, we only utilize a subset which is near to the automated vehicles. The sight block will not be an issue when it is close to the automated vehicle.

This simplification will not be able to fully utilize the observation from the automated vehicle but will significantly simplify the observation model without considering the sight block issue. Besides, as aforementioned, the most useful information comes from the opposing traffic, we will not lose much as long as the opposing traffic is not missed. Based on this intuition, we can come up with a simple criterion for the minimum length of the truncated region. Let Δt be the sample time and v_{max} be the maximum speed for both directions, the length of the truncated region L should be:

$$L > 2v_{max}\Delta t$$

such that no opposing traffic will be missed between two adjacent sample times.

Here we will not provide the details of the mathematical formulation of the observation model. It will be similar to the example in Figure 2.5. The main difference is that the observation matrix should reflect the truncated region illustrated by the red block in Figure 2.6. Even if the sight blocking will be mitigated by using the truncated region, there are still bad cases that cannot be ignored. Figure 2.7 is an illustration of two special cases that need additional considerations. For case A, if the opposing traffic is full of stopped vehicles, the vehicle in the faraway lane will be likely blocked by the vehicles in the nearby lane. In this case, we will assume that both lanes will be full of stopped vehicles if one of the lanes is occupied. For case B, if the automated vehicle is within the queueing

area, its sight will also be severely blocked by the surrounding stopped vehicles. In this case, we will directly assume this automated vehicle observes nothing. It will become a connected vehicle without surrounding observation.

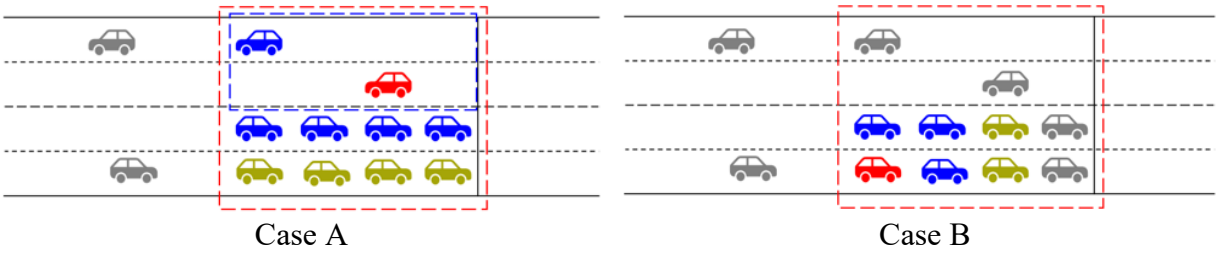


Figure 2.7: Special cases for the observation from automated vehicles

Other than the observation model for the automated vehicle, there are two another domain information that can be applied including the stop event and free-flow event. Figure 2.8 is an illustration of the stop event. It will be jam density between two observed stopped vehicles. The red light will also be regarded as a dummy stopped vehicle. Similarly, if one observed vehicle has a speed which is larger than a certain threshold, this vehicle will be considered as a free-flow vehicle and the density of the corresponding cell will be the critical density. The stop event and free-flow event can significantly improve the estimation accuracy, particularly for connected vehicles.

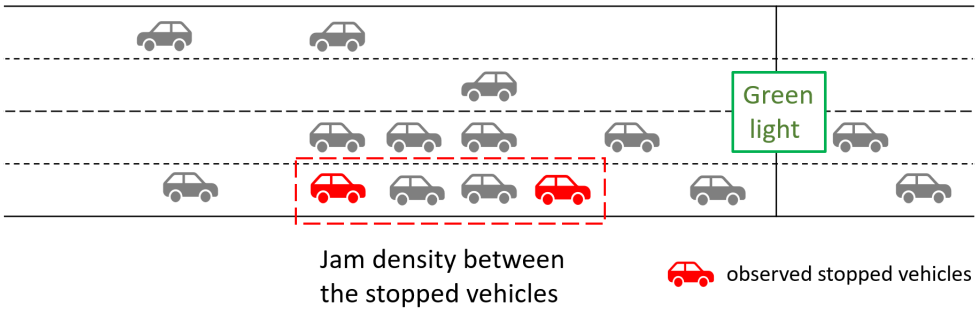


Figure 2.8: Stop event

Figure 2.9 is an illustration of the overall observation model including 1) automated vehicle detection; 2) free-flow event; and 3) stop event. If the vehicle is an automated vehicle with surrounding observations, we will apply both AV detection and the stop event while ignoring the free-flow event. In this case, the free-flow event will be a subset of the AV detection as illustrated in the figure below. If the vehicle is only a connected vehicle without surrounding observations, we will apply the free-flow event and stop event. This completes the observation model for both connected and automated vehicle observations. We will also have the

complete formulation for the hidden Markov model. Since both the stochastic traffic flow model and the observation model are Gaussian, a Kalman filter can be used to estimate the hidden state. The Kalman filter is a standard algorithm, please refer to Welch and Bishop [2] for more details.

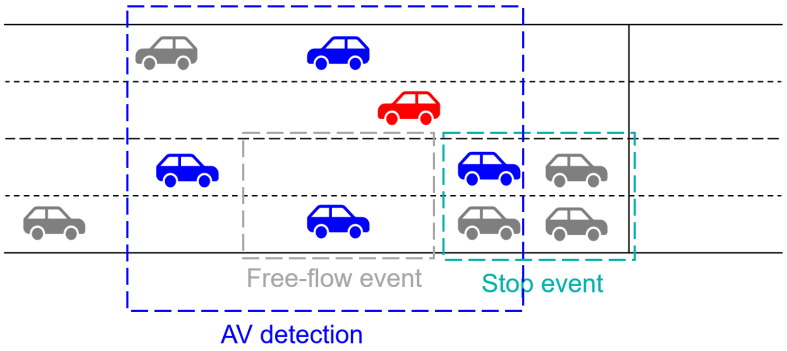


Figure 2.9: Overall estimation model: 1) AV detection, 2) stop event, and 3) free-flow event

2.4 Experiment results

2.4.1 Simulation setup

We test the proposed methods in the SUMO simulation environment. The roadway has two directions, and each direction has two lanes. There is a signalized intersection at the center of the roadway. Connected and automated vehicles are generated according to a certain probability, i.e., penetration rate. Here we assume they are either connected or automated, which means that there will not be both at the same time.

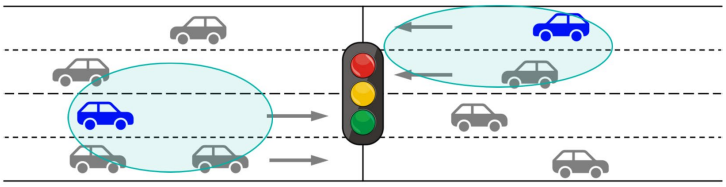


Figure 2.10: Simulation environment setup

2.4.2 AV observation model

We test the proposed methods in the SUMO simulation environment. The roadway has two directions, and each direction

SUMO itself is a traffic simulator without automated vehicle, not to mention any observation model. We design our own automated vehicle observation model as illustrated by Figure 2.11. The basic idea is that a background vehicle can be observed by the automated vehicle. if and only if it is within the detection range and not blocked by other vehicles. Figure 2.11 show the

bird-eye view and each vehicle is assumed to be a rectangular area. We ignore the height of the vehicle. All background vehicles can be projected to the polar coordinates system. The horizontal and vertical distance will be s and h accordingly. The distance will be:

$$r = \sqrt{s^2 + h^2}$$

The range of the angle will be $\Psi_A = [\alpha, \beta]$ where α and β are determined by:

$$\alpha = \operatorname{atan}\frac{2h - w}{2s + b}, \quad \beta = \operatorname{atan}\frac{2h + w}{2s - b}$$

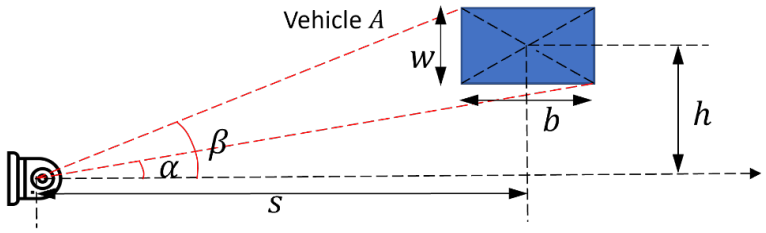


Figure 2.11: Added automated vehicle observation model in SUMO

For a specific automated vehicle, let r_i and Ψ_i be the distance and angle range of the background vehicle i in the polar coordinates centered by the automated vehicle. The following procedure is used to find observable vehicles for a given automated vehicle:

(1) Sort the vehicle (within the detection range) by distance to the sensor ($r_1 \leq r_2 \leq \cdots \leq r_N$)

(2) Initiation and preparation: $\Phi_1 = \Phi_0$, Ψ_i, $i = 1,2, \ldots N$

(3) Iterate for each $i = 1, 2, \ldots, N$:

- If $|\Phi_{i-1} \cap \Psi_i| \geq \psi_m$, set vehicle i as observable, otherwise not.
- Update the occupied angle range: $\Phi_i = \Phi_{i-1} - \Psi_i$

The general idea of this algorithm is to first sort the vehicle according to the distance to the automated vehicle. For all these vehicles within the detection range, starting from the closest vehicle, the proposed algorithm above finds the intersection between the available angle range Φ and the angle range of this vehicle Ψ_i. If the intersection is larger than a certain threshold ψ_m, this vehicle will be labeled as an observable background vehicle; otherwise, it will be blocked. At last, we subtract Ψ_i from the overall observable range Φ since it will be occupied. In this way, we will be able to find all the observable background vehicles that are not blocked for each automated vehicle.

