Navigating Urban Traffic: From Data to Simulations to Real-World Impacts

By

Alben Rome Bagabaldo

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexandre M. Bayen, Chair
Associate Professor Marta C. González
Professor Joan Walker

Summer 2024

Navigating Urban Traffic: From Data to Simulations to Real-World Impacts

Abstract

Navigating Urban Traffic: From Data to Simulations to Real-World Impacts

by

Alben Rome Bagabaldo

Doctor of Philosophy in Engineering - Civil and Environmental Engineering

University of California, Berkeley

Professor Alexandre M. Bayen, Chair

Transportation systems face increasing challenges in managing traffic congestion and improving mobility. Recent advancements in sensor networks and cyber-physical systems have led to an exponential increase in data collection, often presenting issues such as ambiguity, inconsistency, inaccuracy, and problematic data formats. This dissertation investigates leveraging data from multiple sources to develop traffic simulations. The developed simulations help understand dynamic navigation strategies in urban environments, focusing on how information-aware routing impacts congestion dynamics. This work explores the influence of dynamic routing algorithms on overall traffic performance and congestion levels. By employing the SIR model and average marginal regret, the research quantifies the effects of navigation apps on traffic patterns, highlighting the potential for increased congestion and its implications. Moreover, two case studies are presented, capturing the effects of traffic light coordination on routing and using speed limits as a control parameter to improve traffic performance. On another note, sensor data can be incomplete due to malfunctions. This research demonstrates how machine learning techniques can accurately fill in the missing data. Beyond macroscopic traffic analysis, this dissertation examines the individual behavior of drivers, which can lead to 'phantom congestion'. With the growing interest in automated vehicles (AVs), a chapter evaluates the impact of microscopic traffic behavior on longitudinal AV control policies in a ring road setting. The investigation of stop-and-go waves in closed-circuit ring road traffic reveals that improvements are possible using specific AV controllers, which could be affected by their distribution in mixed traffic settings. Additionally, this dissertation presents a chapter on designing bus routes using location-based services data. By understanding routing behavior and integrating existing transit data with shortest path algorithms and machine learning techniques, we can design new bus routes and potentially enhance existing ones. Overall, through advanced traffic simulation techniques and data-driven analyses, this research demonstrates that a balanced mix of app and non-app users can improve traffic conditions, AVs can mitigate stop-and-go waves, and innovative bus route design can enhance urban transit systems.

To my family and friends...

Contents

List of Figures

List of Tables

Acknowledgments

While a Ph.D. degree is conferred to one person, it takes the support of many to earn one. This dissertation would not have been possible without the help and support of the people around me.

First and foremost, I am deeply grateful to my research adviser, Prof. Alexandre Bayen, for welcoming me into his lab and dedicating his time to helping me grow as a researcher. His support has greatly enriched my academic journey.

I extend my heartfelt thanks to Prof. Marta Gonzalez, who served as my second adviser. Her mentorship has profoundly shaped my work, providing invaluable insights. I am also grateful for the opportunity she provided me to serve as a Graduate Student Instructor (GSI) for three semesters in her classes at Cal.

I wish to express my appreciation to Prof. Joan Walker for her pivotal role as the committee chair during my qualifying exam and for serving on my dissertation committee. My thanks also go to Prof. Alexander Skabardonis for being a member of my qualifying exam committee, and to Dr. Jane Macfarlane for her mentorship.

My doctoral studies would not have been possible without the financial support from the Philippine-California Advanced Research Institutes Ph.D. Scholarship program, funded by the Commission on Higher Education and administered by the Department of Science and Technology. I am especially thankful to Ms. Regina Beleno for ensuring that our school fees and stipends were processed on time. I also appreciate Mapua University for its additional financial assistance, and I am grateful to Dr. Francis Uy for informing me about the scholarship opportunity and supporting my application. Likewise, I thank Dr. Reynaldo Vea for his assistance with my scholarship appeal just before I embarked on my Ph.D. journey.

I owe a great debt of gratitude to my friends and colleagues in Prof. Bayen's lab, including past and present members: Alexander Keimer, Yashar Farid, Jonathan Lee, Joy Carpio, Fangyu Wu, Nathan Lichtle, Zhe Fu, Arwa Alanqary, Aboudy Kreidieh, Sulaiman Almatrudi, Yiling You, Shuxia Tang, Saleh Albeaik, Marsalis Gibson, Jessica Lazarus, Kathy Jang, Eugene Vinitsky, and Ashkan Yousefpour. A special thanks to Theophile Cabannes and Fang-Chieh Chou for their instrumental support in completing this dissertation.

I am also thankful to all the members of the HumNet lab for welcoming me: Minho Kim, Jiaman Wu, Tugba Ozturk, Albert Cao, Abdullah Alhadlaq, Aparimit Kasliwal, Giuseppe Perona, Eileen Chen, Weixin Li, Ariel Salgado, and others whom I had the pleasure of meeting during my time at Berkeley.

I am extremely grateful to all the people I had the opportunity to work with and mentor, especially John Lee, Carl Gan, Preston Fu, Arya Bakhtiar, Ayush Jain, and Yuhan Tang. I would also like to thank Aimsun for granting me a free license to use their software.

To my fellow founding officers and members of the Representation of Asian and Pacific Islander in the Department of Civil and Environmental Engineering (RAPID-CEE), thank you for your camaraderie and support.

I am thankful to the Department of City and Regional Planning and the Physics Department for the opportunity to serve as a GSI, and to the Industrial Engineering and Operations Research Department for allowing me to contribute as a Reader. I am also grateful for the privilege of serving as a GSI within my home department, Civil and Environmental Engineering. I am especially thankful to Dr. Daniel Arnold, who believed in my capabilities.

I am grateful to all my friends who helped me navigate the challenging years: to Theophile and his wife, Marine, for the weekend cooking sessions and all the other fun we shared; to Guillaume and Adele, and Elly, who even helped me get home from the hospital after my surgeries; to Xuan, Yuhan, and Clarkson, who have been great friends and roommates at conferences; to Joy and Renz, Michael, and Mira, for all the unplanned trips and movie marathons; to Ryan, Christian, Brett, Richard, and JR, for all the great Bay Area adventures; and to all the people I met at CrossFit, especially the ever-dedicated coaches. I also appreciate my roommates at Jackson House and all the acquaintances I made in the San Francisco Bay Area, which has been my home for the past five years.

My deep gratitude extends to my Bay Area family — Tita Ellen and Tito Nong, my cousins, Ninang Jojie and Tito Dinan — for all their support during my time in the area.

Lastly, my deepest appreciation goes to my family — especially my parents, Romulo and Marita, and my siblings, Almir and Alyssa — for their unwavering love and support throughout the years. Without them, this journey would have been nearly impossible.

Chapter 1

Introduction

1.1 Unscrambling the data

In transportation engineering and city planning, the integration of data analytics and simulation techniques offers a significant opportunity to understand the complexities of traffic dynamics and routing decisions. Understanding routing is crucial in modern transportation engineering [170]. This dissertation lays the groundwork for a comprehensive exploration of how big data analytics in sensor networks and cyber-physical systems can revolutionize transportation systems.

The dissertation scrutinizes the interaction among user behaviors, routing strategies, and traffic flow dynamics, digging into the foundational elements of data analysis and simulation methodologies. By analyzing vast amounts of data from various sources, valuable insights are obtained into the fundamental patterns and trends that shape the transportation landscape. These insights not only inform decision-making but also serve as the basis for strategic interventions aimed at improving mobility, efficiency, and sustainability [134] [4].

The creation of traffic simulation models is considered essential as they allow for the development of virtual replicas of real-world traffic scenarios, enabling hypothesis testing, strategy evaluation, and outcome forecasting with precision. These models help in understanding how factors such as road infrastructure, vehicle types, and driver behaviors interact to impact traffic flow and in evaluating interventions from a sustainability perspective [73] [97] [139].

The rise of sensor networks and cyber-physical systems has resulted in a tremendous increase of data volume and variety available for transportation analysis. These systems collect data from sources like GPS devices, traffic cameras, and mobile devices, providing real-time insights into traffic conditions, congestion patterns, and user mobility behaviors. Harnessing this data abundance enables the creation of advanced algorithms and predictive models to optimize traffic management strategies and enhance overall system performance.

Recent trends in transportation include the rise of shared mobility services, such as ride-sharing and micro-mobility options like e-scooters and e-bikes. These new modes of trans-

portation introduce complexities in understanding their impact on traffic flow, infrastructure requirements, and user behavior [132]. Additionally, the increasing adoption of electric vehicles and autonomous vehicles poses challenges in terms of infrastructure readiness, charging station placement, and integration with existing transportation networks.

Addressing environmental impacts, equity, and accessibility challenges in transportation is paramount. Data-driven approaches can help mitigate these issues by optimizing traffic flow, encouraging public transportation usage, and transitioning to more sustainable transportation modes. Data analytics and simulation tools play a crucial role in identifying underserved areas and ensuring fair access to mobility options

The integration of data analytics and simulation tools not only enables us to understand current traffic patterns but also empowers us to anticipate and mitigate future challenges. By leveraging historical data, real-time information, and predictive modeling techniques, we can forecast traffic demand, identify potential bottlenecks, and design more resilient transportation networks. This proactive approach allows us to implement targeted interventions and allocate resources efficiently, ultimately enhancing the reliability, sustainability, and inclusivity of transportation systems.

1.2 Overview and Contributions

In this dissertation, we explore in greater depth the concepts of information-aware routing and dynamic navigation strategies. We explore how these innovative approaches leverage data from multiple sources and advanced algorithms to optimize route planning, reduce travel times, and minimize congestion. Through a combination of theoretical insights and practical case studies, we aim to demonstrate the transformative potential of data-driven approaches in shaping the future of transportation.

Chapter 2 discusses the often underestimated challenges of creating, calibrating, and validating a traffic microsimulation model. Using a case study of the San Jose Mission district in Fremont, California, USA, this chapter provides a detailed process for developing a large-scale traffic microsimulation using the Aimsun microsimulation software. We made the necessary codes and data openly available for reproduction. The calibration process involved simulating the movement of 130,000 vehicles through a Fremont subnetwork with over 4,000 links, using a representative six-hour afternoon demand from 2019. The resulting simulation showed a strong correlation with real data, with a linear regression slope of 0.976 and an R^2 value of 0.845, measured across 83 sensors at 15-minute intervals. This chapter has been co-authored with Theophile Cabannes (among others) and is also included in his dissertation.

Chapter 3 focuses on traffic flow, an essential aspect of traffic analysis typically measured using loop detectors. The study highlights the California Department of Transportation's Performance Measurement System (PeMS), which uses these detectors but sometimes suffers from data gaps due to faulty detectors. To address this, the study aims to predict traffic flow in the absence of data on some detectors. Principal component analysis demonstrated

that traffic flow could be reconstructed using a few eigenvectors. The study then applied machine learning techniques, including linear regression and random forest, to predict traffic flow. Using two weeks of flow data from I-680 South as a case study, the results showed that simple feature selection from the data improved prediction accuracy, which further improved with k-Means clustering. However, the random forest model outperformed all other methods.

Chapter 4 investigates the impact of navigation app usage on local traffic patterns using microscopic traffic simulations. The study compares static routing, where users follow predetermined fastest routes, and dynamic routing, where users adjust their routes based on real-time app guidance. Key traffic metrics such as flow, density, speed, travel time, delay time, and queue lengths were analyzed. The research also applied a susceptible-infected-recovered (SIR) model to understand congestion dynamics, finding optimal system performance when 30-60% of users adopted dynamic routing. The SIR model indicated that a 40% adoption rate of dynamic routing achieved the best congestion propagation-to-dissipation ratio, providing valuable insights for transportation planning and management. This chapter also uses average marginal regret to evaluate the impact of navigation app usage.

The next chapter explores two additional case studies using traffic simulations: signal coordination and speed limit regulation. Chapter 5 examines the role of traffic signals in urban traffic management in the Mission San Jose District of Fremont, California, using 2019 demand data and microscopic traffic simulation models to assess the impact of uncoordinated versus reconfigured signal timings on Mission Boulevard. Findings show that improved signal coordination affects traffic flow and congestion. This chapter also explores reducing speed limits as a Pigovian tax, proposing that viewing travel time as a price and lowering speed limits on local roads can act as a uniform tax to achieve a socially optimal traffic equilibrium, highlighting the potential of speed limit regulations to balance travel time and traffic flow for more desirable urban traffic outcomes.

Previous chapters capture the macroscopic effect of certain schemes on traffic. Meanwhile, Chapter 6 explores the phenomenon of stop-and-go waves in closed-circuit ring road traffic and evaluates various vehicle longitudinal control models. Human-driven vehicles (HVs) can induce persistent stop-and-go waves, but these can be mitigated by introducing automated vehicles (AVs). The study assessed ten AV control algorithms using the Flow platform, with experiments varying the distribution and penetration rates of AVs. Results from over 3,000 simulations showed that higher AV penetration rates generally improved traffic conditions, regardless of AV distribution. The reinforcement learning (RL) controller consistently outperformed other algorithms in terms of time to stabilize, maximum headway, vehicle miles traveled, and fuel economy, indicating significant potential for improving traffic flow with AV integration. This chapter has been co-authored with Fang-Chieh Chou (and Alexandre Bayen) and is also included in his dissertation.

We believe that understanding the collective benefits of each routing scheme is the first step toward developing new algorithms or design mechanisms for routing apps and vehicle automation. However, this is limited to car travel. Instead, it might be more effective to implement policies that promote ecological transportation options, such as the use of public transportation systems, or other more sustainable methods. Hence, Chapter 7 addresses

the inefficiencies caused by discrepancies between transit agency predictions and real-world travel behavior. By leveraging Location-based Services (LBS) data, the study analyzed vehicle routing behavior in Dallas, Texas. Using map matching techniques, the data was overlaid onto existing road networks and transit lines information by using General Transit Feed Specification (GTFS) data, identifying underserved areas with high LBS data density but insufficient transit infrastructure. A shortest path-based clustering algorithm was then applied to design new transit lines and route extensions. The results showed that the new transit lines effectively covered these underserved areas and demonstrated potential reductions in vehicle miles traveled (VMT) when new bus lines were added to the existing routes. This chapter has been co-authored with Yuhan Tang (together with Abdullah Alhadlaq and Marta Gonzalez) and is included in his master's thesis.

Finally, we close this dissertation with the concluding remarks in Chapter 8 where a recap of findings and an outlook of future research are given.

Chapter 2

Development of Traffic Simulation

2.1 Overview

Motivation

[1]Each year, individuals residing in U.S. cities lose 84 hours and 33 gallons (124.92 liters) of fuel due to traffic congestion [92]. The productivity loss from traffic congestion is estimated at $15 billion annually [92], while road transportation accounts for approximately 15% of global energy consumption [1]. These statistics highlight significant efficiency gaps that remain unresolved, placing the responsibility on cities to implement transportation policies aimed at reducing traffic congestion and enhancing the efficiency of transportation systems.

To assess the impact of a city's proposed transportation policy, transportation planners have three primary options. First, they can conduct case studies. For instance, if the city of Fremont, CA, seeks to understand the effect of modifying traffic signal timing plans on Mission Boulevard (a major street in the city) on cut-through traffic [145], traffic engineers might extrapolate from the actions taken by the city of Leonia for Ford Lee Road [81] or by the city of Pleasanton for Dublin Canyon Road [80]. If a sufficient number of case studies exist, standardized machine learning techniques can be employed to make statistically significant predictions about how a chosen policy or set of policies would impact a given traffic state.

Second, cities can employ a trial-and-error approach (often referred to as evidence-based practice or A/B experiments). Over time, as more trials are conducted, reinforcement learning can be used to explore and identify optimal policies. For example, the city of Fremont might experiment with altering traffic signal plans, implementing turn and access restrictions, activating ramp metering, and other strategies [145], retaining only those policies that yield the most favorable public outcomes.

Third, a city can create a digital twin of its road traffic network to test and refine policies

[1]This chapter is co-authored with Theophile Cabannes (among others) and is also included in his dissertation. The work was presented at the Transportation Research Board Annual Meeting in 2023 [29].

in a virtual environment [99]. Fremont might choose this method to avoid inconveniencing its residents with frequent experimental changes to traffic signal plans. Digital twins are particularly valuable in addressing complex transportation challenges where there are limited case studies, or where real-world testing of potential solutions is impractical due to time or cost constraints. Such challenges include planning sustainable transportation systems that require coordination among connected vehicles, transportation planners, and traffic managers.

Depending on a city's objectives and the type of candidate policies that traffic engineers intend to evaluate and implement, various types of digital twins or simulations can be developed. These include land use models [162], which assist in defining zoning, or activity-based models [21], which illustrate commutes or the accessibility and attractiveness of a district, as depicted in figure 2.1. Cities may develop microsimulations if they aim to replicate individual driving behaviors and achieve fine-grain traffic resolution.

One of the motivations for our work is the desire to understand the impact of information-aware routing and traffic-calming measures (*e.g.*, traffic signal timing changes) on congestion and business attractiveness in the Mission San Jose district in Fremont, CA. To our knowledge, within the options for digital twins, only microsimulations can effectively assess the impact of changes in traffic signal timing plans on road traffic at a small network scale.

This chapter aims to equip city planners and traffic engineers with the necessary tools and methodologies to create, calibrate, and validate a large-scale road traffic microsimulation (*i.e.,* where routing behaviors significantly influence the state of traffic). An accurate traffic simulation model can support transportation engineering in various areas, from improving traffic congestion [131] to applications in autonomous driving [40].

Existing literature offers a high-level overview and comparison of traffic simulation and its development [115, 6, 40, 131], along with insights into the benefits and applications of traffic simulation [40] and an abstract framework for constructing simulations [89, 149, 107]. Additionally, the literature delves into simulation models, providing various techniques for the calibration and validation of simulations under different scenarios [96, 20, 75, 111, 143, 28].

This chapter follows a pattern similar to [149] in creating a microsimulation but introduces a machine learning-based method for model calibration (see section 2.6). We have observed that none of the aforementioned literature provides a generalizable or transferable blueprint for the end-to-end process of creating, calibrating, and validating a microsimulation. However, we recognize that some transportation consulting firms may have proprietary internal methodologies that are not publicly available.

Given that the processes for developing traffic microsimulations are quite similar across different cities, this chapter seeks to offer a comprehensive handbook and a publicly available code source for creating a microsimulation in general.

Note that traffic microsimulation only makes sense when fine-grain traffic data needs to be modeled and when case studies or A/B experiments are unavailable or unrealistic. Traffic microsimulations cannot be used for demand analysis (such as for assessing the impact of ride-hailing companies with the respect to the number of trips [63]) or mode shift analysis [21,

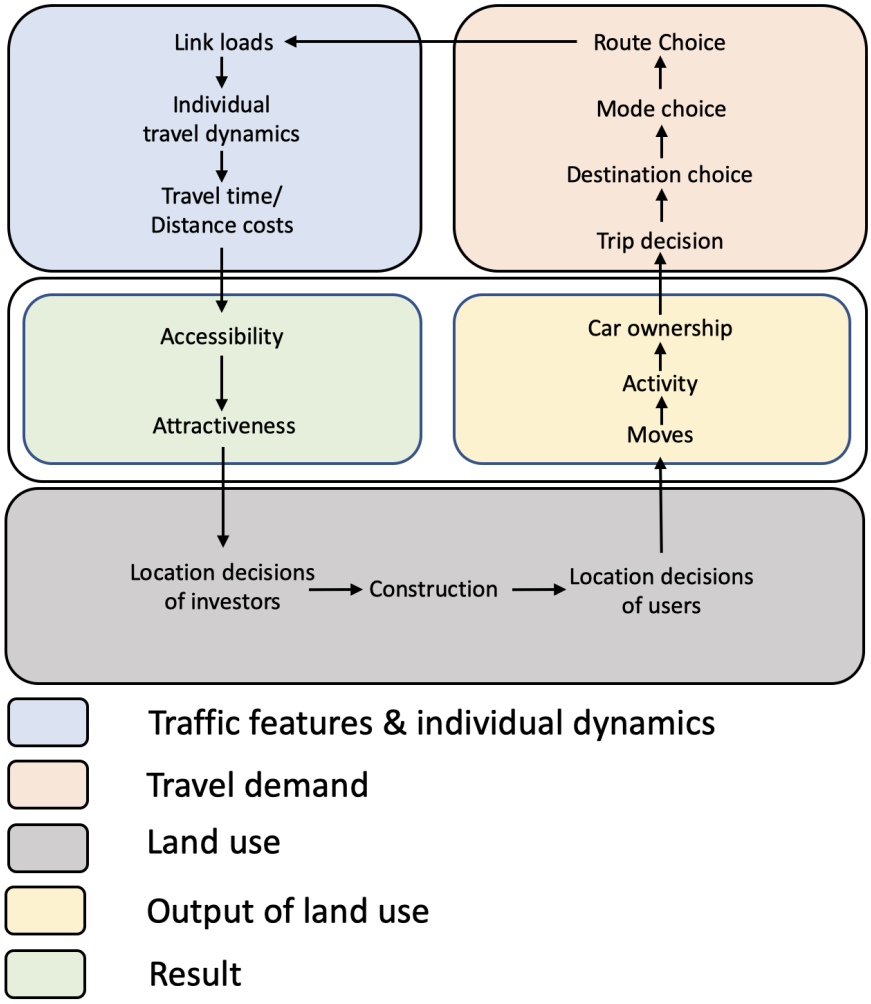


Figure 2.1: The Land Use Cycle [162] shows the relationship between transportation and city planning. Each color represents a type of model that planners can use. Microsimulations fall across the red, blue, and green boxes. Using OD demand input, it models route choice, link loads, as well as time and distance costs. Its output can be used to compute accessibility and attractiveness indicators, among others.

107, 176]. In addition, microsimulations are not relevant when data is missing to calibrate the simulation.

Contributions

This work makes two key contributions. First, it provides insights, processes, and lessons on developing a traffic microsimulator, aimed at individuals who are interested in creating their own. It offers a comprehensive overview of the full design process for traffic microsimulation, highlighting challenges that often become apparent only when one builds their own system.

Second, it offers a practical, open-source simulation featuring a network with nearly 5,000 links, representing the Mission San Jose district in Fremont. This simulation models traffic conditions and dynamic routing for an estimated peak demand of 130,000 vehicles during the afternoon (2 PM-6 PM). It is designed for researchers who want to experiment with and test ideas in a similar setting.

Lastly, this chapter provides a cohesive framework for creating, calibrating, validating, and running a traffic microsimulation, including open-source code to replicate each described step.

Organization

This chapter first gives an overview of the different type of simulations a city can perform (Section 2.2. It describes the process to create, calibrate and validate traffic microsimulations (Section 2.3). Then, we specify the input data needed (Section 2.4) and the steps to follow (Section 2.5) to run an Aimsun simulation. Finally, it describes the calibration process (Section 2.6) and the output analysis that can be done after running the simulation (Section 2.7). Each section is accompanied by references to the open-source code repository[2] such that the reader can replicate the processes discussed in this chapter. In addition, data and results from Fremont, California, USA, are included to provide motivating examples for the reader.

2.2 Existing traffic simulator

To understand the type of simulations that cities can use, it is important to understand how traffic is shaped by cities.

First, transportation planning is inherently related and interconnected to urban planning. Specific interaction within the land use cycle (Figure 2.1) can be cast using different type of models, depending on the problem a planner might want to solve.

[2] `https://github.com/Fremont-project/traffic-microsimulation`

Land use models

Land use models [162] estimate how accessibility and attractiveness impact the land use. As an illustration, using traffic data and demographics as input, UrbanSim [157] models household, employment, real estate, and potential evolution in job location, urban density, and building constructions.

Travel demand models and simulations

Using land use data coupled with demographics data, travel demand models estimate the need for transportation [107, 129]. More specific, they estimate and forecast the travel and traffic demand. These models are relevant to estimate mode-shifting opportunities or latent demand, for example. There are three main approaches of transport demand modeling.

Trip-based models [107] are the historical way to generate facility demand from sociodemographic data. In particular, since the 1950s, the classical four-step model [100] has been used extensively by planners. In this model, a trip is an origin, and a destination. Trips are aggregated across space in origin-destination demand matrix (**OD matrix**) using transportation analysis zones (**TAZs**). The model first generates a distribution of outbound and inbound trips across transportation analysis zones (TAZs) (*trips generation*). Second, the inbound trips and outbound trips across all TAZs are matched into origin-destination (OD) matrices (*trip distribution*). Third, for each OD, the modal split is estimated (*modal split*). Finally, the traffic demand is assigned to routes using the static traffic assignment (*route assignment*). The four-step model is a static model, that estimates total travel during a day without modeling any dynamics in the network.

Activity-based models [21] extends the trip-based models by associating to each trip a purpose (work, shopping, or leisure). This feature enables modeling heterogeneous departure times. It also enables taking into account the demand elasticity with respect to the network traffic conditions. MATSim [13], TransCAD [94], Visum, or CEMDAP [117] are few software products that enable solving activity-based models.

Finally, **agent-based models** extend the activity-based model by modeling each agent, instead of each trip [176]. Therefore, agent-based models can reproduce carpooling behaviors, and model that most commuters come back to their home after working during the day. MATSim [13], BEAM [10] (that was built on MATSim), or ActivitySim [59] are few software products that enable solving agent-based models. Agent-based models can also be solved by researchers using their own algorithm, like [99].

The travel demand models do not represent the traffic dynamics on individual links.

Traffic models and simulators

Assigning the OD demand to dynamic routes is done by **dynamic traffic assignment** models [42, 154]. Once the route of each vehicle is known, computing the dynamic road section traffic loads in the network is called **dynamic network loading** [42].

In this dissertation, we refer to the traffic demand as the demand that travels using the road network, without a route being specified. However, in the transportation engineering academic literature, the traffic demand might refer to a different notion: in the four-step model [100], it includes the route assignment. Our choice is motivated by the fact that we are interested in understanding the impact of information-aware routing on traffic. With the rise of information-aware routing behaviors, we believe that the **traffic demand should not include the individual route choice** anymore. Consequently, traffic models might refer to models that perform both the dynamic traffic assignment and the dynamic network loading, but it might also refer to models that only perform dynamic network loading.

Three types of traffic flow modeling exist [40, 42].

Macroscopic models aggregate vehicles into traffic flow, making the internal assumption that traffic behaves like a fluid [42]. These models can be static (like the four-step model [100], or the static traffic assignment ([113])) or dynamic. Aimsun [148], Cube, POLARIS [9], Visum, TransCAD [94] are few macrosimulation software products.

Microscopic models represent each vehicle in the network [42]. Microsimulation reproduces individual driving behaviors and fine-grain traffic resolution. The interaction of each vehicle with its environment is modeled using car-following models, lane-changing models, and route-choice models [42]. Aimsun [148], Cube, TransModeler, POLARIS [9], SUMO [17], Vissim [53], MISIMLab, Synchro, BEAM [10], Paramics [35], ActivitySim [59] are few microsimulation software products.

Mesoscopic models fit in between and represent a compromise between macroscopic and microscopic modelling [42]. They model every vehicle, but only the interaction between a vehicle and the traffic flow is modeled. The interaction between a vehicle and the traffic flow can be modeled with fundamental diagrams of traffic flow, exit-functions, or queuing models [42]. Some mesoscopic models are event-based models: they model interaction between vehicles and the traffic flow as events that might not require to model time continuously. As an example, the mean-field routing game introduced in [30] is an event-based mesoscopic model that uses fundamental diagrams of traffic flow. Aimsun [148], Cube, TransModeler, POLARIS [9], SUMO [17], Vissim [53], MISIMLab, Synchro, BEAM [10], Paramics [35], ActivitySim [59] are few mesosimulation software products.

In the team's knowledge, within the traffic simulations options, only microsimulations can help understand the impact of a change in traffic signal timing plans on road traffic at scale. Within the available traffic microsimulator, Aimsun is used in this work because it includes macroscopic, mesoscopic and microscopic simulation options, and it can be used with an academic license.

2.3 Simulation overview and its creation process

Traffic simulations [40] provide traffic information visualizations and related figures, which include vehicle hours traveled (VHT), vehicle miles traveled (VMT), mean delay per vehicle,

gas emission, accessibility index, etc., for a comprehensive analysis of the design and efficiency of the transportation system in question (Section 2.7).

In this chapter, we refer to traffic simulations as the simulation of road traffic (vehicle flows in the network over time) with key inputs of traffic demand (people's origin, destination, and departure time grouped by timed origin-destination matrices) and a road network (including road sections, lanes, intersections, road signs, and traffic signal timing plans).

Simulations can be aggregated macroscopically, mesoscopically, and microscopically [40]. Macrosimulation focuses on the aggregation of traffic flow and demand, while mesosimulation breaks traffic flow into smaller groups and examines the behavior of the whole in those groups. In this work, Aimsun Next 22 [148] is used to perform microsimulations, where the focus is on the individual elements in a transportation system.

In a microsimulation, individual vehicles are generated and assigned to a route, which is then simulated across the lanes of the input network's road sections [40]. Before being generated, each vehicle is defined by an origin, destination, departure time, and optionally, a vehicle type. The vehicle input data are aggregated across space and time into timed origin-destination (OD) matrices for each vehicle type. Space aggregation uses transportation analysis zones (TAZs), while time aggregation uses time buckets. Lanes of contiguous road sections are connected through unsignalized intersections (with yield or stop signs) or signalized intersections (with given traffic signal timing plans and a master control plan). Assigning each vehicle to a route is sometimes referred to as route assignment, while the simulation of vehicle movement through the network is commonly referred to as dynamic network loading [40]. Simulated link flows and network traversal times can be compared to ground data with which to calibrate and validate a proposed model. For the example Fremont San Jose Mission district microsimulation, input data are described in (Section 2.4).

Generally, when it comes to modeling transportation systems, there exists a notable tradeoff between the number of model variables and the risk of overfitting, a result of the large quantity of data needed to calibrate complex transportation models [28]. With this in mind, real data set aside for model calibration should be further split into training and testing data to decrease the overfitting risk [171].

Before calibrating a microsimulation (Section 2.6), one needs first to fix any existing network and demand issues (connectivity issues, wrong number of lanes, wrong traffic signal plans, small mistakes in the master traffic control plan, obvious error in the demand data). The first phase of calibration is done without simulation by matching simulated and ground total counts of vehicles entering or exiting the network. This is then followed up by the second phase of calibration, done through macrosimulation. Once the OD demand is calibrated, the driving behaviors (routing, car-following, lane-changing models), and microsimulation parameters (like simulation time step) can be calibrated using optimization algorithms that work with expensive function evaluations (this work uses a genetic algorithm that is highly parallelizable).

Once calibrated, the microsimulation can be validated using eyeball estimation or concrete metrics alike. Eyeballing here mainly consists of understanding where and when the congestion occurs in the input network and checking for consistency with any prior knowl-

edge about the network's congestion. Metrics of effectiveness (MOE) can then be used for a more rigorous second validation. For example, the mean delay per vehicle over time in the network indicates when the peak hour happens in the network and is a strong indicator of the global quality of the simulation. Finally, more specific data like detector flows and network traversal times can similarly be used to validate the simulation against ground data.

Once the simulation is created, calibrated, and validated, it can be used for analysis (Section 2.7). For example, causes of congestion can be derived and policies to mitigate these can be tested.

2.4 Input Data Description

The required inputs for a microsimulation are a network (Section 2.4) and a dataset of timed origin-destination demand (Section 2.4). Traffic data (Section 2.4) may also be used to calibrate and validate the simulation. The data we used to simulate the traffic in the Mission San Jose district around Interstate 680 and Mission Boulevard (State Route 238/262) in Fremont, CA is openly available and the process of calibrating the input data and importing it in Aimsun is reproducible[2].

Network

The road network is made up of constituent road sections connected through signalized or unsignalized intersections. To create this network, we downloaded the OpenStreetMap (OSM)[70] network model using the bounding box defined by the following coordinates: North: 37.5524, East: -121.9089, South: 37.4907, and West: -121.9544 and first cleaned it in ArcGIS[26] (see figure 2.2). After importing the network into Aimsun [148], Google satellite, Maps, and StreetView images were used to perform manual adjustments to ensure the accuracy of connectivity, yield and stop sign locations, and lane counts (see figure 2.3). Speed limits are inferred using the data provided by the City of Fremont and road capacities are adjusted using the data from the Behavior, Energy, Autonomy, and Mobility (BEAM) model which is an open-source agent-based regional transportation model [10]. Then, traffic signal plans (including the ramp meters and the master control plan) from the city and CalTrans were added using the Aimsun graphical user interface (GUI). Finally, traffic-calming measures (primarily turn-restrictions) were created in the simulator. In summary, the modeled network has 4,705 links with a total of 393.27 km section length. This includes 111 freeway sections, 373 primary road sections, 2,916 residential road sections, and 2,013 nodes (intersections), 313 of which have stop signs and 37 of which have traffic lights (26 operated by the city and 11 operated by CalTrans). The overall process to create and fix the network (with traffic signal plans) took our team about 600 person-hours to complete.

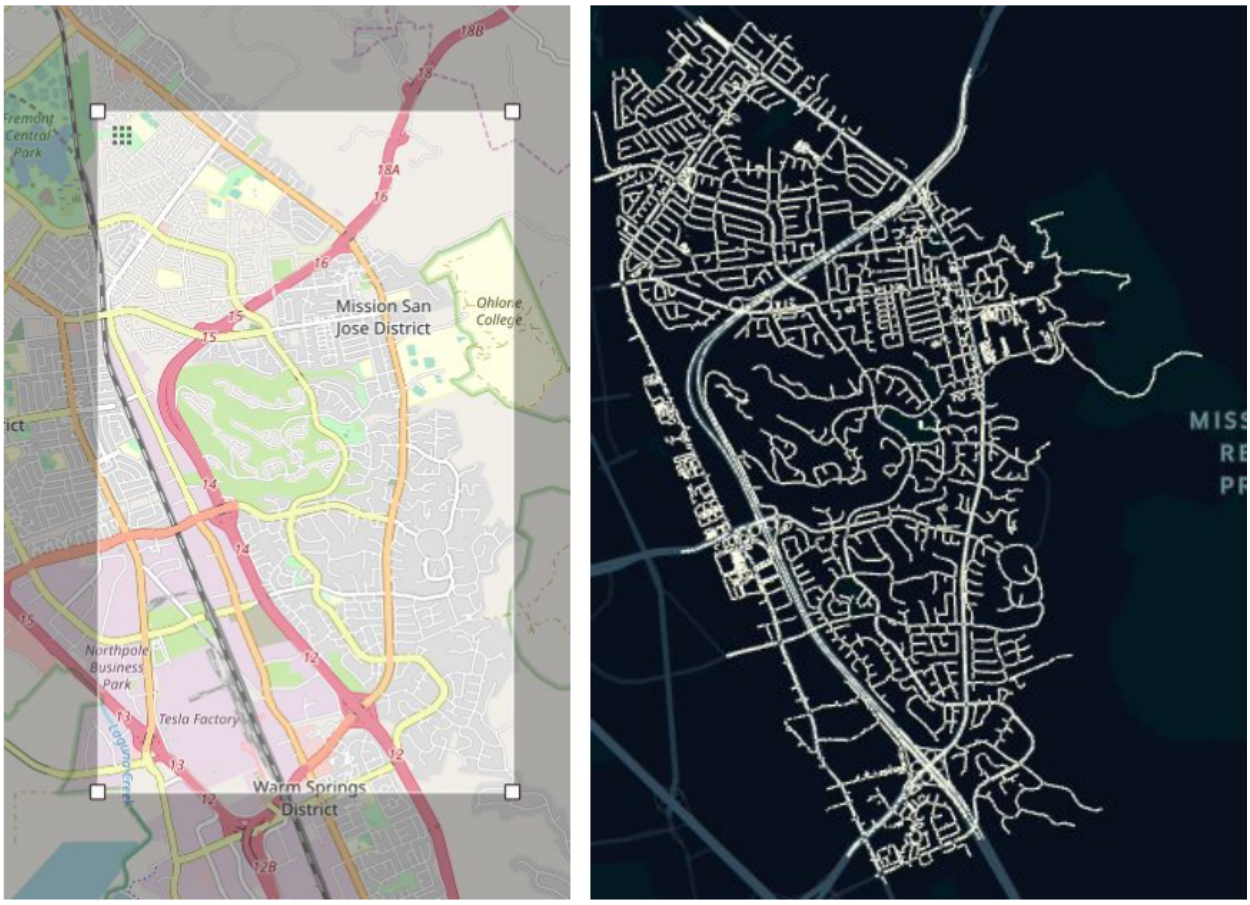


Figure 2.2: OSM network with the bounding box on the left and corresponding Aimsun network after cleaning on the right.

Origin-Destination Demand

The origins, destinations, and departure times for every vehicle are aggregated into timed origin-destination demand (TODD) matrices. Origins (or destinations) are clustered into transportation analysis zones (TAZ), which are bijective to the set of centroids connected to internal or external entry/exit points in the network. The connections between the centroids and these points (nodes) in the network are called centroid connectors [58]. The 20.8 square kilometers network area is divided into 76 internal centroids and 10 external centroids (see figure 2.4). Departure times are aggregated into 15-minutes time intervals. Between 2pm and 8pm, 130,000 vehicles are modeled (including 62,000 commuters and 68,000 residents). In this work, the demand data was derived from the SF-CHAMP demand model [82] from the San Francisco County Transportation Authority and from a StreetLight study performed

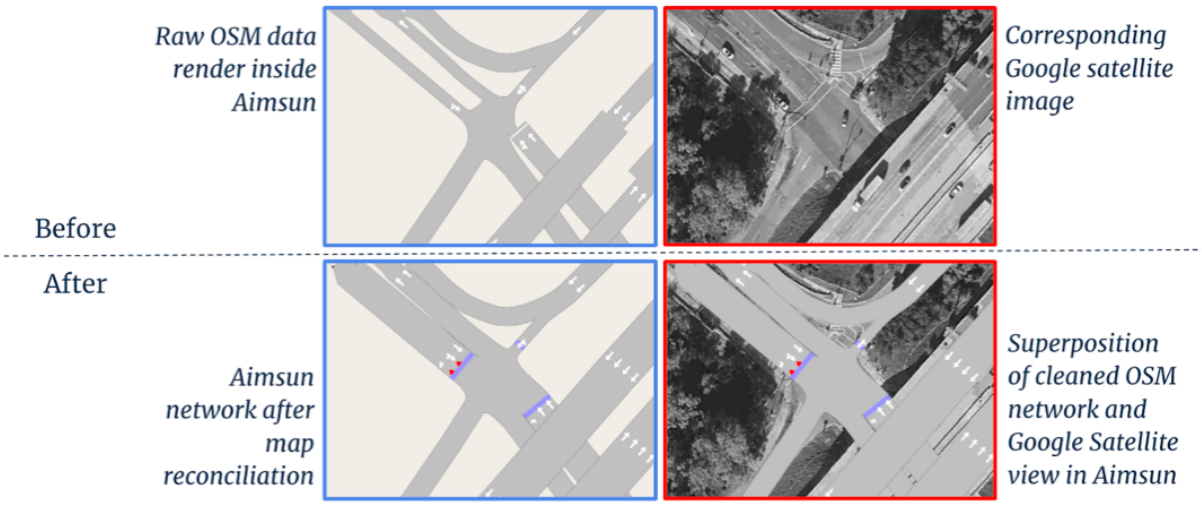


Figure 2.3: Before *(top)* and after *(bottom)* manual editing of the OSM network in Aimsun with comparison to the Google Satellite image.

for the City of Fremont. Unfortunately, there is no reproducible process to create accurate demand data as of now, which is where most of the major challenges of realistic traffic simulation remain. However, demand data accuracy can still be slightly improved through calibration against ground data (Section 2.6).

The overall process to create and calibrate the origin-destination demand took our team around 600 person-hours to complete.

Traffic data

To calibrate against ground data, one can utilize ground flow, speed, or/and travel time data, each of which can be directly imported within Aimsun as a Real Data Set. In this chapter, flow data is generated from 56 city flow detectors and 27 CalTrans Performance Measurement Systems (PeMS) detectors [41]. Speed and travel time data can be acquired using the Google Maps API[3]. The overall process to create traffic data took our team 400 person-hours to complete.

[3]`https://developers.google.com/maps/documentation/directions`

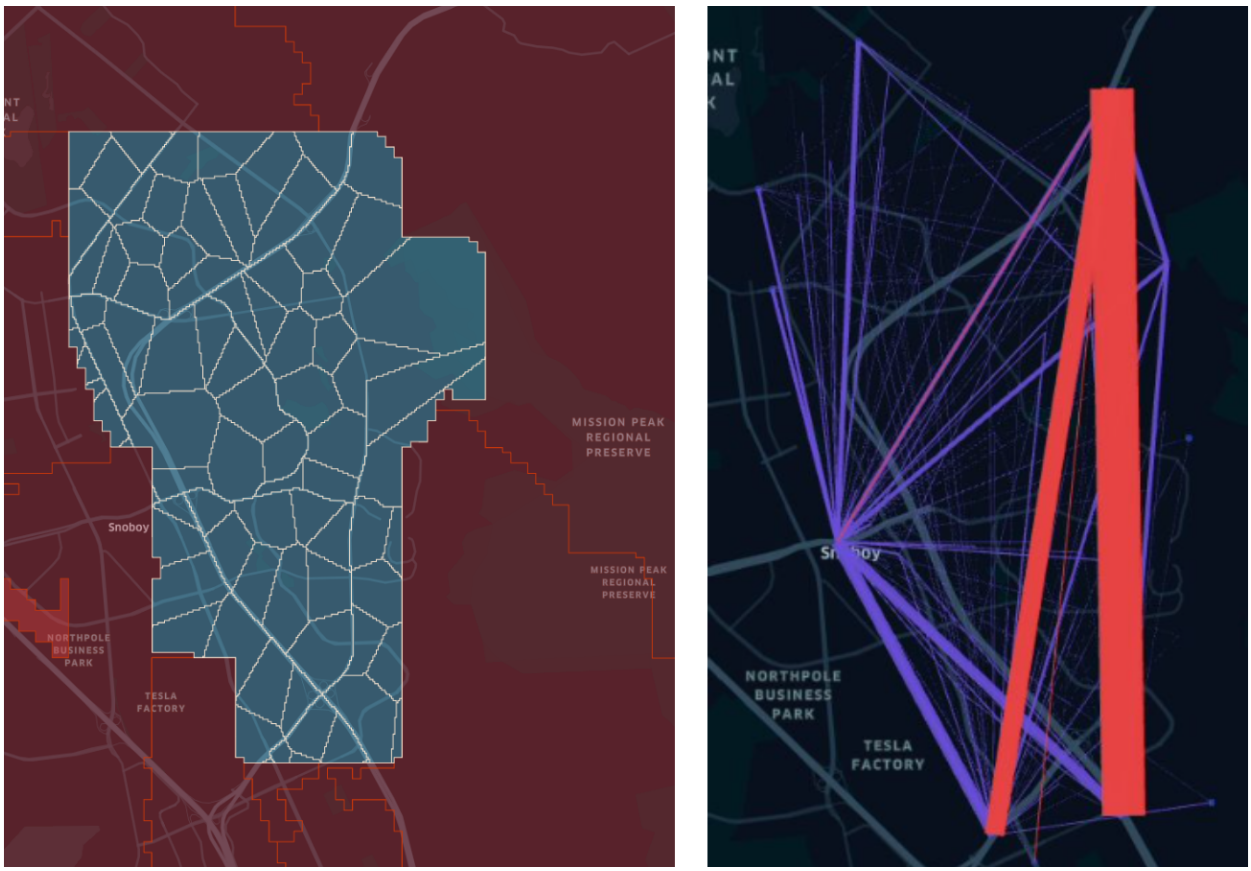


Figure 2.4: Transportation analysis zones (TAZ) (on the left) and demand plotted with desire lines (on the right). A commuter (aggregated into red lines on the right plot) is a vehicle with an origin and a destination, which are both external centroids (red TAZ on the left plot). A resident (aggregated into blue lines on the right plot) is a vehicle departing or/and arriving from or/and to an internal centroid (blue TAZ on the left plot).

2.5 Simulation

Once the input data is imported into the Aimsun simulator, simulations can be run, generating simulated traffic data as output.

Running a simulation

To run a traffic simulation in Aimsun, one first needs to create/import the network. Second, the OD demand data should be imported, and a traffic demand (a list of timed OD demand matrices with scaling factors) should be generated. Optionally, traffic data can also be imported. Once all imports are complete, a traffic simulation can be created, with many mutable parameters (see full list in Section 2.6). From here, the simulation can be run.

Creating and running a simulation can be done using the Aimsun's GUI, but, for the sake of reproducibility, we opted to write Python scripts for each step of the process[2]. The open-source repository is self-contained, and any readers with an Aimsun 22.0.1 license should be able to reproduce all the steps explained below and run the same simulations.

First, the network can be imported directly from OpenStreetMap (OSM) in Aimsun. For this work, we processed the OSM network data in ArcGIS before importing the OSM network from an external file in Aimsun. Speed limits and vehicle capacity for each of the road sections can then be updated, followed by importing the ramp meters, traffic control plans, the master control plan, and traffic management strategies. Among these, correctly configuring the traffic control plans took most of the time because of the many parameters that needed to be changed for each traffic light to consider various settings and actuation. The demand data is also imported into Aimsun, which is created by importing the centroid data with the centroid connectors. The centroid data contains the OD demand data which is converted to traffic demand The real flow data, the final bit of input, is then imported as a real data set inside Aimsun.

Then, a Jupyter Notebook is designed to define any configurable model characteristics outside Aimsun (including time step size, routing model, driving behavior, or output database location) needed to generate the simulation. Finally, the simulation can be run. Running the 6-hour-long simulation takes between 30 minutes and 4 hours, with total runtime depending on the list of data to output and the input data size.

Simulation Output Data Description

Microsimulation models can generate detailed data for every vehicle corresponding to its car following, lane-changing, and gap acceptance behavior. These characteristics can be observed through simulation playback, using Aimsun's GUI to visualize each vehicle's motion through the network. These results can also be aggregated to compare the macroscopic simulation results vs. real data sets with respect to factors such as flow, speed, and travel time.

In Aimsun, while the outputs can be accessed using the GUI, it is also possible to save them as SQLite tables. The output tables[4] contained in the output database are defined in the microsimulation configuration. Each table contains different types of statistics, and it is important to identify which tables are necessary to generate before running the simulation to prevent data cluttering. For example, the MISYS (microsimulation system) table contains system-level statistics about the entire network, such as VMT, VHT, total gas emissions, or average delay across all road sections. The SQLite output databases are used as primary data sources for simulation outputs throughout calibration (Section 2.6), validation, and analysis (Section 2.7).

2.6 Calibration

The challenges of creating a realistic microsimulation of city traffic lie in its calibration. To calibrate a microsimulation, one first needs to fix all the network issues (Section 2.6). Then, OD demand can be calibrated without simulation by matching the total counts of vehicles entering or exiting the network (Section 2.6) followed by calibration with macrosimulation by matching all detector counts in the network (Section 2.6). Once the OD demand is calibrated in this manner and after having chosen a route choice model (Section 2.6), the driving behaviors are then calibrated (Section 2.6) using a genetic algorithm (Section 2.6). The OD demand calibration and driving behavior calibration procedures can be reproduced from the provided open-source code[2].

Network and demand apparent issue fixes

A first test to ensure that the input network and demand data are not flawed can be done by running a simulation with 50% of the demand and checking that no congestion occurs in the network. Congestion can be detected with the Aimsun GUI by playing the simulation. It can also be detected with total delay in the network, or mean travel time per vehicle-mile. A second check – running the full demand – can be done by looking at gaps between the ground flow and the simulated flow that are below 50% or above 200%.

We used GitHub issues[5] to report, follow and solve any apparent network or demand issues. On average, such checks found 4 to 5 issues. Each issue was assigned to a team member, and around 2 issues were solved per team member per week. 43 issues of this variety were reported in total. The overall apparent issues fixes process took around 500 person-hours. The issues included:

- Updating incorrect lane connections at intersections.
- Changing erroneous road section geometries.

[4]Described in `https://docs.aimsun.com/next/22.0.1/UsersManual/OutputDatabaseDefinition.html`

[5]`https://github.com/features/issues`

- Changing wrong numbers of lanes on road sections.
- Removing parking lots, where cut-throughs were performed in the simulation to avoid congestion at traffic lights.
- Updating improperly imported traffic signal plans.
- Updating master traffic control plans to solve missing synchronization between traffic signal plans. We found that there was one ground truth congestion issue that could be solved by synchronizing traffic lights operated by the state with the traffic lights operated by the city on Auto-Mall Pkwy and I-680.

We realized during this process that the input demand data used was biased towards the northwest, which is likely a byproduct of the demand being sourced from the SFCTA CHAMP demand model, which aims to replicate traffic in San Francisco (which is to the northwest of Fremont). An accurate initial demand data is key for a realistic simulation.

OD demand calibration without simulation

A first calibration of the OD demand described in (Section 2.4) can be done without simulation. To do so, the total demand is scaled up or down such that the ground flow data at every external entry or exit point matches the demand data that enter or exit the network at said point (each of which is represented by an external centroid derived from external TAZs). This approach was inspired by the scaling problem formulation approach presented in [65]. The objective function of the minimization problem is shown in equation 2.1:

$$\min_{\alpha \in \mathbb{R}} \sum_{t \in \mathcal{T}} \sum_{d \in \mathcal{C}_{ext}} \left(f_{t,mapd(d)} - \alpha \sum_{o \in \mathcal{C}} \boldsymbol{d}(o,d,t) \right)^2 + \sum_{t \in \mathcal{T}} \sum_{o \in \mathcal{C}_{ext}} \left(f_{t,mapo(o)} - \alpha \sum_{d \in \mathcal{C}} \boldsymbol{d}(o,d,t) \right)^2 \quad (2.1)$$

Where:

- α is the demand scaling factor.
- $\mathcal{C}$ is the set of centroids.
- $\mathcal{C}_{ext}$ is the subset of external centroids.
- o and d are origin and destination centroids.
- $\mathcal{T}$ is the set of time bucket, and t is one time bucket.
- $mapd(d)$ is the detector associated with the destination centroid d.
- $mapo(o)$ is the detector associated with the origin centroid o.

- $f_{t,l}$ is the ground flow data for the time bucket t on the detector l
- $\boldsymbol{d}(o, d, t)$ is the number of vehicle that exit the origin o to reach the destination d during the time bucket t.

This approach can be used to derive how the demand changes over time in cases when the current OD demand matrices and the flow data over the years are available.

OD demand calibration with macrosimulation

Once the OD demand is calibrated against entry or exit flows, it can be calibrated against all detector flows in the network by assigning the OD demand to routes and counting the number of vehicles going over each detector. To assign the OD demand to routes without simulating each individual dynamics, the static traffic assignment can be used [113]. OD-demand calibration aims to better align the simulated and ground detector flows. This step can be done inside Aimsun directly with the static OD demand adjustment scenarios. OD adjustment is done by solving the constrained generalized least-squares (GLS) described in equation 2.2 as adopted from [19]:

$$\min_{\hat{\boldsymbol{d}} \in \mathbb{R}_+^{|\mathcal{C}|\times|\mathcal{C}|\times|\mathcal{T}|}} \sum_{t\in\mathcal{T}} \sum_{l\in\mathcal{L}} (f_{t,l} - \hat{f}_{t,l}(\hat{\boldsymbol{d}}))^2 + \gamma \sum_{t\in\mathcal{T}} \sum_{o,d\in\mathcal{C}} (\boldsymbol{d}(o,d,t) - \hat{\boldsymbol{d}}(o,d,t))^2 \tag{2.2}$$

Where:

- $\hat{\boldsymbol{d}}$ is the calibrated timed OD demand.
- $\mathcal{L}$ is the set of detectors.
- $\hat{f}_{t,l}(\hat{\boldsymbol{d}})$ is the simulated flow on detector l during the time bucket t given the demand $\hat{\boldsymbol{d}}$. In this subsection, the simulation is the static traffic assignment.
- $\boldsymbol{d}$ is the prior demand before the calibration with macrosimulation and after the calibration without simulation.
- γ is a scaling factor to avoid overfitting the flow data.

Because of the high number of variables that can be calibrated (namely, each element of each OD demand matrix) and the relatively low volume of ground data (flow for each time step for each detector), overfitting is a major risk and must be accounted for. Therefore, the flow data is divided into training data and testing data. Then, the objective function is minimized against the training data and tested against testing data. If the demand is changed such that the training data is perfectly fitted, but the testing data is badly reproduced, then the calibration has overfitted the training data.

The comparison of results between the macrosimulation results after OD adjustment using the training and testing sets is shown in figure 2.5, where overfitting can be observed.

To reduce the risk of overfitting, a regularization term that penalizes large modifications of the prior demand was added in equation 2.2 to provide a balancing effect [19], formulated as the Frobenius norm [18] of the difference between the calibrated and the original OD demand matrices. This approach is not exclusive to the Frobenius norm – other norms such as the nuclear norm [18] could be considered for regularization. In [102] the l_1 norm is used to compare the OD matrix elements.

In this work, to avoid overfitting, the regularization term was scaled by a large factor $\gamma = 10$ such that the OD demand matrix after the macrosimulation calibration was very similar to the OD demand matrix after the calibration without simulation.

Finally, the validation of the calibrated matrix was done with flow regression plots. Flow regression plots compare simulated values with real-world values by scatter-plotting them as y and x-axes, respectively. A linear regression is then fitted onto to the data points to draw the line of best fit [50]. The slope and intercept of the regression can then be compared to the ideal y=x line to determine whether the simulation model tends to over/underestimate the plotted metrics and whether there is bias in the model. Performance metrics of the linear regression [50], such as the coefficient of determination (R^2), root-mean-square error (RMSE, or nRMSE when normalized), and the mean absolute percentage error (MAPE) can be computed to determine the accuracy of the simulated flow.

Choice of the routing model

Once the network is bug-free and the OD demand is calibrated, the microsimulation can be run. By design, the microsimulation has many modifiable parameters to calibrate individual driver behaviors. Some of the most important microsimulation parameters are about routing behaviors.

A routing model assigns travelers to a series of links to get from one centroid (origin) to another (destination). There exist two types of routing models [42]:

1. The one-shot assignment model assigns routes and runs the simulation once. When assigning vehicles to route, only past and current information are used, and no assumptions are made about the future. The route is given following a stochastic route choice (SRC) model.

2. The iterative assignment model assigns routes and runs the simulation iteratively until the travel cost experienced by each vehicle at the end of their trip cannot be minimized by unilaterally changing the route of the vehicle. This equilibrium state is referred to as the dynamic user equilibrium [42] (sometimes called Wardrop equilibrium or Nash equilibrium).

Because running many simulations iteratively takes a lot of time, we opted for a stochastic route choice (SRC) model. Several SRC models are available in Aimsun (fixed-route under free-flow conditions, fixed-route under warm-up period traffic conditions, binomial model, proportional model, logit model, and C-logit model). Considering the tradeoff between

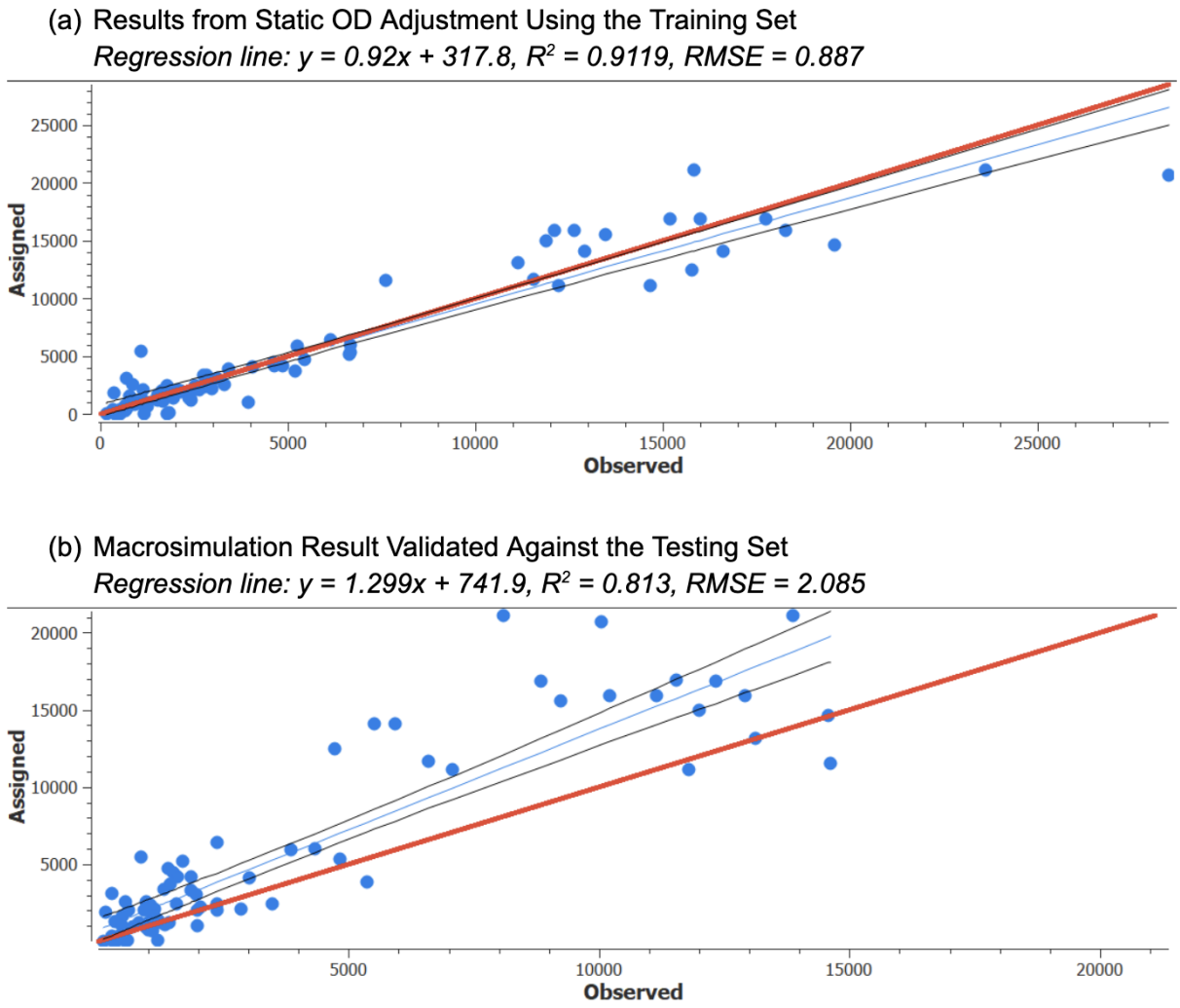


Figure 2.5: Assigned/Simulated Traffic vs. Observed/Actual Traffic Flow Linear Regression Plot after Aimsun's default OD-demand calibration with macrosimulation. Training results are reported on the top figure, while testing results are on the bottom figure. Very good training results (slope of 0.92 and R^2 of 0.9119, both close to 1) accompanying poor-quality testing results (slope of 1.299 and R^2 of 0.813, further away to 1) show that the calibration has over-fitted the training demand data.

accuracy and simulation run time, we chose to use the C-logit model [38] after experimenting with the different models.

The C-logit route choice parameters include:

- Number of alternate routes considered at each routing time step.
- Size of each rerouting time interval.
- Percentage of vehicles allowed to reroute en route.
- Averaging parameters for past and current (instantaneous) travel cost parameters.
- Route cost function parameters (utility, scaling cross-factor, overlap parameter).
- The calibration of the C-logit route choice parameters was done as part of the microsimulation calibration.

Driving behavior calibration

Once the routing choice model has been chosen, the microsimulation-specific parameters (like simulation time step length) and the driving behavior can be calibrated. The driving behavior parameters include the routing behavior parameters (like rerouting time interval), the car following parameters (like reaction time), and the lane changing parameters (like aggressiveness). The full list of parameters that can be calibrated can be found online[2].

The routing calibration aims to find the configuration of parameters that minimizes the difference between simulated data and ground truth data, without overfitting. A first manual calibration can be done based on intuition with simulation recording (for example, reaction time can be adjusted if the output to input flow ratio at some intersection seems low). Then, bounds can be set for each parameter based on physical intuitions (reaction time is between 0.2 to 3 seconds) and a systematic calibration can be performed.

In the systematic calibration, an objective function, described in equation 2.3, is set to be minimized (similar to OD demand calibration in equation 2.2):

$$\min_{\theta} \sum_{t \in \mathcal{T}} \sum_{l \in \mathcal{L}} (f_{t,l} - \tilde{f}_{t,l}(\theta))^2 \tag{2.3}$$

where:

- θ is a set of parameters to calibrate. If we have any prior on the set of parameters theta, a regularization term can be added to avoid overfitting.
- $\tilde{f}_{t,l}(\theta)$ is the simulated flow on detector l for the time bucket t given the calibrated demand, and the set of parameters θ.

An optimization algorithm can be run to find the optimal microsimulation-specific parameters and driving behavior parameters. Optimization algorithms include brute force algorithms like grid search, random search [122], classical optimization algorithms that can be found in the SciPy.Optimize toolbox [156], neural-network [85], or genetic algorithm [104]. Because evaluating the objective function given the input parameters is costly, since it requires running a microsimulation, we decided to use a genetic algorithm. Genetic algorithms are particularly efficient for opaque box functions with a high stochastic effect. In addition, they are easily parallelizable and can handle multi-criteria optimization [104].

Genetic algorithm for microsimulation calibration

We followed the approach presented by [96] to fine-tune the driving and microsimulation parameters.

Because of the significant number of model parameters (35 for micro-simulation using the C-logit stochastic route choice model) and the run-time for each simulation in our computer (30-40 minutes), the search space (*i.e.,* number of model parameters to calibrate) has been decreased through a sensitivity analysis. Only the 10 parameters that have the most impact on the measures of effectiveness have been selected by comparing the relative metric evolution over small increments of each parameter using the Latin hypercube sampling (LHS) algorithm [78]. The calibration of the 10 selected parameters was done using the Non-dominated Sorting Genetic Algorithm-III [48] from the DEAP Python library [56] using the objective function described in equation 2.3. The initial value of the parameters was set to the Aimsun default ones if not updated based on intuition after few initial simulations. Similar to subsection 2.6, the RMSE over the flow bi-plots is minimized.

2.7 Post-Processing Analysis

After the simulation is created, one needs to validate the accuracy of the simulation to ensure the credibility of its results. Once the accuracy of the simulation is satisfactory, traffic analysis can be conducted to observe how certain metrics change in different scenarios.

Validation

The first step in validating the simulation is to use Measures of Effectiveness (MOE) [24, 89], which serve as indicators for the general correctness of system-wide results. Some examples of metrics that can be used for MOE are average delay time, total distance traveled by vehicles in the entire network, and the average number of vehicles in the virtual queue at each time step.

After determining that MOE correctness is validated, the next step is to validate system-wide and location-specific metrics for the simulation. System-wide metrics denote data encapsulating property or properties across the entire simulation network, such as flow at all

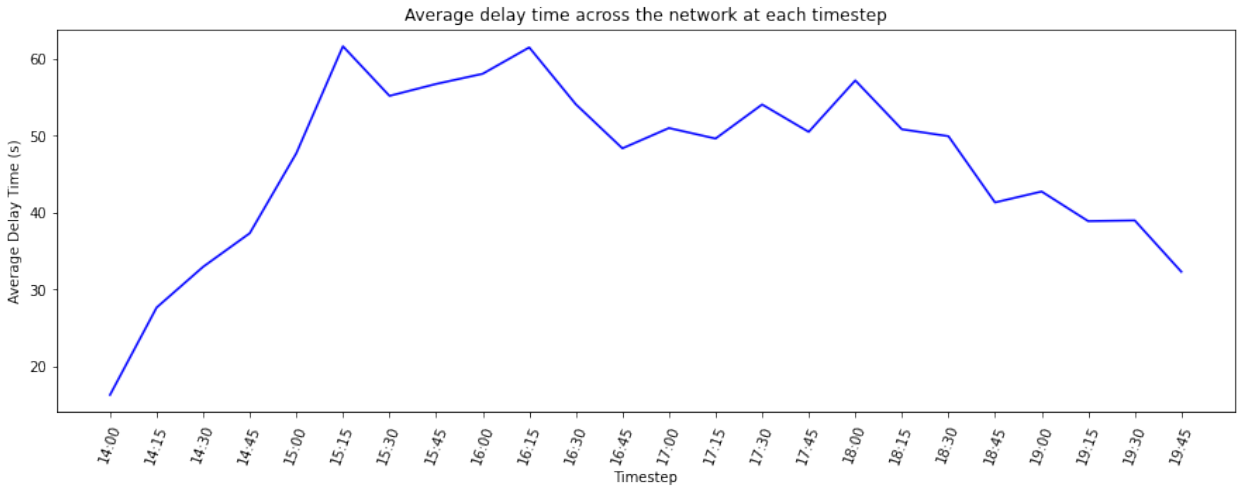


Figure 2.6: Time-series of average delay time across the entire network in Fremont, CA. We identify that peak congestion happens during 3:15PM to 4:15PM.

detectors. Location-specific metrics, on the other hand, deal with data specific to a subset of the network, such as a corridor.

Validation of system-wide metrics

For validation of system-wide metrics, linear regression plots comparing simulated versus real-world data are commonly used across previous literature [131, 7, 39]. In this work, detector flow, OD travel times, and OD route distances were system-wide metrics used to validate the simulation results (Figure 2.7).

To conduct a more detailed analysis of linear regression plots, one can create a separate regression plot of system-wide data points for each time frame. Then, a time series of regression statistics for each regression plot can be drawn to gain insight into specific points in time when the simulation needs more calibration.

In addition, it might be useful to identify the times for each detector in which the simulated flow was over 1.2 or below 0.8 of the ground flow to know which detectors need more calibration.

Distributions of actual and simulated values can also be used for eyeball validation of the simulation. Though distributions show that the general trend that the simulated data is correctly approached, they provide less insight into the individuality of each data point than biplots.

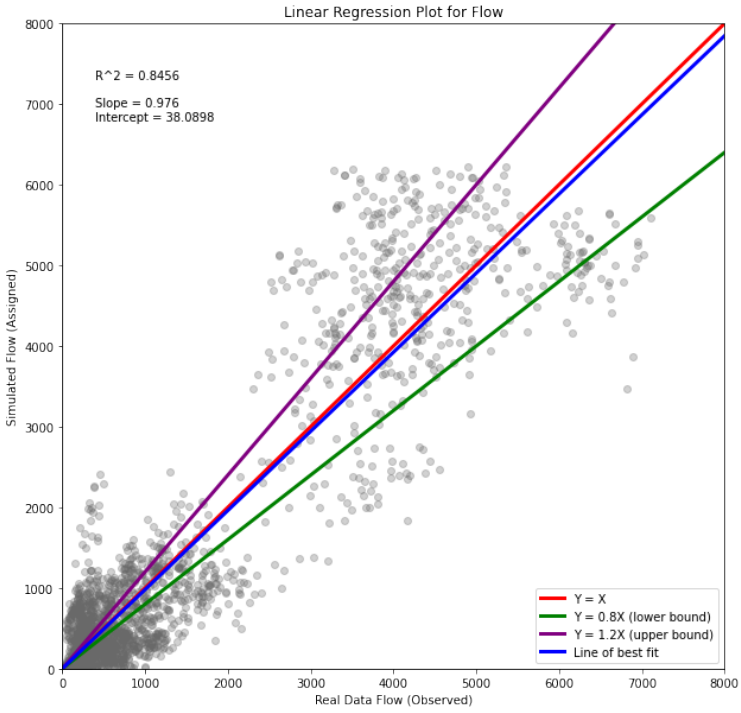


Figure 2.7: Linear regression plot of flow across 83 detectors in the entire network at each 15 minute time step.

Validation of specific location metrics

After validation of system-wide metrics, one can validate location-specific metrics in areas of high importance to ensure the accuracy of the simulation. To do so, we need to identify the scope of the location critical to our study. In this example, we set the I-680 corridor as the scope of specific location metric validation. Metrics one can validate include flow (veh/h), speed (km/h), delay time (h or h/veh), and density of vehicles (veh/km) observed at each detector within the corridor.

We divided up validation of location-specific metrics into two processes. The first process is to visualize a time-series of each metric observed in each detector, like Figure 2.10. The trend and values observed at each timestep help determine the accuracy of each metric at each critical geographic point at the granular level.

Then, a time-space diagram is used to validate the macroscopic spatio-temporal relations across detectors in one corridor, like Figure 2.11. Time-space graphs describe the relationship between the location of vehicles in a traffic stream and the time as the vehicles progress along the highway [61]. Note that on or off-ramp detectors are not included when creating a time-space diagram of detectors on a highway.

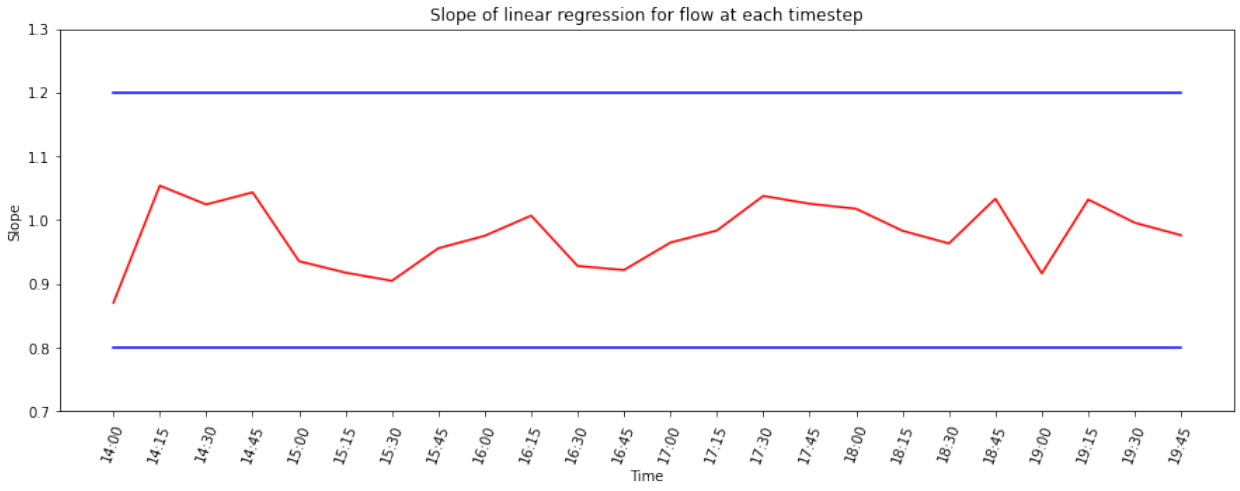


Figure 2.8: Slope of the linear regression for flow at each timestep (red). Setting arbitrary lower and upper bounds of 0.8 and 1.2 (blue), respectively, shows that the simulation does not over- or under-estimate the real flow.

Analysis

After successful validation, the microsimulation can be used for traffic analysis study. This work can help understand the current traffic in a city or predict the impact of a "what-if" scenario. Note that microsimulation can only be used to study effects that do not require changing the input demand data. Changing the demand data will result in comparisons of two different traffic simulation models.

Some examples of scenario changes include implementing new traffic-calming strategies or changes in routing behaviors or driving behaviors. Motivating examples of scenarios that can be analyzed are as follows:

- Changing traffic signal timing plans [145].
- Changing speed limit or adding speed bumps [145].
- Adding turn and/or access restrictions [145].
- Understand the impact of increase in usage of Navigational apps on traffic [33, 31, 32]
- Changing cost function that a portion of the drivers minimizes in their routing choice to understand the impact of eco-routing adoption [8, 64].
- Changing the type of some vehicles to study the impact of mixed-autonomy in traffic [166].

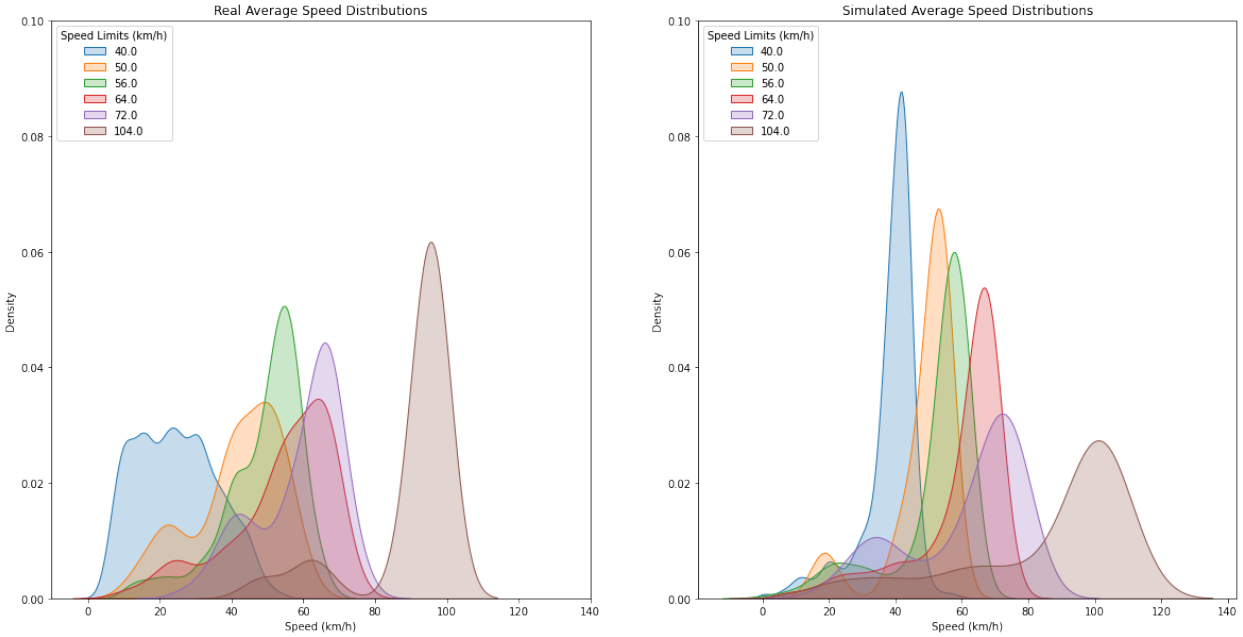


Figure 2.9: Kernel density estimation (KDE) plot of real (on the left) versus simulated (on the right) speeds at each road section. Each distribution is grouped by the speed limit on the road where the speed was observed. The simulation is able to capture a majority of trends seen in real average speed distributions. The simulation better performs on highway data than local road data.

Table 2.1 enumerates some of the metrics that can be used to analyze the impact of such scenarios, alongside with the granularity of measurement.

2.8 Summary

Through the development of a traffic microsimulation of the San Jose Mission district in Fremont, CA, we designed and shared a reproducible process to create, calibrate and validate a large-scale microsimulation. The development of a large-scale traffic microsimulation is a tedious process that took us around 2,500 person-hours, and it is relevant only if case study or A/B experiments cannot be performed and if enough data is available to accurately reproduce demand data. A realistic traffic microsimulation can be used to understand current traffic and estimate policies that might impact routing or driving behaviors without changing the traffic demand (number of trips, departure times, and origins and destinations). If the simulation quality is very high compared to existing literature (flow nRMSE of 47%), it is not good enough yet to be used off-the-shelves by the city of Fremont traffic engineers.

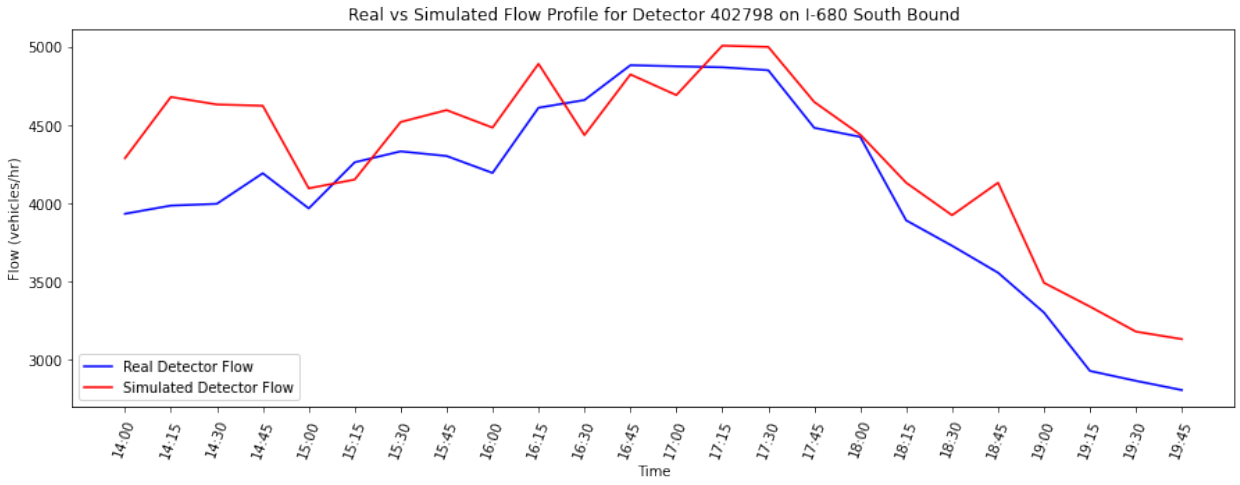


Figure 2.10: Flow profile for a detector on the I-680 South corridor. The trends of observed flow in real and simulated detectors align with each other.

We envision that future research directions about traffic microsimulation should include:

- Considering physical constraints on OD demand calibration regularization terms (like physics-informed neural network [135], or nuclear matrix norms [36]).
- Building on this work to develop a standardized validation toolbox for traffic microsimulation.
- Building on this work to develop a standardized calibration toolbox for microsimulation.
- Establishing clear use cases for when each type of traffic simulation are relevant and for when simulation are not relevant and case study, A/B experiment or simple models should be used instead of simulation.
- Continue developing large-scale simulation models and techniques to continue improving traffic at the metropolitan scale.

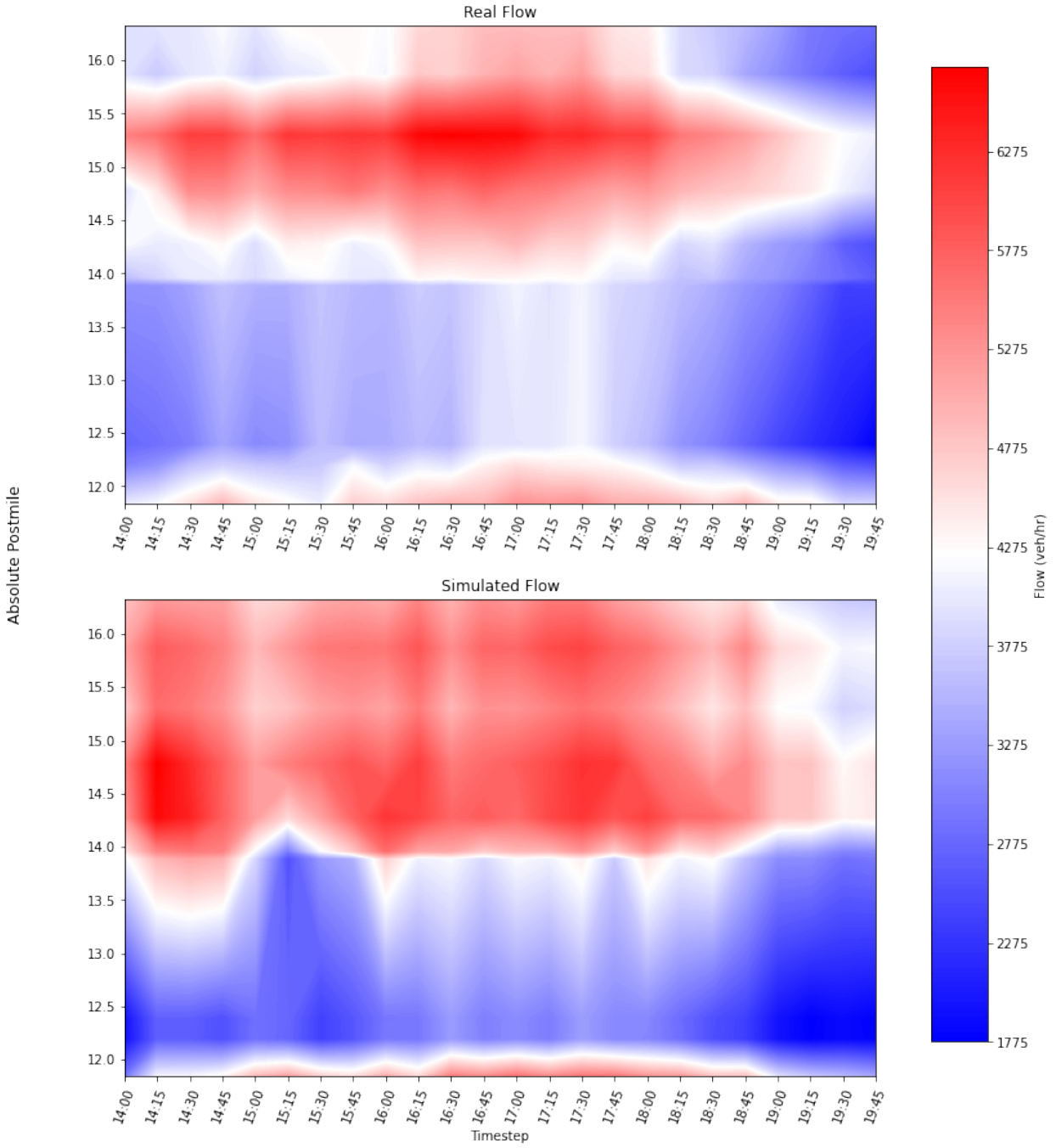


Figure 2.11: Time-space diagram of real versus simulated flow in I-680 South. Patterns of congestion are similar across real and simulated plots.