2.4.3 Numerical example of input data

Figure 2.12 is an example of the SUMO simulation with the implemented automated vehicle observation model. The red vehicle denotes the automated vehicle while the blue vehicle

denotes the observable background vehicles. The yellow vehicle is a normal vehicle that cannot be observed. In this case, we assume that the automated vehicle can only see the vehicle in front of it. This could be changed by using a different initial observable angle range Ψ_o. The current Ψ_0 is set as $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The right-hand-side figure shows the corresponding time-space diagram. Correspondingly, the red, blue, and yellow colors denote the automated vehicle, observable background vehicle, and unobservable background vehicle.

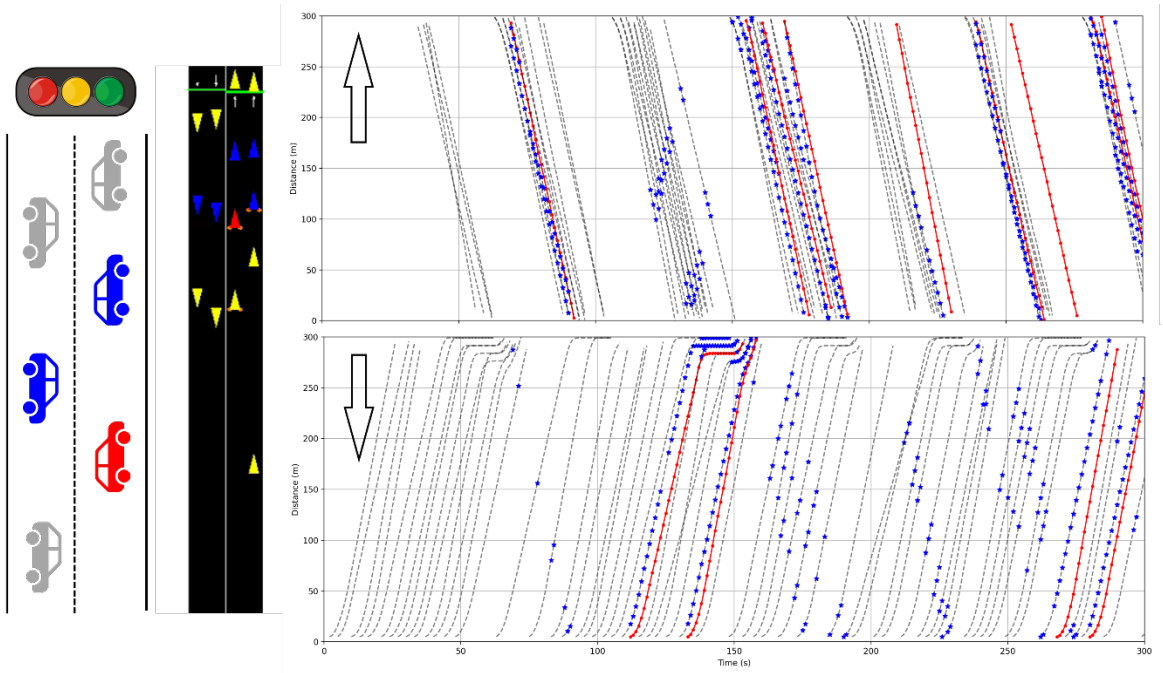
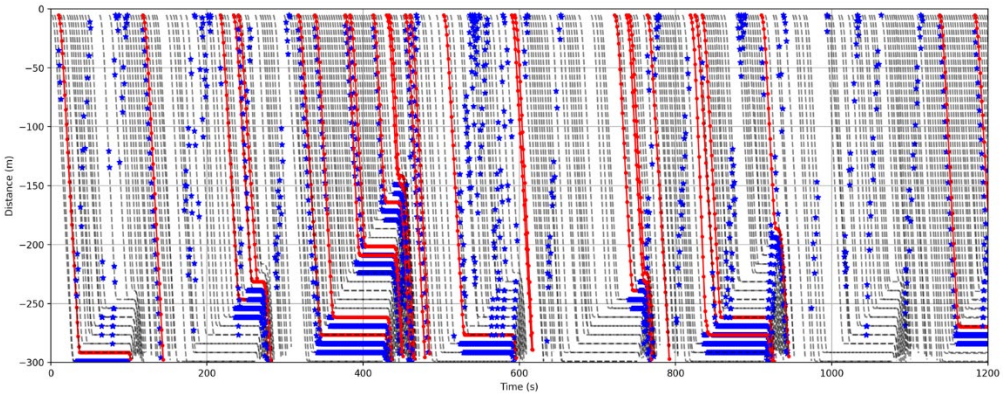
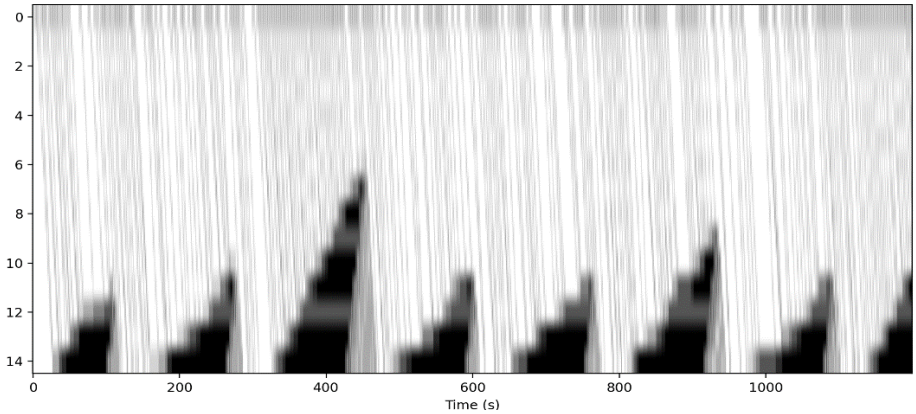


Figure 2.12: SUMO simulation example

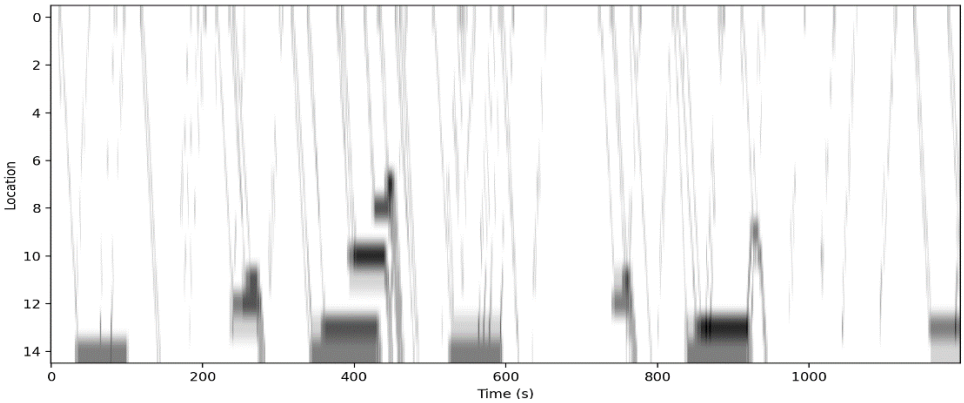
The proposed traffic state estimation method is to estimate the traffic density of each cell in the spatial-temporal space. Therefore, we need to convert the time-space vehicle trajectories to traffic densities. The following figures show converted traffic densities from the time-space diagram. Figure 2.13a shows the original time-space diagram, Figure 2.13b is the corresponding traffic density of overall traffic, while Figure 2.13c is the traffic density of observed traffic. The estimation algorithm is to reconstruct overall traffic density (Figure 2.13b) based on the observed traffic density (Figure 2.13c).



a. Time-space diagram



b. Traffic density of overall traffic



c. Traffic density of observed traffic

Figure 2.13: Time-space diagram and the corresponding traffic density diagram

2.4.4 Case studies

Before we introduce the final estimation results, this subsection will show some case studies that can explain how the different observations can contribute to the overall traffic state estimation. Figure 2.14 is an illustration of the traffic state estimation with automated vehicles. This figure uses the same example in the previous subsection, so the ground truth is given by Figure 2.13b. According to the proposed estimation methods, the final estimation result is a

fusion of three different parts: prior stochastic traffic flow model, automated vehicle observations, and the stop event. On the top left is the prior traffic state, which is the same for each cycle since there is no observation with it. The estimation will be improved by adding observations to it.