Type	Output Statistic (Units)	Granularity of Measurement
Transportation Effectiveness	Flow (veh/h)	Network, Centroid, Road, Lane, Turn, Detector
	Density (veh/km)	Network, Centroid, Road, Lane, Detector
	Travel Time (s)	Network, Centroid, Road, Lane, Turn
	Delay Time (s)	Network, Centroid, Road, Lane, Turn
	Stop Time (sec/km)	Network, Centroid, Road, Lane, Turn
	Mean Speed (km/h)	Network, Centroid, Road, Lane, Turn
	Number of Vehicles (veh)	Network, Centroid, Road, Lane, Turn
	Mean Virtual Queue (veh)	Network, Centroid, Road, Lane, Turn
	Number of Stops per Vehicle (#/veh)	Network, Centroid, Road
	Total Distance Traveled (km)	Network, Centroid
	Instantaneous Relative Gap	Network
	Experienced Relative Gap	Network
	Total Number of Lane Changes	Road
	Total Number of Missed Turns	Turn
	Total Green Light Time (s)	Turn
	Total Red Light Time (s)	Turn
Economic	Total Fuel Consumption (L)	Network, Centroid, Road, Turn
	Total Battery Consumption (kWh)	Network, Centroid, Road, Turn
Social & Environmental	Pollutant Weight (kg)	Network, Centroid, Road, Turn
	CO2 Emission (g/km)	Network, Road
	NO Emission (g/km)	Network, Road
	VOC Emission (g/km)	Network, Road
	Particulate Emission (g/km)	Network, Road
	Accessibility	Can be directly assessed through GUI

Table 2.1: Examples of metrics that can be used to analyze scenario changes. Relative Gap refers to the comparison between the current assignment solution to the ideal shortest-route time for all O-D pairs and all departure intervals [68]. The granularity of measurement indicates at which scales the output statistic is computed by Aimsun. "Network" refers to the entire traffic network, "Centroid" each OD centroid pair, "Road" each road section in the network, "Lane" each lane in the road section, "Turn" each turn section of road intersections, and "Detector" each detector on road sections.

Chapter 3

Filling in the missing data: Predicting Traffic Flow on Faulty Traffic Detectors using Machine Learning Techniques

3.1 Overview

[1] Among the primary elements of traffic streams are flow, density, and speed. These elements and their relationship to each other are studied mathematically in traffic flow theory [61]. Traffic flow, together with other elements of traffic, is beneficial in understanding mobility, and flow data is often used in planning and designing highways. When analyzed, it can help guide traffic management measures implemented in a city. Flow data can also help measure delay and congestion in the network, as well as estimate energy consumption and pollution levels.

Flow is usually measured in terms of the number of vehicles per unit time (e.g., vehicles per hour), and the data is usually gathered through empirical studies. Methods for gathering flow data include the use of loop detectors, radars, cameras, closed-circuit televisions (CCTVs), and unmanned aerial vehicles (drones).

Despite advancements in these methods, any of these systems can be subject to failure. For instance, detectors (e.g., loops) occasionally malfunction, cease working, or halt sending data. One example is the California Department of Transportation (Caltrans) Performance Measurement System (PeMS), which occasionally experiences errors due to communications loss or faulty connections, resulting in gaps in the dataset. If these gaps are not filled, PeMS cannot serve its main purpose, which is to provide accurate freeway conditions. Therefore, this chapter aims to predict traffic flow when no data are recorded by the sensors located at

[1] This chapter is part of the proceedings of the International Conference on Transportation and Development 2022 [12].

any specific vehicle detection station (VDS) at a given time.

As a case study, Interstate 680 Southbound direction (I-680 S) was randomly selected. I-680 is a north-to-south interstate highway in the San Francisco Bay Area, running from San Jose, CA in the south to Fairfield, CA in the north.

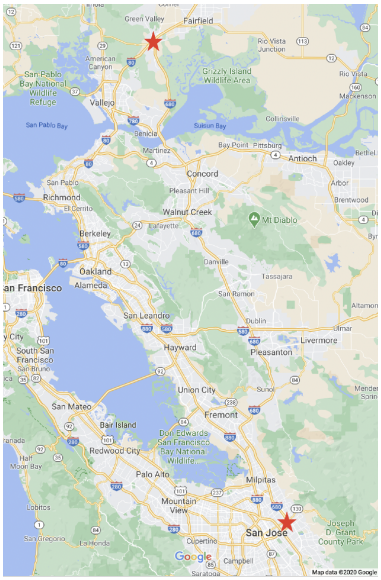


Figure 3.1: The start and end of Interstate 680 (I-680) as marked.

To illustrate the issue of malfunctioning sensors, the percentage of good and bad detectors for I-680 S is shown in Figure 3.2.

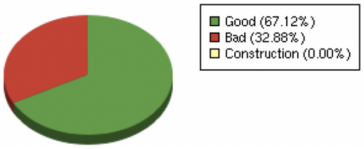


Figure 3.2: Percentage of good and bad detectors for I-680 S.

The work of Tsekeris and Stathopoulos [152] inspired the initial method used in this paper. They analyzed traffic data from traffic detectors in several arterial links of an extended urban network and used principal component analysis (PCA) to model the time series of these data. This technique has been proven to provide a plausible and powerful tool in showing the common patterns of temporal variability. They recognized in their work that the "temporal trend thresholding of the given eigenflow together with the corresponding band of $\pm$2SE is suitable for the long-term analysis of the expected or predictable variations

of traffic flow" which can also capture outliers in cases of unexpected events in the network like bus breakdowns and accidents. In this research, we aim to predict traffic flow using PCA as well as linear regression and random forest models.

This is interesting to investigate given that the current methods being used by PeMS to fill in the gap between the data set is by the use of linear regression from neighbors based on local coefficients, linear regression from neighbors based on global coefficients, temporal medians wherein PeMS looks at data values at similar times and days of the week over a long period of time where the medians of those data values are used to fill gaps and cluster medians [34], [22].

Furthermore, given that the current imputation techniques require information from the neighbors, it has also been the objective of this chapter to introduce models that may not depend on the neighbors but features that could be inherent to the VDS itself such as its ID, postmile (which is the relative location/distance from the start of the freeway), and other features such as time and day of the week using linear regression and a more advanced machine learning technique like a random forest. The use of just a few eigenvectors from PCA as features in a linear regression model, if proven to result in high accuracy of the flow estimates can then help reduce the dimensionality of the data required to estimate traffic flow when no data is available at any VDS and timestep. Otherwise, if this does not meet our desired performance, other techniques introduced can be considered useful. Meanwhile, it is also good to note that this study has been limited to having predictions for the aggregated flow for every five (5) minutes.

The remainder of the chapter is organized as follows: Data which describes the details of our data including their visualization. This is followed by Methods. Next are the Results and the last part is dedicated for the summary.

3.2 Data

The data used in this chapter are five-minute aggregated flow data of each day on I-680 S over a two-week period from March 17, 2019 to March 30, 2019. Before downloading them from the PeMS website, which are filtered to consider only the data from the mainline excluding those from the on and off-ramps. The data downloaded from PeMS have the following columns: 'Time', 'Postmile (Abs)', 'Postmile (CA)', 'VDS', 'AggFlow', 'No. of Lane Points', and '% Observed'. Time is divided into five-minute time steps within the entire day starting from 00:00 to 23:55. Absolute postmile which has its column named 'Postmile (Abs)' indicates the actual distance along a freeway from its beginning to its end. These postmile values increase monotonically with the length of the freeway. Absolute postmiles have been converted by PeMS from the Caltrans Postmiles. Then, it also has the column for the Caltrans Postmiles ('Postmile (CA)') which is then considered as jurisdictional because these are reset to zero at every county line and are assigned to physical boxes and geometric features on freeways when they are built. 'VDS' column gives the ID of the vehicle detector stations. We also have the aggregated flow (AggFlow) at each VDS which is the number

of vehicles for every five-minutes. Then, the number of lane points ('No. of Lane Points') at each VDS location is also given within the data which represents the number of points used to generate the data. The last column is the '% Observed' which tells us whether the aggregated flow has been generated using actual data or partially/fully inferred from the points where there is collected data. The data is then restructured for use in the principal component analysis as described in Section 3.3.

This gives a total of 580,608 flow observations at five-minute time interval on 144 mainline VDS. But, among these observations, only about 63% is based on 100% observation, 27% was imputed after no observations were recorded, and the remaining 10% was also imputed after observations of less than 100% but greater than 0%. After filtering the rows with less than 100% observed flow at each VDS, we end up with 366269 rows out of the original 580608. Then, 19 out of the 114 sensors are not able to record any flow at all.

In addition, we can generate a time series of flow passing through any given VDS corresponding to a specific section of the freeway. The inventory of these VDS from PeMS website can help you locate this approximately on the map. An example of a time series plot is shown for VDS 418817 (Absolute Postmile 0.200) in Figure 3.3. The time series can easily help us identify the variation in terms of peak and off-peak hours of traffic and can later be used to compare the reconstructed flow.

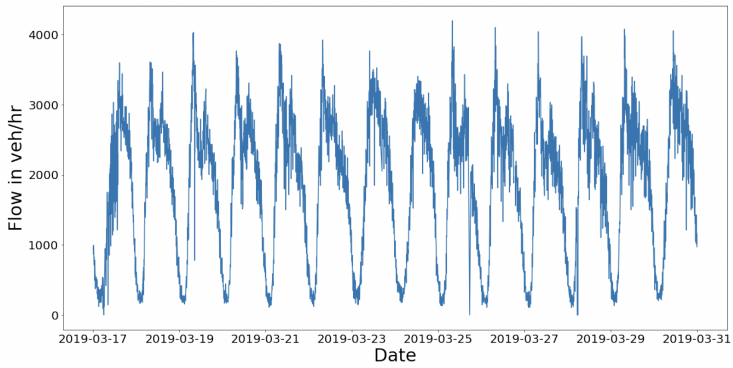


Figure 3.3: Flow time series at VDS 418817 (Absolute Postmile 0.200.

Also, we used k-means clustering to see if it can easily divide the data into clusters using only the two columns of the data: VDS ID ('VDS') and aggregated flow ('AggFlow'). The no. of clusters used is four (4) based on the elbow rule using the plot in Figure 3.4 of the variation of the sum of squared distances in the function of the number of clusters. Meanwhile, the results of k-means clustering can be seen in Figure 3.5. From among the clusters, it is only Cluster 1 which is very easy to interpret showing all the flow when they are below 4000 vehicles per hour, while the other three (3) clusters aren't giving any very intuitive interpretation, this might be because of the limited no. of inputs used in clustering. Despite this, the clusters were used to see if they can add a predictive power in the linear

regression model which is almost similar to considering the clusters as described in Caltrans 2020.

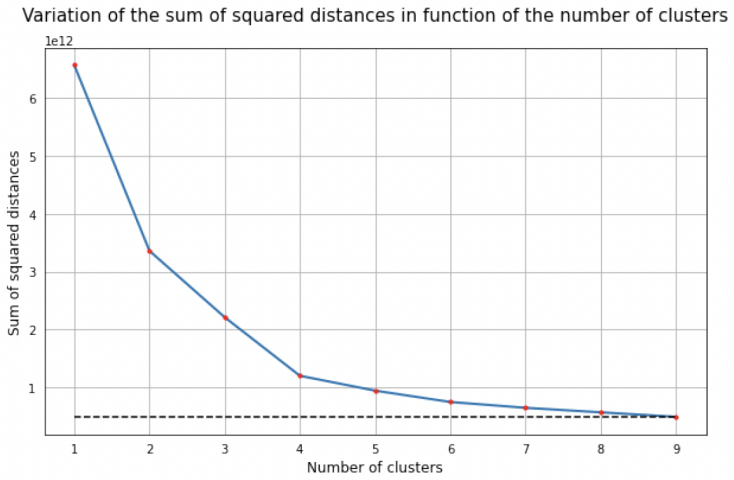


Figure 3.4: Variation of the sum of squared distances in function of the number of clusters.

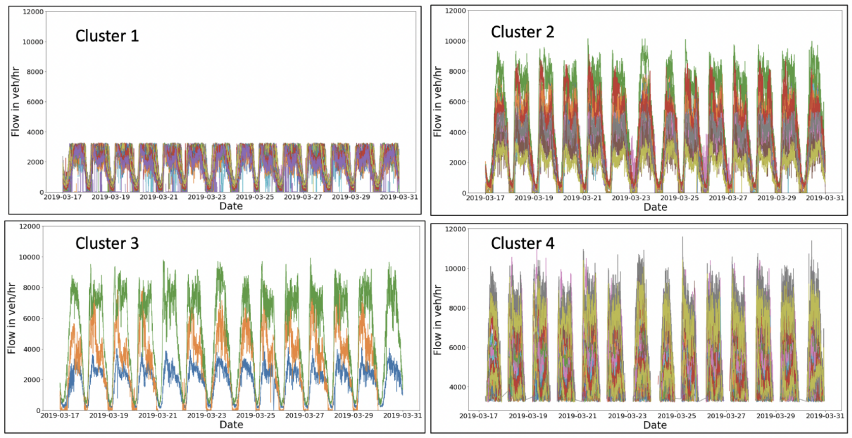


Figure 3.5: Four clusters with time series of flow.

3.3 Methods

We first used principal component analysis on traffic flow as adopted from the method introduced in the work of [152]. They calculated for eigenflows which they defined as "a

time series that captures a common pattern (or source) of temporal variability in traffic flow at the network level. Each traffic flow time series is expressed as a weighted sum of eigenflows and the corresponding weights reflect the extent to which each source of temporal variability is present in the given traffic flow."

Similarly, m is used as the number of columns corresponding to the mainline detectors/VDS of I-680 S. Then, t is the number of successive days in which the detector data are collected which in this case is 14 days (*i.e.,* two weeks from Mar. 17 to Mar. 30, 2019), and τ is the number of time intervals where flow data was collected. Then, matrix X with dimension $p \times m$ is defined which is then referred to as measurement matrix where $p = \ no.\ of\ rows = \tau \times t$. From here, we have column vectors X_i which then represents the i-th traffic flow time series, and each row j denotes the particular point in the time series in which traffic flows have been collected at interval j which in this case is at every five (5) minutes. In short, one data point (*i.e.,* row) is the flow on every road section for five (5) minutes for a given time of the day while one feature (*i.e.,* column) is the flow on one VDS. After rearranging the data conforming to the description above which results in a final dimension of 3996×95 (there appear to be 32 missing time steps). This is then divided into train and test set at a ratio of 1:1 to make sure that I can use the eigenvectors from the train set as features into the linear regression to predict flow on the test set. So, the dimension for both subsets is 3996×47. We needed to exclude the data from one VDS so they have the same no. of columns that address the compatibility issue when adopting the eigenvectors from the train set to the test set. Then, the reconstruction of the training set from a smaller number of eigenvectors is calculated and its overall accuracy is calculated using equation 3.1, where X corresponds to the data, $\bar{X}$ is the mean, and $\hat{X}$ is the reconstruction value from PCA.

$$\text{Accuracy } = 1 - \sqrt{\frac{\sum_{i=1}^{96}(X - \hat{X})^2}{\sum_{i=1}^{96}(X - \bar{X})^2}} \tag{3.1}$$

Then, the histogram of the flow data is visualized in Figure 3.6. It can be seen from the histogram that there are many flow observations that fall within the range of lower values (skewed to the right). Hence, the data was standardized by removing the mean and scaling to unit variance which was done using the 'StandardScaler' module of sklearn [114]. It calculates the standard score of sample x by having $z = \frac{(x-u)}{s}$ where u is the mean of the training samples or zero if $with\_mean = False$, and s is the standard deviation of the training samples or one if $with\_stdev = False$. The resulting histogram after standardization of data is shown in Figure 3.7.

Using only the no. of eigenvectors required to explain at least 90% of the variance, we trained a linear regression model using the eigenvectors and This might be considered as a naive approach but as a baseline, we are interested to see if this somewhat has a predictive power. The linear regression equation using PCA has its equation described in equation 3.2.

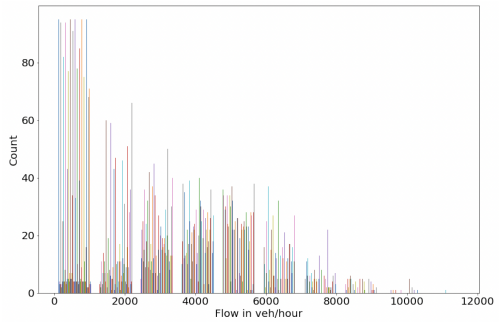


Figure 3.6: Histogram of flow data (no. of bins = 10).

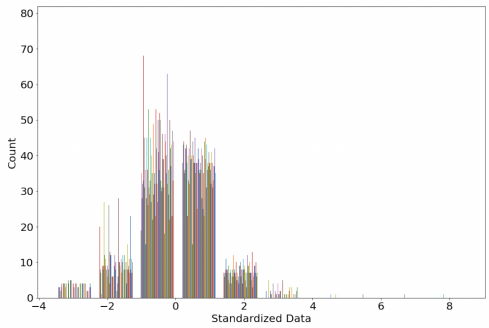


Figure 3.7: Histogram of standardize flow data (no. of bins = 10).

$$\text{AggFlow}_i = \beta_0 + \beta_i \sum_{j=1}^{n} \text{value\_from\_eigenvector}_{ij} + \epsilon \tag{3.2}$$

In equation 3.2, i represents the row location of the value from the corresponding eigenvector based from j which represents the order of eigenvectors while n is the no. of eigenvectors used in the model.

Meanwhile, for the other machine learning techniques, the original structure of the data with columns described in Section 3.2 was used with an added column corresponding to the day of the week as another feature. This time, 80% of the data was used for training while 20% for testing. We can note in the linear regression equation below that time and day of the week are both treated as categorical variables.

$$\text{AggFlow}_{\ i} = \beta_0 + \beta_1(\text{ VDS }) + \beta_2 \text{ Abs\_Postmile } + \beta_j \sum_{j=3}^{299} \text{ Time}_{\ j} + \beta_k \sum_{k=300}^{305} \text{ Day}_{\ k} + \epsilon \tag{3.3}$$

Similar form of the equation above was used in another linear regression model where we added the resulting clusters from k-means clustering that was previously applied as described in Section 3.2 to see if it improves the performance of the model. The corresponding linear regression equation is shown in equation 3.4.

$$\begin{aligned} \text{AggFlow}_i = \beta_0 + \beta_1(\text{VDS}) + \beta_2(\text{AbsPostmile}) + \beta_3(\text{Cluster}) \\ + \beta_j \sum_{j=4}^{300} \text{Time}_j + \beta_k \sum_{k=301}^{306} \text{Day}_k + \epsilon \end{aligned} \tag{3.4}$$

The regression type of random forest is also introduced to see if a more advanced machine learning technique would be able to generate a higher accuracy on both train and test set. The module 'sklearn.ensemble.RandomForestRegressor' [114] was used with its default parameters where we have the number of trees in the forest being equal to 100. This method uses random decision forests as an ensemble learning method, regression in this case, where we use the mean or average of the individual trees to generate prediction. This model is known to be hard to interpret but usually results to high level of accuracy when applied on the test set. The features considered in random forest is similar to that of the second linear regression model with its features described in equation 3.3.

'Score' was the metric chosen to evaluate all the four models considered on this paper. The score of the models are calculated for both the train and test set for each model as the coefficient of determination (R2) of the prediction. Root mean square error between the predicted and true values of each model on the test set are also reported in this paper.

3.4 Results

On the left of Figure 3.8 is the cumulative sum of the variance explained as we increase the number of components used in PCA. It can be noticed that it takes at least 4 components to explain at least 90% of the variance, with principal component (PC) 1, PC2, PC3, and PC4 explain 79.82%, 4.76%, 4.44% and 1.88% of the variance respectively. On the right of Figure 3.8 are the corresponding plots of PCs 1, 2, 3, and 4. The plot gives us an idea how the detectors are compensating for the variations in traffic flow based on each eigenvector.

Then, in Figure 3.9 is an example of flow reconstruction in one VDS from the training data using four (4) eigenvectors from PCA. Only four (4) was used since it's enough to explain at least 90% of the variation in flow. The overall accuracy of the flow reconstruction is calculated being equal to 0.7254. Meanwhile, as discussed in Section 3.3, we developed three (3) linear regression models and one random forest model. Their performance based on the corresponding score for both train and test set of each model is summarized in Table 3.1 and the biplots of the predicted and true values of the test set are shown in Figure 3.10.

It can easily be noticed that random forest stands out in terms of all the metrics considered as seen in Table 3.1. Also, the biplot of random forest in 3.10.d. is the best in

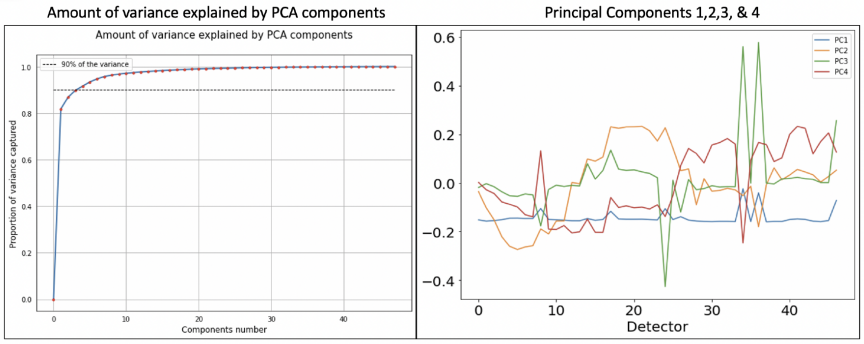


Figure 3.8: Cumulative sum of the variance explained at increasing no. of components (L); Principal components 1, 2, 3, and 4 (R).

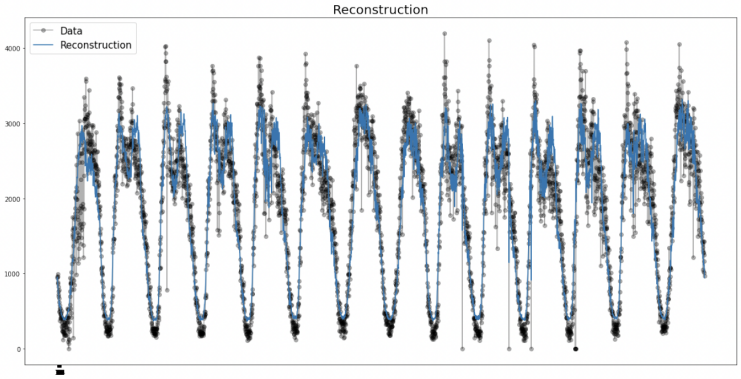


Figure 3.9: Example of flow reconstruction for one VDS using 4 eigenvectors.

Model	RMSE	Train score	Test score
Linear regression with PCA	1945.68	0.9120	0.1372
Linear regression w/ select features	1385.65	0.6130	0.6120
Linear regression w/ clustering	1166.22	0.7263	0.7252
Random forest (no. of trees = 100)	1089.06	0.9513	0.7595

Table 3.1: Comparative summary of scores and RMSE of the models.

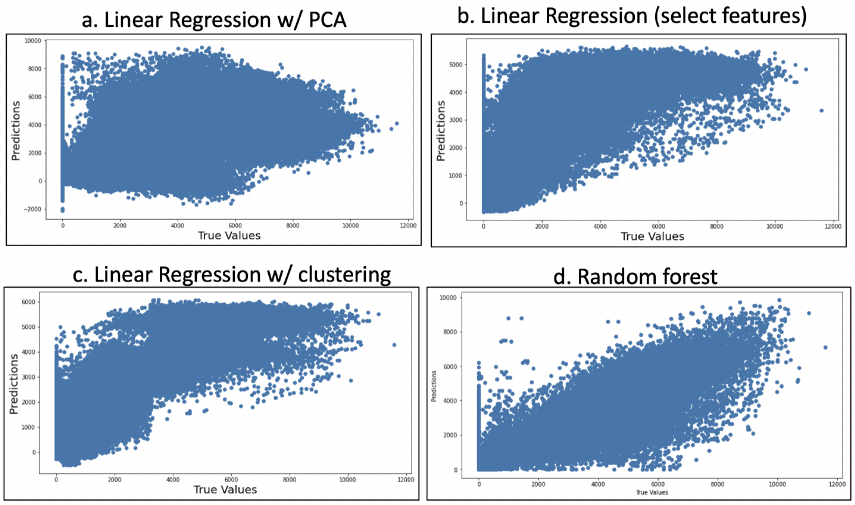


Figure 3.10: Biplots of predicted vs. true flow values of the test set.

comparison with the three other models. But, its performance isn't too far from the linear regression model where we added the cluster from k-means as feature considering both the RMSE and test score making it the second best performing model among the four. Meanwhile, the model which performed the least on the test set is the linear regression where we used the values from the eigenvectors of the training as features but its train score comes second to random forest in terms of performance.

In addition, we also illustrated in Figure 3.11 the feature importance to see which among them matters the most based on the random forest model, making the the top five are postmile, VDS, when it's a weekend (Saturday and Sunday), and when the time is 01:55 AM.

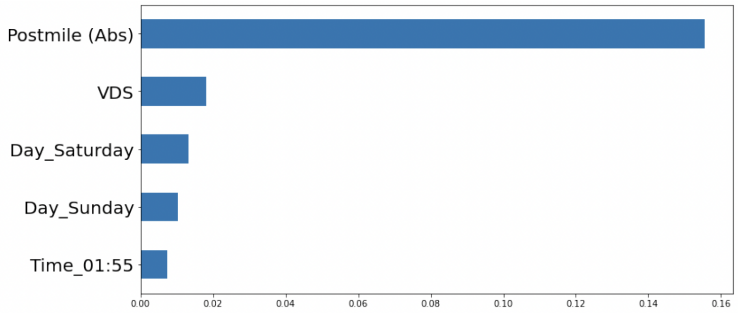


Figure 3.11: Five (5) Most Important Features.

Finally, illustrated in Figure 3.12 an example time series of the predicted flow for one of

the faulty VDS which in this case is for VDS 407194. It recorded 0% observation for the two-week period covered. We can see that the prediction is able to capture almost the same variability or pattern in flow when compared to Figure 3.3.

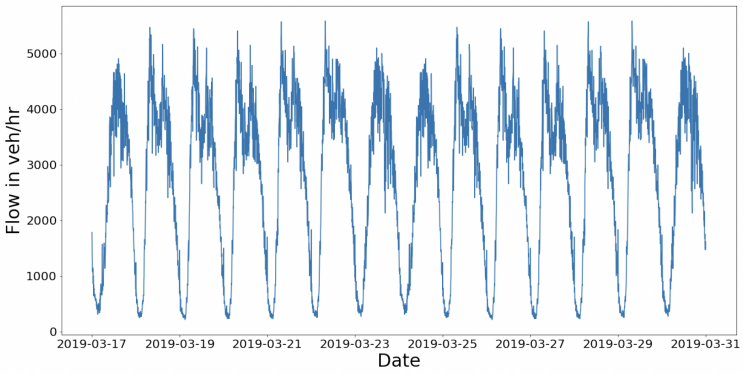


Figure 3.12: Time series of predicted flow for VDS 407194.

3.5 Summary

The linear regression using the eigenvectors from the training set as features to predict flow on the test set was found to overfit the training set. Meanwhile, the linear regression model using time, absolute postmile (relative location of VDS from the start of the freeway), VDS ID, and day of the week as independent variables performs much better on the test set compared to the former. This is further improved when the clusters from k-means were added as feature to the model.

But, random forest with just the default parameters performs the best among the models indicating that 'Postmile (Abs)' and 'VDS ID' as the two most important features. Meanwhile, it appears that distinguishing that the flow is gathered on a weekend, Saturday and Sunday, would be an important feature as well. This makes sense given that in most cases, flow distribution on weekends differs from weekdays.

Despite performing the best on both train and test set, we noticed that the random forest has the longest training time. It took about 37 minutes 48 seconds for it to finish running the cell containing the random forest model (training and testing) while it was just 5.11 seconds for the linear regression with clustering to finish using a 2018 Macbook with 2.6 GHz 6-Core Intel Core i7 and 16 GB RAM with default settings on Jupyter Notebook ran in Google Chrome. So, it may be worthy to revisit if we care for the training time which could depend on how often we may want to train a new model from the data especially the difference in terms of accuracy on the test set between the two models are not too far.

Some future work after making this project could be addressing the lengthy training time for random forest and evaluating the performance of the model, which if we consider the best

performing model here would be the random forest, using other freeways. Also, it might be of interest for many to consider other ways of using the features; like instead of having the days of the week, we may try classifying whether it's a weekday or a weekend; also try other machine learning algorithms to see if they can further improve the accuracy especially on the test set.

Lastly, among the interesting things we discovered is that PeMS generate so much data for all the freeways across California. Scaling this project to all freeways can be a big challenge. In that case, one must really be familiar with its design architecture to be able to successfully implement this kind of study and include it within the features of PeMS.

Chapter 4

Impact of navigation apps on congestion and spread dynamics on a transportation network

4.1 Overview

[1]The rising popularity of traffic navigation applications (apps) such as Google Maps, Apple Maps, and Waze, has resulted in the redirection of an increasing number of vehicles around congested freeways and boulevards, leading them through residential areas with lighter traffic and fewer stoplights that are not intended for large volumes of traffic. In some instances, these apps even divert drivers to streets ill-suited to function as thoroughfares. These effects of navigation app usage have been noticed as early as 2015. Notable examples include Sherman Oaks in Los Angeles, California, USA [98] and, more recently, San Jose Mission District in Fremont, California, according to the city's Public Works Department.

Navigation app usage has not resolved Braess's paradox [32]; rather, it has given rise to a new form of Braess's paradox [23]. In the toy example experiment performed in ref. [32], they found that navigation apps can lead to increased overall travel time for all users. Given that, this study aims to examine the impact of information-aware routing, facilitated by navigation apps, on traffic flow and congestion. In addition, this chapter investigates the extent of the impact by evaluating how selfish routing algorithms contribute to the redistribution of traffic in previously uncongested areas and the consequent implications on traffic congestion. If these effects are not adequately acknowledged and addressed, urban areas may experience increased traffic on roads that are ill-equipped to support heavy traffic flow, leading to exacerbated congestion and the deterioration of infrastructure (*e.g.,* pavement potholes). Consequently, it is crucial to communicate the impact of information-aware routing among residents and other stakeholders. Previous studies illustrating the negative externalities of navigation apps at different adoption levels also include refs. [144] and [54]. Ref. [133]

[1]A version of this chapter was published in Data Science for Transportation journal [11].

emphasizes that the sharing of real-time traffic information with all users of a transportation network is sub-optimal.

This sub-optimality can be associated with the individual decisions made by road users (*i.e.,* drivers). These decisions, known as selfish routing, involve drivers individually optimizing their travel times without considering the impact on others [116]. This local optimization does not achieve global optimality, resulting in decreased network performance [127] and an increased price of anarchy. The price of anarchy is defined as the ratio between user equilibrium (UE) and system optimum (SO) [175]. UE, described by Wardrop's first principle of traffic flow, is achieved when no road users can switch to a road with lower cost (*i.e.,* travel time or distance). In transportation, UE is one form of Nash equilibrium [106]. SO, as defined by Wardrop's second principle, involves users minimizing the total network travel cost [161]. UE is generally considered sub-optimal and greater than SO [175] & [126]. "The use of apps leads to a system-wide convergence towards Nash equilibrium" [33].

To understand what is happening in the network, several models and approaches are commonly used to analyze and predict traffic congestion patterns and how they spread. These include traffic flow models such as the Greenshields Model which is a simple model that assumes a linear relationship between traffic density and traffic speed [67]; Lighthill-Whitham-Richards (LWR) Model which is a macroscopic traffic flow model that describes the evolution of traffic density and speed based on conservation principles [90] & [124]; and, Cell Transmission Model (CTM) which divides the road into discrete cells and models traffic flow as a flow of vehicles between cells [46]. There are also models built on percolation theory [5], machine learning-based approaches, and queueing theory as discussed in the work of Saberi, et al. [128] wherein they highlighted in their work that, although widely employed, existing models are predominantly constructed on input (arrival) and output (departure) rates. Consequently, they introduced a simplified model to examine the spread of traffic congestion within a network, employing a basic contagion model that relies solely on the average speed of road segments — a concept further investigated in this paper by comparing the results to conventional key performance indicators (KPIs) such as flow, density, speed, travel time, delay time, and queue length.

Contribution: We focus on examining the changes and impacts of information-aware routing through the use of traffic assignment models. Our key contribution lies in the adoption of the Susceptible-Infected-Recovered (SIR) model to study congestion propagation and dissipation, drawing an analogy to the spread of diseases as depicted in Figure 4.1. Traditionally employed for disease spread analysis, the SIR model is repurposed to investigate congestion propagation within the network and understand how routing choices influence congestion spread, aiming to identify an optimal user range for dynamic routing to minimize average costs and achieve system optimality. We demonstrate that parameter fitting to this model using negative log-likelihood can effectively explain the spread of congestion in the network, without relying on conventional examination of KPIs.

We used this method to describe how traffic spreads and recovers over time. This helped us investigate how fast congestion spread over the network in the case of the city of Fremont using the results of simulations and comparing them across the scenarios described in Section

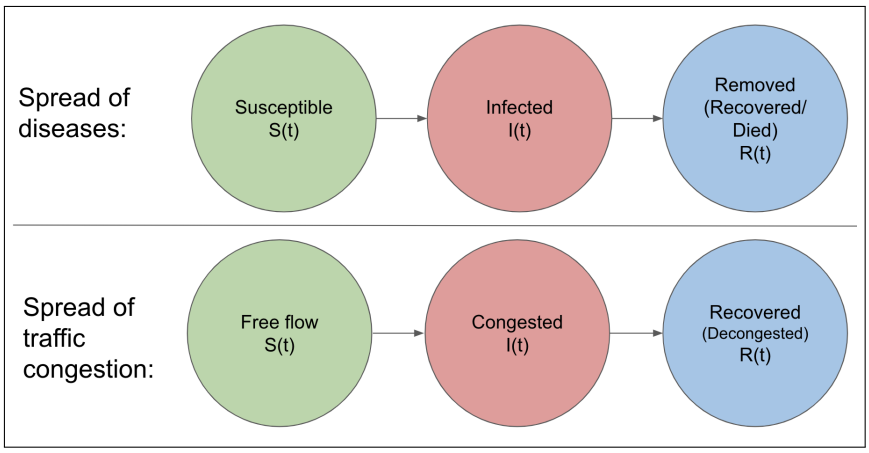


Figure 4.1: Analogy between the spread of diseases and the dynamics of traffic congestion using SIR model.

4.3. To summarize, we take into account the different percentages of users that follow routing from navigation apps which can be represented by dynamic en-route routing behavior, while the rest of the users follow fixed routing behavior. We also discuss how the negative log-likelihood, which falls under the broad category of statistical estimation and optimization methods, particularly within the fields of statistics and machine learning, can be employed to estimate the parameters of the SIR model. This method captures the effects to the overall traffic system, the evaluation we have shown is as a collective or aggregated result rather than the performance of individual cars.

Significance: With the growing interest in autonomous vehicle research, it is essential to recognize that these vehicles will rely on route recommendations and guidance from algorithms employed by navigation apps. Gaining insights into the potential impacts of varying penetration rates on fixed and dynamic en-route routing behaviors could motivate future researchers and scientists to reassess the design of routing algorithms, especially for navigation apps. Analyzing people's movements through information and communication technologies could provide valuable insights into changes in routing patterns at various scales leading to a better understanding of human mobility.

4.2 Data and Simulation Development

To conduct our experiments, we developed a microscopic traffic simulation for the San Jose Mission District, focusing on Interstate Highway 680 (I-680) and Mission Boulevard (State Route 238/262) in Fremont, CA, USA, using Aimsun Next 22.0.1. [2]. The required inputs for a microsimulation are a road network and a dataset of timed origin-destination demand (TODD). Actual traffic data are also used to calibrate and validate the simulation. The

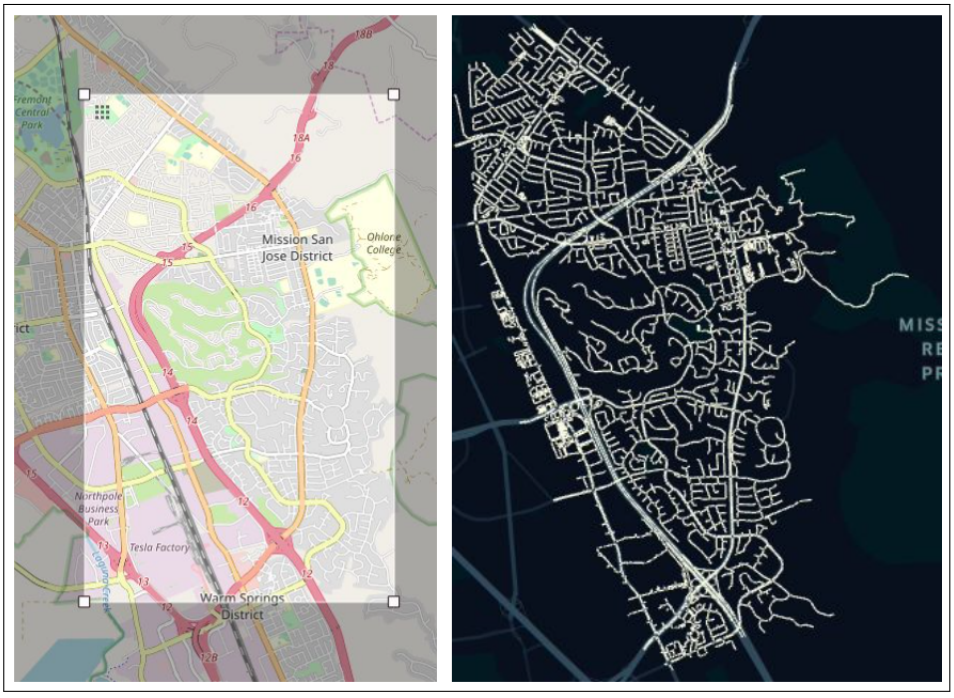


Figure 4.2: OpenStreetMap network with the bounding box on the left and the corresponding Aimsun network after cleaning on the right.

discussion in this section is also detailed in 2.

Road network: The road network is composed of road sections connected through signalized or unsignalized intersections. To create this network, we downloaded the OpenStreetMap (OSM)[70] network model using the bounding box defined by the following coordinates: North: 37.5524, East: -121.9089, South: 37.4907, and West: -121.9544 as shown in figure 4.2, which was cleaned in ArcGIS, a geographic information systems (GIS) software. After importing the network into Aimsun. Google satellite, Maps, and StreetView images were used to perform manual adjustments to ensure the correctness of connectivity, yield and stop sign locations, and lane counts.

Then, we used the data provided by the City of Fremont to set the road speed limits. Meanwhile, road capacities are adjusted using the data from the Behavior, Energy, Autonomy, and Mobility (BEAM) model which is an open-source agent-based regional transportation model [10]. Then, the traffic signal plans including the road detectors, ramp meters and the master control plan from the city and CalTrans were added using the Aimsun graphical user interface (GUI).

In summary, the modeled network has 4,397 links with a total of 393.27 km section length. This includes 111 freeway sections, 1,370 primary road and secondary road sections, 2,916 residential road sections, and 2,013 nodes (intersections), 313 of which have stop signs

and 37 of which have traffic lights (26 operated by the city and 11 operated by CalTrans).

Origin-destination demand: The origins, destinations, and departure times for every vehicle are aggregated into TODD matrices. Origins (or destinations) are clustered into transportation analysis zones (TAZ), which are bijective to the set of centroids connected to internal or external entry or exit points in the network.