As shown in Figure 2.14, the traffic density tends to be underestimated only with the automated vehicle while overestimated only with the stop event. The underestimation caused by the AV observation is mainly due to the sight block issue as aforementioned. If one vehicle is blocked but the automated vehicle regards its location to be observable, the traffic density will be underestimated. Nevertheless, the stop event sets a lower bound of the queue length, that is, the queue length is always larger than the last observable stopped vehicle. As a result, only utilizing the stop event will overestimate the traffic density without setting an upper bound on the other side. Eventually, we will get a good estimation by combining both AV observations and the stop event.

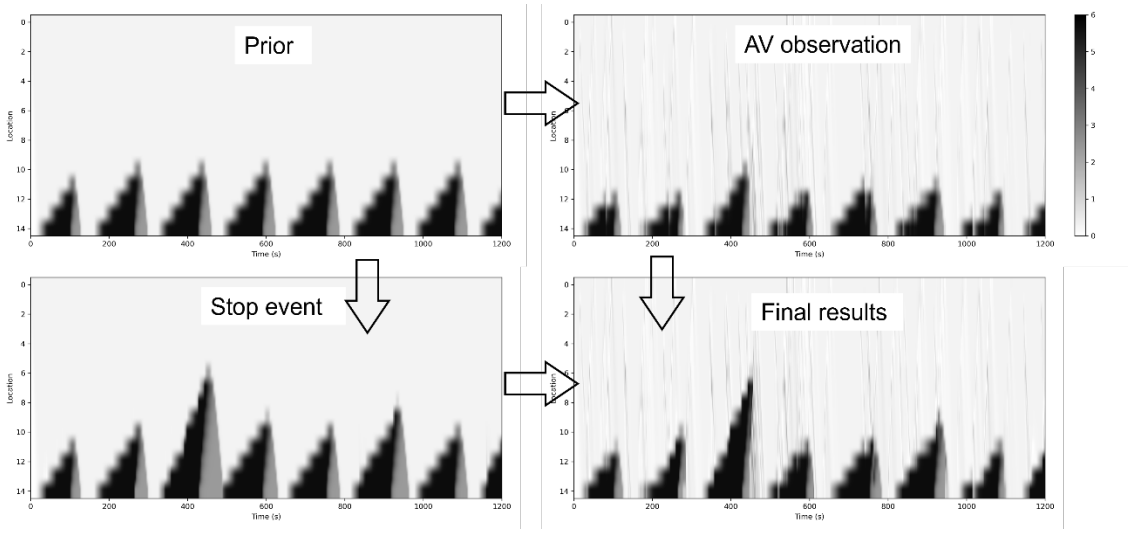


Figure 2.14: Traffic state estimation with automated vehicles: AV observation + stop event

Figure 2.15 is an illustration of the traffic state estimation with connected vehicles. In this case, we will not have the vehicles' observations but only the stop event and the free-flow event. Similarly, the stop event provides a lower bound of the queue length while the free-flow event provides the upper bound of the queue length.

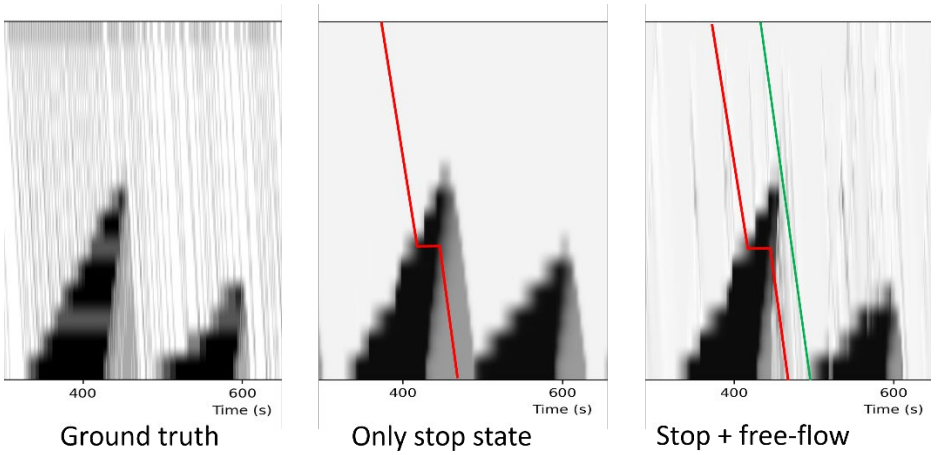


Figure 2.15: Traffic state estimation with connected vehicles: stop event + free flow event

2.4.5 Overall results

Figure 2.16 is the results of the traffic state estimation with different penetration rates of connected and automated vehicles. Here we assume a biased prior with roughly 10% error since the perfect prior is not available in most cases. The metric to evaluate the estimation results is the average of the 2-norm (Frobenius norm) difference of the traffic density at each time.

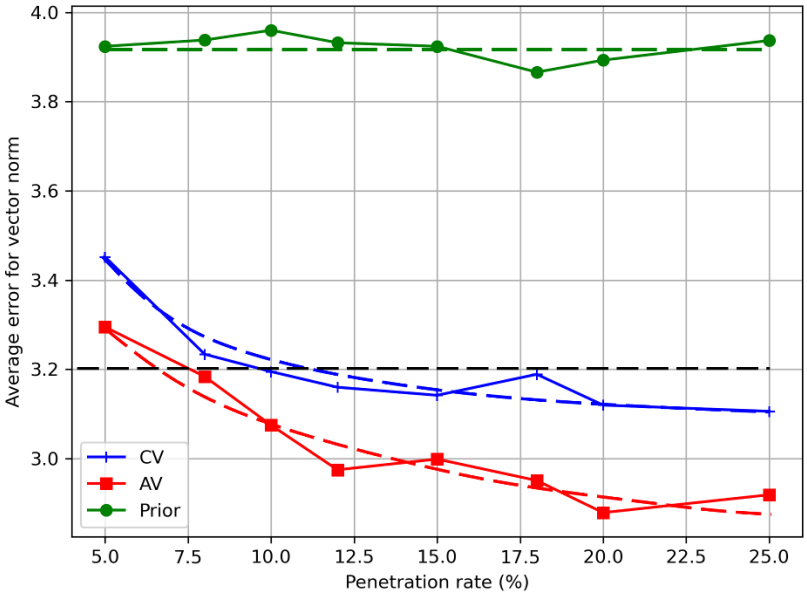


Figure 2.16: Traffic state estimation with different penetration rates of connected and automated vehicles

As shown in the figure, both connected and automated vehicles are helpful to improve the estimation accuracy. At the same penetration rate, the AV observation will have a better performance compared with the connected vehicle, which is consistent with our intuition. Specifically, in this case, a 6% automated vehicle has similar estimation performance with a 12% connected vehicle.

3. Leveraging connected and automated vehicles to influence day-to-day traffic dynamics

3.1 Introduction

Compared to traditional personal vehicles, one crucial distinction of driving automation is that it requires travelers to relinquish part of their travel agency, referring to the capacity to make decisions based on their personal volition. In the early stages of development, driving automation requires travelers to surrender their agency over driving maneuvers, including velocity control and trajectory planning. Utilizing this relinquished agency, we can employ CAVs as traffic stream regulators. For instance, Wu et al. [5] focused on ensuring the string stability of mixed traffic within a single-lane system by employing AVs. Ge and Gabor [6] proposed an optimal connected cruise controller to enhance the performance of mixed traffic, later combined with an estimation method for human drivers [7]. Additionally, Cicic [8] investigated the coordination of CAV platoons to mitigate bottlenecks in highway sections by controlling their formation and velocity. These studies collectively demonstrate the potential of CAVs as effective tools to improve road performance in various local traffic scenarios.

As driving automation continues to advance, we envision that travelers are willing to surrender more travel agency, including their control over route and departure time choices. Under such a level of relinquished agency, CAVs can serve as traffic demand distributors, effectively regulating traffic flow throughout a network. Typically, HVs are often considered as User Equilibrium (UE) users, aiming to minimize their individual travel costs. On the other hand, CAVs operating under control are viewed as System Optimal (SO) users, taking into account the overall system cost. Zhang and Nie [9] investigated the optimal ratio of these two user types, striking a balance between improving the mixed equilibrium and maintaining low control intensity. To solve the complexity arising from the bilevel optimization with an equilibrium condition, Sharon et al. [10] and Chen et al. [11] reformulated the problem as a linear program. They successfully determined the minimal control ratio required to achieve the system optimum. Moreover, Chen et al. [11] demonstrated that CAV-based control can be effectively combined with pricing mechanisms to further enhance traffic management.