The 20.8 square kilometers network area is divided into 76 internal centroids and 10 external centroids. Departure times are aggregated into 15-minute time intervals between 2-8 PM. Each simulation runs for this six-hour period. Based on real-world observations, congestion forms from 2-4 PM, peaks between 4-6:30 PM, and dissipates from 6:30-8 PM. About 130,000 vehicles are modeled (including 62,000 commuters or 'through' traffic and 68,000 residents or local demand). In this work, the demand data was derived from the SF-CHAMP demand model [82] from the San Francisco County Transportation Authority and from a StreetLight study performed for the City of Fremont.

After this, we also performed calibration and validation of the model. To do this, we used actual ground flow data from 56 city flow detectors and 27 CalTrans Performance Measurement Systems (PeMS) detectors [41]. We also used actual speed data from MapBox API. Also, while the model is only showing a six-hour window, it was calibrated against data for three (3) weekdays (Tuesday-Thursday) that, according to the city of Fremont, are known to capture the more typical behavior of traffic and are more of a concern for them, given congestion forms during this time.

4.3 Scenarios

Using the calibrated and validated simulation model, scenarios are developed to understand the impacts of navigation app usage. To do this, the total traffic demand, $\mathcal{T}_{ij}$, within any given origin, i, and destination, j, is split by α which is the percentage of users following fixed routing behavior and the remainder of the demand, $1-\alpha$, follows dynamic en-route routing behavior as described in equation (4.1). The parameter α is varied at 10% interval from 100% to 0%. This setup is similar to the work of Shaqfeh, et al. [133].

$$\tilde{\mathcal{T}}_{ij} = \begin{cases} \alpha \mathcal{T}_{ij} \\ (1-\alpha)\,\mathcal{T}_{ij} \end{cases} \tag{4.1}$$

To determine the paths that will be taken by users following fixed routing behavior, the static user equilibrium (SUE) is first solved, and the resulting paths are then saved, serving as input to the simulation. We assume that the path assignment from SUE represents the paths that drivers will take when they remain on their known routes from their origin to their destination, ideally resulting in the shortest path. Given that it was solved using SUE, there can be more than one path between any given origin-destination (OD) pair, ij. A lower percentage of SUE path assignment users corresponds to a higher percentage of app users in the scenario. For the dynamic en-route routing behavior, we used stochastic route choice (SRC) for route assignment to allow the possibility of users changing paths while on

their way to their destinations, which typically occurs when people actively use navigation apps while driving. The probabilities of being assigned to a path are calculated using the C-Logit model. The routing behavior compliance setup is illustrated in Figure 4.3.

In total, there are 11 scenarios considering $\alpha = 100, 90, 80, ..., 0\%$ in equation (4.1) with three (3) replications per scenario which are varied by the input seed. The results are then averaged.

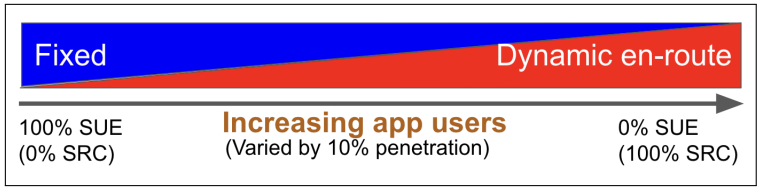


Figure 4.3: Routing behavior compliance setup. The routing behaviors of the users are divided into 'Fixed', where the paths used are determined by static user equilibrium (SUE), and 'Dynamic en-route', which uses route assignment based on stochastic route choice (SRC) varied at 10% intervals.

Routing

As previously mentioned, the drivers can either follow fixed or dynamic en-route routing behavior. The paths to be used by static routing vehicles were determined using static traffic assignment (STA) which was solved using an optimization problem with the classical Rosenthal potential [125] as the objective function and variational inequalities to balance demand and path flows, and ensure non-negativity of flows. This is solved as an optimization problem defined in ref. [113]. The Frank and Wolfe algorithm [57] is used to assign traffic using shortest paths and adjust them based on the cost functions of the links. This gives the traffic assignment results at the end of the set time horizon. This is resolved at 15-minute intervals, where the results from the previous time step serve as input for calculating the updated costs for the current time step. This approach allows us to capture the possibility of a driver following a fixed route, who might be aware of an alternate route upon departure that could offer an equal or less cost to the well-known shortest path.

Meanwhile, the paths used by dynamic en-route users are determined using stochastic route choice (SRC) using the C-logit model, in which the probability of choosing a path p, P_p, is defined in equation (4.2). C-logit considers an additional parameter called the commonality factor, CF_p, which is defined in equation (4.3) to eliminate the drawback of independence of irrelevant alternative (IIA) assumption of the conventional multinomial logit (MNL) model where it considers that the relative odds of choosing one alternative over another are unaffected by the presence or absence of other alternatives [37]. Consistent with the MNL model, it uses the perceived utility for a chosen path p denoted as V_p (changing

it to p' for an alternative path), and a scaling parameter, θ. The perceived utility for both the chosen and alternative paths is the negative of the cost of the path which is equal to $-CP_p/3600$ where CP_p is the cost of path p, measured in hours. CP_p is taken as the sum of the travel time of all the links composing a given path and *3,600* is the conversion factor from hours to seconds. The link costs are updated every one (1) minute, so the probabilities calculated from C-Logit are also updated, and users have the opportunity to switch paths according to these values.

$$P_p = \frac{\exp\left[V_p/\theta - CF_p\right]}{\sum_{p' \in K} \exp\left[V_{p'}/\theta - CF_{p'}\right]} \tag{4.2}$$

$$\mathrm{CF}_p = \zeta \cdot \ln \sum_{p' \in K} \left[\frac{L_{pp'}}{L_p^{1/2} L_{p'}^{1/2}} \right]^{\Psi} \tag{4.3}$$

where $L_{pp'}$ is the cost of links common to paths p and p', and L_p and L'_p are the individual costs of paths p and p' respectively from the set of paths K. Meanwhile, ζ and Ψ are the parameters of CF_p. Larger values of ζ place greater weighting on the commonality factor, and $\Psi \in [0, 2]$ has less influence than ζ and has the opposite effect. The simulations use $\theta = 1, \zeta = 0.3, \Psi = 1$. These parameters were determined in the model calibration and validation as discussed in Chapter 2.

The use of both SUE and C-logit allowed the exploration of other feasible routes between any origin-destination pair considering different values of α, an illustration of this is shown for the the top origin-destination pair as shown in figure 4.4.

4.4 Spread of congestion

The simplest model of the spread of a disease over a network is the Susceptible-Infected-Recovered (SIR) model of epidemic disease. Despite being originally a model for epidemic diseases, it has been used in many applications including the spread of traffic congestion as originally studied in [168] and applied in an empirical study later in [128]. The SIR model is appropriate given that traffic congestion propagates over time and space, similar to an epidemic disease. Variations of this model include the SI model, which eliminates the recovered phase, often applicable to diseases in plants where they would eventually die and not recover, and the SIS model, which considers that anyone who recovers could belong again to the susceptible group. The SIR model is also more appropriate when the analysis is focused on a shorter time span, such as rush hour, which applies to the presented case study of Fremont, CA. Furthermore, in the understanding of routing, we do not anticipate a full gridlock or complete jam. The simulation accounts for the selection of alternate routes and the eventual dissipation of demand.

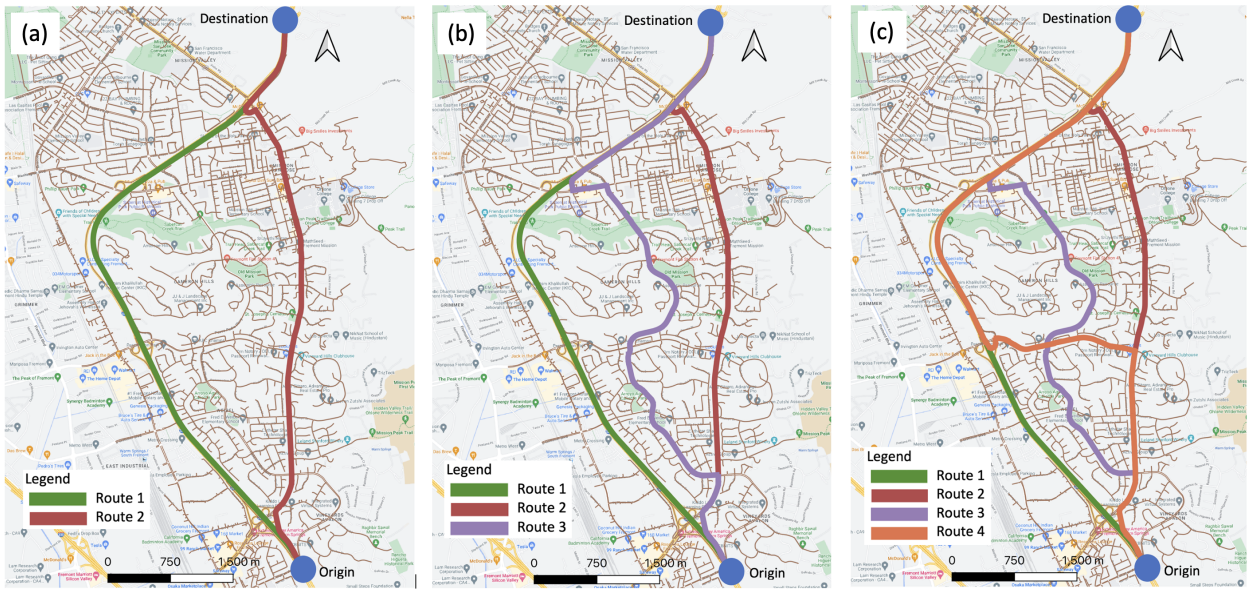


Figure 4.4: Routes for the top origin-destination pair at varied percentages of dynamic en-route users. This illustrates the case when *(a)* dynamic en-route = 0%, *(b)* dynamic en-route = 30%, and *(c)* dynamic en-route = 60%. It can be noticed here how alternate routes would pass through the city and on less-utilized local roads.

SIR Model

The spread of congestion can be described using parameters β and γ, which describe congestion propagation and congestion dissipation rate, respectively. The simple SIR model uses the ordinary differential equations shown in equations (4.4)-(4.6) which can be adopted in understanding congestion propagation and dissipation in a network [168] as supported by empirical studies [128]. The inputs to these equations are the following:

$S(t)$ = fraction of free flow links
$I(t)$ = fraction of congested links
$R(t)$ = fraction of recovered links
These equations describe transmission, propagation, and dissipation rate respectively.

$$\frac{dS(t)}{dt} = -\beta S(t)I(t) \tag{4.4}$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \tag{4.5}$$

Figure 4.5: Spatial distribution of congestion in the Fremont network at 6:00 p.m., showing the congested links *(coded red)* on the network when 90% of the drivers follow dynamic en-route assignment when *(a)* $\rho = 0.3$, *(b)* $\rho = 0.6$, and *(c)* $\rho = 0.9$.

$$\frac{dR(t)}{dt} = \gamma I(t) \tag{4.6}$$

The links are classified whether they are congested or not based on equation (4.7). The congestion classification uses a threshold ρ which defines the tolerance for congestion with $\rho \in [0, 1]$. Higher values of ρ correspond to lower tolerance for congestion, (*e.g.,* at a set speed limit of 60 miles per hour (mph), when ρ is 0.3, a speed of 53 mph is not classified as congested but would be classified as congested if we changed ρ to 0.9). This means that with a higher value of ρ, more links are classified as congested as illustrated in Figure 4.5. To show more links being congested, we chose $\rho = 0.9$.

This allows us to do a binary classification to determine whether a link or section c_i is congested ($c_i(t) = 1$ when $\lambda_i(t) < \rho$) or not ($c_i(t) = 0$ when $\lambda_i(t) \geq \rho$). Then $I(t)$ is calculated using equation (4.8). To consider all the links of the network, it can be taken that $S(t) + I(t) + R(t) = 1$.

$$\lambda_i(t) = \frac{\text{speed of the link } (t)}{\text{speed limit}} \tag{4.7}$$

$$I(t) = \frac{\text{number of congested links } (t)}{\text{total number of links}} \tag{4.8}$$

Using this, the parameters β and γ are estimated. Note that given the equations for SIR, these two parameters are interdependent. So, the final results are the calculated values of the basic reproductive number R_0 which is the ratio of propagation over recovery rates vs. scenarios to explain the extent to which congestion builds up in a network and how fast it recovers. R_0 is calculated using equation (4.9). Higher values of R_0 indicates that congestion is spreading faster and dissipating slower.

$$R_0 = \beta/\gamma \tag{4.9}$$

Parameter Estimation

To estimate the parameters for a SIR model, we employed a three-step approach that involves initializing the parameters, simulating the model, and refining the estimations based on the original data. The estimation of the congestion propagation rate (β) and congestion dissipation rate (γ) is essential for accurately predicting the dynamics of congestion spread.

1. *Initialization:* We provided initial guesses for the model parameters ($\beta_0 = 0.4$, $\gamma_0 = 0.25$) and $R(0) = 0$ assuming there are no links at the recovery stage at the start of the simulation. $I(0)$ which is the fraction of congested links at $t = 0$, is calculated from the sum of c_i divided by the total number of links $N = 4,397$ and the links in free-flow is $S(0) = 1 - I(0) - R(0)$.

2. *Simulation:* With the initial conditions and parameter guesses, we simulated the SIR model to predict the number of congested links at each time step.

3. *Parameter Refinement:* We compared the modeled congested links with the actual data and adjusted the model parameters to minimize the difference the negative log-likelihood function as shown in equation (4.10). This optimization step is achieved by using the Nelder-Mead method, [108] & [60], to find the best parameter values that fit the data.

$$NLL = -\sum_{i=1}^{n} \log(PoissonPMF(y_i, \hat{y}_i)) \tag{4.10}$$

where y_i is the actual data, $\hat{y}_i$ is the model output, and $PoissonPMF$ is the probability mass function (PMF) for Poisson distribution calculated using the SciPy package [156].

4.5 Results

In our study, we explore the behavior of navigation apps in the context of traffic congestion management. Navigation apps can be interpreted as a greedy algorithm in a broader sense, where they recommend the best available routes to individual users at any given time. For instance, during periods of high congestion on highways, the apps often suggest alternative routes through less-utilized local roads. Importantly, this individual behavior has a cumulative effect on the overall traffic system, leading to increased use of alternative routes and alleviation of congestion in previously uncongested areas. To assess the impact on the overall traffic system, our evaluation presented in this section is collective or aggregated results rather than focusing on individual car performance.

Assuming a certain percentage of drivers will receive information while en-route to their destination, while others will not and will stick to a route more familiar to them or one they would have preferred from their point of origin. In reality, navigation will be used, but compliance is something that's hard to determine. The route choice model we selected allowed us to simulate this behavior. The actual level of compliance can be determined if a less disaggregated result is analyzed. Additionally, we understand that actual compliance rates may vary depending on factors such as driver preferences, familiarity with the area, and traffic conditions.

Figure 4.6 illustrates how the different scenarios perform using conventional metrics including mean flow *(veh/h)*, mean density *(veh/km per lane)*, mean speed *(km/h)*, mean travel time *(sec/km)*, mean delay time *(sec)*, and mean queue *(veh)*. It can be seen in the heatmap with the corresponding values that it performs better if there are 30-60% dynamic en-route users. This is mostly consistent across all metrics. Meanwhile, it performs poorly when there are 0%, 90%, and 100% dynamic en-route users. As a sample illustration, Figure 4.7 shows the comparison of links that get congested with 90% and 40% dynamic en-route users at 5:30 pm.

The average delay time had significantly improved when there are 30-60% dynamic en-route users, which indicates a reduction in the difference between the expected travel time (the time a vehicle would take to traverse the route under ideal conditions) and the actual travel time, and thus a more time-efficient network. A similar trend is also observed in the size of the queue, where more vehicles were able to enter the network to finish the travel without being held out due to the congestion. Meanwhile, the mean speed does not vary significantly given that there are still many uncongested sections (mostly residential roads) of the entire network. However, the slight change should be sufficient to indicate that there is indeed an effect with a varied percentage of users following 'Dynamic En-route' when we consider the entirety of the network.

Then, using the SIR model, we fit the parameters to determine the congestion. Figures 4.8 & 4.9 illustrate how the infection curve generated by these parameters performs when compared to actual data. In both figures, it can be observed that while the trend is not perfectly explained by the experimental data from the simulation, when we generated the curves from the SIR model, they show an interesting trend almost similar to how congested

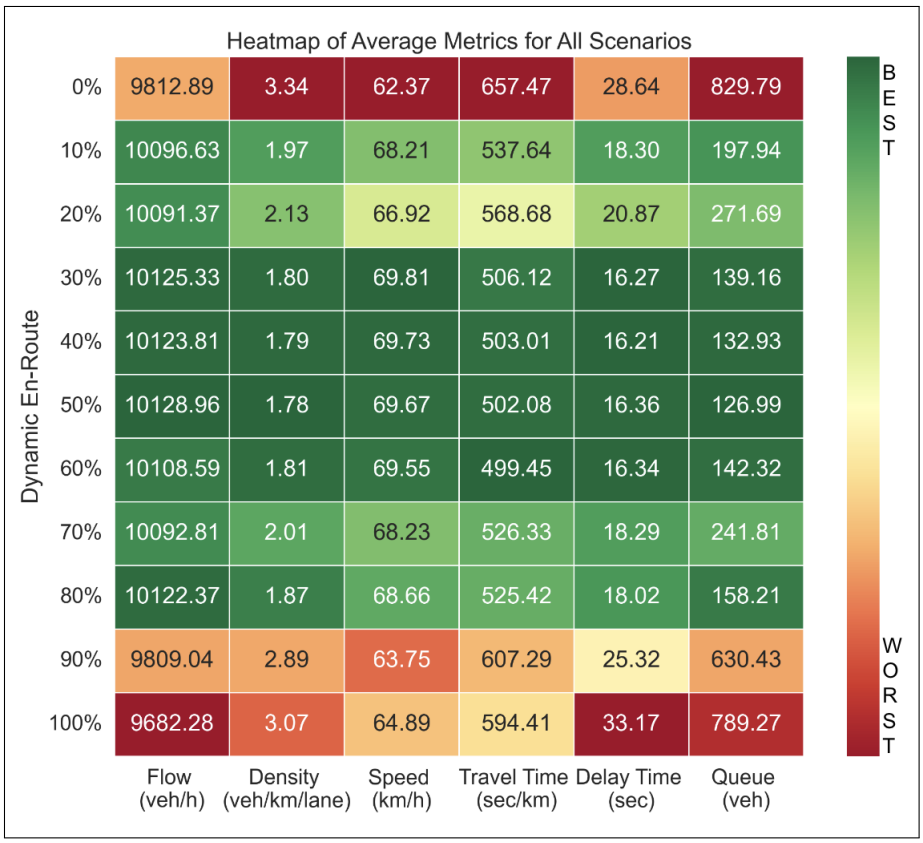


Figure 4.6: Heatmap of metrics for system-wide averages: mean flow *(veh/h)*, mean density *(veh/km per lane)*, mean speed *(km/h)*, mean travel time *(sec/km)*, mean delay time *(sec)*, and mean queue *(veh)*.

links increase toward peak hours and decrease toward the end of the simulation, consistent to how traffic congestion forms and dissipates over time.

Finally, we illustrate the impact of the spread of congestion across different scenarios using the basic reproduction number, R_0, from the SIR model. In Figure 4.10, we can observe that R_0 is generally lower for scenarios with a lower percentage of dynamic en-route users. The least value can be observed when 40% of the users are following dynamic routing. This indicates that the overall performance of the network is improved, and congestion is less likely to persist in the network. We may associate the increase in the R_0 value with the increased percentage of users following dynamic en-route routing, as there are more routes explored by the users that could lead to congestion in previously uncongested areas, particularly within residential areas. Figure 4.11 shows the dynamics of congestion propagation considering all the three traffic congestion states —free flow $S(t)$, congested $I(t)$, and recovered $R(t)$. The trend shown by each curve shows a similar pattern when compared to the SIR model applied in epidemiology.

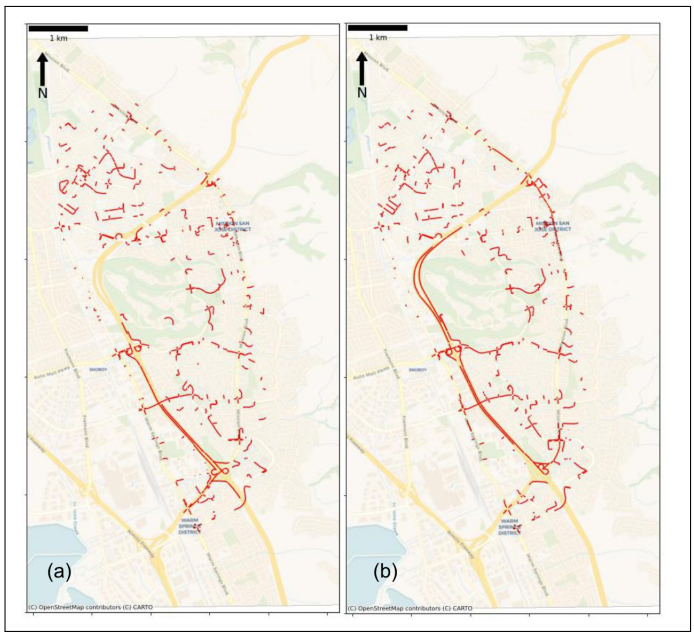


Figure 4.7: Comparison of the spatial distribution of congestion in the Fremont network at 5:30 p.m. showing the congested links (coded red) on the network with $\rho = 0.9$. (a): when 40% of the drivers follow dynamic en-route assignment with different thresholds ρ; (b): when 90% of the drivers follow dynamic en-route assignment.

4.6 Average Marginal Regret

Average Marginal Regret

Besides using the SIR model, we also considered average marginal regret. The marginal regret of a driver is defined as the "difference between their actual travel times and the shortest travel times between their origins to their destinations" [32]. This definition of marginal regret can be written as shown in equation 4.11.

$$r_p(\mathbf{f}) = t_{\mathbf{p}}(\mathbf{f}) - \min_{p \in \mathcal{P}_{od}} t_{\tilde{p}}(\mathbf{f}) \tag{4.11}$$

where

r_p = marginal regret

t_p = travel time of each path p which is the sum of the travel time of every link used by a given path p

min $t_{\tilde{p}}$ = shortest travel time for all the feasible paths between any given origin o and destination d

P_{od} = set of paths between all origin o to destination d point

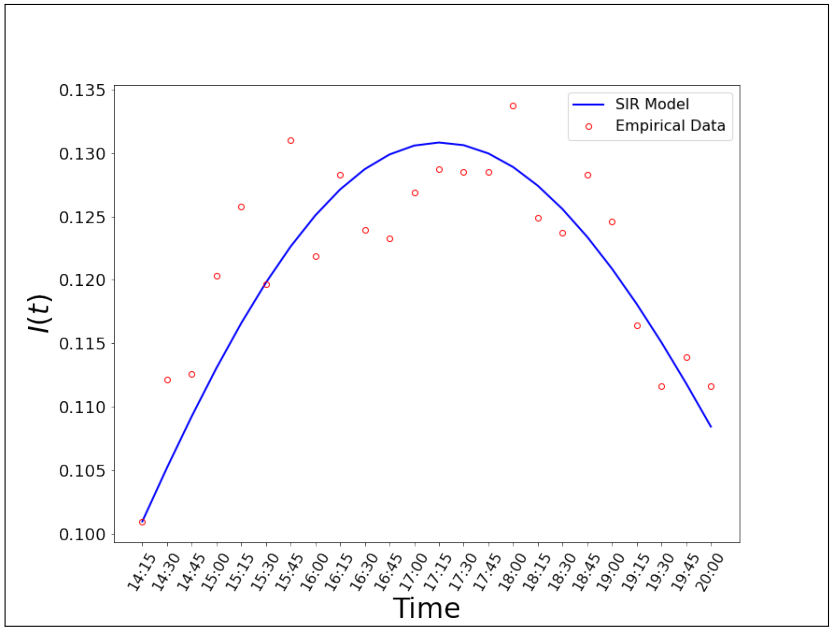


Figure 4.8: Fitted SIR model to dynamics of the system with estimated parameters $\beta = 0.229558$ and $\gamma = 0.171211$ when $\rho = 0.9$ with 50% Dynamic En-Route Users.

Following equation 4.11, the average marginal regret can be calculated by summing the marginal regret for all origin-destination pairs in the network and all paths and divided by the total demand trips as described in equation 4.12 and originally formulated in [32].

$$\overline{\mathcal{R}}(\hat{\mathbf{f}}) = \frac{1}{\|\mathbf{d}\|_1} \cdot \sum_{(o,d)\in\mathcal{V}^2} \sum_{\mathbf{p}\in\mathcal{P}_{od}} h_{\mathrm{p}} \cdot \left(t_{\mathrm{p}}(\hat{\mathbf{f}}) - \pi_{od}(\hat{\mathbf{f}}) \right) \tag{4.12}$$

where
$\hat{\mathbf{f}}$ = observed flow
h_p = path flow of vehicles using path p
$\|\mathbf{d}\|_1 = \sum_{o,d\in\mathcal{V}} \mathrm{d}_{od}$ = total demand trips for all origin-destination pairs
$\pi_{od}(\hat{\mathbf{f}}) = \min_{\tilde{p}\in\mathcal{P}_{od}} t_{\mathrm{p}}(\hat{\mathbf{f}})$ = minimum travel time

In this work, the definition of average marginal regret in equation 4.12 which is non-time varying (*i.e.,* static) is extended to time-dependent calculation of average marginal regret (*i.e.,* dynamic). This can be extended by using time-dynamical models and considering the average marginal regret as a function of time. According to [42], in DTA, there are two definitions of travel time which are: (*1* instantaneous) and (*2*) experienced. The two are illustrated and explained in figure 4.12. Given this, dynamic average marginal regret can be calculated using equation 4.13 where travel time t can be differentiated between the two definitions of travel time.

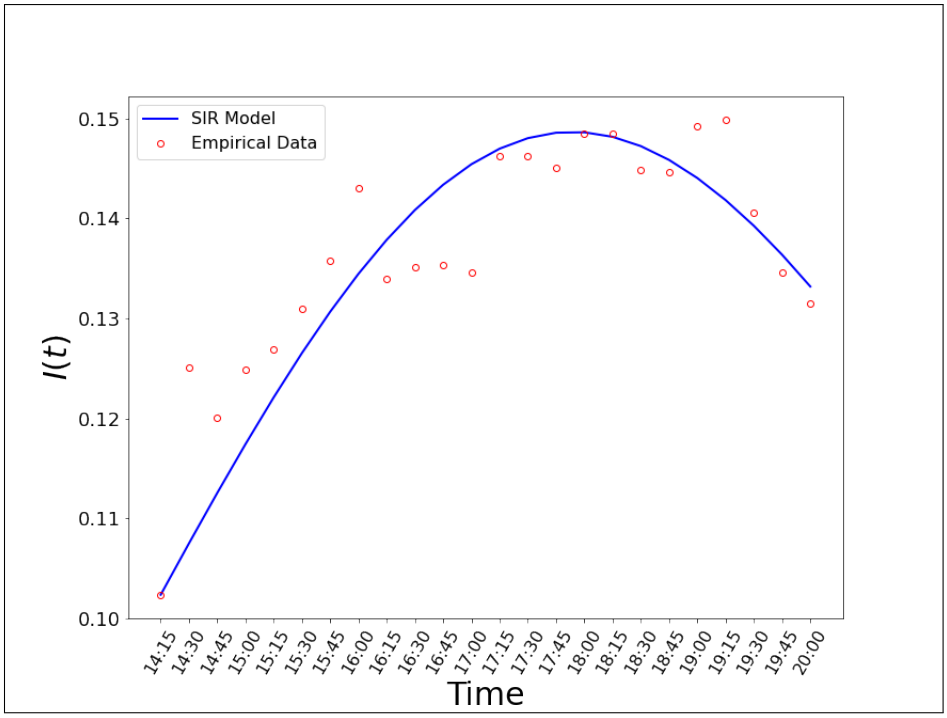


Figure 4.9: Fitted SIR model to dynamics of the system with estimated parameters $\beta = 0.213965$ and $\gamma = 0.147720$ when $\rho = 0.9$ with 100% Dynamic En-Route Users.

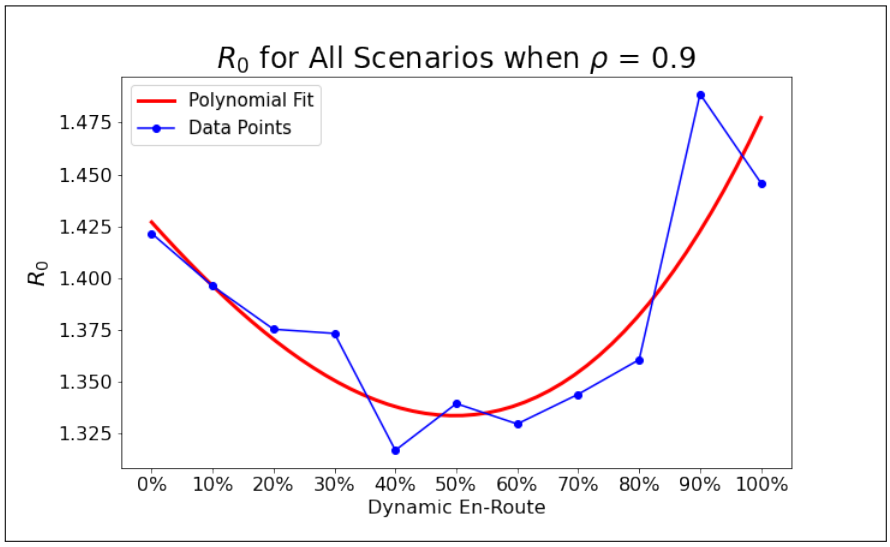


Figure 4.10: SIR model estimated basic reproductive number, R_0, for all simulation scenarios using $\rho = 0.9$, with a plot for polynomial fit.

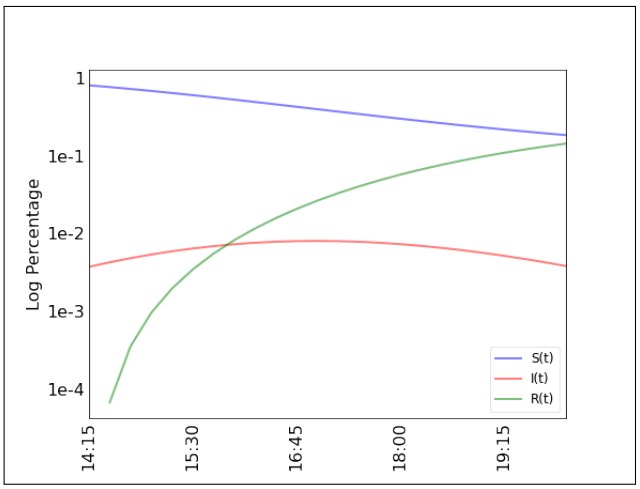


Figure 4.11: Three-state model showing how congestion spread for the scenario with the least R_0 value in log scale (90% dynamic en-route, $\beta = 0.252400$, $\gamma = 0.169535$).

The paper of Janson [79] defined a form that is almost similar to this which they called the impedance gap of a dynamic assignment. This is also adopted in Aimsun [148] where it calculates the marginal regret at the end of each time interval and uses it as convergence criterion in dynamic user equilibrium. They called this as the relative gap (Rgap). Analogous to the work of Cabannes, et al., we adopted equation 4.13 as a way to compare the results of the experiments to evaluate the impact of information-aware routing.

$$\overline{\mathcal{R}}(\mathbf{t}) = \frac{\sum_t \sum_{(o,d)\in\mathcal{J} p\in P_{od}(t)} h_{odp}(t)\left[\tau_{odp}(t) - \theta_{od}(t)\right]}{\sum_t \sum_{(o,d)\in\mathcal{J}} \mathbf{d}_{od}(t)\theta_{od}(t)} \tag{4.13}$$

where
$h_{odp}(t)$ is the flow on path p from o to d at time t
$\tau_{odp}(t)$ is the instantaneous cost of path p from o to d
$\mathbf{d}_{od}(t)$ is the demand (trips) from o to d at time t
$\theta_{od}(t)$ is the shortest path cost from o to d at time t
$\mathcal{J}$ is the set of origin-destination pairs
$\tau_{odp}(t) - \theta_{od}(t)$ is the excess cost the traveler had experienced due to using a path p instead of the shortest path

$$\tau_{od}(t) = \lambda c(t) + (1-\lambda)\tilde{c}(t-1) \tag{4.14}$$

where
$c(t)$ = expected cost
$\tilde{c}(t-1)$ = experienced cost at t-1

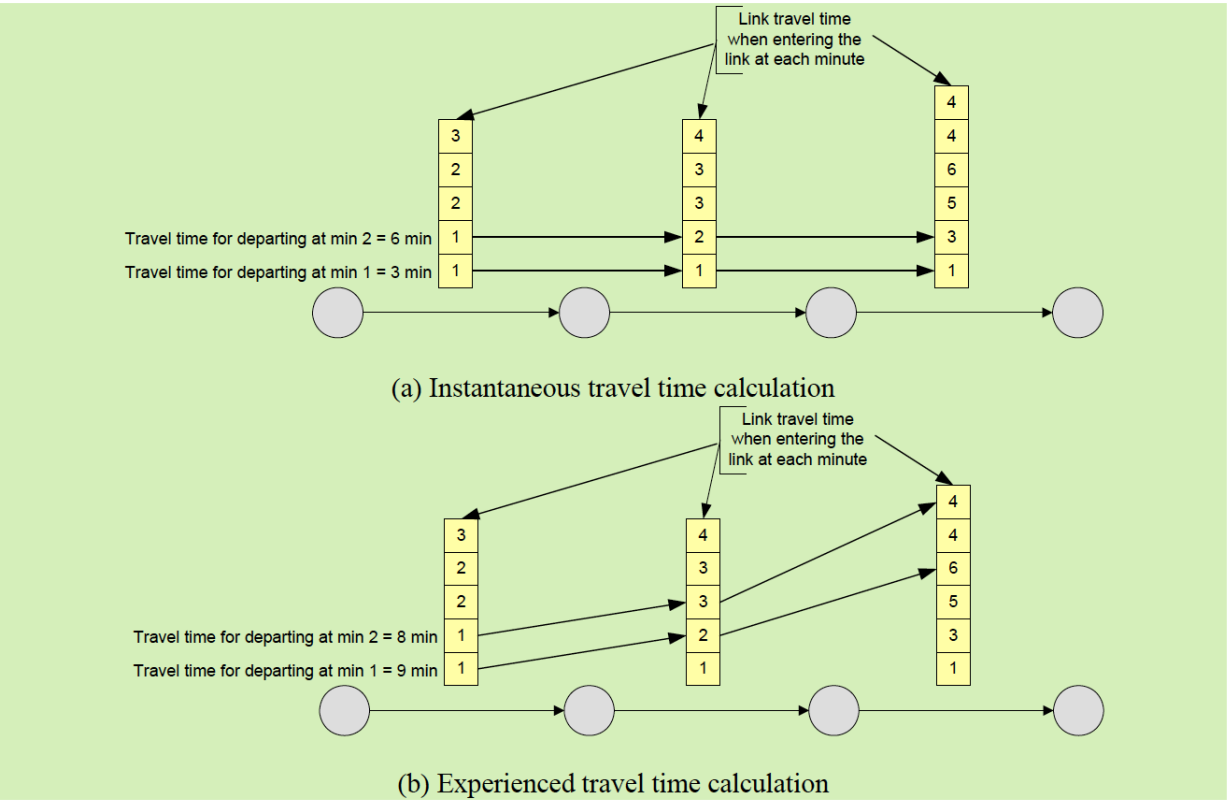


Figure 4.12: Experienced travel time versus instantaneous travel time determination. *(a)* Instantaneous travel time is calculated by summing up the link travel time corresponding to that same departure time for all links comprising the route for the entire route at each different departure time *(b)* Experienced travel time calculation accounts for the time needed for traversing one link, and looks up the downstream link travel time based on the time of entering that downstream link (assuming that traversing a node takes no time). Both the figure and defitions are from [42].

Through this metric, we are able to generate the results shown in Figure 4.13. The result is very much comparable with the I-210 simulation study in the work of Cabannes, et. al in 2018 [31] where the average marginal regret is initially high at a low penetration rate of dynamic routing users then it significantly decreases within certain penetration rates and increases again as more people use dynamic routing. This trend is also consistent with what we have seen in the use of the SIR model.

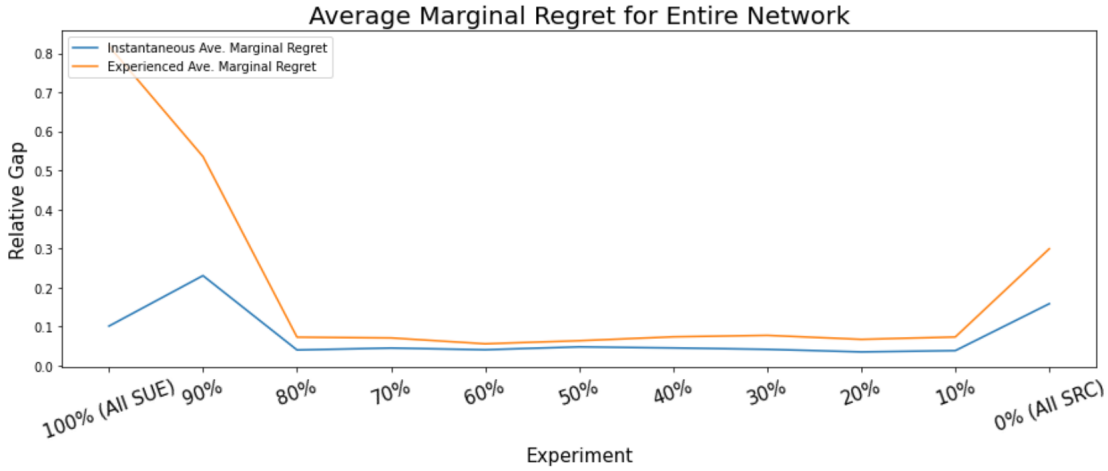


Figure 4.13: Average Marginal Regret. Results show that the average marginal regret decreases at higher penetration rates of dynamically-routed users, although it increases again when 100% of the users follow dynamic routing. The trend is similar when considering both instantaneous and experienced travel times for calculating regret.

4.7 Summary

Overall, our research contributes to the understanding of navigation app usage and its potential consequences on traffic flow and how congestion forms by employing advanced large-scale microscopic traffic simulation techniques. Our results augment existing studies that highlight the possible negative effects of navigation apps due to their growing usage. We demonstrated these effects by varying the proportion of users adopting static and dynamic routing in our simulations.

Our findings indicate that the simulation produces optimal results when approximately 30-60% of users employ dynamic en-route navigation. This observation provides valuable insights into the impact of routing strategies on overall traffic performance and congestion. There could be policy implications on how to encourage dynamic routing adoption and how to limit it when a certain percentage of users have switched to alternative routes suggested by the navigation app. We believe that understanding the collective benefits of each routing scheme is the first step in motivating new algorithms or design mechanisms for routing apps.

We have successfully supported our results by applying the SIR model, specifically by calculating the basic reproductive number, R_0, with trends closely matching those obtained using average marginal regret. This approach allows us to quantify the effects of navigation apps on traffic patterns, emphasizing the potential for increased congestion and its subsequent implications.

The observed trend of the average marginal regret, initially high at low penetration rates

of dynamic routing users, decreasing with certain penetration rates and then increasing with higher usage, aligns closely with the findings from the I-210 simulation study by Cabannes et al. [33] and the SIR model analysis.

Although these findings may be comparable to those of other similar studies using numerical experiments, our study focuses on a specific area and involves specific assumptions about drivers' behavior, such as the choice between fixed and dynamic routing. Therefore, one should exercise caution when extrapolating these findings to understand the broader implications of navigation apps on traffic congestion.

Given this, future work in this field may include examining the reproducibility of such results in other areas. There could also be more work done to determine the optimal percentage of road users that should be routed dynamically instead of setting a range; but, because there are a lot of elements affecting traffic conditions leading to uncertainties, setting a target range might be more beneficial over time instead of trying to determine a fixed value. If a fixed optimal value is determined, it will be ideal to perform a sensitivity analysis to see how the behavior might change under different conditions (*e.g.,* weather conditions, the effect of road accidents, etc.). A study on driver compliance with navigation apps could also be considered as part of future work.