However, in reality, traffic networks are seldom in an equilibrium state [12]. Instead, the network flow experiences day-to-day evolution as travelers make adjustments to their travel choices. This dynamic behavior necessitates controlling the evolution process rather than solely developing a static demand distributor. To address this issue, one needs a model for the day-to-day traffic dynamics of HVs and a control scheme for CAVs to drive the system to equilibrium. For example, Li et al. [13] assumed the human driver behavior follows a logit-SUE assignment with inertia, and demonstrated that CAVs can be effectively controlled to drive the system toward a desired mixed equilibrium. Guo et al. [14] adopted a different approach, assuming that all vehicles, including CAVs, act as bounded rational agents and are willing to sacrifice their interests only to a certain extent. Under this setting, they adopted the BRUE (Boundedly Rational User Equilibrium)-based routing adjustment process introduced in [15] to model HV behavior, and proposed a routing scheme for CAVs to drive the system to the best BRUE.

Nevertheless, these previous studies do not really support real-world implementation of CAVs as traffic demand distributors. For one thing, previous model-based methods demand perfect information about the underlying day-to-day dynamical process, which is usually not available to traffic management agencies. Moreover, the success of these methods largely hinges on the

monotone cost property, which ensures the convergence of day-to-day traffic dynamics. However, it is crucial to recognize that the real-world flow dynamics do not necessarily converge to equilibrium, especially when travelers involve departure time choices, as pointed out in [16].

In contrast, in this chapter, we propose a distributed, model-free approach to derive an optimal control policy of controlling a fraction of CAVs to improve the system performance over an infinite horizon. The proposed approach can better support the implementation of CAVs as traffic demand distributors as it does not rely on exhaustive knowledge of the underlying day-to-day dynamic process. In addition, the control policy will be implemented at the individual level, instructing individual CAVs to act based on their local information. Besides being model-free and distributed, the proposed scheme also significantly differs from [13] and [14] in terms of the control objective. Instead of pushing the process to a desired state, we focus on minimizing the total system cost induced by the day-to-day dynamical process. Therefore, unlike previous approaches, our method is flexible on the choice of day-to-day dynamical models, irrespective of their convergence property.

To present our approach, we first consider a scenario with homogeneous travelers and model the problem as a finite agent control problem. However, it becomes intractable due to the large number of travelers. To overcome this limitation, we shift our focus to the limiting case with an infinite number of travelers and formulate the problem within the major-minor mean field control (MFC) framework [1]. Each CAV forms a minor agent, which has its state and action, while the entire HVs (and uncontrolled CAVs) are aggregately modeled by a single major agent. Furthermore, we discuss how to extend the model to accommodate traveler heterogeneity, broadening its applicability to real-world scenarios. We then leverage reinforcement learning algorithms to compute the optimal control policy.

Our model-free and distributed control scheme offers a versatile and adaptable solution, capable of handling a range of scenarios, including departure time and route choices, while effectively accommodating different levels of CAV penetration rates. This broad applicability makes it a valuable tool for enhancing traffic management and optimizing day-to-day traffic dynamics in mixed traffic scenarios. The remainder of this chapter is structured as follows. Section 2.2 presents the model. Section 2.3 discusses the control algorithm and section 2.4 presents numerical examples. Lastly, section 2.5 concludes the chapter.

3.2 Model

In this section, we present a comprehensive model that encompasses various travel choices, including route choices, departure time choices, or both. To better illustrate the proposed method, we start by introducing a simplified case where all travelers are homogeneous. Subsequently, we discuss the extension of the model to accommodate heterogeneous travelers.

3.2.1 Finite-agent control model

Consider a transportation system with multiple travelers, consisting of N controllable CAVs and M uncontrollable vehicles (either HV or uncontrolled CAVs). To distinguish whether a traveler prioritizes their individual interests or the overall system's efficiency, we refer to the controllable CAVs as system users and the uncontrollable vehicles as selfish users.

We model the process of sequentially choosing travel options as a Markov decision process. One's travel choice on a given day is considered as their state at that time. Let $x_t^i \in \mathcal{X}$

represents the travel choice of system user $i \in [N] := \{1, \ldots, N\}$ on day t, where $\mathcal{X}$ is the finite set of all allowable travel choices. For example, the set $\mathcal{X}$ corresponds to the path set in route choice scenarios. Similarly, $s_t^j \in \mathcal{X}$ denotes the travel choice of selfish user $j \in [M]$ on day t. Each traveler's state contributes to an empirical distribution over the entire population. We denote the penetration rate of system users as $\theta = \frac{M}{M+N}$.

For the two groups of travelers, we define the empirical distributions μ_t^N and ν_t^M as follows:

$$\mu_t^N = \frac{1}{N} \sum_{i \in [N]} \delta_{x_t^i}$$

$$\nu_t^N = \frac{1}{M} \sum_{j \in [M]} \delta_{s_t^j}$$

where the superscripts, M and N, are used to clarify that we are dealing with a finite agent model. To provide an illustrative example, μ_t^N is equivalent to the path flow for system users in route choices.

For selfish users, we assume that they implicitly follow certain day-to-day traffic dynamics, which may not necessarily be revealed to the management agency. Therefore, selfish users can be modeled aggregately as a major agent, whose day-to-day dynamical model is denoted as $\nu_{t+1}^M \sim q(\cdot\,|\mu_t^N, \nu_t^M)$, where selfish users' behavior on day $t+1$ is determined by their experience in the previous day, which is further dictated by μ_t^N and ν_t^M.

On the contrary, since each system user has their own assignment, they are modeled separately as many minor agents. The action of a minor agent taken on day t is choosing the travel option for the next day, i.e. day $t+1$, which is denoted as $a_t^i \in \mathcal{A}$. In the homogeneous case, we have $\mathcal{X} = \mathcal{A}$. The action a_t^i is sampled from an assignment policy provided by the management agency, denoted as $\pi(\cdot\,|x_t^i, \mu_t^N, \nu_t^M)$. This policy is a function of the system user's current choice x_t^i and the empirical distributions of both groups. Based on the actions selected, the state of the system user evolves according to the transition kernel $x_{t+1}^i \sim p(\cdot\,|x_t^i, a_t^i, \mu_t^N, \nu_t^M)$. The formulation of the transition kernel is flexible. For example, it can be used to capture the following cases:

- Full compliance: system users are perfectly compliant with the assignment. In such case, $p\left(x \,\middle|\, x_t^i, a_t^i, \mu_t^N, \nu_t^M\right) = p\left(x \,\middle|\, a_t^i\right) = \begin{cases} 1, & \text{if } x = a_t^i \\ 0, & \text{otherwise} \end{cases}$
- Partial compliance due to inertia: system users may have reluctance in switching travel choices and prefer to stay on their previous choices. This can be modeled by

$$p\left(x \,\middle|\, x_t^i, a_t^i, \mu_t^N, \nu_t^M\right) = p\left(x \,\middle|\, x_t^i, a_t^i\right) = \begin{cases} 1-\varepsilon, & \text{if } x = a_t^i \\ \dfrac{\varepsilon}{|\mathcal{X}|-1}, & \text{otherwise} \end{cases}$$

- Partial compliance due to self-interests: we can also manipulate the transition kernel to model the behavior considered in [14], where system users are only willing to sacrifice interest within a threshold ϵ

$$p\left(x \middle| x_t^i,\ a_t^i,\ \mu_t^N,\ \nu_t^M\right) = \begin{cases} 1, & if\ x = a_t^i\ and\ a_t^i \in \Omega^{\epsilon\_BR}_{\mu_t^N, \nu_t^M} \\ 0, & if\ x \neq a_t^i\ and\ a_t^i \in \Omega^{\epsilon\_BR}_{\mu_t^N, \nu_t^M} \\ 1, & if\ x = x_t^i\ and\ a_t^i \notin \Omega^{\epsilon\_BR}_{\mu_t^N, \nu_t^M} \\ 0, & if\ x \neq a_t^i\ and\ a_t^i \notin \Omega^{\epsilon\_BR}_{\mu_t^N, \nu_t^M} \end{cases}$$

where $\Omega^{\epsilon\_BR}_{\mu_t^N, \nu_t^M}$ refers to the set of acceptable choices for ϵ-bounded rational travelers.

Denote the travel cost of travel choice x as $c_x(\mu_t^N, \nu_t^M)$. Additionally, we define the system's total travel cost as follows

$$C\left(\mu_t^N,\ \nu_t^M\right) = \sum\nolimits_{x \in \mathcal{X}} \left(N\mu_t^N(x) + M\nu_t^M(x)\right) c_x\left(\mu_t^N, \nu_t^M\right)$$

where $\mu_t^N(x)$ is the proportion of system users that choose travel choice x. It is worth noting that the system's total travel cost can also be expressed as

$$C\left(\mu_t^N,\ \nu_t^M\right) = (N+M) \sum\nolimits_{x \in \mathcal{X}} \left(\theta\mu_t^N(x) + (1-\theta)\nu_t^M(x)\right) c_x\left(\mu_t^N, \nu_t^M\right)$$

which implicitly reflects the impact of the penetration rate θ on the system's total travel cost. Moreover, the penetration rate also affects the transition kernels by empowering the CAVs with higher influence as the penetration rate increases.