Finally, since our study primarily focuses on the impact of navigation apps on traffic flow and congestion, we acknowledge that there could be additional consequences, such as increased inequity in air and noise pollution. We recognize the importance of these broader effects on urban life, which can also be considered in future research.

Chapter 5

Other Case Studies: Signal Coordination and Speed Limit Regulation

5.1 Overview

In this chapter, we explore other case studies where we used the simulation model developed in Chapter 2. The two case studies are pivotal aspects of urban traffic management: traffic light coordination and the implementation of speed limits as a form of Pigovian tax (also spelled Pigouvian tax).

The first section examines the role of traffic signals in urban traffic management, specifically focusing on the Mission San Jose District of Fremont, California. Using validated and calibrated microscopic traffic simulation models with 2019 demand data, we assess the impact of both uncoordinated signal timings and strategically reconfigured traffic signals on Mission Boulevard, a key arterial road. The findings underscore the potential benefits of improved signal coordination in optimizing traffic flow and reducing congestion.

In the second section, we explore the concept of reducing speed limits as a Pigovian tax. By interpreting travel time as a price, we propose that lowering speed limits on local roads can act as a uniform tax on traffic, aiming to achieve a socially optimal state of traffic equilibrium. This analysis highlights the potential of speed limit regulations to balance the trade-off between travel time and traffic flow, ultimately guiding urban traffic towards more desirable and efficient outcomes.

5.2 Traffic lights (un)-coordination

Traffic signals play a crucial role in urban traffic management. They are designed to maintain order, enhance safety, and minimize delays caused by queuing vehicles during red signal

phases. Often, traffic signals are strategically coordinated along key corridors, typically arterial roads, to facilitate the smooth movement of vehicle platoons through these areas.

In this section, we investigated the implementation of traffic signal timings in the Mission San Jose District of Fremont, California, using the validated and calibrated microscopic traffic simulation model considering 2019 demand data. Employing a dynamic traffic assignment mesoscopic simulation model with the same logit-based route choice model described in section 4.3, we examined the effects of both uncoordinated signal timings and the strategic reconfiguration of traffic signals on Mission Boulevard, one of Fremont's primary arterial roads. Our analysis assesses the impact of these scenarios on the overall network performance as well as their specific effects on individual roadways. Our experiments reveal that improved signal coordination fosters increased traffic flow along Mission Boulevard, redirecting traffic flow.

Signal traffic light alterations

Figure 5.1 shows Fremont's signalized nodes. We optimized signals along Mission Blvd alone, intending to improve metrics both system-wide and locally. This means that signals that are not on Mission Blvd were not altered and the settings were retained based on the 2019 Fremont signal plans.

The main idea for coordination along Mission Blvd is to set a common cycle time (or half-multiples of the cycle time) along the road and to set offsets optimally. Offsets help establish the beginning of the sequence of phases relative to a control plan's initial time, enabling synchronization. Given two reference points x and y, where flow in the $x \to y$ direction is greater than that of $y \to x$, it is typical to set the offset between x and y as L/v, where L is their spacing and v is the average speed across their incident sections. In this scheme, the platoon of vehicles queued at x reaches y immediately as the light at y turns green, minimizing unused green time.

Fremont's 2019 traffic signal plans use a large variety of cycle lengths, with no apparent patterns and no pairs of adjacent nodes with identical cycle lengths:

{67, 70, 100, 110, 111, 112, 120, 121, 128, 130, 132, 134, 146, 165, 172 seconds.}

Thus, removing coordination is simple: maintain these distinct cycle lengths, set all control plans to fixed, recall to maximum (and offset to 0 for consistency). Then the least common multiple of cycle lengths is quite large, and nodes behave independently.

Accordingly, we consider the following four signal control scenarios:

- Uncoordinated on Mission Boulevard (Blvd).: All city- and state-controlled traffic signals are uncoordinated, i.e. maximum recall and zero offset.
- Status Quo: The simulation use existing Fremont's 2019 traffic signal plans [29].

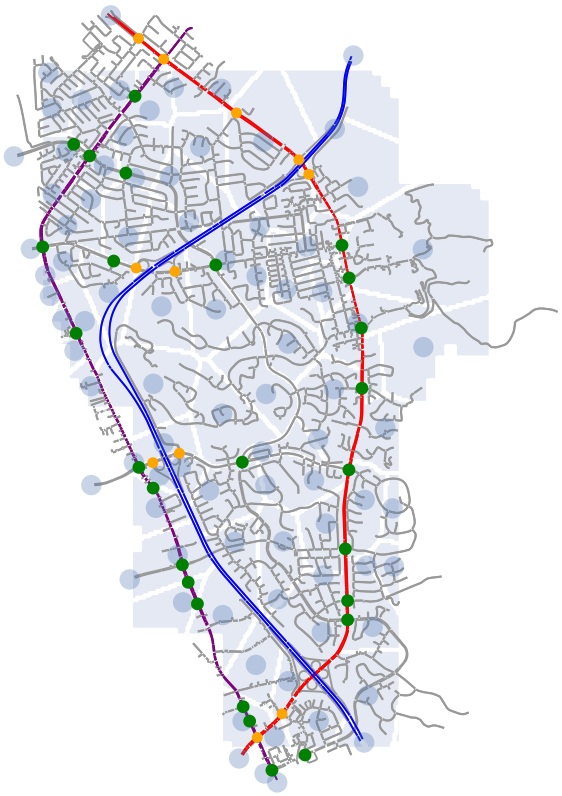


Figure 5.1: The Mission San Jose district's 37 signalized nodes. Among them, 11 (purple) are state-managed via CalTrans (*i.e.,* by the state); the remaining 26 are city-managed (green), including the 4 red nodes on Mission Blvd and 4 purple nodes on Osgood Road. Each centroid is bijective to a TAZ (blue-gray) that contains that centroid.

- Uniform Cycle Lengths on Mission Blvd.: The traffic light configurations for Mission Blvd were modified and changed to have the same cycle lengths but the offsets were set to zero.
- ℓ/v-coordinated on Mission Blvd.: The simulation is identical to Status Quo, except:
  - All cycle times along Mission Blvd are set to the nearest multiple of 60 seconds; they are all in $120, 180$.
  - Number the signalized nodes along Mission Blvd from south to north. Set $\text{offset}(0) = 0$, and for $i > 0$, $\text{offset}(i) = \text{offset}(i-1) + \ell_i/v_i$, where ℓ_i is the spacing between nodes $i-1$ and i, and v_i is the mean speed across the sections between them and over all timesteps in Coordinated.

Each scenario was run thrice, on three different random seeds (which were shared across different scenarios).

Results

We considered four scenarios discussed in section 5.2 namely 'Uncoordinated', 'Status Quo', 'Zero Offset Uniform Cycle Lengths', and 'L/V Coordination'. Among the results presented here are flow, speed, and travel time. Each figure is delineated by red dashed lines, indicating the period of congestion onset and offset, which spans from 4:30 PM to 6:30 PM. This time interval serves as the focal point of our analysis. Presented in figures 5.2-5.5 are the results from our preliminary experiments.

This section requires additional investigation, which may involve running more simulation experiments to generate more data, considering other metrics, and evaluating the reproducibility of these results. These steps will help in reaching a more robust conclusion.

5.3 Speed limit as Pigovian tax

The concept of a Pigovian tax revolves around the idea of internalizing the external costs associated with certain market activities [116]. These taxes are designed to correct market failures by aligning private incentives with societal costs, particularly addressing negative externalities. One prominent example is the taxation of goods or activities that generate pollution, such as carbon emissions from industrial production or vehicle exhaust. By imposing a tax on the market that generates negative externalities, policymakers aim to achieve several objectives such as efficient allocation of resources, behavioral change, revenue generation, and environmental protection [16] [66] [71] [138].

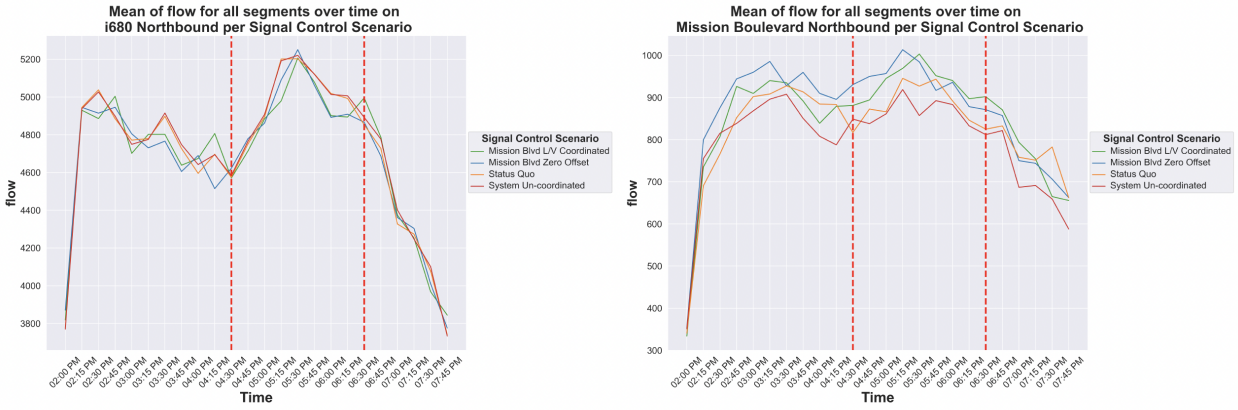
Decreasing speed limit as a Pigovian Tax

We consider that a unique regulator, such as the city government, can change the speed limit of some links (or streets) in the network, particularly those within the city. As the speed limit is reduced, we can anticipate higher travel times for vehicles, which could affect the demand on local roads and potentially steer traffic equilibrium toward a socially optimal state. We define the socially optimal state of traffic as a condition where a trade-off is made between short travel times and a low amount of flow on local roads [161] [155]. By interpreting travel time as a price, we show that decreasing the speed limit on local roads can be seen as a uniform Pigovian tax on traffic [136] [112]. This Pigovian tax is uniform in the sense that it taxes every driver the same way, even if cut-through travelers have a higher negative externality on local roads than residential drivers, creating a case of imperfect tax differentiation [119].

Figure 5.2: This figure shows Route 1 via I-680, highlighted in yellow, and Route 2 via Mission Boulevard, marked in red. It represents the main routes connecting the top Origin-Destination pair in the simulation, indicated by blue circles. Please note that there are overlaps between the two routes, particularly on the highway.

Toy model

We demonstrate how the mechanism of how changing the speed limit can be interpreted as Pigovian tax using a toy model as shown in Fig. 5.6. This somehow resembles the traffic scenario of Fremont, California, patterns of residents of Fremont (O-D2) and commuters (O-D1, potentially through D2). We introduce concepts and notations, calculate the Wardrop equilibrium of traffic, show how Fremont's control is a Pigovian tax, calculate the optimal control with an affine latency function, discuss some parameter studies, and compute the optimal control for Fremont in the general case.



(a) Flow for Route 1 via I-680

(b) Flow for Route 2 via Mission Blvd.

Figure 5.3: Flow for Route 1 via I-680 vs. Route 2 via Mission Blvd. Considering the onset and offset of peak hours, which occur from 4:30 pm to 6:30 pm, we can observe in *(a)* that there is an increase in traffic flow on I-680 when there is no coordination and a decrease in flow when traffic lights are re-strategized. In comparison to the status quo scenario, on average, the uncoordinated scenario adds 10 vehicles per hour, while 30 and 40 vehicles per hour are removed from the highway with Zero Offset and L/V Coordination, respectively. Meanwhile, the situation on Mission Blvd is quite the opposite, as we observe an average increase of 20 vehicles per hour for the uncoordinated scenario, which is twice the decrease on I-680, indicating that a lack of coordination discourages additional flow on Mission Blvd. Improved coordination on Mission Blvd results in an increase of approximately 50 vehicles per hour for both re-strategized scenarios.

In the toy example, we used the following notations and equations:

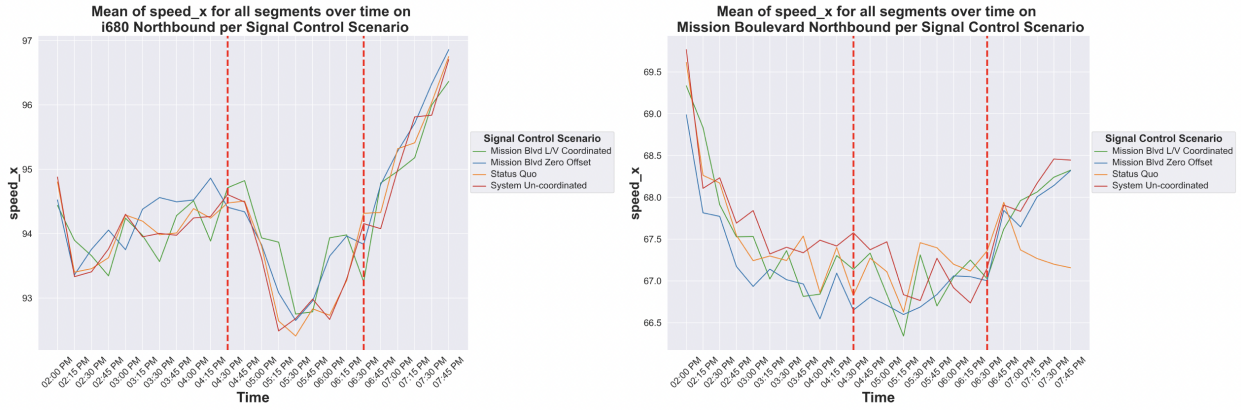
$$d_{OD1} = demand\ from\ O\ to\ D_1$$

$$d_{OD2} = demand\ from\ O\ to\ D_2$$

$$t_{OD_1} = travel\ time\ from\ O\ to\ D_1$$

$$t_{OD_2} = travel\ time\ from\ O\ to\ D_2$$

$$t_{D_1D_2} = travel\ time\ from\ D_1\ to\ D_2$$



(a) Mean Speed for Route 1 via I-680

(b) Mean Speed for Route 2 via Mission Blvd.

Figure 5.4: Speed comparison between two routes. The impact of diverted traffic on the average speed of the freeway is not very significant. Despite the increase in flow on I-680 resulting from the lack of coordination, the effect on speed is negligible (the calculated average decrease is very close to zero at -0.0049). This pattern is also observed in the impact on speed on Mission Boulevard, although we can see in the plot that the speed is noticeably higher for the uncoordinated scenario during certain time steps before the onset of peak hours.

$$t_{f1}, t_{f2}, t_{f2} = free - flow\ travel\ time$$

$$c_{p1} = travel\ time\ for\ path\ 1: \ c_{p1} = t_{OD_1}$$

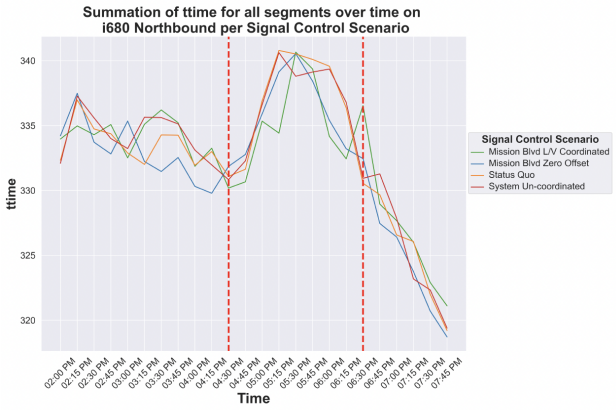
$$c_{p2} = travel\ time\ for\ path\ 2: \ c_{p2} = t_{OD_2} + T_{D_1D_2}$$

$$c_{p2} = travel\ time\ for\ path\ 3: \ c_{p3} = t_{OD_2}$$

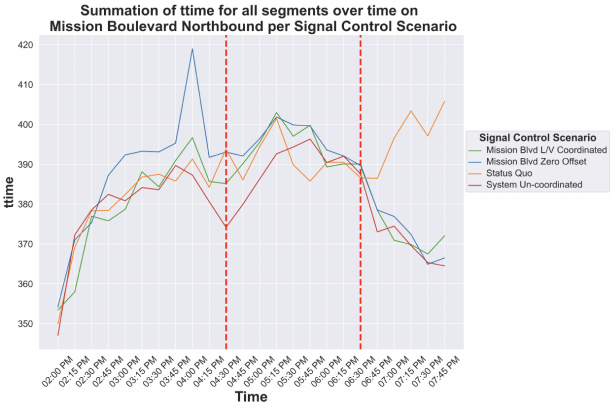
$$f_{OD_1} = \ link\ flow\ on\ OD_1$$

$$f_{OD_2} = \ link\ flow\ on\ OD_2$$

$$f_{D_1D_2} = \ link\ flow\ on\ D_1D_2$$



(a) Instantaneous Travel Time for Route 1 via I-680



(b) Instantaneous Travel Time for Route 2 via Mission Blvd.

Figure 5.5: Instantaneous travel time comparison between two routes. In these figures, we can observe that the instantaneous travel time decreases significantly during the re-strategized scenarios, particularly between 5:30 and 6:15 PM for Route 1 via I-680. Conversely, there is an increase in travel time on Route 2 via Mission Blvd.

$$h_p = path\ \ flow$$

Then, we considered that link flows are functions of path flows:

$$f_{OD_1} = h_{p1}$$

$$f_{OD_2} = h_{p2} + h_{p3}$$

$$f_{D_1D_2} = h_{p2}$$

Given the notations above and having ρ as the capacity of the local road and $\beta\rho$ as the capacity of the highway, we calculate the travel time using the affine latency functions as shown in equations 5.1-5.3.

$$t_{OD_1}\left(f_{OD_1}\right) = t_{f_1}\left(1 + \frac{f_{OD_1}}{\beta\rho}\right) \tag{5.1}$$

$$t_{OD_2}\left(f_{OD_2}\right) = t_{f_2}\left(1 + \frac{f_{OD_2}}{\rho}\right) \tag{5.2}$$

$$t_{D_2D_1}\left(f_{D_2D_1}\right) = t_{f_3}\left(1 + \frac{f_{D_2D_1}}{\rho}\right) \tag{5.3}$$

Using the arbitrarily assigned values, we calculated an optimal fraction of the demand from O to D_1, d_{OD_1} that will be assigned to path 1 denoted as γ, using equation 5.4 considering an optimal value of added tax $\hat{t_2}$. The objective function is an arbitrary function that accounts for the cost of using path 2 and a penalty function for a higher travel time on local road where α is a penalty parameter. The first constraint satisfies Wardrop's user-equilibrium where no one can switch to a path of lesser cost having $c_{p1} = c_{p2}$. The third constraint sets the lower bound for free-flow travel time on link 2. The last constraint is used to set the feasible values for γ ranging from 0 to 1.

$$\begin{aligned}
&\min_{\gamma, t_2} J_{Fremont} := (1-\gamma)^2 d_{od_1}^2 + \alpha t_{f_1}\left(1 + \frac{\gamma d_{od_1}}{\rho\beta}\right) - \alpha t_{f_3}\left(1 + \frac{(1-\gamma)d_{od_1}}{\rho}\right) \\
&\text{s.t.} \\
&t_{f_1}\left(1 + \frac{\gamma d_{od_1}}{\beta\rho}\right) = \left(\widehat{t_2} + t_{f_2}\left(1 + \frac{(1-\gamma)d_{od_1} + d_{od_2}}{\rho}\right)\right) + t_{f_3}\left(1 + \frac{(1-\gamma)d_{od_1}}{\rho}\right) \\
&t_f^{\min} \leq t_{f_2} \leq \infty \\
&0 \leq \gamma \leq 1
\end{aligned} \tag{5.4}$$

The results of the toy model experiment clearly show the shift of the equilibrium point between c_{p1} and c_{p2} as shown in Figure 5.7. The values used to generate such results are $d_{od_1} = 60$, $d_{od_2} = 5$, $\beta = 4$, $\rho = 10$, $\alpha = 200$, $t_{f_1} = 2$, $t_{f_2} = 0.3$, and $t_{f_3} = 1$.

Implementation to the Fremont network

The toy model used in section 5.3 somehow resembles the traffic scenario of Fremont, California, patterns of residents of Fremont (O-D2) and commuters/cut-through traffic (O-D1, potentially through D2). Using the simulation we developed, we also attempted to investigate whether a similar effect could be replicated in a realistic network, such as that of Fremont. Considering the two routes: Route 1 via I-680 and Route 2 via Mission Blvd, as shown in Figure 5.2, we performed experiments to see if we could reproduce trends similar to those observed in the toy network example in the previous section.

From the experiments we performed, we noticed this trend when can be replicated when we further divided the demand into additional categories. First, there is a fraction that stays on the highway. The second category is further divided into two groups: the first group consists of additional demand assigned to Route 1 via I-680 (highway route), and the second group comprises those with flexibility, assigned to Route 2 via Mission Blvd (local road route). The first split for the demand is described in equation 5.5 and the equation for the second split is described in equation 5.6. Then, to increase the cost on Route 2, we

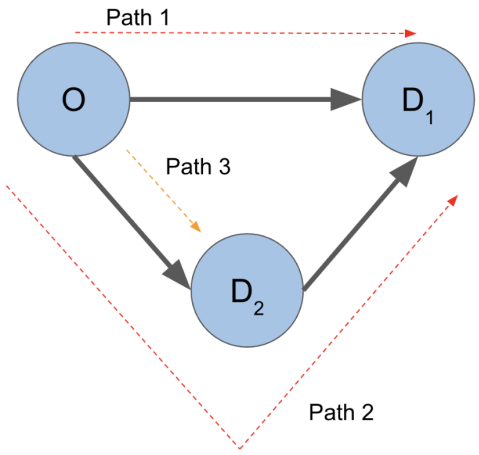


Figure 5.6: Network representation. There is a total demand from O going to D_1, with two feasible paths. Path 1 is a direct route from O to D_1, while Path 2 entails traveling from O to D_2 and then to D_1. Path 1 represents the highway, whereas Path 2 represents taking a local road.

changed the speed limit from 45 mph to 35 mph within the link closest to the I-680 exit to Mission Blvd in the northbound direction.

$$D_{od_1} = \begin{cases} \gamma d_{od_1} \\ \delta d_{od_1} \end{cases} \tag{5.5}$$

$$D'_{od_1} = \begin{cases} \Gamma_1 \delta d_{od_1} \\ \Gamma_2 \delta d_{od_1} \end{cases} \tag{5.6}$$

where

$$\gamma + \delta = 1$$

$$\Gamma_1 + \Gamma_2 = 1$$

From the equation above, the total demand assigned to Route 1 via I-680 is $\gamma d_{od_1} + \Gamma_1 \gamma d_{od_1}$ in addition to the demand from other OD pairs within the network. While Route 2 via Mission Blvd will be having some local demand added to $\Gamma_1 \gamma d_{od_1}$. The results, as shown in Figure 5.8, illustrate a very similar trend to the graph presented in Figure 5.7.

Overall, the results show the possibility of viewing speed limits as a form of Pigovian tax. Reducing the speed limit may discourage cut-through traffic while also promoting safety.

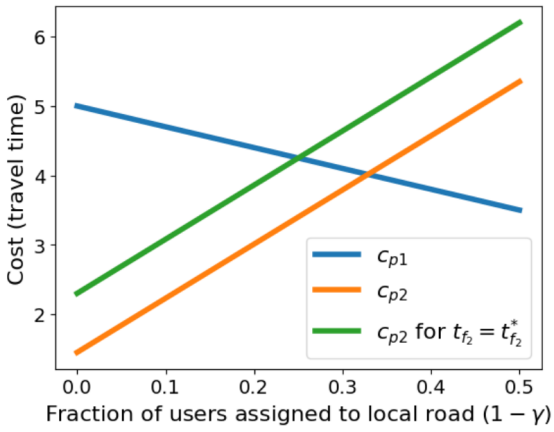


Figure 5.7: Effect of Pigovian tax on the toy network. We can see that the cost curve is shifted after increasing the free-flow travel time on the link from O to D_2 (green curve), and there is also a shift for the equilibrium point between that and the link from O to D_1 (blue curve). The original cost curve for c_{p2} does not have any tax ($\hat{t_2} = 0$, while the new cost curve c_{p2} accounts for the calculated optimal tax ($\hat{t_2} = 0.85$) with the optimal value for γ being equal to 0.75.

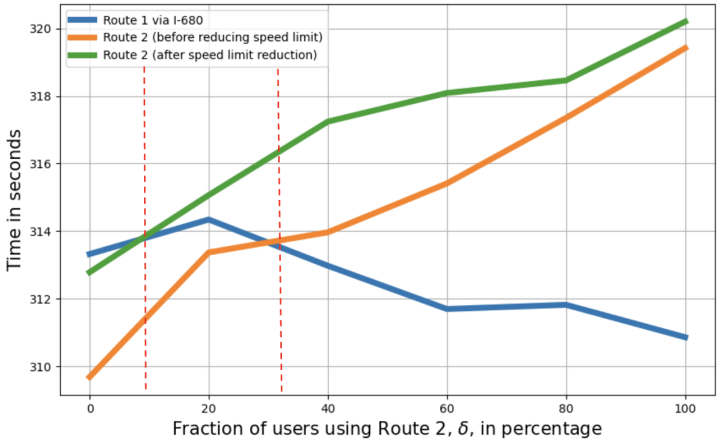


Figure 5.8: Effect of Pigovian tax on the Fremont network. We can see a similar trend as seen in Figure 5.7 where a smaller fraction of users routed to Route 2 via Mission Blvd shows the equilibrium point between the cost curves for using Routes 1 and 2.

Chapter 6

Evaluation of autonomous control policies for traffic in a ring road

6.1 Overview

[1] *"One ring to rule them all,"* The Lord of the Rings, J.R.R. Tolkien (1954) [146]. The future of traffic autonomy lands to the power of the *Ring*.

While mobile navigation apps have been shown to impact traffic performance on a large scale, this chapter focuses on a detailed investigation of a specific traffic phenomenon that occurs on closed-circuit ring roads. In these cases, even when other factors are held constant (*e.g.,* vehicle numbers and road dimensions), stop-and-go waves can propagate upstream against the flow of traffic and travel backward along the road, creating what is known as 'phantom congestion'. This chapter examines the ring road using fundamental mathematical equations that describe car-following models and reviews findings from other researchers through a thorough literature survey. Studying this traffic phenomenon on ring roads provides insights into traffic jams that also occur on open straight roads [84] [159], which can inform strategies for mitigating congestion

Car-Following Models (CFMs) describe the driving behavior of human-driven vehicles (HVs) and are considered the most important representatives of microscopic traffic flow models for studying traffic behavior and addressing congestion [151]. The development of CFMs dates back to the 1950s, with Reuschel in 1950 [123] and Pipes in 1953 [118] being the first researchers to introduce the dynamical elements of a line of vehicles. Their work included a critical component of modern microscopic modeling: the safety distance between vehicles (i.e., the minimum bumper-to-bumper distance a vehicle maintains while following a leading vehicle). Their models focus on the dynamic behavior of a stream of vehicles as they accelerate or decelerate, with each leader-follower pair interacting, making it foundational to many recent advances in CFMs. Subsequent researchers, including Kometani and Sasaki

[1] This chapter is co-authored with Fang-Chieh Chou and is also included in his dissertation. This chapter is published in ACM Transactions on Cyber-Physical Systems [43].

in 1958 [83] and Herman and his colleagues at *General Motors* [74], extended the work of Reuschel and Pipes. These early models were referred to as *minimal models* because they are "not complete and can't describe either free traffic or approaches to standing obstacles[151, Chapter 10]."

In the earlier years, solving time-continuous models was more challenging because they were computationally expensive. To address this, discrete-time models were developed. One of the simplest discrete-time models is Newell's CFM [109], which considers a leading vehicle following a preceding vehicle on a homogeneous highway. In this model, the time-space trajectory of the leading vehicle is assumed to mirror that of the preceding vehicle. The leading vehicle adjusts its speed based on the preceding vehicle at any time-space point along the traffic wave speed. However, advancements in computing technologies have made time-continuous models more feasible and flexible. The increasing popularity of time-continuous models, such as the Intelligent Driver Model (IDM), which according to [151, Chapter 11] is "probably the simplest, complete, and accident-free model producing realistic acceleration profiles and a plausible behavior in essentially all single-lane traffic situations," has led to their widespread application in traffic modeling and vehicle controller design.

Since the early 1990s, numerous researchers have demonstrated traffic congestion without bottlenecks in closed networks. Nagel *et al.* [105] showed spontaneous traffic jams on a closed ring road with cellular automaton model. Sugiyama *et al.* [140] experimentally showed the so-called phantom traffic jam phenomenon on a closed circuit road. Tadaki *et al.* [142] also demonstrated stop-and-go waves on a ring road with a similar real world experimental setup. Various numbers of vehicles on rings with different ring radii were experimented. They showed that critical density for emergence of traffic jam is consistent with critical density found on real highway.

The emergence of AVs was pushed forward when Defense Advanced Research Projects Agency, the research arm of the United States Department of Defense, organized a series of challenges in 2004-2013. This interest for AVs grew further when Tesla's Autopilot was launched in 2014. The transportation network companies like Uber and Lyft are also now invested in research for autonomous cars. It is expected that AVs has potential to improve safety[163] [95][55], but also efficiency[3][153] and mobility [164] [47]. The idea of using AVs to stabilize a stop-and-go waves on a ring road had been studied. The emergence of stop-and-go waves on a ring road can be viewed as instability of a dynamical system. Theoretically, the ring road is not controllable, but it is stabilizable [158]. Therefore, the ring road can be stabilized with an automated vehicle as a state feedback controller, although full state feedback controller may not be necessarily needed for stabilizing the system. The dynamics of the ring road can be totally different by replacing a HV by an AV, and so does the stability. If the the AV model is selected carefully, the ring road stability may be achieved even without a full state feedback controller. Cui *et al.*[45] carried out a stability analysis and showed the possibility of smoothing traffic flow with a single automated vehicle on ring road in the simulation. Stern *et al.*[137] verified the idea of smoothing traffic on a ring road. Horn *et al.* [76] proposed bilateral controller to stabilize the ring road traffic. Wu *et al.* [166] showed using the reinforcement learning controller to stabilize a ring road traffic. Delle

Monache et al. [49] proposed Lyapunov based controller for dissipating traffic waves and validated it on the ring road. Zheng *et al.*[177] considered a full state feedback controller and derived a AV controller based on optimal control to stabilize a ring road. Li *et al.* [86] further investigates the impact of the penetration rates and distributions of the optimal control method, in which optimal distributions based on $\mathcal{H}_2$ norm are shown dependent on penetration rates and the set up of car-following models. While many controllers have been successfully demonstrated the capabilities of improving traffic flows on the ring road in the literature, most of them are not generally benchmarked under different penetration rates and distributions. Since the penetration rates and distribution may have great impact to the traffic [169][86], it is essential to evaluate controllers thoroughly under different penetration rates and distributions. Given that the analysis may generally not be tractable for all all controller, analysis based on simulations is done in this work. This work aims to make a general comparisons of these controllers under different penetration rates and distributions, particularly focusing on controllers that have been validated against ring road traffic in simulations or real world experiments. Kreidieh *et al.* [84] found that controllers generated in closed network scenarios having otherwise similar densities and perturbing behaviors confirms that closed network policies are transferable to open network scenarios. While Wang *et al.* [159] was able to verify in their work in leading cruise control the potential to control traffic using connected AVs in an open road.

Contribution In this chapter, we assess the improvement of mixed autonomy traffic with AVs on a ring road under different penetration rates and distributions, while all AV algorithms remain unchanged under different penetration rates and distributions. The time to stabilize, maximum headway, vehicle miles traveled, and fuel economy are used to evaluate their performances. From the simulation experiments, we find that traffic condition improvement is generally independent of distribution of AVs particularly when no cooperation among them is considered, except for two of the AV controllers considered in this work. The traffic condition is generally improved with a higher penetration rate, while one of the AV algorithms shows a contrary trend. We show that the VMT and fuel economy are also improved with more AVs because of smoother traffic flow, except for one AV controller which shows reduced fuel economy even with improved traffic flow. Finally, among all AV algorithms in this study, the reinforcement learning controller shows the most consistent improvement under all distributions and penetration rates.

The remainder of the chapter is organized as follows: Section 6.2, Mathematics, details the notations and preliminaries in section 6.2, followed by the detailed mathematical CFM as HV in section 6.2 and various mathematical models of car-following controller as AVs in section 6.2. In section 6.3, Simulation, we studied the performance of AVs by carrying out experiments on a ring road using state-of-the-art traffic simulation platform considering various penetration rates and distribution of AVs for each AV model. Time to stabilize, maximum headway, vehicle miles traveled (VMT), and fuel economy are used to evaluate their performances. Lastly, section 6.4 summarizes the chapter.

6.2 Mathematics

This section is divided into three subsections. The first subsections define the notations and preliminaries, while the second subsection deals with typical human behavior CFMs, and the third subsection is dedicated to controllers designed for AVs.

Notations and preliminaries

In this chapter, vehicle CFMs are discussed. Subscripts of variables indicate to which vehicle the variables are related. Vehicles are generally labeled with a number, increasing sequentially against traffic flow. Vehicles in the downstream are numbered with smaller numbers, and vehicles upstream are numbered with larger numbers. In a platoon of N vehicles, the vehicle indexed 1 is the leading vehicle in the platoon, and the last vehicle is indexed N. An illustration of the platoon is shown in figure 6.1. The image show three cars that in platoon traveling to the same direction. For convenience, a few commonly used variables are listed below:

- $x_i : [0, \infty) \to \mathbb{R}$: rear end position of the vehicle i;
- $v_i : [0, \infty) \to \mathbb{R}$: speed of the vehicle i;
- $a_i : [0, \infty) \to \mathbb{R}$: acceleration of the vehicle i;
- $L_i \in \mathbb{R}$: length of the vehicle i;
- $s_i : [0, \infty) \to \mathbb{R}$: space headway between the vehicle i and the vehicle $i - 1$, which is equivalent to $x_{i-1} - x_i - L_i$.
- $v_{\text{rel},i} : [0, \infty) \to \mathbb{R}$: relative speed between the leading vehicle and the subject vehicle i; $v_{\text{rel},i} := v_{i-1} - v_i$

Discrete models are considered in this chapter. Discrete dynamics of the vehicle motion is:

$$x_i[k+1] = x_i[k] + v_i[k]\Delta t, \tag{6.1}$$
$$v_i[k+1] = v_i[k] + a_i[k]\Delta t, \tag{6.2}$$

where $\Delta t \in \mathbb{R}$ is discritization step size. Acceleration of the vehicle will be determined based on car following models and AV algorithms described in the following sections.

A list of abbreviations and acronyms used in this chapter is summarized in table 6.1.

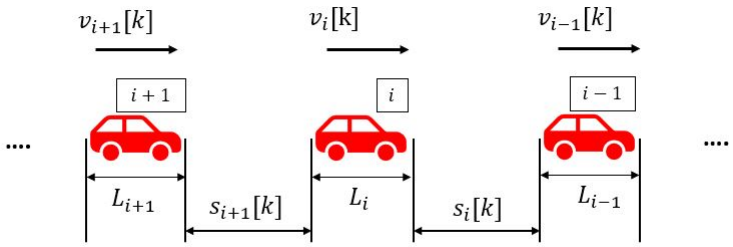


Figure 6.1: Car following model notation representation. The image shows three cars that are spaced which acts as headway with corresponding speed traveling to the same direction. The first car is $i-1$, followed by the subject vehicle i and the vehicle behind the subject vehicle represented as $i+1$.

Intelligent Driving Model (IDM) as Human Driver CFM

As mentioned in section 6.1, there are many extensive studies to model how people drive on the road. The conceptual bases of these models are supported by empirical data [27]. Despite the numerous models developed, this is still an active field of research given the modeling challenges caused by heterogeneity of traffic and varying characteristics of drivers. Other examples of CFMs include Optimal Velocity Model [15][14], second-order linear model [**Orosz2010**] and IDM [150][151]. Among these models, IDM is recognized for being capable of accurately representing realistic driver behavior [150]. Hence, IDM is adopted in this work and used to represent human-driven cars.

M. Treiber *et al.* [150][151] provided a comprehensive and instructive coverage of vehicular traffic flow dynamics and modeling in their work leading to the formulation of IDM, a CFM for the simulation of freeway and urban traffic. It describes the dynamics of the positions and velocities of single vehicles. The acceleration of IDM is shown below:

$$a_i[k] = a_{\max}\left[1 - \left(\frac{v_i[k]}{v_0}\right)^{\delta} - \left(\frac{s^*\left(v_i[k], v_{\mathrm{rel},i}[k]\right)}{s_i[k]}\right)^2\right], \tag{6.3}$$

where $v_0 \in \mathbb{R}$ is the desired velocity, $a_{\max} \in \mathbb{R}$ is the maximum acceleration, $\delta \in \mathbb{R}$ is the acceleration exponent, $b \in \mathbb{R}$ is the desired deceleration, and $s^* : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the desired headway of the vehicle for a given vehicle speed and a relative speed, denoted by:

$$s^*\left(v_i[k], v_{\mathrm{rel},i}[k]\right) = s_0 + \max\left(0, v_i[k]T - \frac{v_i[k]v_{\mathrm{rel},i}[k]}{2\sqrt{a_{\max}b}}\right). \tag{6.4}$$

where $T \in \mathbb{R}$ is the safe time headway, and $s_0 \in \mathbb{R}$ is the jam distance or minimum gap. The parameters used in IDM are presented in section 6.3. In order to replicate stop-and-go waves with IDM, the selection of the parameters are picked such the criterion for ring road stability is not satisfied.

AVs	Automated vehicles	ACC	Adaptive cruise control
HVs	Human-driven vehicles	LACC	Linear adaptive cruise control
CFM	Car following model	MLYAU1	Modified Lyapunov 1 controller
RL	Reinforcement learning	MLYAU2	Modified Lyapunov 2 controller
IDM	Intelligent driver model	LinOpt	Linear optimal controller
VMT	Vehicle miles traveled	MLP	Multilayer perceptron
AUG	Augmented Optimal Velocity Follow-the-Leader	GRU	Gated recurrent unit
BCM	Bilateral control model	PPO	Proximal policy optimization
FS	FollowerStopper controller	SUMO	Simulation of Urban Mobility
PI	Proportional-integral with saturation controller	ICs	Initial conditions
HBEFA	Handbook Emission for Road Transportation	FUZ	Fuzzy controller

Table 6.1: Acronyms and abbreviations

Automated Vehicle Car-Following Models

This section discusses the known models which include controllers mainly developed for automated vehicles, starting from Level 1 (driving assistance capabilities) to having algorithms that may apply up to Level 5 (fully) automated vehicles. Note that as the level of autonomy increases, there are also increased capabilities with respect to what the vehicles on its own like the ones described for Level 1 and Level 2 here. Unlike human behavior CFMs, models introduced here are not meant to replicate human driving behavior. These AV CFMs were designed and meant to have better properties, e.g., stability, safety, efficiency, etc.

Augmented OV-FTL model

Cui et al. [45] augmented the optimal-velocity-follow-the-leader (*OV-FTL*) model with a term penalizing the difference between the automated vehicle speed and the equilibrium speed $v_{\text{eq}} \in \mathbb{R}$:

$$a[k] = k_{\text{a}}(V(s_i[k]) - v_i[k]) + k_{\text{b}}\frac{v_{i-1}[k] - v_i[k]}{s_i[k]^2} + k_{\text{c}}(v_{\text{eq}} - v_i[k]), \tag{6.5}$$

where $k_{\mathrm{a}} \in \mathbb{R}$, $k_{\mathrm{b}} \in \mathbb{R}$, and $k_{\mathrm{c}} \in \mathbb{R}$ are positive parameters; $V(\cdot) : \mathbb{R} \to \mathbb{R}$ is a monotonically non-decreasing function:

$$V(s_n) = \begin{cases} 0, & \text{if} \quad s_n \leqslant s_{\mathrm{st}}, \\ \frac{v_{\max}}{2}\left(1 - \cos\left(\pi \frac{s_n - s_{\mathrm{st}}}{s_{go} - s_{\mathrm{st}}}\right)\right), & \text{if} \quad s_{\mathrm{st}} < s_n < s_{\mathrm{go}}, \\ v_{\max}, & \text{if} \quad s_n \geqslant s_{\mathrm{go}} \end{cases} \tag{6.6}$$

where $s_{st} \in \mathbb{R}$, $s_{go} \in \mathbb{R}$ are parameters.

Bilateral Control Model

This controller is almost similar to the linear ACC, except that instead of considering the relative speed and the spacing headway only with respect to the preceding vehicle as applied in the linear ACC, it is now considering relative speeds and the spacing headways with respect to its preceding vehicle and its following vehicle wherein it is assumed that these information can be obtained by the subject vehicle using sensors in the front and in the back. By that, the vehicle tries to be halfway between the leading and the following vehicle. Following this, the ith vehicle uses the following acceleration equation [76]:

$$a[k] = k_{\mathrm{d}}\left(s_i[k] - s_{i+1}[k]\right) + k_{\mathrm{v}}\Big(\left(v_{i-1}[k] - v_i[k]\right) - \left(v_i[k] - v_{i+1}[k]\right)\Big) + k_{\mathrm{p}}(v_{\mathrm{des}} - v_i[k]), \tag{6.7}$$

where $k_{\mathrm{d}} \in \mathbb{R}$, $k_{\mathrm{v}} \in \mathbb{R}$, and $k_{\mathrm{p}} \in \mathbb{R}$ are parameters where with any arbitrary positive constants will always lead to ring stability [77] [160]; $v_{\mathrm{des}} \in \mathbb{R}$ is a design parameter representing the desired speed.

FollowerStopper

The FollowerStopper controller [137] generates the speed command $v^{\mathrm{cmd}}(\cdot,\cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ based on the combinative conditions of the headway $s_i[k]$ and the relative speed $v_{\mathrm{rel},i}[k]$. The condition of the headway and the relative speed are divided into four regions. In each region, safety speed command is determined in order to prevent the automated from crashing into the vehicle in front of it. Speed command v^{cmd} in each region is shown below, followed be the definition of each region:

$$v_i^{\mathrm{cmd}}(s_i[k], v_{\mathrm{rel},i}[k]) = \begin{cases} 0 & \text{if } s_i[k] \leq \Delta x_1 \text{ (Stopping region)} \\ \hat{v}[k]\frac{s_i[k] - \Delta x_1}{\Delta x_2 - \Delta x_1} & \text{if } \Delta x_1 < s_i[k] \leq \Delta x_2 \text{ (Adaptation region I)} \\ \hat{v}[k] + (U - \hat{v}[k])\frac{s_i[k] - \Delta x_2}{\Delta x_3 - \Delta x_2} & \text{if } \Delta x_2 < s_i[k] \leq \Delta x_3 \text{ (Adaptation region II)} \\ U & \text{if } \Delta x_3 < s_i[k] \text{ (Safe region)} \end{cases}, \tag{6.8}$$

where $\hat{v}[k] = \min\{\max\{v_{i-1}[k], 0\}, U\}$. A safe cruise speed $U \in \mathbb{R}$ is assigned as reference speed in the safe region. In the stopping region, zero velocity is commanded. In the adaptation region (two parts), some average of desired and lead vehicle velocity is commanded. Boundary of each region is defined as follows:

$$\Delta x_j[k] = \Delta x_j^0 + \frac{1}{2d_j} \left(\Delta v_-[k]\right)^2, \quad \forall j = 1, 2, 3, \tag{6.9}$$

where $\Delta v_-[k] = \min(v_{\mathrm{rel},i}[k], 0)$. $\Delta x_1^0 \in \mathbb{R}$, $\Delta x_2^0 \in \mathbb{R}$, $\Delta x_3^0 \in \mathbb{R}$, $d_1 \in \mathbb{R}$, $d_2 \in \mathbb{R}$, and $d_3 \in \mathbb{R}$ are design parameters. Given the command speed, corresponding acceleration of the vehicle is

$$a_i[k] = \frac{v_i^{\mathrm{cmd}}[k] - v_i[k]}{\Delta t}.$$

Proportional-Integral with Saturation (PI)

The *(PI)* controller is a controller for speed control. It has been implemented on the real vehicle and used to dissipate stop and go waves in the circuit ring road [49][137]. The controller is modeled in the discrete manner and $v^{\mathrm{cmd}}[k]$ denotes the speed command at time step $k \in \mathbb{Z}_{\geq 0}$:

$$v^{\mathrm{cmd}}[k+1] = \beta[k] \left(\alpha[k] v^{\mathrm{target}}[k] + (1 - \alpha[k])\, v^{\mathrm{lead}}[k]\right) + (1 - \beta[k])\, v^{\mathrm{cmd}}[k]. \tag{6.10}$$

The speed command $v^{\mathrm{cmd}}[k] \in \mathbb{R}$ is basically the low-pass filtered output of the weighted average of the preceding vehicle speed $v^{\mathrm{lead}}[k] \in \mathbb{R}$ and the subject vehicle's own target speed $v^{\mathrm{target}}[k] \in \mathbb{R}$. The automated vehicle's target speed is defined as follows:

$$v^{\mathrm{target}}[k] = \bar{U}[k] + v^{\,\mathrm{catch}} \times \min\left\{\max\left\{\frac{s_i[k] - g_{\mathrm{l}}}{g_{\mathrm{u}} - g_{\mathrm{l}}}, 0\right\}, 1\right\}, \tag{6.11}$$

where $\bar{U}[k] \in \mathbb{R}$ is the temporal average of own speed over an historic interval, which is $\bar{U}[k] = \frac{1}{m} \sum_{j=k-1}^{k-m} v^{\mathrm{AV}}[j]$; $v^{\mathrm{catch}} \in \mathbb{R}$, $g_{\mathrm{l}} \in \mathbb{R}$, and $g_{\mathrm{u}} \in \mathbb{R}$ are design parameters. $v^{\mathrm{catch}} \in \mathbb{R}$ is the catch-up speed that allow the target speed to be larger than average speed in the past, so that the automated vehicle can speed up when the headway is large. The parameters $\alpha[k] \in \mathbb{R}$ and $\beta[k] \in \mathbb{R}$ are formulated to be dependent on the headway $s_i[k] \in \mathbb{R}$ at time step k, which are shown as follows:

$$\alpha[k] = \min\left\{\max\left\{\frac{s_i[k] - \Delta x^{\mathrm{s}}[k]}{\gamma}, 0\right\}, 1\right\}, \tag{6.12}$$

$$\beta[k] = 1 - \frac{1}{2}\alpha[k], \tag{6.13}$$

where $\gamma \in \mathbb{R}$ is a design parameter and $\Delta x^{\mathrm{s}}[k] \in \mathbb{R}$ is defined as:

$$\Delta x^{\mathrm{s}}[k] = \max\left\{2\left(v^{\mathrm{lead}}[k] - v^{\mathrm{AV}}[k]\right), 4\right\}. \tag{6.14}$$

This controller has also been called as the MLB controller in [49]. Given the speed command, corresponding acceleration command is

$$a_i[k] = \frac{v_i^{\text{cmd}}[k] - v_i[k]}{\Delta t}.$$

Linear Adaptive Cruise Control

Adaptive cruise control is a driver-assistance system which can automatically adjust the vehicle's speed to maintain a safe spacing to the preceding vehicle, which is commonly seen on many production vehicles. Linear models can be used to approximate the dynamics of the adaptive cruise control system [103][44][172][87][91][121]. In this work, we study the performance of adaptive cruise control using approximation of a first order ordinary differential equation. The model described in [120] is adapted in this work:

$$a_i[k+1] = (1 - \frac{\Delta t}{\tau})a_i[k] + \frac{\Delta t}{\tau}a_{\text{cmd},i}[k], \tag{6.15}$$

$$a_{\text{cmd},i}[k] = k_1 e_{\text{x},i}[k] + k_2 v_{\text{rel},i}[k], \tag{6.16}$$

$$e_{\text{x},i}[k] = s_i[k] - h v_i[k], \tag{6.17}$$

where $a_{\text{cmd},i}[k] \in \mathbb{R}$ is the command acceleration of the vehicle; $e_{\text{x},i}[k] \in \mathbb{R}$ is the gap error; $\tau \in \mathbb{R}$ is the lag time of the system; h is the desired time gap; $k_1 \in \mathbb{R}$ and $k_2 \in \mathbb{R}$ are positive design parameters. These parameter can be calibrated using the collected data from the field experiment.

Optimal Control Strategy

Zheng et al. [177] proposed a strategy of optimal control. The human driver CFM is firstly linearized. Suppose that vehicle-1 is the automated vehicle, linear optimal control is derived in the follows form:

$$a_i[k] = -\left[k_{1,1}(s_i[k] - s_{\text{c}}^*) + k_{1,2}(v_i[k] - v^*) + \sum_{j=2}^{N} k_{j,1}(s_{i+j-1}[k] - s^*) + k_{j,2}(v_{i+j-1}[k] - v^*)\right]. \tag{6.18}$$

On a ring road, where vehicle 1 is following the vehicle N, the index of the vehicle is cyclic, i.e. $s_i = s_{i+N}$, $v_i = v_{i+N}$. Parameters $k_{j,1} \in \mathbb{R}$, $k_{j,2} \in \mathbb{R}, \forall j \in \{1, \ldots, N\}$ are obtained by optimal control. $s_{\text{c}}^* \in \mathbb{R}$ is the average following distance of every vehicle on the ring road. To obtain these parameters, we define $K = [k_{1,1}, k_{1,2}, k_{2,1}, k_{2,2}, \cdots, k_{N,1}, k_{N,2}] \in \mathbb{R}^{1\times 2N}$ and let K equals ZX^{-1}, where $Z \in \mathbb{R}^{1\times 2N}$ and $X \in \mathbb{R}^{2N\times 2N}$ are optimizer of the following

optimization problem:

$$\begin{aligned} &\min_{X,Y,Z} Trace(QX) + Trace(RY) \\ \text{subject to} \quad &(AX - BZ) + (AX - BZ)^T + HH^T \preceq 0, \\ &\begin{bmatrix} Y & Z \\ Z^T & X \end{bmatrix} \succeq 0, \qquad\qquad X \succeq 0, \end{aligned}$$

where $A \in \mathbb{R}^{2N \times 2N}$, $B \in \mathbb{R}^{2N}$, $H \in \mathbb{R}^{2N}$ are matrices for the state space model of the linearized ring road dynamics. Q and R are design parameters, which are defined as $Q = diag(\gamma_{\mathrm{s}}^2, \gamma_{\mathrm{v}}^2, \ldots, \gamma_{\mathrm{s}}^2, \gamma_{\mathrm{v}}^2) \in \mathbb{R}^{2N \times 2N}$ and $R = \gamma_{\mathrm{u}}^2 \in \mathbb{R}$. Although the optimal control can be redesigned to achieve an optimal performance under different penetration rates and different distributions [86], in this work, for our purpose, we do not redesign the controller for different penetration rates and distributions. The optimal controller described in this section is derived based traffic of 1 AV and 22 HVs and tested under all scenarios in the simulation studies.

Lyapunov Based Controller

Delle Monache et al. [49] proposed two types of Lyapunov based controller. The controllers are firstly derived using Lyapunov like function. Controllers are then discretized considering that the controllers are deployed on the digital system, where target speed at each time step is issued as the input to the lower system of the automated vehicle. The first Lyapunov based controller is as follows:

$$v_i^{\text{target},1}[k+1] = (u[k] - \bar{v}[k])exp(-\Delta t) + \bar{v}[k]. \tag{6.19}$$

The second controller is as shown below:

$$v_i^{\text{target},2}[k+1] = (u[k] - \frac{v_{i-1}[k] + \bar{v}[k]}{2})exp(-\Delta t) + \frac{v_{i-1}[k] + \bar{v}[k]}{2}. \tag{6.20}$$

$v_i^{\text{target},1}[k+1] \in \mathbb{R}$, $v_i^{\text{target},2}[k+1] \in \mathbb{R}$ are target speeds of the i-th vehicle of the first and the second controller at time step $k+1$. $u[k] \in \mathbb{R}$ and $\bar{v}[k] \in \mathbb{R}$ are defined as follows:

$$u[k+1] = \beta[k](\alpha[k]v_i^{\text{target},j}[k] + (1 - \alpha[k])v_{i-1}[k]) + (1 - \beta[k])u[k], \qquad (j = 1, 2), \tag{6.21}$$

$$\bar{v}[k] = \min\left\{ \frac{\sum_{m=1}^{k-1} v_{i-1}[m]}{k-1}, \frac{\sum_{m=1}^{k-1} u[m]}{k-1} \right\}. \tag{6.22}$$

Controller gains $\alpha[k] \in \mathbb{R}$ and $\beta[k] \in \mathbb{R}$ are defined as follows:

$$\alpha[k] = \min\left\{ \max\left\{ \frac{s_i[k] - \Delta x^{\mathrm{s}}[k]}{\gamma}, 0 \right\}, 1 \right\}, \tag{6.23}$$

$$\beta[k] = 1 - \frac{1}{2}\alpha[k], \tag{6.24}$$

where

$$\Delta x^{\mathrm{s}}[k] = \max\left\{2\Big(v_{i-1}[k] - v_i[k]\Big), 4\right\}.$$

Given the target speed, corresponding acceleration command is

$$a_i[k] = \frac{v_i^{\mathrm{target},j}[k] - v_i[k]}{\Delta t}.$$

Fuzzy Controller

Haulcy et al. [72] proposed fuzzy controller for stabilizing a ring road. In the fuzzy logic, the space headway and the relative speed are processed to determine the desired speed change. Triangular shape membership functions are constructed to represent the levels of the different classes of the distances and the speed differences. The speed change command is determined following the inference process and the defuzzification process.

Reinforcement Learning Control

Reinforcement learning is learning to do tasks wherein the agent (*i.e.* the learner) does some action without having told what action to do. The agent does random actions at first and maps out each of these actions to determine which among them maximizes the numerical rewards. Mathematically, an agent learns a policy $\pi(state) = action$, maps from states *state* to actions *action*, to achieve a goal in an environment under uncertainty. Through repeated environment interactions, a reinforcement learning agent strives to develop an optimal policy π^*, which maximizes the sum of the rewards. RL is often used in solving sequential decision-making problems [141]. To deal with the traffic control with the reinforcement learning, Wu *et al.* [165][166] proposed framework for traffic control using deep reinforcement learning . The framework integrates the traffic simulation environment SUMO and the reinforcement learning library such as rllab [51] and rllib [88] so that the policy cna be learnt to optimize the cumulative reward using sampled data from SUMO. The policy usually consists of neural networks, and may be of several forms. Two policies: the Multilayer Perceptron and Gated Recurrent Unit are proposed for a ring road problem in the [166]. MLP is a classical (feedforward) artificial neural network with multiple hidden layers and utilizes back propagation to optimize its parameters. GRUs are recurrent neural network capable of storing memory on the previous states of the system through the use of parameterized update and reset gates, which are also optimized by the policy gradient method. One of the advanced reinforcement learning algorithms is the Proximal Policy Optimization [130] which is said to perform comparably or better than state-of-the-art approaches. Other advanced reinforcement learning, such as soft-actor critic [69], not covered in this work, can also be investigated in the near future. The RL control tested in this work is trained on the ring road of 1 agent and 21 HVs. The *state* are the distance between the agent and its leading vehicle ($s_i[k]$), the leading vehicle speed ($v_{i-1}[k]$), and the agent speed ($v_i[k]$). The *action* is the agent acceleration ($a_i[k]$).

Suppose, the RL agent is i-the vehicle in the ring road, the *reward* $r_i[k] \in \mathbb{R}$ is designed in a way that high average speed of all vehicles is rewarded, while high acceleration of the agent is penalized. The reward at each time step is:

$$r_i[k] = \eta_1 \sum_{j=1}^{22} v_j[k] - \eta_2 \max\{0, a_i[k]\}.$$

Once the policy is trained on the ring road of 1 agent and 21 HVs, it is tested under different penetration rates and distributions with no modification.

Summary Table for AV Controllers

In the table 6.2, properties of the automated vehicle models considered in this chapter are summarized, including the needed input, the controller type, the design parameters, and the design method. In terms of input needed for the controllers, all the controllers need at least the speed measurements of the leading vehicle and the subject vehicle, and most of them also use the measurement of the spacing ahead of the subject vehicle, except MLYAU1 and MLYAU2. Without using the feedback of the spacing ahead, the following distance of MLYAU1 and MLYAU2 cannot be controlled. In addition to the spacing ahead and the speed of the leading vehicle, the BCM also uses the spacing behind and the speed of the following vehicle. The optimal control needs the most information, not only the spacing and the speeds of the vehicles near by, but also spacing and speeds of all the vehicles on the ring road. To access these information, wireless communication may be needed between vehicles. Third column shows the types of the controller. Four of the controllers are nonlinear controllers, and the rest of the controllers are linear. The design parameters of the controllers are listed in the forth column; except FUZ and RL, because the parameters these two controllers are too many to show here. The references for these two controllers are provided instead. Inspired by the classification in [158], the last column summarized the ways controllers are derived. Three controllers are designed based on observations in the real world. RL is the only controller that is based on learning. The rest of the controllers are based on analyses of the car following models.

6.3 Simulation

Simulation platform

Simulation of Urban Mobility [93] is mainly used to simulate traffic flow in a ring road. We used this simulation software since it is an open-source enabling interested parties to reproduce the results we have. SUMO has the capability to generate microscopic models of inter-modal traffic systems including road vehicles, public transport and pedestrians. It allows customized models and has various APIs to control the simulations remotely. It also

AV models	Input	Controller type	Design parameters	Design Method
AUG	s_i, v_i, v_{i-1}	Linear	$k_{\mathrm{a}}, k_{\mathrm{b}}, k_{\mathrm{c}}, v_{\mathrm{eq}}$	Model-based
BCM	$s_i, s_{i+1}, v_i, v_{i-1}, v_{i+1}$	Linear	$k_{\mathrm{d}}, k_{\mathrm{v}}, k_{\mathrm{p}}, v_{\mathrm{des}}$	Model-based
FS	s_i, v_i, v_{i-1}	Nonlinear	$U, \Delta x_1^0, \Delta x_2^0, \Delta x_3^0, d_1, d_2, d_3$	Heuristic
PI	s_i, v_i, v_{i-1}	Nonlinear	$v^{\mathrm{catch}}, g_{\mathrm{u}}, g_{\mathrm{l}}, \gamma$	Heuristic
LACC	s_i, v_i, v_{i-1}	Linear	k_1, k_2, h	Model-based
LinOpt	$s_i, v_i, \forall i \in \{1, 2, \ldots, N\}$	Linear	$\gamma_{\mathrm{s}}, \gamma_{\mathrm{v}}, \gamma_{\mathrm{u}}$	Model-based
MLYAU1	v_i, v_{i-1}	Linear	γ	Model-based
MLYAU2	v_i, v_{i-1}	Linear	γ	Model-based
FUZ	s_i, v_i, v_{i-1}	Nonlinear	see details in [72]	Heuristic
RL	s_i, v_i, v_{i-1}	Nonlinear	see details in [166]	Learning-based

Table 6.2: Summary of the AV models on a ring road.

allowed us to calculate the fuel economy which calculates it based on the Handbook Emission Factors for Road Transport 3 Euro 4 passenger car emission model.

To run the simulations in SUMO, we also utilized existing libraries of code available in Flow [166][165] which is a computational framework mainly developed to enable the use of reinforcement learning methods such as policy gradient methods for traffic control and enables benchmarking of the performance of classical controllers which is then built to work the SUMO microsimulator. Flow is developed by the Mobile Sensing Lab at the University of California, Berkeley.

This study is the general investigation of behaviour of vehicles driving on the closed circuit road. Since the ring itself is a closed loop, we used the continuous router which ensures the continuous rerouting of the vehicles in a closed loop. This class is useful if vehicles are expected to continuously follow the same route, and repeat said route once it reaches its end.

Simulation set up and simulation parameters

We are interested in the dynamic behaviour of the closed circuit ring road. The simulation environment is set up similar to the real experiment shown in [137] [167]. There are 22 vehicles on the single lane circuit ring road, which the radius is 41.38m (circumference is 260m). At the beginning of the simulation, the vehicle are placed uniformly, which each headway being the same; and the initial speeds are all zero. Depends on the scenarios (section 6.3) running, each vehicle is assigned as human-driving vehicle or as a different type

of automated vehicle. In each simulation run, there is always 300 seconds warm-up, in which all vehicles are running with IDM to make the ring traffic in the status of having stop-and-go waves. After that, the assigned automated vehicles switch to the automated control. For each model, values of parameters used in the simulations in this work are summarized in this section. Selection of parameters may have influence on the performances. For the purpose of bench-marking, values of the original articles are used if they are available. Otherwise, values based on design criteria shown in the original articles are used. None of these values are fine-tuned, although performances may be improved after careful tuning.

- **Intelligent Driver Model (IDM)** The parameters chosen for the IDM are as follows: acceleration component $\delta = 4$, safe time headway $T = 1$, maximum acceleration $a_{\max} = 1$, desired deceleration $b = 1.5$, jam distance $s_0 = 2$, and maximum speed $v_0 = 30$ [151, Chapter 11].

- **Augmented OV-FTL (AUG)** $k_{\mathrm{a}} = 1$, $k_{\mathrm{b}} = 1$, $k_{\mathrm{c}} = 11.0$, $h_{\mathrm{st}} = 2$, $h_{\mathrm{go}} = 15$, $v_{\max} = 30$, and $v_{\mathrm{eq}} = 4.8$, where k_{a}, k_{b} are arbitrary positive numbers and k_{c} is selected based on the criterion shown in [45]; h_{st}, h_{go}, and $v_{\max}$ are default value in the FLOW [166]; v_{eq} is the equilibrium speed of 22 IDMs on the ring road.

- **Bilateral Control Model (BCM)** The bilateral CFM (section 6.2) and the parameters of the controller used are: $k_d = 1$, $k_v = 1$, $k_p = 1$, $v_{\mathrm{des}} = 4.8$. It has been proven in [160] that the system of BCM vehicles is chain stable for arbitrary values of k_{d} and k_{v} that are greater than 0. v_{des} is picked as the equilibrium speed of 22 IDMs on the ring road.

- **FollowerStopper (FS)** The parameters defining the four regions of the controller are as follows: $\Delta x_1^0 = 4.5$, $\Delta x_2^0 = 5.0$,and $\Delta x_3^0 = 6.0$. Also, the deceleration rates are defined to be: $d_1 = 1.5$, $d_2 = 1.0$, $d_3 = 0.5$, and $U = 4.8$. Parameters shown in [137] are used. U is the target speed of the FollowerStopper. It is picked as the equilibrium speed of 22 IDMs on the ring road.

- **Proportional-Integral with saturation (PI)** Parameters used in simulation are: $\gamma = 2$, $g_{\mathrm{l}} = 7$, $g_{\mathrm{u}} = 30$, and $v^{\mathrm{catch}} = 1$. Parameters shown in [137] are used.

- **Linear Adaptive Cruise Control** All the ACC controlled vehicles have the same parameters for dynamics and control: $\tau = 0.1$, $h = 1.4$s, $k_1 = 0.4$, and $k_2 = 0.7$. These parameters are picked such that its equilibrium speed is close to IDM's; and string stability condition for a linear car following model can be satisfied.

- **Linear optimal control (LinOpt)** The controller gains are designed using the gains as follows: $\gamma_{\mathrm{s}} = 1$, $\gamma_{\mathrm{v}} = 1$, $\gamma_{\mathrm{u}} = 1$. These values are not specified in the original paper and are arbitrary selected here

- **Lyapunov-based controllers (MLYAU1 and MLYAU2)** For both the type 1 and the type 2, they have the same parameter: $\gamma = 2$. The parameter value shown in [49] is used.

Scenario

Given the goal of this chapter, we designed our simulations to include cases of both mixed-autonomy traffic with varying percentages of AVs in the system, and also the case where we have full-autonomy.

Baseline case (No Automation) In this chapter, the baseline scenario we are considering is all human driving vehicles on the ring. We are using IDM to model the human driving behaviour. In order to stimulate instability of ring road model to reproduce stop-and-go waves, acceleration noises are intentionally injected to all the car-following models. Another way of stimulating speed fluctuations is by inducing a sudden speed changes of vehicles in the traffic, e.g.[62][76][177]. In this works, persistent excitation of the traffic with acceleration noises are used, because we can produce stop-and-go waves more easily by doing this. Speed profiles of the 22 vehicles of IDM on the ring is shown in the figure 6.2. The plot shows how the speed profiles of the vehicles greatly vary with time especially after a certain time greater than 200s.

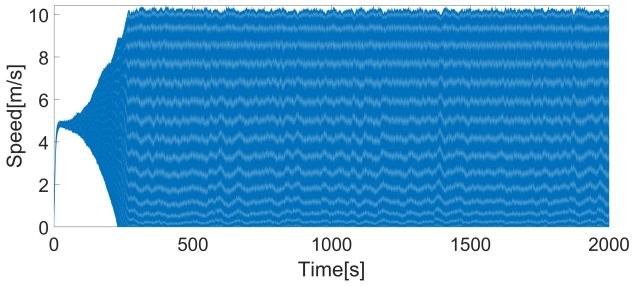


Figure 6.2: Speed profiles of all IDM vehicles on the ring. The stop-and-go waves are fully developed at around 300 seconds and persist for the rest of the time.

Mixed Autonomy Traffic (Varying Percentages of AVs Penetration) As an integral part of the simulations, we included a variation in the number of AVs in the system to test how the autonomous controllers perform in mixed-autonomy condition to see if there is also a significance in their performance depending on the percentage of penetration. Besides the penetration rate, we also investigate the impact of different distributions of AVs among HVs. Table 6.3 summarized the setups for scenarios we studied. In the *scenario I: platooned*, automated vehicles are placed in the clustered manner, which AVs are placed consecutively

and HVs are also placed consecutively. Penetration of 1 AVs all the way to 22 AVs are studied. Then, in *scenario II: evenly distributed* is the case where the AVs are placed as evenly distributed as possible. That is the minimum number of HVs between any two AVs is maximized. A maximum of 11 AVs are placed on the ring is studied in this scenario, because any number more than 11 will no longer keep AVs separated evenly by HVs. Simulation results of these scenarios are shown in the next section. To help reader visualize scenarios, table 6.4 and table 6.5 are provided, showing the types of each vehicle on the road under different penetration rate at different scenarios. In order to compare the performances of controllers in terms of capabilities of dissipating stop-and-go waves, the controllers would only be activated after 300s, a time at which the stop-and-go waves are fully formed.

	scenario I	*scenario II*
AV(s) activation time	300	300
ICs (positions)	$s_i[0] = s_{\text{eq}} + \tilde{s}_i, \forall i = \{1, \ldots, 22\}$	$s_i[0] = s_{\text{eq}} + \tilde{s}_i, \forall i = \{1, \ldots, 22\}$
ICs (speeds)	$v_i[0] = 0, \forall i = \{1, \ldots, 22\}$	$v_i[0] = 0, \forall i = \{1, \ldots, 22\}$
AV distribution	clustered	evenly distributed
Number of AVs (= n)	$n \in \{1, 2, \cdots, 22\}$	$n \in \{2, \cdots, 11\}$
IDM noise	0.1	0.1

Table 6.3: Scenarios of experiments in FLOW. $s_{\text{eq}} := (2\pi R_{\text{ring}} - \sum_{i=1}^{22} L_i)/22$, where $R_{\text{ring}} \in \mathbb{R}$ is radius of the ring road; and $\tilde{s}_i \in \mathbb{R}$ are random variables and are sampled such that $\sum_{i=1}^{22} \tilde{s}_i = 0$ to keep the sum of the headway matches the perimeter of the ring road. *AV(s) activation time* is the time at which AV controllers start to actively control the vehicles. *ICs(positions)* are the initial conditions of vehicle positions on the ring road. *ICs(speeds)* are the initial conditions of vehicle speeds on the ring road. *AV distribution* is the way AVs are distributed among other vehicles. *Number of AVs* is the number of AVs being placed on the ring road. *IDM noise* is the magnitude of the acceleration noise (in $\mathrm{m\,s^{-2}}$) added to vehicles. The distributions of AVs for *scenario I* are shown below in the table 6.4, and the distributions of AVs for *scenario II* are shown in below in the table 6.5.

Results

Some simulation results are shown in this section and performances of the AV algorithms are evaluated. The metrics we are using for evaluation including : *Time to stabilize*, *Maximum final gap*, *Vehicle Miles of Travel (VMT)*, and *fuel economy*.

Definition (Time to stabilize) The minimum time it takes for the standard deviation of the speeds across all vehicles to become smaller than 0.1, which is the noise we intentionally inject into the acceleration command. To be more precise, mathematical description is shown

(%)	Vehicle Type																					
4.55	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
9.09	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
13.64	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
18.18	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
22.73	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
27.27	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
31.82	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
36.36	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
40.91	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
45.45	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
50.00	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
54.55	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
59.09	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV
63.64	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV	HV
68.38	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV	HV
72.73	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV	HV
77.27	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV	HV
81.82	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV	HV
86.36	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV	HV
90.91	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV	HV
95.45	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	HV
100.00	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV	AV

Table 6.4: AV distribution for scenario I-platooned under different penetration rates.

%	Vehicle Type																					
9.09	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV	AV	HV	HV	HV	HV	HV	HV	HV	HV	HV	HV
13.64	AV	HV	HV	HV	HV	HV	HV	AV	HV	HV	HV	HV	HV	HV	HV	AV	HV	HV	HV	HV	HV	HV
18.18	AV	HV	HV	HV	HV	AV	HV	HV	HV	HV	HV	AV	HV	HV	HV	HV	AV	HV	HV	HV	HV	HV
22.73	AV	HV	HV	HV	AV	HV	HV	HV	HV	AV	HV	HV	HV	AV	HV	HV	HV	HV	AV	HV	HV	HV
27.27	AV	HV	HV	HV	AV	HV	HV	AV	HV	HV	HV	AV	HV	HV	HV	AV	HV	HV	AV	HV	HV	HV
31.82	AV	HV	HV	AV	HV	HV	AV	HV	HV	AV	HV	HV	HV	AV	HV	HV	AV	HV	HV	AV	HV	HV
36.36	AV	HV	HV	AV	HV	AV	HV	HV	AV	HV	HV	AV	HV	HV	AV	HV	AV	HV	HV	AV	HV	HV
40.91	AV	HV	AV	HV	HV	AV	HV	AV	HV	HV	AV	HV	AV	HV	HV	AV	HV	AV	HV	HV	AV	HV
45.45	AV	HV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	HV	AV	HV	AV	HV	AV	HV	AV	HV
50.00	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV	AV	HV

Table 6.5: AV distribution for scenario II-evenly distributed under different penetration rates.

below:

$$T_{\text{stable}} = arg\min_{t} \left\{ t : \sqrt{\frac{1}{21}\sum_{i=1}^{22}(v_i[k] - \mu[k])^2} \leq 0.1, \mu[k] = \sum_{i=1}^{22} v_i[k] \right\} - t_{activation}, \quad (6.25)$$

where $t_{activation}$ is the time when we switch on the automated vehicle, which is 300s in our simulation. The shorter time implies better wave dissipation capability.

Definition (Maximum final gap) The maximum gap between any of two consecutive vehicles on the ring if a ring road is stable. This can be used to evaluate the efficiency of the use of spacing. The smaller is better. Formal mathematical definition is as follows:

$$\bar{S}_{\text{final}} = \max_{t \in [t_{\text{activation}}+T_{\text{stable}}, T_{\text{final}}]} \max_{\forall \in i=1,2,\ldots,22} s_i[k], \tag{6.26}$$

where T_{final} is the time step at the end of the simulation. To reduce the variations of these values because of randomness of the microsimulation, the values we are showing below are averages of 10 simulation runs. Because of the randomness, for a few of the controllers, some of the stable(unstable) results are not 100% reproducible. That is, for a few AV, under the same distribution and the same penetration rate, the simulations may not always stable(unstable). For those marginal cases, we can say they are stable if more than 50% of the simulation runs are stable.

Scenario I

Given the metrics defined, we can now evaluate the performance of different AV controllers. In the first scenario, AV penetration rate from 0 to 100 percent are studied, where AVs are placed in the platoon. Some AV controllers can successfully stabilize the ring at relative low penetration rate while some need higher penetration rate to stabilize the ring. Table 6.6 shows the minimum number of AVs needed to stabilize the ring and their corresponding *time to stabilize*, *max gap*, *VMT*, and *fuel economy*. To be noted, FUZ cannot stabilize the ring for any penetration rate. Therefore, no values are available. To illustrate the simulation, speed profiles of each vehicle of each case in the table are shown in the figure 6.3; and headways of each case in the table are shown in figure 6.4, where red curves represent AVs and blue curves represent HVs. Results of Fuzzy controller are not shown here because the ring is not stabilized for any penetration rates. Table 6.6 also shows the VMT and the fuel economy for AVs that stabilized the traffic on the ring. The fuel economy is calculated using the built-in fuel consumption model of SUMO which uses the HBEFA 3 Euro 4 passenger car emission model.

Based on table 6.6, we can tell that the following controllers are able to stabilize the traffic on the ring with a minimum of one (1) AV: FollowerStopper, PI, Linear Optimal Controller, Lyapunov-based Controllers Type 1 and Type 2, and Reinforcement Learning. While AUG and BCM needs four AVs and LACC needs 9 AVs to stabilize the ring, the rest of the controllers need only one AV to stabilize the ring. Among these controllers, the Lyapunov-based Controller Type 1 (6.2) yields the fastest time to stabilize. However, from figure 6.3, it is notable that after a few seconds that the vehicles are stable, all of them simply stops. This is a trivial solution to be stable. Because of this, we can consider that the RL controller performs the best in terms of *time to stabilize*.

By looking at the headway, we can observe that whether the traffic flow density on the ring road is uniform. We can observe that headway between vehicles converge to a steady state when the traffic on the ring road is stable for all cases in figure 6.4. Most AV controllers

Controller	Min no. of AVs to stabilize	Time to stabilize in *sec*	Max gap in *m*	VMT	Fuel economy in *mpg*
AUG	4	88.29	12.19	118.23	16.21
BCM	4	319.68	12.11	128.59	21.46
FS	1	270.85	12.96	126.54	20.51
FUZ	*Unstable*	*N/A*	*N/A*	*N/A*	*N/A*
LACC	9	1104.77	12.14	130.15	21.15
LinOpt	1	471.23	12.19	125.78	20.89
MLYAU1	1	62.64	116.49	21.83	7.39
MLYAU2	1	625.66	39.65	99.83	16.15
PI	1	263.74	12.03	125.67	20.71
RL	1	102.20	13.12	142.64	20.64

Table 6.6: Average performance comparison of scenario I

have similar headway, except MLYAU1 and MLYAU2 where they have relative large headway. MLYAU1 gives the highest headway since the vehicles go to a complete stop a few seconds after being stable creating a huge gap between the AV and the HVs. The baseline values of VMT and fuel economy are 96.71[*miles*] and 13.43[*mpg*] when all the vehicles are HVs. In terms of VMT, all cases in the table perform better than the baseline, because the flow is more smoother, except that MLYAU1 is basically block the traffic, as mentioned. Fuel economy for all cases are generally better than baseline as well, except MLYAU1.

Figure 6.5 further shows *time to stabilize*, *maximum final gap*, *VMT*, and *fuel economy* across different number of automated vehicle on the ring, respectively. In figure 6.5-(a) and figure 6.5-(b), we neglect data points of unstable ring road.. In the figure 6.5-(a), it can be observed that generally, the time need to be stable is shorter with more AVs on the ring, except PI. The PI eventually make the traffic into totally chaos when there are more than 8 AVs (˜36% penetration rate) on the ring.

In figure 6.5-(b), we can also observe that the maximum headway is also not much dependent on the number of AVs on the ring, but is relatively dependent on the AV types. To be noted, that the MLYAU1 has relatively large headway because what is doing is basically stop all the traffic behind.

Figure 6.5-(c) shows the comparison of VMT and figure 6.5-(d) show the fuel economy, respectively. In terms of VMT, we can see that the AVs generally improves the VMT, except the FUZ, which is not stable, and the MLYAU1, which basically stops the traffic as we saw previously. The VMT are improved because the traffic flow are smoother than the baseline scenario. The VMT is generally positively related to the stability of ring road. The VMT are almost the same for AVs whose stabilizing time is almost the same for different penetration rates. For AVs that get more stable (shorter stabilizing time) with increasing penetration rate, the VMT is increasing with penetration rates, e.g. BCM, LACC, AUG. On the other

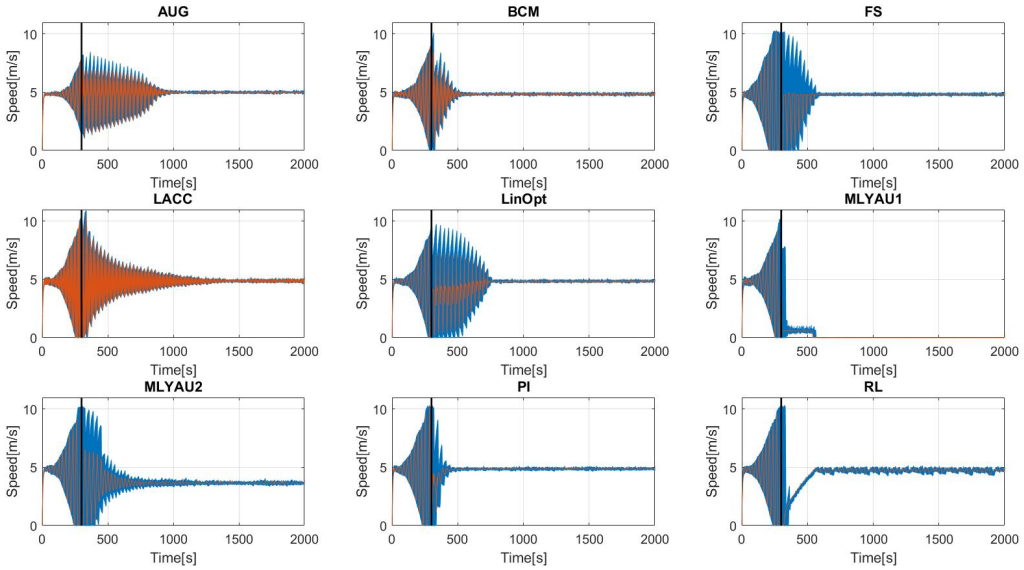


Figure 6.3: Speed profiles of the stabilized results; red: automated vehicle; blue: human driving vehicle; black: indication of beginning of AV control. For the results of AUG and BCM, four AVs are placed on the ring road. For the result of LACC, nine AVs are placed on the ring road. For the result of the rest of the AVs, only one AV is placed on the ring road.

hand, for the AV getting less stable at higher penetration rates, the VMT decreases, e.g. PI. For fuel economy, we can observe that the fuel economy is improved for AVs with better stability. While most AVs have generally improved fuel economy than the baseline case, except some AVs, both FUZ and MLYAU1 have worse fuel economy, because they either make the traffic less stable or block the traffic. Fuel economy generally gets better with increase penetration rates. It is particularly significant for the AVs that are not stable at lower penetration rates, e.g. AUG, BCM, LACC. There are two exceptions that the fuel economy is getting worse at higher penetration rates: FS, and PI. For PI, as we have seen, the stability is getting worse at higher penetration rates; hence, the fuel economy is also getting worse at higher penetration rates. It is interesting to notice that the fuel economy of FS also gets worse even though the traffic flow is getting smoother. The possible reason is that FS is actually doing a lot of subtle acceleration and deceleration. This is because FS is essentially a speed controller, the process of converting speed command to acceleration introduces 'noise' into acceleration command, which causes higher fuel consumption.

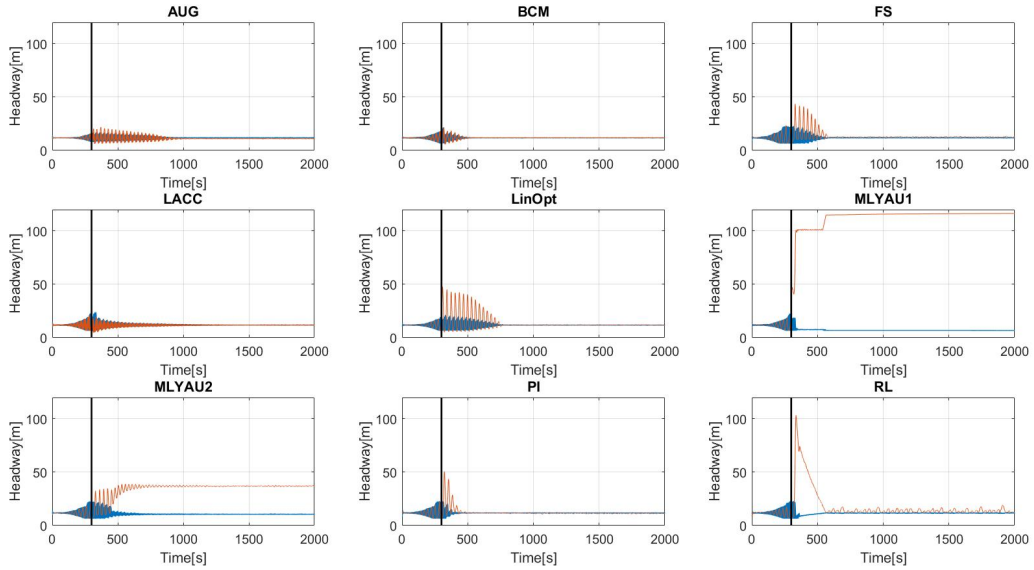


Figure 6.4: Headway profiles of the stabilized results;red: automated vehicle; blue: human driving vehicle; black: indication of beginning of AV control. For the results of AUG and BCM, four AVs are placed on the ring road. For the result of LACC, nine AVs are placed on the ring road. For the result of the rest of the AVs, only one AV is placed on the ring road.