Here, we make two assumptions regarding the transition kernel and the system cost, which will be used in later parts.

Assumption 1. The transition kernels $p(\cdot\,|x, a, \mu, \nu)$ and $q(\cdot\,|\mu, \nu)$ are Lipschtiz continuous with respect to μ and ν.

Assumption 2. The system cost function $C(\mu, \nu)$ is Lipschtiz continuous with respect to μ and ν.

These assumptions are mild and are widely used in literature. The control objective of the management agency is to find the optimal policy to minimize the total discounted cost

$$min_{\pi}\ J(\pi) = E\left[\sum\nolimits_{t=0}^{\infty} \gamma^t C\left(\mu_t^N,\ \nu_t^M\right)\right]$$

$$s.t.\quad \nu_{t+1}^M = q\left(\cdot \middle| \mu_t^N,\ \nu_t^M\right)$$

$$a_t^i \sim \pi\left(\cdot \middle| x_t^i, \mu_t^N,\ \nu_t^M\right), \forall i \in [N]$$

$$x_{t+1}^i \sim p\left(\cdot \middle| x_t^i,\ a_t^i, \mu_t^N,\ \nu_t^M\right), \forall i \in [N]$$

3.2.2 Major-minor mean field control model

A limitation of the model in the above subsection is that as the number of agents grows large, finding the optimal control policy becomes intractable [17]. This complexity is especially evident in the application of participatory control, where the number of controlled CAVs is

large, even at low penetration rates, due to the high vehicle ownership. As a result, scalable methods are essential to efficiently solve the model and overcome the computational challenge.

To address this issue, we adopt the mean field approximation by considering the limiting case with an infinite number of travelers. This approximation technique was first proposed in mean field games [18], [19], which leverages the "smoothing" effect to simplify the model and avoid complex mutual interactions among agents. Under the homogeneity assumption, the group of agents can be effectively abstracted by a representative agent. By employing the mean field approximation, we reformulate the model within the major-minor mean field control (MFC) framework proposed by Cui et al. [1], [20].

To be more specific, consider when N and $M \to \infty$, under the law of large numbers, the empirical distribution μ_t^N and ν_t^M becomes the mean field (MF) distribution $\mu_t, \nu_t \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ refers to the set of all probability mass functions defined on the state space. With this mean field approximation, we no longer need to track the state of each individual system user. Instead, we can only focus on the representative agent, whose state is now treated as a random variable whose distribution matches the MF distribution. To maintain consistency, we continue to use the same notations as before. Specifically, $q(\cdot\,|\mu_t, \nu_t)$ represents the transition kernel for selfish users, while $p(\cdot\,|x, a, \mu_t, \nu_t)$ denotes the transition kernel for system users. The assignment policy from the management agency and the system cost are denoted as $\pi(\cdot\,|x, \mu_t, \nu_t)$ and $C(\mu_t, \nu_t)$ respectively.

Due to the homogeneity assumption, every system user follows the same transition kernel. Consequently, its mean field (MF) distribution evolves deterministically according to the following equation

$$\mu_{t+1} = \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} p\left(\cdot \,\middle|\, x, a, \mu_t, \nu_t\right) \mu_t(x)\, \pi_t(a|x, \mu_t, \nu_t)$$

Here, $\mu_t(x)\pi_t(a|x, \mu_t, \nu_t)$ corresponds to the proportion of system users that choose the state-action pair (x, a) on day t. By introducing $\pi_t(\mu_t) = \pi_t(\cdot|\,\cdot, \mu\_t, \nu\_t)$, we use $\mu_t \otimes \pi_t(\mu_t)$ to denote the joint distribution of states and actions, where $\otimes$ refers to the element-wise product. Then, we can express the evolution of μ_t as a function

$$\mu_{t+1} = T(\mu_t, \nu_t, \mu_t \otimes \pi_t(\mu_t))$$

In this sense, we can consider the entire population as a whole, aggregating their behavior using the new state (μ_t, ν_t). The overall assignment is considered as a new action $h_t = \mu_t \otimes \pi_t(\mu_t) \in \mathcal{H}(\mu_t)$, where $\mathcal{H}(\mu_t)$ refers to the joint distribution whose state marginal matches μ_t. Consider the action h_t being sampled from a new policy $\hat{\pi}$, then the system is transferred to the following problem of a single-agent MFC MDP

$$\min_{\hat{\pi}} J(\hat{\pi}) = E\left[\sum_{t=0}^{\infty} \gamma^t C\left(\mu_t, \nu_t\right)\right]$$

$$s.t. \quad \nu_{t+1} = q\left(\cdot \,\middle|\, \mu_t, \nu_t\right)$$

$$h_t \sim \hat{\pi}\left(\cdot \,\middle|\, x_t, \mu_t, \nu_t\right)$$

$$\mu_{t+1} = T\left(\mu_t, \nu_t, h_t\right)$$

For the new model, we have the following proposition

Proposition 1. [Theorem B.4 in [1]]

Under Assumption 1 and 2, the MFC MDP always exists an optimal stationary policy $\widehat{\pi}$.

3.2.3 Relaxing homogeneity assumption

Under the homogeneity assumption, the control problem was formulated as a major-minor MFC model in the previous subsection. However, it is important to acknowledge that travelers in real-world transportation systems are often inhomogeneous. Thus, we need to relax this assumption to account for heterogeneity in travel choices.

One way to address this issue is to consider different state spaces between travelers. In the previously discussed model, the state (i.e., travel choice) of all travelers was assumed to be from a uniform state space. However, in practical scenarios such as routing, travelers can only choose from a specific set of route choices corresponding to their origin-destination (OD) pairs.

To accommodate this heterogeneity, we classify agents into J types, where each type $j \in \mathcal{J} = \{1, \ldots, J\}$ has its own state space denoted as $\mathcal{X}^j$. For ease of notation, we introduce the function $J(x)$ to indicate the type of state $x \in \mathcal{X}$. Specifically, if x belongs to the state space $\mathcal{X}^j$, then $J(x) = j$. For instance, in routing scenarios, $\mathcal{X}^j$ represents the path set of OD pair j. The overall state space is defined as $\mathcal{X} = \bigcup_{j \in \mathcal{J}} \mathcal{X}^j$. The action space is set as $\mathcal{A} = \mathcal{X}$, and we also use $J(a)$ to denote the type of action $a \in \mathcal{A}$ with minor abuse of notation.

In this extended model, it is possible for an action executed by an agent to not belong to their type. To handle this situation, we introduce the new transition kernel $\hat{p}(x'|x, a, \mu, \nu)$, which satisfies the following condition

$$\widehat{p}(x' \mid x, a, \mu, \nu) = \begin{cases} p(x' \mid x, a, \mu, \nu), & \text{if } J(x) = J(a) = J(x') \\ 1, & \text{if } J(x) \neq J(a),\ x = x' \\ 0, & \text{otherwise} \end{cases}$$

which ensures that an action that does not belong to the type of the current state is considered invalid or ineffective. In such cases, the agent is not impacted by the invalid control actions, and the state remains unchanged. However, if the control action is valid, the transition is the same as in the previous model. Intuitively, it is desirable for the optimal policy π to ensure that actions executed for every state are valid, allowing the control effect to be maximized while respecting the heterogeneity of travelers. Additionally, we require the transition kernel for selfish users, $q(\cdot\,|\mu, \nu)$, to be valid. Specifically, if $q(\nu'|\mu, \nu) > 0$, it must satisfy $\sum_{x \in \mathcal{X}} \nu'(x) = \sum_{x \in \mathcal{X}^j} \nu(x)$ for all types j, and for all μ, ν. Note that the validity of q is typically ensured since it is a model-based exogenous kernel, for example, derived from Smith's dynamic. Therefore, no explicit revisions are needed to enforce this validity.

By revising the transition kernel, the heterogeneous model with different state spaces can be equivalently represented by the previous homogenous model with the overall state space $\mathcal{X}$. The only difference is that now the system user's state evolves according to the revised transition kernel $\hat{p}$. As a result, the consideration of agent types becomes unnecessary.