Scenario II

Results of scenario II are presented in this section. Similar to the previous section, for scenario II, the corresponding values when the traffic on the ring is stable are presented in table 6.7. In this scenario, the penetration rate is only limited to about 9% to 50% of AV penetration in the system given that the placement of the AVs has to follow an even distribution (as shown in table 6.5).

From table 6.7, we can see that the overall results appear to be similar in Scenario I with minimal changes in the values. Again, it failed to stabilize when the FUZ is used. It takes the most number of AVs for LACC to work while the number of AUG needs to be increased to six (6) from three (3) and the number of BCM needs to be increased to five (5) from three (3) in Scenario I for the system to be stable.

To further differentiate the controllers based on their performance at different penetration rates, results across different penetration rates also presented here is figure 6.6. Again, we can see the same trend as in the previous scenario if we look at figure 6.6-(a) where the time to stabilize for the ring generally decreases as AV penetration rate increases, except for PI where it is also could get unstable at high penetration rates. In figures 6.6-(b), we can restate our prior observations that the maximum gap is invariant with respect to the

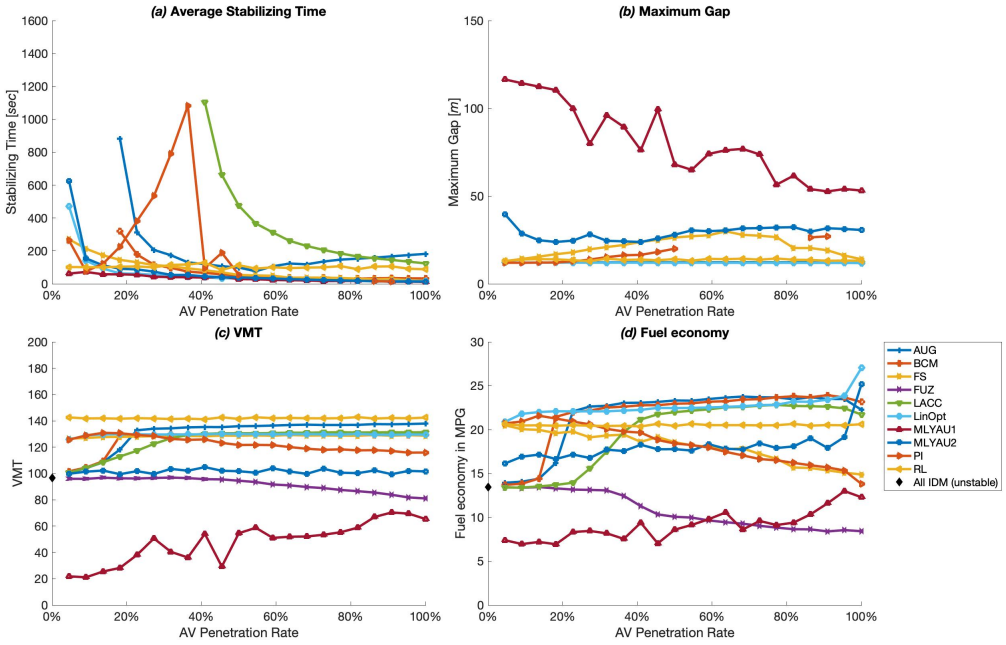


Figure 6.5: *(a)* Time to stabilize; *(b)* Maximum final gap; *(c)* VMT; *(d)* Fuel economy for the clustered case scenario. For (c) and (d), black dots are the baseline scenario values, where all vehicles are HVs.

penetration rate except for the two Lyapunov-based controllers for the same reasons that are previously mentioned. In figure 6.6-(c) and 6.6-(d), the trends of VMT and fuel economy are also similar to the previous scenario.

Comparison across two scenarios

To further compare the influences of distributions of AVs, figure 6.7-figure 6.10 arrange the results in a different way. We overlay the results of different scenarios on top of each other scenario for different types of automated vehicles. Figure 6.7 shows the time to stabilize for each AV. For most of the AVs, time to stabilize is decreasing with increasing penetration rate, or kept the same. However, PI shows that as penetration rate increases, traffic becomes less stable. This may imply that the PI is more appropriate when AV penetration rate is low. For most of the AVs, It can be noted that variation of the stabilizing time is not significant across different AV distributions while it is more relevant to AV types and penetration rate. The BCM shows significant difference between two distributions. The performance of BCM is much better when they are clustered together than being separated.

Figure 6.8 shows maximum gap for each AV. Only the results of FS show that the

Controller	Min no. of AVs to stabilize	Time to stabilize in *sec*	Max gap in *m*	VMT	Fuel economy in *mpg*
AUG	6	272.04	12.19	133.83	22.28
BCM	5	1516.9	12.29	127.06	19.32
FS	1	196.23	12.70	127.17	20.18
FUZ	*Unstable*	*N/A*	*N/A*	*N/A*	*N/A*
LACC	9	1213.55	12.13	129.98	21.05
LinOpt	1	120.50	12.12	128.99	21.82
MLYAU1	1	61.41	114.97	24.27	7.70
MLYAU2	1	151.23	36.87	101.13	16.82
PI	1	86.98	12.01	128.69	21.13
RL	1	241.03	14.52	138.69	19.70

Table 6.7: Average performance comparison of scenario II - Evenly Distributed

maximum gap is a little shorter in the case of the evenly distributed scenario. There is no significant discrepancy for other type of AVs.

Figure 6.9 shows the VMT for each AV. VMT is also not sensitive to the variation of AV distributions. With the help of AVs smoothing traffic flow, the VMTs are significantly increased compared to the baseline case, except for FUZ and MLYAU1. FUZ results in slightly lower VMT because it was not able to stabilize the traffic flow under any penetration rate. MLYAU1 stops the traffic so that the VMT is much smaller than the baseline. Figure 6.10 shows the fuel economy for each AV. For AVs, fuel economy does not show much change with the change in penetration rates and AV distributions.

6.4 Summary

In this chapter, automated vehicles dissipating stop and go waves on a ring road are studied under different distributions and penetration rates. To evaluate their performances, experiments are carried out in the state of the art simulation framework. Our findings are summarized as follows:

- Out of 10 automated vehicles, 6 automated vehicles models are able to stabilize the ring road traffic at penetration rate as low as 5 percent. They are FS, LinOpt, MLYAU1, MLYAU2, PI, and RL.
- Generally, *stabilizing time* is shorter if more AVs are placed in the traffic. However, for PI, the performance is degraded when more AVs are placed.

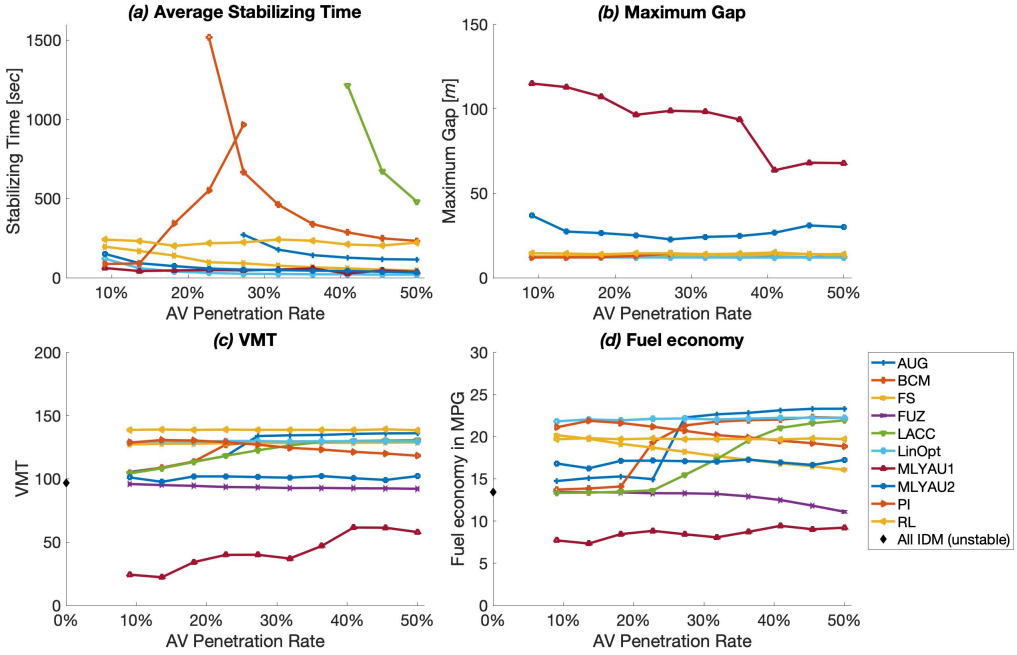


Figure 6.6: *(a)* Time to stabilize; *(b)* Maximum final gap; *(c)* VMT; *(d)* Fuel economy for the evenly distributed scenario. For (c) and (d), black dots are the baseline scenario values, where all vehicles are HVs.

- FUZ is the only controller that is not able to meet stability based on the set criterion but it is able to ensure that the traffic moves on the ring without encountering any vehicular crashes.
- RL controller shows the most consistent traffic improvement under all circumstances for all four metrics. It shows more than 40 percent VMT improvement while maintaining good fuel economy.
- MLYAU1 stabilizes the ring road with a trivial speed, where it almost stops all the vehicles in the system.
- For most of the AVs, the performance of the AVs are not much related to the way they are distributed among the traffic (except PI and BCM). No matter whether they are clustered (*i.e.,* platooned) or evenly distributed, performance of AVs is pretty much the same. However, this shall be treated with caveat. This might only be applicable when we treat control of AVs individually (*i.e.,* no cooperation/communication among them) as a recent study by Li *et al.*[86] mentions that the distribution of AVs may have a big impact when we consider the cooperation of multiple AVs.

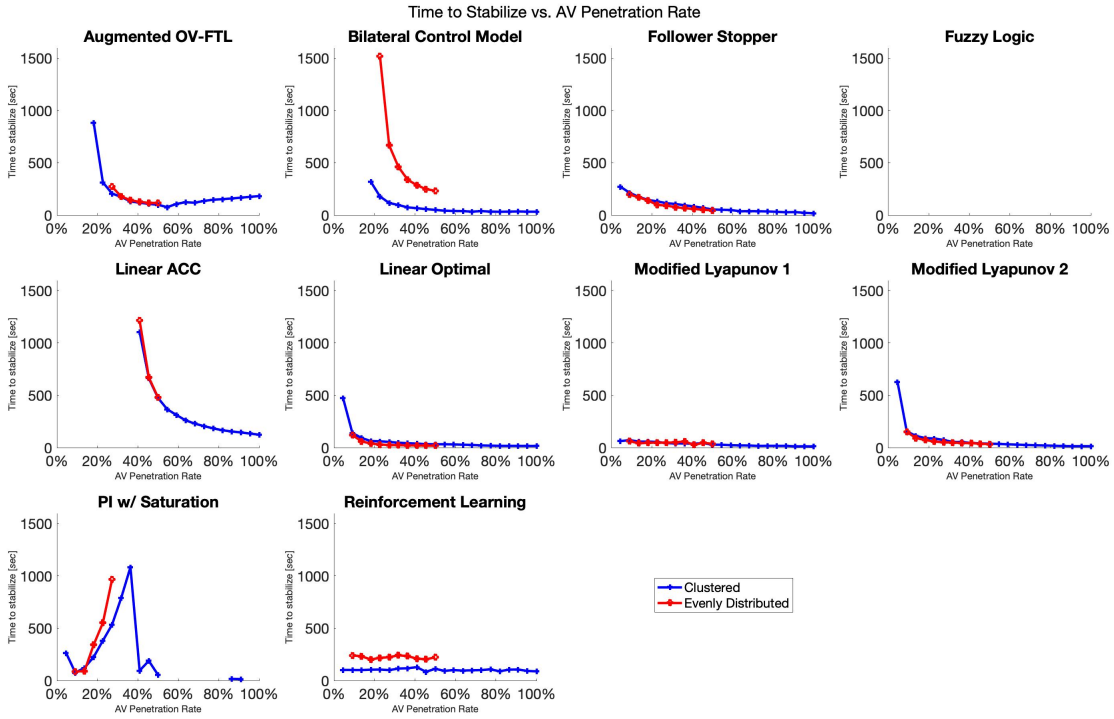


Figure 6.7: Time to stabilize the ring road for AV controllers under different penetration rates and different distributions.

- PI is one of the two controllers that shows some difference between two distributions. It is interesting to see that the performance of the PI is not as good as in the *clustered* case compared to being *evenly distributed*. When the PI is placed in the manner of scenario II, the performance of PI is degraded severely as multiple AVs are placed in the traffic.
- Another controller that shows quite different results for different distributions is BCM. The results of *time to stabilize* also show that it performs better under clustered scenario.

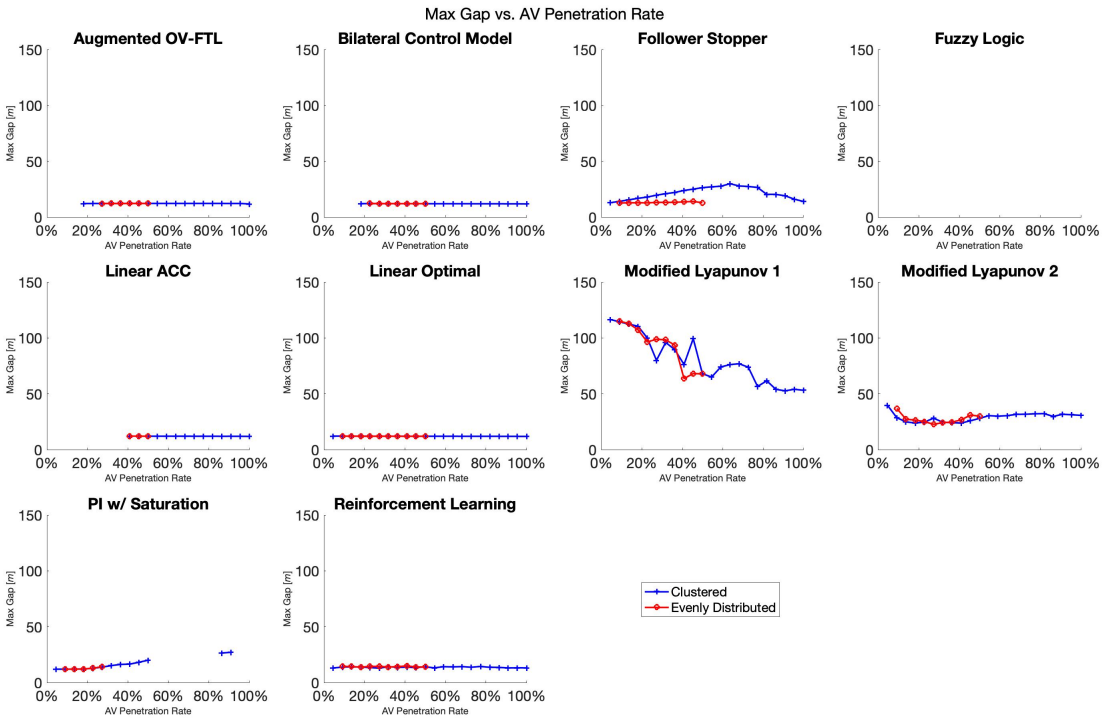


Figure 6.8: Maximum gap for AV controllers under different penetration rates and different distributions.

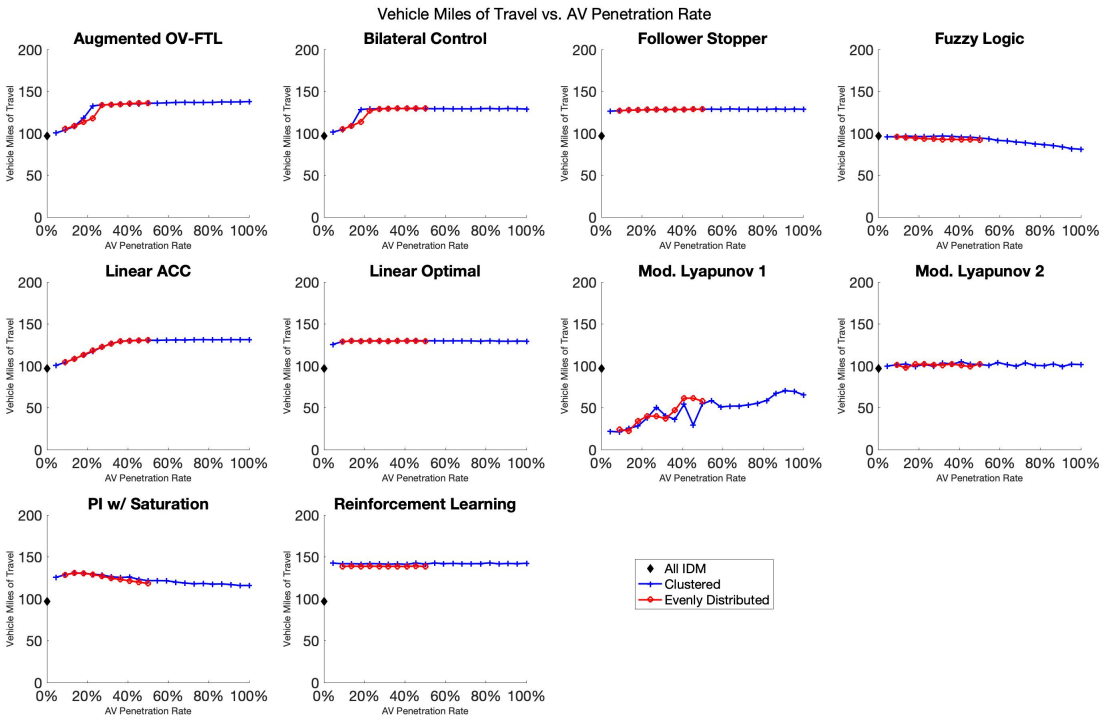


Figure 6.9: Vehicle Miles of Travel for AV controllers under different penetration rates and different distributions. The baseline VMT is 96.71.

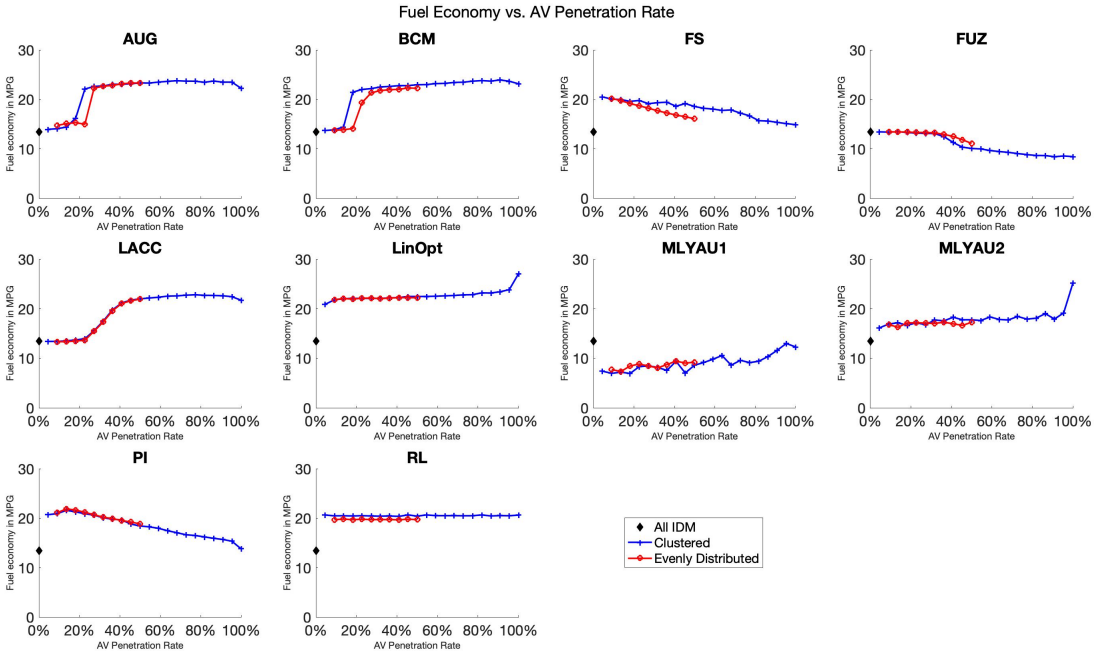


Figure 6.10: Fuel economy in MPG under different penetration rates and different distributions. The baseline fuel economy is 13.12 MPG.

Chapter 7

Designing Transit Routes based on Vehicle Routing Behavior Determined through Location-Based Services Data

7.1 Overview

Motivation

As cities expand, the need for sustainable and accessible transit systems becomes increasingly urgent. The efficiency and effectiveness of public transportation have a significant impact on the quality of life for both residents and visitors. Routing, which forms the backbone of transit systems, determines their efficiency and reliability. Well-designed routes can reduce congestion, minimize environmental impact, and shorten travel times.

One prominent concern is the selection of transit routes, which can sometimes overlook key areas, leading to inaccessibility. Inefficient systems may result in overcrowded vehicles in some areas and underutilized vehicles in others, leading to low ridership. Accurately predicting commuter demand remains a challenge, affecting service planning and optimization.

Addressing these challenges is crucial, as increased public transit usage can alleviate traffic congestion and is more environmentally sustainable than individual car travel. Despite efforts to promote alternative modes of transportation, the benefits of optimized public transit are often underestimated.

Many commuters are seeking relief from the burden of driving, indicating a demand for a system where navigating traffic becomes someone else's responsibility. An efficient transit system could provide such a solution. To enhance bus transit efficiency, one of the most recent works by Yoo and Lee [174] on bus route planning offers a detailed discussion on current practices for optimizing bus routes and introduces a reinforcement learning (RL) approach as their contribution. They found that RL shows considerable potential in managing the complex, multifactorial interactions within a bus transport system. Additionally, they mention metaheuristic techniques, such as genetic algorithms and simulated annealing, which are

being considered to improve bus routes. While most of the literature focuses on optimizing existing bus routes, our work aims to introduce a method for identifying potential locations for route extensions or new bus routes, particularly in cities planning to enhance their services. According to Yin and Zhang [173], common urban bus corridor location methods primarily rely on empirical judgment, travel expectations, and two-step clustering methods.

Understanding vehicular routing behavior with location-based service data

Location-Based Services (LBS) data is collected via applications installed on smart mobile devices. These applications gather users' geographical locations using a global positioning system (GPS) or Wi-Fi positioning system (WPS). In this study, LBS data was provided by Cuebiq, a location intelligence and measurement platform.

Xu et al. [170] devised a method for understanding vehicular routing behavior using LBS data. To process the data, a series of algorithms were employed to extract vehicular trips and reconstruct the origin-destination matrix. Initially, non-vehicular trips, such as walking or biking, were filtered out. Next, a clustering algorithm was applied to group GPS points into trips based on their spatial and temporal proximity. A speed filter was also utilized to exclude trips with unrealistic speeds. The origin-destination matrices were then reconstructed by aggregating trips between each pair of locations. These procedures established a robust foundation for analyzing vehicular routing behavior through LBS data. The processed data from their study served as the input for our work.

The potential uses of LBS data are extensive, with one of the most promising being the enhancement of public transit efficiency and the reduction of traffic congestion. By analyzing LBS data, transportation planners can gain valuable insights into travel patterns and identify areas where public transit could be optimized. For instance, LBS data can help pinpoint regions with high demand for public transit, allowing for the optimization of transit routes and schedules to better serve these areas. Moreover, LBS data can identify high congestion areas, enabling the development of strategies to alleviate traffic and improve transit efficiency. Overall, LBS data holds significant potential for providing insights into human behavior and informing policy decisions that can contribute to more efficient and sustainable transportation systems.

Contribution

In our research, we investigate decentralized planning by leveraging location-based data from mobile devices. This methodology enables a bottom-up transit planning process that prioritizes demand, particularly in underserved areas, while considering the existing transit infrastructure. Such an approach can enhance inclusivity and equity, allowing all users of location-enabled apps to contribute to the optimization of the transit system.

The bottom-up approach we proposed is detailed in the steps outlined below and is visually represented in Figure 7.1.

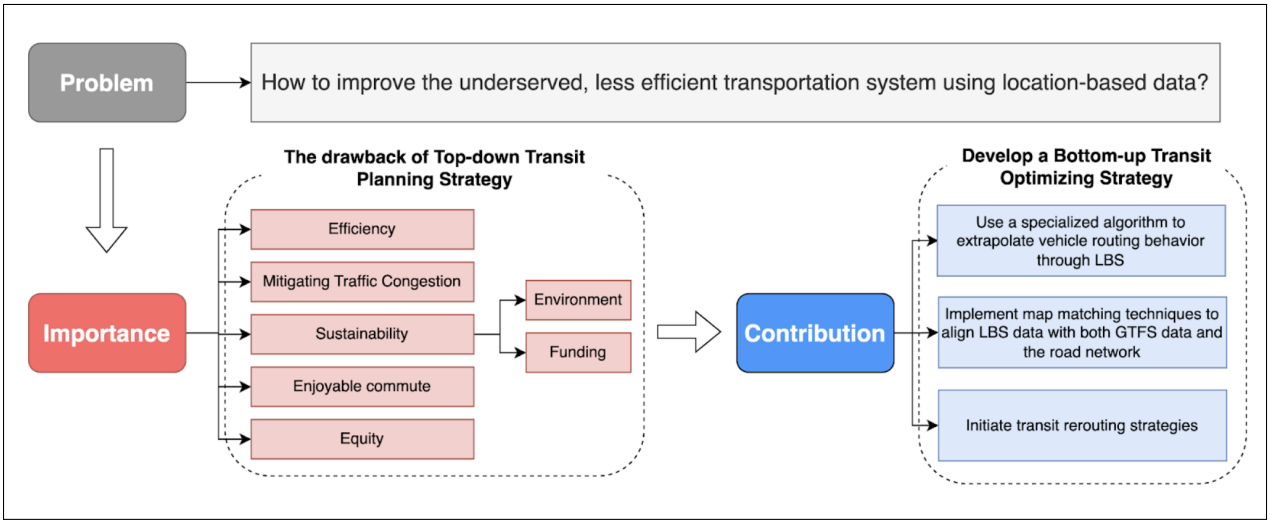


Figure 7.1: Research Framework. The research framework highlights the problem, importance, and contributions of our work which is focused on developing a bottom-up transit optimization strategy in comparison to the top-down planning strategy.

1. Acquisition of road network data from OpenStreetMap, subsequently facilitating the construction of a detailed road network representation for Dallas.
2. Synthesis of a network leveraging the GTFS data, which encompasses public transit routing information.
3. Implementation of map matching techniques to align LBS data with both GTFS data and the aforementioned road network, ensuring comprehensive overlap and data congruence.
4. Initiation of bus routing strategies, prioritizing those segments of the road network characterized by a high user count yet lacking in transit options.

Through these steps, we aspire to bridge the gap between observed demand and available transit options, fostering a more responsive and equitable transportation ecosystem. However, our work does not cover operations, thus we do not address frequency, operational costs, or resource allocation for operating the proposed bus lines or potential extensions. Our method serves as a data ingestion mechanism to test and initiate a mainstream decision framework.

7.2 Data

GTFS Data

General Transit Feed Specification (GTFS) is a widely adopted format for public transit data, which enables the representation of comprehensive information about transit systems. GTFS data, typically provided in a collection of text files, with detailed aspects of public transit operations such as routes, stops, schedules, and trip planning information. For the city of Dallas, the GTFS data was sourced from TransitFeeds, an extensive repository of public transit data [147]. Table 7.1 provides a concise summary of the primary files used in our analysis, including information about transit stops, routes, trip schedules, geographical paths of routes, and the timing at each stop. This granular level of data is essential to analyze the city's transit system, enabling a deeper understanding of its efficiency and coverage.

Table 7.1: Description of GTFS Data Files

File Name	**Description**
stops.txt	Contains information about individual transit stops including their IDs, names, and geographical coordinates.
routes.txt	Includes details about the transit routes such as route IDs, route names, and route types.
trips.txt	Describes trips associated with each route, including trip IDs, route IDs, and service IDs.
shapes.txt	Provides geographical paths for each route, described through a series of latitude and longitude points.
stop_times.txt	Lists the arrival and departure times of buses at each stop for each trip.

LBS Data in Dallas, Texas

The dataset we used is the post-processed location-based services (LBS) data from the DFW Metroplex in Texas that was generated in the work of Xu et al. [170]. The dataset is collected by Cuebiq over a 6-month period from November 1st, 2016, to April 30th, 2017, in Texas, and includes around 6.5 million users and 12.43 billion records. Each record comprises a `timestamp`, unique `userID`, `tripId`, `datetime`, `longitude`, `latitude`, `grid`, and `geometry`. The description of the dataset's format is shown in Table 7.2.

Since the primary goal is to determine the road segments (or edges) used by the commuters, the first step is to perform map-matching to determine road usage.

Table 7.2: The Dallas LBS dataset format

Parameter	Description	Example Value
`userId`	A unique identifier for the user whose location data is being collected	3
`tripId`	A unique identifier for a user's collected trip	0
`datetime`	The date and time when the location data was collected	2016-11-01 06:46:55
`timestamp`	A numerical value representing the exact moment an event occurs	1478000815
`lon`	The geographic coordinate that specifies the east-west position of the user	-96.415063
`lat`	The geographic coordinate that specifies the north-south position of the user	32.728888
`grid`	A spatial grid or a set of predefined geographic cells	113393
`geometry`	The geographic coordinate of the user	POINT (-96.4150631, 32.7288877)

7.3 Methodology

Preprocessing of GTFS

The preprocessing of Dallas's GTFS data for bus transit focused on extracting bus-specific details from the extensive dataset. Each bus route was geospatially mapped onto the Dallas map, applying buffering to the route shapes. This was followed by calculating the number of trips per bus to determine the frequency of buses along each edge of the map.

In the spatial analysis of bus routes, buffering played a pivotal role, serving two primary purposes. Firstly, it compensated for potential inaccuracies in the geographical data, ensuring a more robust analysis. More importantly, buffering allowed the transit paths to encompass nearby parallel routes. This approach acknowledges the practical impact of a transit route on its immediate vicinity, effectively considering a single transit path as serving adjacent parallel paths. This methodology was chosen as it represents a more realistic description of the bus network's coverage, especially in areas with closely situated parallel routes.

The end result of this analysis is a detailed mapping of each edge in the Dallas network, indicating the frequency of bus services. This includes not just the number of different buses, but also accounts for the same bus passing multiple times, providing a comprehensive view of service regularity.

Map Matching

Hidden Markov Model (HMM) [110] is used to find the most likely road route for the time-stamped sequence of latitude and longitude pairs of LBS data. Through this method, it is expected that we are able to determine the road segments frequently used by the users. The fundamental idea behind HMM involves discretizing the map into a series of states, where

each state corresponds to a location or road segment on the map. The observed values are the position points along the trajectory, and the model determines the most likely path by considering the transition probabilities between states and the observation probabilities at each state. This modeling approach enables HMM-MP to provide a reasonable mapping of trajectory data onto the map while accounting for uncertainty and noise.

In a more concrete sense, HMM represents the road network as a graph composed of nodes and edges, where nodes are located at intersections, dead ends, and road name changes, and edges represent road segments between nodes. Each node has associated latitude and longitude coordinates, and each edge is represented by a sequence of latitude and longitude pairs outlining its geometry. Considering measurement noise and road network layout, HMM matches the latitude and longitude point sequence to the roads, identifying the most likely route.

In this process, each edge on the map is treated as a state in the HMM. The state set $X = \{X_1, X_2, \ldots, X_N\}$ includes all possible segments on which a vehicle could be located at any given time. These are the N hidden states, which are not directly observable. The system transitions among these states. The position points on the trajectory are considered as observation values, which are the T observations that are visible. Each observation y_t for $t = 1, 2, \ldots, T$ is associated with one of the states and represents a location point at a specific time, but the direct correspondence is not known. The transition probability matrix A is a $N \times N$ matrix where each element a_{ij} represents the probability of moving from state X_i to state X_j, which means the probability that vehicle actually goes from road X_i to road X_j. The probabilities from each state must sum to 1:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$$

The algorithm aims to determine the most likely trajectory on the map, providing an estimate of the actual path of a vehicle or moving object.

The matching process starts by selecting the initial position of the trajectory and matching it to an edge on the map. Then, by considering transition probabilities a_{ij} and updating the current state's position based on observation probabilities, the algorithm infers the next location on which edge the vehicle is likely to be. This step relies on the topological structure of the map, where edges represent road segments, and states represent positions on these segments.

Throughout the matching process, observation values, i.e., LBS points on the trajectory, update the position associated with the current state through observation probabilities, refining the matching results. By iteratively applying state transitions and observation updates, the algorithm forms the most likely path on the map, ultimately matching the trajectory points to edges on the map.

The key idea behind this matching method is to consider the uncertainty and noise between the map and the trajectory. The HMM has the capability to model uncertainty, adapting flexibly to various uncertainties present in the real environment, such as LBS

data measurement errors and the complexity of the road network. By learning the model parameters and utilizing techniques like the forward-backwards algorithm, the HMM-MP algorithm effectively matches trajectory data to the map, providing a reasonable estimation of the most likely path. The strength of this method lies in its ability to model and match complex trajectory data.

Algorithms based on the HMM can handle additional measurement noise with a standard deviation of up to 50 meters, roughly equivalent to the accuracy of WiFi-based multi-range measurements. The graph demonstrates the algorithm's robustness in handling additional noise with different sampling periods. In longer sampling periods, the algorithm exhibits robustness against measurement noise with a standard deviation of up to 50 meters.

The road network where the LBS data will be matched was extracted from OpenStreetMap using the OSMNx package by Boeing [25]. In this experiment, we have chosen a geographical region with latitude boundaries of 32.8 to 32.7 degrees (north to south) and longitude boundaries of -96.7 to -96.8 degrees (east to west) for Map Matching. This area encompasses downtown Dallas. Utilizing OpenStreetMap, we have successfully generated a map image of the specified area.

We used the DistanceMatcher from the Leuven Map Mathing package [101] for map matching, which considers the distance between matched locations on the map compared to the distance between the observations matched to these locations. The description of our parameter settings is in Table 7.3.

Table 7.3: Parameter Settings

Parameter	**Description**	**Value**
`max_dist`	Maximum distance from the start location to points on the path	500
`max_dist_init`	Maximum distance from the start location to the path	170
`min_prob_norm`	Minimum normalized probability of observations	0.0001
`non_emitting_states`	Allow non-emitting states	True
`non_emitting_length_factor`	Reduce the probability of a sequence of non-emitting states	0.95
`max_lattice_width`	Restrict the lattice to this value	20
`obs_noise`	Standard deviation of observation noise	50
`obs_noise_ne`	Standard deviation of observation noise for non-emitting states	50
`dist_noise`	Difference between state and observation distance's standard deviation	50
`dist_noise_ne`	Standard deviation of difference between distance for non-emitting states	50

After processing the travel data, we applied the aforementioned method to perform map matching on two travel records. Subsequently, we generated a dictionary that documents how many LBS data points are matched to each node or edge. We employed this dictionary for visualization on the map as illustrated in Figure 7.2. In the figure, the red and blue lines represent the routes of two travel record trajectories after map matching.

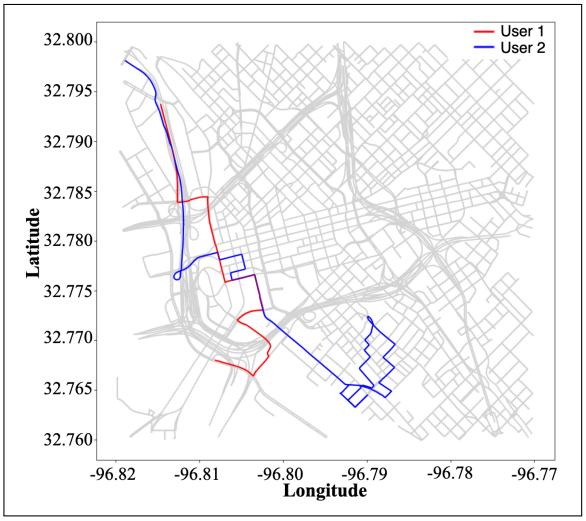


Figure 7.2: Travel Routes Map Matching Visualization. This figure shows the map-matched routes taken by two different users with overlapping edges.

Identification of Underserved Transit Regions

To identify regions within Downtown Dallas that are underserved by transit services, we employed a data-driven approach visualized through a scatter plot, as shown in Figure 7.3. The methodology involves a comparative analysis between the normalized Location-Based Services (LBS) frequency, which serves as a proxy for user demand, and the normalized frequency of bus transit, representing the supply.

The scatter plot as shown in Figure 7.3 was generated by plotting the normalized LBS frequency on the x-axis against the normalized bus frequency on the y-axis for each edge in the network. Edges that fall below a predetermined threshold indicate areas where the demand for transit, inferred from the LBS data, outstrips the supply of bus services. These points are indicative of potential transit deserts—areas where the transit network is not adequately serving the population's needs.

This analytical framework allows us to pinpoint specific geographic locations that require attention from urban planners and transit authorities to improve service frequency and ensure equitable transit access across the city.

7.4 Transit Planning and Evaluation

In Section 7.3, we have successfully identified the underserved edges within our transportation network that are potentially in need of enhanced transit service. Building upon this foundation, this section delves into the methodologies employed to transition from recog-

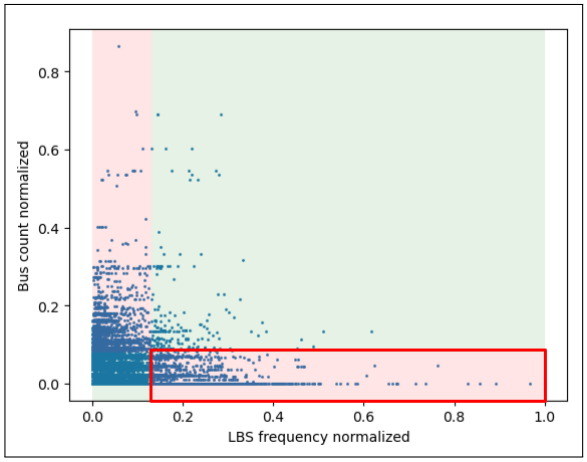


Figure 7.3: Scatter plot of LBS data and bus frequency for each edge in Downtown Dallas, illustrating the relationship between user demand and bus service supply. The bottom right region is the critical region in question.

nizing these critical areas to proposing practical recommendations for new transit routes or extending existing ones. This section will follow up on evaluations of the proposed method.

Underserved Edge Clustering Based on Shortest Path

In this study, we initiate the analysis by segmenting the study area into distinct regions or clusters of interest. This segmentation is achieved through the application of k-means clustering on the identified underserved edges, utilizing Euclidean distance as the optimization metric. Each resulting cluster corresponds to a specific region targeted for proposed transit routes. The clustering method was implemented using scikit-learn [114].

Figure 7.4 shows the different clusters identified for Dallas. Following this initial clustering, we perform further clustering within each region based on the shortest path distance. The purpose of this step is to pinpoint critical points that will dictate the trajectory of the proposed transit routes. Subsequently, we find the best route that passes through the points using a greedy approach as shown in Algorithm 1. Therefore, we use a two-layer k-Means clustering with the first layer to divide the whole area into regions by minimizing the Euclidean distance between the centroids and datapoints, and the second layer to determine centroid locations within the isolated region by implementing the shortest path-based clustering algorithm.

As a test case, we examine one of the previously identified regions in the Far North Dallas area shown in Figure 7.5 where centroids of the road links that represent underserved roads are in blue. The red points are obtained by performing k-means clustering, which represent the ciritcal points near which the recommended transit path should travel. Then, we filter

out outliers which are centroids that have less 5 edges assigned to them. After applying Algorithm 1 on the remaining clusters, we obtain the suggested route shown in Figure 7.6.