Everything is taken care of by the extended state space and revised transition kernel, which results in the following optimal control problem

$$min_{\pi}\ J(\pi) = E\left[\sum_{t=0}^{\infty}\gamma^t C\left(\mu_t^N,\ \nu_t^M\right)\right]$$

$$s.t.\quad \nu_{t+1}^M = q\left(\cdot\ \middle|\ \mu_t^N,\ \nu_t^M\right)$$

$$a_t^i \sim \pi\left(\cdot\ \middle|\ x_t^i, \mu_t^N,\ \nu_t^M\right), \forall i \in [N]$$

$$x_{t+1}^i \sim \hat{p}\left(\cdot\ \middle|\ x_t^i,\ a_t^i, \mu_t^N,\ \nu_t^M\right), \forall i \in [N]$$

As before, letting $N, M \to \infty$, we can get the limiting MFC model

$$min_{\hat{\pi}}\ J(\hat{\pi}) = E\left[\sum_{t=0}^{\infty}\gamma^t C\left(\mu_t,\ \nu_t\right)\right]$$

$$s.t.\quad \nu_{t+1} = q\left(\cdot\ \middle|\ \mu_t,\ \nu_t\right)$$

$$h_t \sim \hat{\pi}\left(\cdot\ \middle|\ x_t, \mu_t,\ \nu_t\right)$$

$$\mu_{t+1} = \widehat{T}\left(\mu_t,\ \nu_t\right)$$

where $\widehat{T}(\mu_t, \nu_t, h_t) = \sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}}\hat{p}(\cdot\ |x, a, \mu, \nu)\,\mu_t(x)\,\pi_t(a|x, \mu_t, \nu_t)$.

Note that the new transition kernel also has the Lipschitz continuity

Proposition 2. If Assumption 1 holds for every type, then the transition kernel $\hat{p}$ is Lipschitz continuous with respect to μ and ν.

Proof. For each x, a, x' , if $J(x) = J(a) = J(x')$, then $\hat{p}(x'|x, a, \mu, \nu) = p(x'|x, a, \mu, \nu)$. Since kernel p, $\hat{p}$ is also Lipschitz continuous in this case. Otherwise, $\hat{p}(x'|x, a, \mu, \nu)$ is a constant, whose value is fixed regardless of the distribution μ and ν. In such cases, the Lipschitz continuity naturally holds.

This proposition further leads to the existence of the optimal policy of the new model

Proposition 3. If Assumption 2 holds for every type, then the MFC MDP with heterogeneity in state spaces always exists an optimal stationary policy $\hat{\pi}$.

3.3 Algorithm

So far, we have obtained a single-agent MDP by formulating the problem as a major-minor MFC model, and we have successfully proved the existence of the optimal policy under mild conditions. However, to practically solve for the optimal policy, we need an algorithm that meets certain criteria. Specifically, the solution algorithm should be decentralized, model-free, and applicable to finite agent cases, as the presence of an infinite number of travelers is unrealistic in reality. To address these requirements, we turn to the MFC reinforcement learning (RL) approach based on the work in [1], which is outlined as follows

Algorithm 1. MFC-RL algorithm framework

Input: Initialize policy $\hat{\pi}^{\theta}$

1: **for** iterations $n = 1,2, \ldots$ **do**

2: Sample MFC action $h_t \sim \hat{\pi}^{\theta}\left(\cdot \middle| \mu_t, \nu_t \right)$

3: Retrieve individual policy π_t from h_t

4: **for** minor agent $i = 1, \ldots, N$ **do**

5: Sample and execute action $a_t^i \sim \pi_t\left(\cdot \middle| x_t^i \right)$

6: **end for**

7: Observe system cost C_t, next MF distributions μ_{t+1}, ν_{t+1}

8: Update policy $\hat{\pi}^{\theta}$

9: **end for**

Since the MFC action h_t represents the joint distribution of states and actions, it can be used to recover the original individual policy π_t, as mentioned in Line 3. Note that the individual policy requires only local information, such as the current state, rather than global information like the MF distribution. Consequently, the algorithm lies in the paradigm of centralized training decentralized execution (CTDE) [21]. In the CTDE paradigm, the management agency broadcasts the individual policy π_t to all system users, and each system user independently selects its own action. This design effectively relieves the management agency from the burden of managing individual assignments, significantly reducing the computation complexity of the process. Moreover, the MFC framework does not require knowledge regarding the specific transition kernels $p$and q. The transition process is essentially induced by the accumulated effect of individual behaviors, and as an observer, the management agency can solely observe the results of the transition without explicit knowledge of the underlying transition process. Furthermore, the proposed model exhibits flexibility in the choice of RL algorithms to update the policy (Line 8).

3.4 Numerical examples

In this section, we apply the proposed model and algorithm to two examples: one for route choices and the other for departure time choices. In both cases, we use the discrete-time version of the Smith dynamic [22] to model human behavior. The Smith dynamic assumes that individuals will switch to lower-cost options based on their experience from the previous day, which takes the following form

$$\nu_{t+1}(x) - \nu_t(x) = \eta \sum_{x' \in \mathcal{X}} \left(\nu_t(x') \left[c_{x'}\left(\mu_t, \nu_t\right) - c_x\left(\mu_t, \nu_t\right) \right]^+ - \nu_t(x) \left[c_x\left(\mu_t, \nu_t\right) - c_{x'}\left(\mu_t, \nu_t\right) \right]^+ \right)$$

where $\nu_t(x)$ represents the proportion of individuals choosing travel choice x on day t, $[\cdot]^+ = \max\{0, \cdot\}$, and η captures the effect of user inertia. It is worth noting that the model can be readily applied to other types of dynamical models as well.

For both examples, we set the discount factor γ to 0.99. Although the cost function in the model assumes an infinite horizon, it is impractical to account for an infinite number of days in real-world scenarios. Therefore, we truncate the horizon length to 200 days. Thus, the training process consists of iterations of 200-day episodes. To ensure robustness and adaptability, we initialize the system randomly at the beginning of each episode. It allows

our algorithm to be trained on various scenarios, ensuring the capability to handle diverse cases. We normalize the system cost by dividing it by the average cost of 10 random distributions and set the inertia weight $\eta = 0.02$ to make the value problem-independent. The value of η matches the experiment setting in [23]. In both examples, we use Proximal Policy Optimization (PPO) [24] as the RL algorithm, where the values of hyperparameters are given as follows

Table 3.1: Hyperparameter values

Hyperparameter	Value
GAE lambda	1
KL coefficient	0.01
Clip parameter	0.2
Learning rate	0.00005
Training batch size	24,000
Mini-batch size	4,000
Gradient steps per batch	5

3.4.1 Route choices

The first example considers the Baraess network in Figure 3.1 and follows the experimental settings outlined in [11]. The total demand is 6 from node 1 to node 9. The link travel time of the five links is

$$t_1 = 10v_1$$

$$t_2 = 50 + v_2$$

$$t_3 = 50 + v_3$$

$$t_4 = 10 + v_4$$

$$t_5 = 10v_5$$

where v represents the link flow. In total, travelers have 3 path choices. The path-link relationship is provided in Table 3.2. As demonstrated in Chen et al. [11], the minimal control ratio of this network is 1, indicating that the network is highly challenging to regulate.

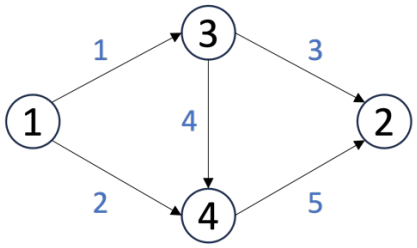


Figure 3.1: Braess network

Table 3.2: Path-link relationship

Path	Link
1	1, 3
2	2, 5
3	1, 4, 5

In this example, we investigate three different system user penetration levels: $\theta = 0$, 50%, and 100%. To evaluate the algorithm's performance, we use the total undiscounted cost sum over the 200-day horizon. The training curves of the algorithm under the three penetration levels are illustrated in Figure 3.2. The orange curve represents the baseline, which corresponds to the scenario with pure human response under the Smith dynamic. Due to random initialization, there might be slight fluctuations in the curve. To facilitate comparison, we normalize its average to 1. The blue curve corresponds to the scenario where the management agency has control over all vehicles. In this case, the trained algorithm converges to the theoretical lower bound, represented by the red dotted curve. The theoretical lower bound corresponds to the scenario where the system achieves and maintains the SO flow from the second day onward. The training results demonstrate that the proposed method has the potential to fully harness the control power of the system users, achieving an optimal and efficient traffic flow. The green curve represents the penetration rate at 50%. The training result falls between the other two experiments, showing a trade-off between the control capabilities and the control intensity.

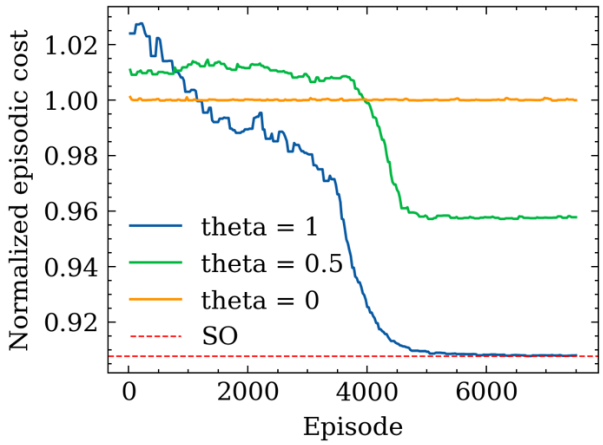


Figure 3.2: Training curve of the routing experiment

To visually demonstrate the trained policy, we conducted 10 experiments with a 100% penetration level of system users, and we implemented the trained policy on the system, tracking the path flow evolution. The results are plotted in Figure 3.3, where the dark curve represents the average path flow, and the light colors represent the range of variation. From the figure, it is evident that, regardless of the initial path flow distribution, the path flow on paths 1 and 2 roughly converges to 3 on the second day and maintains that level for the subsequent days. On the other hand, no travelers use path 3 from the second day onward. This result closely aligns with the SO flow, where half of the travelers adopt path 1, and the other half choose path 2. The consistency between the experimental results and the SO flow demonstrates the capability of the proposed control scheme to effectively manage traffic dynamics and direct traffic flow toward an optimal state.

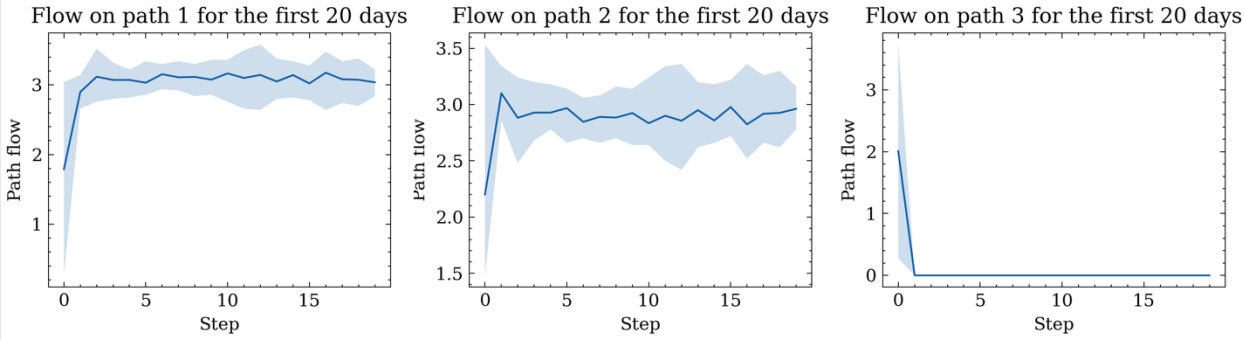


Figure 3.3: Path flow evolution under the trained policy

3.4.2 Departure time choices

In this example, we apply the model to departure time choices based on the setting outlined in [16]. The scenario involves a single bottleneck with a total demand of 6,000 vehicles and a bottleneck capacity of 3,000 vehicles per hour. The penalty factors for travel time, early arrival, and late arrival are 10, 5, and 15 respectively. The departure time window for each day ranges from 0 to 3 hours, and it is further discretized into 60 slices. The desired arrival time for all travelers is set at 2 hours. It is noteworthy that Guo et al. [16] demonstrated that the Smith dynamic fails to converge to equilibrium in this case, indicating that the day-to-day traffic dynamic here is more complex and chaotic compared to the route choice example, which makes it more difficult to learn the optimal policy.

The training curve of the algorithm for the departure time choices example is shown in Figure 3.4. As usual, the red solid curve represents the baseline scenario without system users, and we have normalized its value to 1 for comparison. The red dotted curve shows the theoretical lower bound, where the system reaches and maintains the optimal SO departure rate since the second day. The blue curve corresponds to the case with a 100% penetration rate. Given the complexity of the dynamic and the larger size of the state space compared to the previous example, the algorithm becomes more challenging to train. However, it still shows progress towards the theoretical lower bound. Although we have not trained the algorithm to full convergence, it exhibits remarkable performance. Further improvements are anticipated with a longer training duration. Moreover, the figure also includes the training curves for penetration

rates of 20% and 50%. As observed from the results, a higher penetration rate leads to better overall system performance.

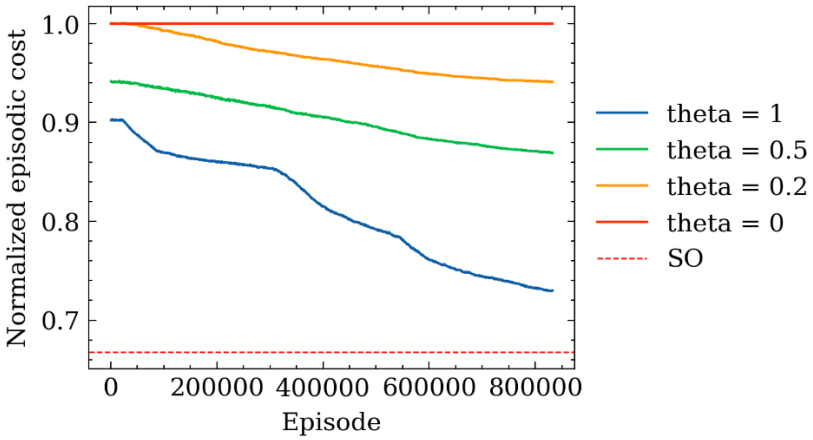


Figure 3.4: Training curve of the departure time choice experiment

Based on the training curves in Figure 3.2 and 3.4, another intriguing observation can be made. In route choices, when the penetration rate of system users is higher, the performance of the initial random policy tends to deteriorate. This is because that pure selfish user will eventually converge to an equilibrium, which might not be optimal but still represents a relatively stable and predictable behavior. On the other hand, the random initial policy has no performance guarantee and lacks optimization, leading to potentially less efficient traffic flow and higher system costs. Therefore, as the proportion of random policy increases, the overall system performance declines accordingly. Conversely, the situation is different in departure time choices, where the Smith dynamic can exhibit chaotic behavior. In such cases, a higher proportion of random policy surprisingly improves the system's performance. This could be due to the demand redistribution introduced by the initial policy, which promotes more diverse and distributed departure time choices, resulting in a more efficient system.

It is essential to acknowledge that the training process is time-consuming, especially for the departure time choice experiment. Considering the practical challenges of real-time policy training, we propose an alternative approach for the management agency. We suggest that the agency employs a simulator that accurately models the human response dynamics, where the agency can conduct extensive policy training and optimization in a controlled virtual environment.

As in the previous case, we proceed to implement the trained policy on a transportation system with a 100% penetration rate and random initial departure time profile. We conduct 20 experiments and track the evolution of the departure time profile over consecutive days. The average result is depicted in Figure 3.5. At the beginning of each experiment, the system is initialized with a random departure time profile, represented by the blue curve in the figure. However, as the trained policy takes effect, the departure time profile quickly transitions to the green curve, which closely aligns with the theoretical SO profile. Once the departure time

profile reaches the SO state, it remains relatively stable for the subsequent day. This stability is a desirable outcome, indicating that the policy has successfully managed the transportation system to maintain an optimal departure time distribution, minimizing congestion and travel delays. Also, the close resemblance of the resulting departure profile to the theoretical SO profile again affirms the effectiveness and accuracy of the participatory control approach.

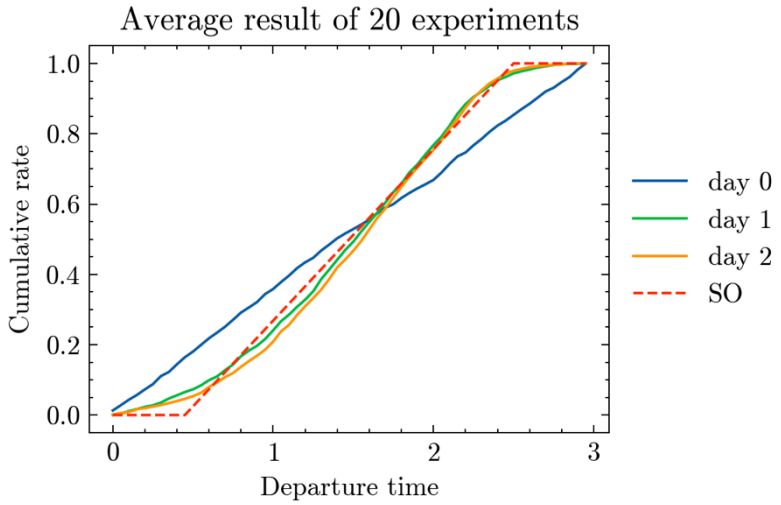


Figure 3.5: Departure profile evolution under the trained policy

4. Findings and Conclusions

This study has focused on the traffic state estimation with connected and automated vehicles, particularly on the automated vehicle with surrounding observation. As a moving observer, the automated vehicle could provide much more useful information compared with a connected vehicle. Other than the trajectory itself, automated vehicles can observe the surrounding background vehicles, especially the vehicle from the opposing direction. In the ideal case, the automated vehicle could see every vehicle passing through in the opposite direction, at least the number of vehicles. We formulate the traffic state estimation problem as a hidden Markov model: the hidden state is the overall traffic while the observable state is all the available observations. An existing stochastic traffic flow model is used to predict the traffic flow dynamics. We formulate three different observation models including: 1) automated vehicle observations, 2) stop event, and 3) free-flow event. The main difficulty of the observation model is that we need to consider the change of the observable range of the automated vehicle considering the sight blocking. We propose to use a truncated observable range to mitigate this effect. Finally, this study uses SUMO simulation environment to test the proposed methods. It is demonstrated that both connected vehicle and automated vehicle data can be used to significantly improve the estimation accuracy, and intuitively, the automated vehicle data is more useful than the connected vehicle.

In addition, we have proposed a novel traffic control scheme that leverages CAVs to indirectly influence the day-to-day adjustment process of human drivers, thereby improving overall system performance. Specifically, we first model the problem as a finite agent control problem. To overcome the intractability due to the large number of agents, we consider the limiting case with an infinite number of travelers and formulate the problem within the major-minor mean field control framework. Under mild conditions, we prove the existence of the optimal control policy, which can be computed by leveraging reinforcement learning algorithms. The numerical examples demonstrate that the proposed method has the potential to fully harness the control power of CAVs, achieving an optimal and efficient traffic flow.

5. Recommendations

This report serves as an initial exploration into the development of a participatory traffic control system. It successfully demonstrates that CAVs can be leveraged to accurately assess current traffic conditions and subsequently optimize them. There are several avenues for future research that could further enhance the effectiveness of participatory traffic control systems.

Firstly, the estimation techniques used in the initial section of the report are constrained to a two-lane roadway in a simulated environment. For a more comprehensive validation, it would be essential to evaluate these methods using real-world data.

Secondly, there is considerable potential for synergy between our newly proposed control scheme, detailed in the latter part of the report, and traditional traffic management strategies such as congestion pricing. Further investigation into how these varying control measures can be seamlessly integrated could lead to even more significant improvements in traffic efficiency.

6. Outputs, Outcomes and Impacts

This research serves as a proof of concepts for the participatory traffic control. We demonstrate that leveraging connected and automated vehicle data can significantly improve the accuracy of traffic state estimation. Besides, it is theoretically proved that the connected and automated vehicles can be utilized to strategically regulate the traffic demand.

The proposed model has significant advantages and real-world implementation impacts. This work paves the way for the development of an innovative traffic management method. Upon the implementation of the participatory traffic control, it has the potential to alleviating traffic congestion, and reducing travel time, which will have positive implications for transportation efficiency.

The following outputs were generated during the performance of this project:

- Conference presentation at 9th International Symposium on Dynamic Traffic Assignment (DTA 2023)
- A paper submitted for presentation at 2024 TRB Annual Meeting

References

[1] K. Cui, C. Fabian, and H. Koeppl, "Multi-Agent Reinforcement Learning via Mean Field Control: Common Noise, Major Agents and Approximation Properties," 2023, Accessed: Apr. 17, 2023. [Online]. Available: http://arxiv.org/abs/2303.10665

[2] G. Welch and G. Bishop, *An introduction to the Kalman filter*. 1995. doi: 10.1109/LAWP.2018.2818058.

[3] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handb. nonlinear Filter.*, vol. 12, no. 3, pp. 656–704, 2009, [Online]. Available: https://www.stats.ox.ac.uk/~doucet/doucet_johansen_tutorialPF2011.pdf

[4] S. E. Jabari and H. X. Liu, "A stochastic model of traffic flow: Gaussian approximation and estimation," *Transp. Res. Part B Methodol.*, vol. 47, pp. 15–41, Jan. 2013, doi: 10.1016/j.trb.2012.09.004.

[5] C. Wu, A. M. Bayen, and A. Mehta, "Stabilizing Traffic with Autonomous Vehicles," in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., Sep. 2018, pp. 6012–6018. doi: 10.1109/ICRA.2018.8460567.

[6] J. I. Ge and G. Orosz, "Optimal Control of Connected Vehicle Systems With Communication Delay and Driver Reaction Time," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2056–2070, 2017, doi: 10.1109/TITS.2016.2633164.

[7] J. I. Ge and G. Orosz, "Connected cruise control among human-driven vehicles: Experiment-based parameter estimation and optimal control design," *Transp. Res. Part C Emerg. Technol.*, vol. 95, no. July, pp. 445–459, 2018, doi: 10.1016/j.trc.2018.07.021.

[8] M. Cicic, X. Xiong, L. Jin, and K. H. Johansson, "Coordinating Vehicle Platoons for Highway Bottleneck Decongestion and Throughput Improvement," *IEEE Trans. Intell. Transp. Syst.*, 2021, doi: 10.1109/TITS.2021.3088775.

[9] K. Zhang and Y. (Marco) Nie, "Mitigating the impact of selfish routing: An optimal-ratio control scheme (ORCS) inspired by autonomous driving," *Transp. Res. Part C Emerg. Technol.*, vol. 87, pp. 75–90, Feb. 2018, doi: 10.1016/j.trc.2017.12.011.

[10] G. Sharon, M. Albert, T. Rambha, S. Boyles, and P. Stone, "Traffic optimization for a mixture of self-interested and compliant agents," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, pp. 1202–1209, 2018, doi: 10.1609/aaai.v32i1.11444.

[11] Z. Chen, X. Lin, Y. Yin, and M. Li, "Path controlling of automated vehicles for system optimum on transportation networks with heterogeneous traffic stream," *Transp. Res. Part C Emerg. Technol.*, vol. 110, pp. 312–329, Jan. 2020, doi: 10.1016/j.trc.2019.11.017.

[12] T. L. Friesz, D. Bernstein, N. J. Mehta, R. L. Tobin, and S. Ganjalizadeh, "Day-to-day dynamic network disequilibria and idealized traveler information systems," *Oper. Res.*, vol. 42, no. 6, pp. 1120–1136, 1994, doi: 10.1287/opre.42.6.1120.

[13] R. Li, X. Liu, and Y. (Marco) Nie, "Managing partially automated network traffic flow: Efficiency vs. stability," *Transp. Res. Part B Methodol.*, vol. 114, pp. 300–324, Aug. 2018, doi: 10.1016/j.trb.2018.06.004.

[14] Z. Guo, D. Z. W. Wang, and D. Wang, "Managing mixed traffic with autonomous vehicles – A day-to-day routing allocation scheme," *Transp. Res. Part C Emerg. Technol.*, vol. 140, Jul. 2022, doi: 10.1016/j.trc.2022.103726.

[15] X. Guo and H. X. Liu, "Bounded rationality and irreversible network change," *Transp. Res. Part B Methodol.*, vol. 45, no. 10, pp. 1606–1618, 2011, doi: 10.1016/j.trb.2011.05.026.

[16] R. Y. Guo, H. Yang, and H. J. Huang, "Are we really solving the dynamic traffic equilibrium

problem with a departure time choice?," *Transp. Sci.*, vol. 52, no. 3, pp. 603–620, 2018, doi: 10.1287/trsc.2017.0764.
[17] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *35th International Conference on Machine Learning, ICML 2018*, 2018, pp. 8869–8886.
[18] M. Huang, P. E. Caines, and R. P. Malhamé, "Large-population cost-coupled LQG problems with nonuniform agents: Individual-mass behavior and decentralized ε-nash equilibria," *IEEE Trans. Automat. Contr.*, vol. 52, no. 9, pp. 1560–1571, Sep. 2007, doi: 10.1109/TAC.2007.904450.
[19] J. M. Lasry and P. L. Lions, "Mean field games," *Japanese J. Math.*, vol. 2, no. 1, pp. 229–260, 2007, doi: 10.1007/s11537-007-0657-8.
[20] K. Cui, A. Tahir, M. Sinzger, and H. Koeppl, "Discrete-Time Mean Field Control with Environment States," in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, Dec. 2022, pp. 5239–5246. doi: 10.1109/cdc45484.2021.9683749.
[21] K. Zhang, Z. Yang, and T. Başar, "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms," in *Studies in Systems, Decision and Control*, 2021, pp. 321–384. doi: 10.1007/978-3-030-60990-0_12.
[22] M. J. Smith, "The Stability of a Dynamic Model of Traffic Assignment—An Application of a Method of Lyapunov," *Transp. Sci.*, vol. 18, no. 3, pp. 245–252, Aug. 1984, doi: 10.1287/trsc.18.3.245.
[23] R.-Y. Guo, H. Yang, and H.-J. Huang, "The Day-to-Day Departure Time Choice of Heterogeneous Commuters Under an Anonymous Toll Charge for System Optimum," *Transp. Sci.*, 2023, doi: 10.1287/trsc.2022.1191.
[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12, Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.06347