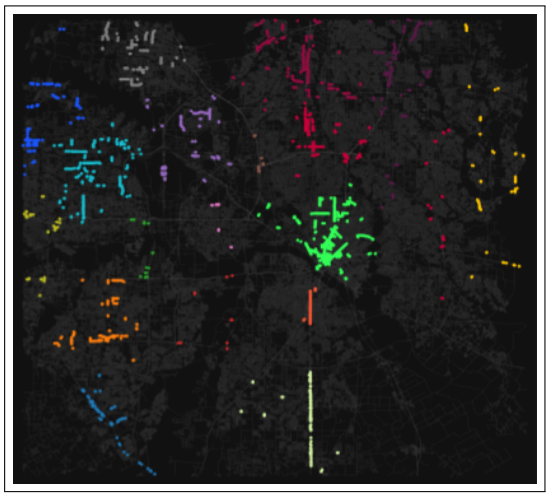


Figure 7.4: Clustering result of the Dallas transportation network after the application of the k-means clustering algorithm. Different colors represent unique clusters, each corresponding to a group of road segments identified as underserved based on LBS data frequency. These clusters signify potential areas where transit service improvements are needed.

Model Evaluation

We evaluate the model's effectiveness by focusing on a case study within a clustered region where Algorithm 1 identifies a transit route as seen in Figure 7.6. To assess the impact of this route on the region, we estimate the potential Vehicle Miles Traveled (VMT) reduction. This is done by creating a 1 km buffer around the route shown in Figure 7.7 and then having a rough estimate of the potential demand covered using the LBS data within this buffer. The Federal Highway Administration's [52] estimates are used to convert this coverage into VMT saved, using the equation shown:

$$VMT_{reduced} = VMT_{total} \times \frac{\sum_{e \in S} f_e}{\sum_{e \in E} f_e}$$

Where E is the set of all road links in the network, and f_e denotes the frequency of LBS data points associated with edge e, serving as an indicator of transit demand. The set S is the set of edges in the examined cluster. VMT_{total} is the estimate of the total VMT for Dallas from the Federal Highway Administration. After this analysis, we find that the proposed line has the potential for a 760,000 VMT reduction, assuming every driver on or near the route took the bus instead of driving. This translates into a substantial reduction of vehicular travel, contributing to decreased traffic congestion and emissions within the region.

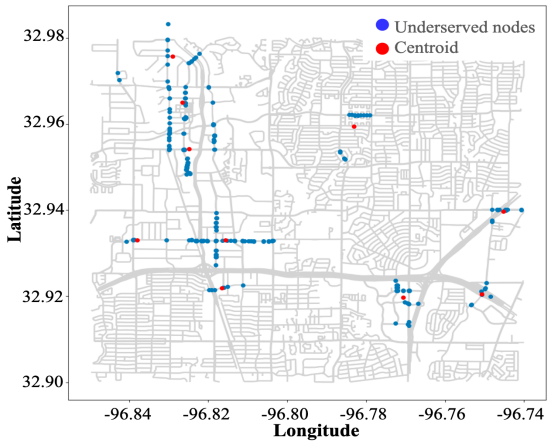


Figure 7.5: Clusters of underserved edges in a region located in Far North Dallas. Blue markers represent underserved road segments, with the critical points (*i.e.,* centroids) highlighted in red. These key nodes, determined through k-means clustering, indicate priority areas to help with transit route design.

Algorithm 1 Shortest Path Clustering

Input: Road network R, number of clusters k
Output: Set of shortest path routes for each cluster C
Label edges of R as 'underserved' based on the results in Figure 7.3
Initialize matrix M to store shortest path distances
for each edge e_i in R **do**
  for each edge e_j in R **do**
    Calculate shortest path distance d_{ij} from e_i to e_j
    $M[i][j] \leftarrow d_{ij}$
  end for
end for
Perform k-means clustering on M to form k clusters
Initialize C as an empty set
for each cluster $\mathcal{C}_n$ in the k clusters **do**
  Identify the most apart points p_a and p_b in $\mathcal{C}_n$
  Calculate shortest path P_{ab} between p_a and p_b in R
  Add P_{ab} to C
end for
return C

Figure 7.6: Proposed transit lines overlaid on a road network with varying intensities of yellow, indicating the frequency of LBS data. The red lines show the suggested routes in alignment with roads exhibiting higher LBS frequencies, reflecting higher demand for transit services.

Figure 7.7: Buffer zone coverage represented by the yellow shading around the proposed transit routes, demonstrating the potential reach of these routes in meeting the transit demand across adjacent road segments.

7.5 Summary

Our work demonstrates the potential of using LBS data in bus route planning, enabling a more profound understanding of urban mobility patterns. The integration of this data with GTFS data allowed for an inclusive analysis of public transit accessibility and efficiency.

The adoption of map-matching algorithms based on HMM aligns LBS data with road networks well. Then, our approach included the application of two-layer k-Means clustering, which effectively identified regions of high demand for public transit, offering a foundation for targeted transit routes within those regions.

Our findings suggest that the proposed approach not only meets the majority of transit needs but also offers a pathway to substantial VMT reductions. The intelligent use of data analytics in transit planning provides a compelling case for the potential improvements in

traffic efficiency and environmental benefits through decreased vehicle usage.

In potential future extensions of our study, incorporating operational factors—such as resource allocation, service frequency, and stop placement—is a potential direction to further enhance the practicality of transit planning. Additionally, adopting advanced data validation methods through emerging data sources and sensor technologies could significantly bolster the accuracy of demand prediction and route optimization in public transportation systems.

Chapter 8

Concluding Remarks

In the field of transportation engineering and urban planning, the investigation into the effects of information-aware routing on traffic systems has been both informative and challenging. As we conclude this dissertation, it is essential to reflect on the key insights gained, the contributions made, and the future directions for shaping transportation in the digital age.

The examination of traffic simulation models has been crucial in understanding the complex dynamics of routing decisions and their implications on traffic flow. Through microscopic simulations, we have dissected the complexities of individual driving behaviors, congestion patterns, and system performance, providing an understanding of the interplay between user choices and network efficiency.

The development and calibration of a large-scale traffic microsimulation for the San Jose Mission district in Fremont, CA, are detailed in Chapter 2. This microsimulation process, which took approximately 2,500 person-hours, provides a reproducible framework for creating, calibrating, and validating large-scale microscopic simulations. These simulations are valuable for real-world case studies particularly what-if scenario experiments, provided that sufficient data is available to accurately reproduce demand. Our realistic traffic microsimulation aids in understanding current traffic patterns and estimating the potential impacts of new policies on routing and driving behaviors.

Chapter 3 explores various predictive models for traffic flow. Linear regression models that used time, absolute postmile, Vehicle Detection Station (VDS) ID, and day of the week as features performed better on test data than models based on eigenvectors from the training set. Including k-means clusters further improved these models. However, random forest models, even with default parameters, outperformed the other models, identifying 'Postmile (Abs)' and 'VDS ID' as the most critical features. Despite its superior performance, the random forest model had the longest training time among the models considered, suggesting a trade-off between accuracy and computational efficiency.

Meanwhile, our investigation into the effects of navigation app usage on traffic flow revealed that dynamic routing significantly influences overall traffic performance and congestion. Using the SIR model, our simulations indicated that optimal results occur when 30-60%

of users employ dynamic routing. These findings highlight the potential for increased congestion due to widespread navigation app usage and suggest the need for policies promoting dynamic routing adoption while preventing overuse. Understanding the collective benefits of routing schemes is crucial for developing new algorithms or mechanisms for navigation apps, though this poses challenges related to compliance and route suggestion limitations.

In chapter 5 where we have seen two (2) case studies, our analysis of the four traffic scenarios — Uncoordinated, Status Quo, Zero Offset Uniform Cycle Lengths, and L/V Coordination — reveals differences in traffic flow, speed, and travel time, especially during peak congestion hours. The Uncoordinated scenario led to increased traffic flow on Route 1 via I-680 and reduced average flow on Mission Blvd, whereas the Zero Offset and L/V Coordination scenarios effectively reduced flow and improved travel times for Route 1. Although speed changes were minimal, travel time improvements for Route 1 were notable, highlighting the effectiveness of strategic traffic signal coordination. Further simulation experiments are needed to validate these findings and explore their broader applicability. Additionally, in our second case study, our results demonstrate that traffic patterns observed in simplified models can be extended to complex networks like Fremont's. By using concepts such as Pigovian taxes, our simulations suggest that adjusting speed limits can influence traffic distribution which may potentially enhance road safety, offering practical insights for real-world traffic management and policy.

We also examined the impact of automated vehicles on traffic stability. Our experiments showed that automated vehicle models could stabilize traffic with low penetration rates. The performance of these models varied with their distribution in traffic, suggesting that the impact of automated vehicles depends on both their control algorithms and their deployment strategies. Notably, the reinforcement learning (RL) controller consistently improved traffic under all conditions, while the performance of other controllers varied with their distribution.

With the goal of also promoting a more sustainable transportation option, Chapter 6 demonstrates the potential of using location-based services (LBS) data in bus route planning, enabling a more profound understanding of urban mobility patterns. The integration of this data with General Transit Feed Specification (GTFS) data allowed for an inclusive analysis of public transit accessibility and efficiency. Our findings suggest that the proposed approach not only meets the majority of transit needs but also offers a pathway to substantial vehicle miles of travel (VMT) reductions. The intelligent use of data analytics in transit planning provides a compelling case for potential improvements in traffic efficiency and environmental benefits through decreased vehicle usage.

8.1 Future Work

As we conclude this dissertation, we envision several promising directions for future research:

Improving training efficiency for machine learning models, including random forests, is crucial. Addressing the lengthy training times by exploring more efficient algorithms or optimizing current models could significantly reduce computational costs. Additionally, evaluat-

ing model performance on various freeways is important for validating their generalizability and robustness. This evaluation could provide insights into freeway-specific traffic dynamics and help refine the models.

Exploring alternative features and algorithms is another key area. Experimenting with different feature sets, such as classifying days as weekdays or weekends, and testing other machine learning algorithms could enhance model accuracy and efficiency. Scaling traffic simulations to other areas involves understanding design architecture and regional traffic characteristics. Developing standardized validation and calibration toolboxes could facilitate this process.

Further research into optimal routing strategies could determine the ideal percentage of dynamically routed users to minimize congestion. Sensitivity analyses might explore how this percentage changes under different conditions, such as weather or accidents. Moreover, investigating driver compliance with navigation apps and the broader impacts of traffic flow changes, including inequities in air and noise pollution, would offer a more comprehensive understanding of navigation apps' effects.

Future work in traffic light coordination should focus on evaluating the long-term impacts of various coordination strategies on traffic patterns and network performance. Integrating real-time data and adaptive control systems, as well as exploring multi-modal traffic interactions, is essential. Assessing speed limits as a Pigovian tax involves examining their long-term impacts on traffic flow, congestion, and safety, as well as their effects on different types of road users and traffic patterns. Integrating real-time traffic data to refine speed limit adjustments and exploring variable speed limits based on traffic conditions could provide more nuanced insights.

Integrating operational factors into transit planning models, such as resource allocation, service frequency, and stop placement, could enhance their practical applicability. Advanced data validation methods from emerging data sources could improve demand prediction and route optimization. Additionally, developing advanced calibration techniques that incorporate physical constraints on Origin-Destination demand, such as physics-informed neural networks or nuclear matrix norms, could be valuable.

Beyond the scope of this dissertation, we hope that the implications of our research are extensive, resonating with policymakers, practitioners, and researchers alike. The insights derived from our study of information-aware routing can serve as a catalyst for rethinking traditional paradigms, fostering innovation, and promoting sustainable change in the transportation sector. We hope that the insights from this dissertation will inspire all those dedicated to transforming the way we move, connect, and thrive in an ever-evolving urban landscape.

Bibliography

[1] International Energy Agency. *World Energy Outlook 2018 / International Energy Agency*. eng. Organisation for Economic Co-operation and Development Publishing, 2018. ISBN: 92-64-30677-3.

[2] Aimsun. *Aimsun Next 22 User's Manual*. Aimsun Next 22.0.1. Barcelona, Spain, 2022. [Online].

[3] Assad Alam et al. "Heavy-duty vehicle platooning for sustainable freight transportation: A cooperative method to enhance safety and efficiency". In: *IEEE Control Systems* 35.6 (2015), pp. 34–56.

[4] Javad Jomehpour Chahar Aman, Myriam Zakhem, and Janille Smith-Colin. *Towards Equity in Micromobility: Spatial Analysis of Access to Bikes and Scooters Amongst Disadvantaged Populations*. 2021. DOI: 10.3390/su132111856.

[5] Lukas Ambühl, Monica Menendez, and Marta C González. "Understanding congestion propagation by combining percolation theory with the macroscopic fundamental diagram". In: *Communications Physics* (2023). ISSN: 2399-3650.

[6] C Antoniou et al. *Fundamentals of Traffic Simulation*. Ed. by Jaume Barceló. Vol. 145. International Series in Operations Research & Management Science. New York, NY: Springer New York, 2010. ISBN: 978-1-4419-6141-9. DOI: 10.1007/978-1-4419-6142-6. URL: http://iet.jrc.ec.europa.eu/%20http://link.springer.com/10.1007/978-1-4419-6142-6.

[7] Neha Arora et al. "An Efficient Simulation-Based Travel Demand Calibration Algorithm for Large-Scale Metropolitan Traffic Models". In: *arXiv* (2021). DOI: https://doi.org/10.48550/arXiv.2109.11392. URL: https://arxiv.org/abs/2109.11392.

[8] Neha Arora et al. "Quantifying the sustainability impact of Google Maps: A case study of Salt Lake City". In: *arXiv preprint arXiv:2111.03426* (2021).

[9] Joshua Auld et al. "POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations". In: *Transportation Research Part C: Emerging Technologies* 64 (2016), pp. 101–116.

[10] Sangjae Bae et al. *Behavior, Energy, Autonomy, Mobility Modeling Framework*. Tech. rep. 7 Summits IT AG LTD, Zurich, Zurich; Skylite Networks, Fremont, CA; Lawrence . . ., 2019.

[11] Alben Rome Bagabaldo et al. "Impact of navigation apps on congestion and spread dynamics on a transportation network". In: *Data science for Transportation* 6.2 (2024). ISSN: 2948-135X.

[12] Alben Rome B. Bagabaldo and Marta C. González. "Predicting Traffic Flow on Faulty Traffic Detectors Using Machine Learning Techniques". In: *Proceedings of the International Conference on Transportation and Development 2022* (2022), pp. 202–212. DOI: 10.1061/9780784484319.019.

[13] Michael Balmer et al. "MATSim-T: Architecture and simulation times". In: *Multi-agent systems for traffic and transportation engineering*. IGI Global, 2009, pp. 57–78.

[14] Masako Bando et al. "Dynamical model of traffic congestion and numerical simulation". In: *Physical review E* 51.2 (1995), p. 1035.

[15] Masako Bando et al. "Structure stability of congestion in traffic dynamics". In: *Japan Journal of Industrial and Applied Mathematics* 11.2 (1994), p. 203.

[16] William J Baumol and Wallace E Oates. *The theory of environmental policy*. 2nd ed. Cambridge University Press, 2012. ISBN: 0521322243.

[17] Michael Behrisch et al. "SUMO–simulation of urban mobility: an overview". In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. 2011.

[18] Genrich Belitskii et al. *Matrix norms and their applications*. Vol. 36. Birkhäuser, 2013.

[19] Michael GH Bell. "The estimation of origin-destination matrices by constrained generalised least squares". In: *Transportation Research Part B: Methodological* 25.1 (1991), pp. 13–22.

[20] E. Bert, A. Torday, and A. Dumont. "Calibration of Urban Network Microsimulation Models". In: *Proc., 5th Swiss Transport Research Conf., Ascona, Switzerland* January (2005).

[21] Chandra R. Bhat and Frank S. Koppelman. "Activity-Based Modeling of Travel Demand". In: *Handbook of Transportation Science*. Boston: Kluwer Academic Publishers, 2006, pp. 39–65. DOI: 10.1007/0-306-48058-1_3. URL: http://link.springer.com/10.1007/0-306-48058-1_3.

[22] Peter J. Bickel et al. "Measuring Traffic". eng. In: *Statistical science* 22.4 (2007), pp. 581–597. ISSN: 0883-4237.

[23] S. Bittihn and A. Schadschneider. "Braess' paradox in the age of traffic information". eng. In: *Journal of statistical mechanics* 2021.3 (2021), pp. 33401–. ISSN: 1742-5468.

[24] Transportation Economics Committee Transportation Research Board. *Transportation Benefit-Cost Analysis*. 2022. URL: http://bca.transportationeconomics.org/.

[25] Geoff Boeing. "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks". In: *Computers, Environment and Urban Systems* 65 (2017), pp. 126–139.

[26] Bob Booth, Andy Mitchell, et al. *Getting started with ArcGIS*. 2001.

[27] Mark Brackstone and Mike McDonald. "Car-following: a historical review". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 2.4 (Dec. 1999), pp. 181–196. ISSN: 13698478. DOI: 10.1016/S1369-8478(00)00005-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S136984780000005X.

[28] Elmar Brockfeld, Reinhart D Kühne, and Peter Wagner. "Calibration and Validation of Microscopic Traffic Flow Models". In: *Transportation Research Record* 1876.1 (2004), pp. 62–70. DOI: 10.3141/1876-07. URL: https://doi.org/10.3141/1876-07.

[29] Theophile Cabannes et al. "Creating, Calibrating, and Validating Large-Scale Microscopic Traffic Simulation". In: *Transportation Research Board 102nd Annual Meeting* (2023).

[30] Theophile Cabannes et al. "Solving N-Player Dynamic Routing Games with Congestion: A Mean-Field Approach". In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 2022, pp. 1557–1559.

[31] Théophile Cabannes et al. "Measuring regret in routing: assessing the impact of increased app usage". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 2589–2594.

[32] Théophile Cabannes et al. "Regrets in routing networks: Measuring the impact of routing apps in traffic". In: *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 5.2 (2019), pp. 1–19.

[33] Théophile Cabannes et al. *The Impact of GPS-Enabled Shortest Path Routing on Mobility: A Game Theoretic Approach*. Tech. rep. 2018.

[34] California Department of Transportation (CalTrans). *PeMS User Guide*. 2020.

[35] Gordon DB Cameron and Gordon ID Duncan. "PARAMICS—Parallel microscopic simulation of road traffic". In: *The Journal of Supercomputing* 10.1 (1996), pp. 25–53.

[36] Emmanuel J Candès et al. "Robust principal component analysis?" In: *Journal of the ACM (JACM)* 58.3 (2011), pp. 1–37.

[37] Ennio Cascetta. *Transportation Systems Engineering: Theory and Methods*. Applied Optimization. Springer, 2013. ISBN: 0792367928.

[38] Ennio Cascetta et al. "A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks". In: *Transportation and Traffic Theory. Proceedings of The 13th International Symposium On Transportation And Traffic Theory, Lyon, France, 24-26 July 1996*. 1996.

[39] Cy Chan et al. "Quasi-Dynamic Traffic Assignment using High Performance Computing". In: *Open Access Publications from the University of California* (2021). URL: https://escholarship.org/uc/item/0f09r1x7.

[40] Qianwen Chao et al. "A Survey on Visual Traffic Simulation: Models, Evaluations, and Applications in Autonomous Driving". In: *Computer Graphics Forum* 39.1 (2020), pp. 287–308. DOI: https://doi.org/10.1111/cgf.13803. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13803.

[41] Chao Chen et al. "Freeway performance measurement system: mining loop detector data". In: *Transportation Research Record* 1748.1 (2001), pp. 96–102.

[42] Yi-Chang Chiu et al. "Dynamic traffic assignment: A primer (transportation research circular e-c153)". In: (2011).

[43] Fang-Chieh Chou, Alben Rome Bagabaldo, and Alexandre M. Bayen. "The Lord of the Ring Road: A Review and Evaluation of Autonomous Control Policies for Traffic in a Ring Road". In: *ACM transactions on cyber-physical systems* 6.1 (2022), pp. 1–25. ISSN: 2378-962X.

[44] Fang-Chieh Chou et al. "Modeling Vehicle-Following Dynamics of Heavy Trucks Under Automatic Speed Control Based on Experimental Data". In: *Transportation Research Board 97th Annual Meeting*. Washington DC, United States, 2018.

[45] Shumo Cui et al. "Stabilizing traffic flow via a single autonomous vehicle: Possibilities and limitations". In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1336–1341.

[46] Carlos F. Daganzo. "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory". In: *Transportation research. Part B: methodological* 28.4 (1994), pp. 269–287. ISSN: 0191-2615.

[47] L. C. Davis. "Effect of adaptive cruise control systems on traffic flow". In: *Phys. Rev. E* 69 (6 June 2004), p. 066110. DOI: 10.1103/PhysRevE.69.066110. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.066110.

[48] Kalyanmoy Deb and Himanshu Jain. "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints". In: *IEEE transactions on evolutionary computation* 18.4 (2013), pp. 577–601.

[49] Maria Laura Delle Monache et al. "Feedback control algorithms for the dissipation of traffic waves with autonomous vehicles". In: *Computational Intelligence and Optimization Methods for Control Engineering*. Springer, 2019, pp. 1–20. URL: https://hal.inria.fr/hal-01930724.

[50] N. R. Draper and Harry Smith. *Applied regression analysis*. Third Edition. John Wiley, 1998.

[51] Yan Duan et al. *Benchmarking Deep Reinforcement Learning for Continuous Control*. 2016. arXiv: 1604.06778 [cs.LG].

[52] Federal Highway Administration. *Home Page*. https://www.fhwa.dot.gov. Accessed: 2024-04-27. 2024.

[53] Martin Fellendorf and Peter Vortisch. "Microscopic traffic flow simulator VISSIM". In: *Fundamentals of traffic simulation*. Springer, 2010, pp. 63–93.

[54] Adriano Festa and Paola Goatin. "Modeling the impact of on-line navigation devices in traffic flows". In: *2019 IEEE 58th Conference on Decision and Control (CDC)* (2019), pp. 323–328.

[55] R. Firoozi et al. "Safe Adaptive Cruise Control with Road Grade Preview and **V**2**V** Communication". In: *2019 American Control Conference (ACC)*. 2019, pp. 4448–4453.

[56] Félix-Antoine Fortin et al. "DEAP: Evolutionary Algorithms Made Easy". In: *Journal of Machine Learning Research* 13 (July 2012), pp. 2171–2175.

[57] Marguerite Frank and Philip Wolfe. "An algorithm for quadratic programming". In: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110.

[58] Markus Friedrich and Manuel Galster. "Methods for generating connectors in transport planning models". In: *Transportation Research Record* 2132 (2009), pp. 133–142. ISSN: 03611981. DOI: 10.3141/2132-15.

[59] Emanuele Galli et al. "ActivitySim: large-scale agent-based activity generation for infrastructure simulation". In: *Proceedings of the 2009 spring simulation multiconference*. 2009, pp. 1–9.

[60] Fuchang Gao and Lixing Han. "Implementing the Nelder-Mead simplex algorithm with adaptive parameters". In: *Computational Optimization and Applications* 51 (May 2012), pp. 259–277.

[61] NJ Garber and LA Hoel. "Fundamental Principles of Traffic Flow". In: *Traffic and highway engineering*. 4th Ed. Cengage Learning, 2009, pp. 213–214.

[62] Jin I. Ge and Gábor Orosz. "Dynamics of connected vehicle systems with delayed acceleration feedback". In: *Transportation Research Part C: Emerging Technologies* 46 (2014), pp. 46–64. ISSN: 0968090X. DOI: 10.1016/j.trc.2014.04.014. URL: http://dx.doi.org/10.1016/j.trc.2014.04.014.

[63] Steven R Gehrke, Alison Felix, and Timothy Reardon. "Fare choices: A survey of ride-hailing passengers in metro Boston". In: *Metropolitan Area Planning Council* (Feb. 2018), p. 19. URL: http://www.mapc.org/wp-content/uploads/2018/02/Fare-Choices-MAPC.pdf.

[64] Google Maps. *Google Maps Eco-Friendly Routing: How it works*. 2021. URL: https://www.gstatic.com/gumdrop/sustainability/google-maps-eco-friendly-routing.pdf (visited on 07/09/2022).

[65] Jason B Gordon, Haris N Koutsopoulos, and Nigel HM Wilson. "Estimation of population origin–interchange–destination flows on multimodal transit networks". In: *Transportation Research Part C: Emerging Technologies* 90 (2018), pp. 350–365.

[66] Lawrence Goulder. "Environmental taxation and the double dividend: A reader's guide". In: *International Tax and Public Finance* 2.2 (1995), pp. 157–183. ISSN: 0927-5940.

[67] Bruce D. Greenshields. "A Study of Traffic Capacity. Highway Research Board". In: *Highway Research Board* 14 (1935), pp. 448–477.

[68] *Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling*. May 2020. URL: https://ops.fhwa.dot.gov/publications/fhwahop13015/sec2.htm.

[69] Tuomas Haarnoja et al. "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden, 2018. arXiv: 1801.01290. URL: http://arxiv.org/abs/1801.01290.

[70] Mordechai Haklay and Patrick Weber. "Openstreetmap: User-generated street maps". In: *IEEE Pervasive computing* 7.4 (2008), pp. 12–18.

[71] Nick Hanley, Jason F Shogren, and Benedict White. *Introduction to environmental economics*. Third edition. Oxford, United Kingdom: Oxford University Press, 2019. ISBN: 9780198737230.

[72] R'mani Haulcy et al. "A Fuzzy-based Approach to Dampen Emergent Traffic Waves". In: 2017. URL: https://www.csl.uawebhost.arizona.edu/sites/default/files/A_Fuzzy_based_approach_to_Dampen_Emergent_Traffic_Waves.pdf.

[73] Seham Hemdan, Elsayed Abulela, and Farag Khodary. *Mode Choice Effect on the Number of en-Route Travelers and Arrival Rates in an Urban Transport Network*. 2024. DOI: 10.21608/svusrc.2024.268541.1185.

[74] Robert Herman et al. "Traffic Dynamics: Analysis of Stability in Car Following". In: *Operations Research* 7.1 (1959), pp. 86–106. DOI: 10.1287/opre.7.1.86. eprint: https://doi.org/10.1287/opre.7.1.86. URL: https://doi.org/10.1287/opre.7.1.86.

[75] Yaron Hollander and Ronghui Liu. "The principles of calibrating traffic microsimulation models". In: *Transportation* 35.3 (2008), pp. 347–362. ISSN: 00494488. DOI: 10.1007/s11116-007-9156-2.

[76] Berthold K. P. Horn. "Suppressing traffic flow instabilities". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. Oct. 2013, pp. 13–20. DOI: 10.1109/ITSC.2013.6728204.

[77] Berthold K. P. Horn and Liang Wang. "Wave Equation of Suppressed Traffic Flow Instabilities". In: *IEEE Transactions on Intelligent Transportation Systems* 99 (2017), pp. 1–10.

[78] DE Huntington and CS Lyrintzis. "Improvements to and limitations of Latin hypercube sampling". In: *Probabilistic engineering mechanics* 13.4 (1998), pp. 245–253.

[79] Bruce N. Janson. "Dynamic traffic assignment for urban road networks". In: *Transportation Research Part B* 25.2-3 (1991), pp. 143–161. ISSN: 01912615. DOI: 10.1016/0191-2615(91)90020-J.

[80] Jeremy Walsh for Pleasanton Weekly. *City OKs partial street closure to curtail cut-through traffic*. 2018. URL: https://pleasantonweekly.com/news/2018/02/20/city-oks-partial-street-closure-to-curtail-cut-through-traffic (visited on 07/08/2022).

[81] John Cichowski for North Jersey. *How Leonia found smooth sailing with its last-ditch traffic fix to stop bridge commuters*. 2018. URL: https://www.northjersey.com/story/news/columnists/john-cichowski/2018/01/27/leonias-last-ditch-traffic-fix-stop-george-washington-bridge-commuters/1068170001/ (visited on 08/15/2019).

[82] Alireza Khani et al. *Integration of the FAST-TrIPs person-based dynamic transit assignment model, the SF-CHAMP regional, activity-based travel demand model, and san francisco's citywide dynamic traffic assignment model*. Tech. rep. 2013.

[83] E. Kometani and T. Sasaki. "Dynamic behavior of traffic with a nonlinear spacing-speed relationship". In: *Proceedings of the Symposium on Theory of Traffic Flow*. Research Laboratories, General Motors: New York: Elsevier, 1959, pp. 105–119.

[84] Abdul Rahman Kreidieh, Cathy Wu, and Alexandre M. Bayen. "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning". In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* 2018-November (2018), pp. 1475–1480. DOI: 10.1109/ITSC.2018.8569485.

[85] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[86] Keqiang Li, Jiawei Wang, and Yang Zheng. *Cooperative Formation of Autonomous Vehicles in Mixed Traffic Flow: Beyond Platooning*. 2020. arXiv: 2009.04254 [eess.SY].

[87] Chi-Ying Liang and Huei Peng. "Optimal adaptive cruise control with guaranteed string stability". In: *Vehicle System Dynamics* 32.4-5 (1999), pp. 313–330.

[88] Eric Liang et al. "RLlib: Abstractions for Distributed Reinforcement Learning". In: *International Conference on Machine Learning (ICML)*. 2018.

[89] Edward Lieberman and Ajay K. Rathi. "Traffic simulation". In: *Federal Highway Administration* (1992). ISSN: 0097-8418. DOI: 10.1145/274790.273160.

[90] Michael James Lighthill and Gerald Beresford Whitham. "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads". eng. In: *Proceedings of the Royal Society of London. Series A, Mathematical and physical sciences* 229.1178 (1955), pp. 317–345. ISSN: 1364-5021.

[91] Hao Liu et al. "Impact of cooperative adaptive cruise control on multilane freeway merge capacity". In: *Journal of Intelligent Transportation Systems* 22.3 (2018), pp. 263–275. DOI: 10.1080/15472450.2018.1438275. eprint: https://doi.org/10.1080/15472450.2018.1438275. URL: https://doi.org/10.1080/15472450.2018.1438275.

[92] Tim Lomax, David Schrank, Bill Eisele, et al. *2021 urban mobility report*. Tech. rep. Texas Transportation Institute, 2021.

[93] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. URL: https://elib.dlr.de/124092/.

[94] Hong Q Lu and Priya Nimbole. "Intro to TransCAD GIS". In: *Model Research and Development Unit Transportation Planning Branch* (2002).

[95] Silvia Magdici and Matthias Althoff. "Adaptive Cruise Control with Safety Guarantees for Autonomous Vehicles". In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 5774–5781. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.418. URL: http://www.sciencedirect.com/science/article/pii/S2405896317307681.

[96] Palak Maheshwary et al. "A methodology for calibration of traffic micro-simulator for urban heterogeneous traffic operations". In: *Journal of Traffic and Transportation Engineering (English Edition)* 7.4 (2020), pp. 507–519. ISSN: 20957564. DOI: 10.1016/j.jtte.2018.06.007. URL: https://doi.org/10.1016/j.jtte.2018.06.007.

[97] Matthieu Mastio et al. *Distributed Agent-Based Traffic Simulations*. 2018. DOI: 10.1109/mits.2017.2776162.

[98] Meghan McCarty. "The Road Less Traveled? Not Since Waze Came to Los Angeles". In: *National Public Radio* (2016).

[99] Millard McElwee, Bingyu Zhao, and Kenichi Soga. "Real-time Analysis of City Scale Transportation Networks in New Orleans Metropolitan Area using an Agent Based Model Approach". In: *MATEC Web of Conferences* 271.8 (Apr. 2019). Ed. by H. Sadek, p. 06007. ISSN: 2261-236X. DOI: 10.1051/matecconf/201927106007. URL: http://leader.pubs.asha.org/article.aspx?doi=10.1044/leader.PPL.21082016.20%20https://www.matec-conferences.org/10.1051/matecconf/201927106007.

[100] Michael G McNally. "The four-step model". In: *Handbook of transport modelling*. Emerald Group Publishing Limited, 2007.

[101] Wannes Meert and Mathias Verbeke. "HMM with Non-Emitting States for Map Matching". In: *European Conference on Data Analysis (ECDA)*. Paderborn, Germany, 2018.

[102] Aditya Krishna Menon et al. "Fine-grained OD estimation with automated zoning and sparsity regularisation". In: *Transportation Research Part B: Methodological* 80 (2015), pp. 150–172.

[103] Vicente Milanés and Steven E. Shladover. "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data". In: *Transportation Research Part C: Emerging Technologies* 48 (2014), pp. 285–300.

[104] Seyedali Mirjalili. "Genetic algorithm". In: *Evolutionary algorithms and neural networks*. Springer, 2019, pp. 43–55.

[105] Kai Nagel and Michael Schreckenberg. "A cellular automaton model for freeway traffic". In: *Journal de physique I* 2.12 (1992), pp. 2221–2229.

[106] John F. Nash. "Equilibrium Points in n-Person Games". eng. In: *Proceedings of the National Academy of Sciences - PNAS* 36.1 (1950), pp. 48–49. ISSN: 0027-8424.

[107] National Academies of Sciences, Engineering, and Medicine and others. "Travel demand forecasting: Parameters and techniques". In: (2012).

[108] J. A. Nelder and R. Mead. "A Simplex Method for Function Minimization". In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. ISSN: 0010-4620.

[109] G. F. Newell. "A simplified car-following theory: A lower order model". In: *Transportation Research Part B: Methodological* 36.3 (2002), pp. 195–205. ISSN: 01912615. DOI: 10.1016/S0191-2615(00)00044-8.

[110] Paul Newson and John Krumm. "Hidden Markov map matching through noise and sparseness". In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 336–343. ISBN: 9781605586496.

[111] Irena Ištoka Otković, Aleksandra Deluka-Tibljaš, and Sanja Šurdonja. "Validation of the calibration methodology of the micro-simulation traffic model". In: *Transportation Research Procedia* 45.2019 (2020), pp. 684–691. ISSN: 23521465. DOI: 10.1016/j.trpro.2020.02.110. URL: https://doi.org/10.1016/j.trpro.2020.02.110.

[112] Ian W. H. Parry, Margaret Walls, and Winston Harrington. "Automobile Externalities and Policies". In: *Journal of Economic Literature* 45.2 (2007), pp. 373–399. DOI: 10.1257/jel.45.2.373.

[113] Michael. Patriksson. *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015. ISBN: 0486787907.

[114] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[115] Matti Persula. "Simulation of Traffic Systems - An Overview". In: *Journal of Geographic Information and Decision Analysis* 3.1 (1999), pp. 1–8. ISSN: 1435-5949.

[116] Arthur Cecil Pigou. *The Economics of Welfare, Fourth Edition*. St. Martin's St., London: Macmillan and Co., Limited, 1932.

[117] Abdul Pinjari et al. "Cemdap: Modeling and microsimulation frameworks, software development, and verification". In: *Proceedings of the transportation research board 87th annual meeting, Washington DC*. 2008.

[118] L. A. Pipes. In: *J.App.Phys.* 41 (1953), p. 274.

[119] Stef Proost and Kurt Van Dender. "Optimal urban transport pricing in the presence of congestion, economies of density and costly public funds". In: *Transportation research. Part A, Policy and practice* 42.9 (2008), pp. 1220–1230. DOI: 10.1016/j.tra.2008.03.009.

[120] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[121] Hani Ramezani et al. "Micro-Simulation of Truck Platooning with Cooperative Adaptive Cruise Control: Model Development and a Case Study". In: *Transportation Research Record* 2672.19 (2018), pp. 55–65. DOI: 10.1177/0361198118793257. eprint: https://doi.org/10.1177/0361198118793257. URL: https://doi.org/10.1177/0361198118793257.

[122] LA Rastrigin. "Random search as a method for optimization and adaptation". In: *Stochastic Optimization*. Springer, 1986, pp. 534–544.

[123] A. Reuschel. "Fahrzeugbewegungen in der Kolonne". In: *Österreichisches Ingenieur-Archiv* 4.3-4 (1950), pp. 193–215.

[124] Paul I. Richards. "Shock Waves on the Highway". In: *Operations Research* 4.1 (1956), pp. 42–51. ISSN: 0030364X, 15265463. (Visited on 11/15/2023).

[125] R. W. Rosenthal. "The network equilibrium problem in integers". eng. In: *Networks* 3.1 (1973), pp. 53–59. ISSN: 0028-3045.

[126] Tim Roughgarden. *Selfish routing and the price of anarchy*. Cambridge, Massachusetts: MIT Press, 2005. ISBN: 0262182432.

[127] Timothy Avelin Roughgarden. "Selfish Routing". PhD thesis. USA: Cornell University, 2002. ISBN: 0493660690.

[128] Meead Saberi et al. "A simple contagion process describes spreading of traffic jams in urban networks". eng. In: *Nature Communications* 11.1 (2020), pp. 1616–1616. ISSN: 2041-1723.

[129] San Francisco County Transportation Authority and Cambridge Systematics Inc. "San Francisco Travel Demand Forecasting Model Development: Executive Summary". In: 29 (2002).

[130] John Schulman et al. "Proximal Policy Optimization Algorithms". In: *arXiv* (2017), pp. 1–12. arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[131] Sajjad Shafiei, Ziyuan Gu, and Meead Saberi. "Calibration and validation of a simulation-based dynamic traffic assignment model for a large-scale congested network". In: *Simulation Modelling Practice and Theory* 86 (Aug. 2018), pp. 169–186. ISSN: 1569190X. DOI: 10.1016/j.simpat.2018.04.006.

[132] Susan Shaheen and Adam Cohen. "Shared Mobility: An Overview of Definitions, Current Practices, and Its Relationship to Mobility on Demand and Mobility as a Service". In: *International Encyclopedia of Transportation*. Ed. by Roger Vickerman. Oxford: Elsevier, 2021, pp. 155–159. ISBN: 978-0-08-102672-4. DOI: https://doi.org/10.1016/B978-0-08-102671-7.10420-8.

[133] Mohammad Shaqfeh, Salah Hessien, and Erchin Serpedin. "Utility of Traffic Information in Dynamic Routing: Is Sharing Information Always Useful?" In: *2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS)* (2020), pp. 1–6.

[134] Fei Shi. *Research on Accessibility and Equity of Urban Transport Based on Multisource Big Data*. 2021. DOI: 10.1155/2021/1103331.

[135] Rongye Shi, Zhaobin Mo, and Xuan Di. "Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 540–547.

[136] Kenneth A. Small and Erik T. Verhoef. *The Economics of Urban Transportation*. eng. 2nd updated Edition. Oxford: Routledge, 2007. ISBN: 9780415285155.

[137] Raphael E. Stern et al. "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments". In: *Transportation Research Part C: Emerging Technologies* 89.February (2018), pp. 205–221. ISSN: 0968090X. DOI: 10.1016/j.trc.2018.02.005. arXiv: 1705.01693. URL: https://doi.org/10.1016/j.trc.2018.02.005.

[138] Thomas Sterner. *Policy instruments for environmental and natural resource management*. Washington, DC: Resources for the Future, 2003. ISBN: 1891853139.

[139] Yijie Su, Hadi Ghaderi, and Hussein Dia. "The role of traffic simulation in shaping effective and sustainable innovative urban delivery interventions". In: *EURO Journal on Transportation and Logistics* 13 (2024), p. 100130. ISSN: 2192-4376. DOI: https://doi.org/10.1016/j.ejtl.2024.100130.

[140] Yuki Sugiyama et al. "Traffic jams without bottlenecks experimental evidence for the physical mechanism of the formation of a jam". In: *New journal of physics* 10.3 (2008), p. 033001.

[141] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.

[142] Shin Ichi Tadaki et al. "Phase transition in traffic jam experiment on a circuit". In: *New Journal of Physics* 15 (2013). ISSN: 13672630. DOI: 10.1088/1367-2630/15/10/103034.

[143] Mostafa H. Tawfeek et al. "Calibration and validation of micro-simulation models using measurable variables". In: *12th International Transportation Specialty Conference 2018, Held as Part of the Canadian Society for Civil Engineering Annual Conference 2018* 6 (2019), pp. 12–22.

[144] Jerome Thai, Nicolas Laurent-Brouty, and Alexandre M. Bayen. "Negative externalities of GPS-enabled routing applications: A game theoretical approach". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (2016), pp. 595–601.

[145] The Fremont Mobility Task Force. *Fremont Mobility Action Plan*. 2019. URL: https://www.fremont.gov/DocumentCenter/View/40583/FREMONT-Mobility-Action-Plan-Final-3-1-19?bidId=.

[146] J.R.R. Tolkien. *The Lord of the Rings*. New York: Ballantine Books, 1954.

[147] TransitFeeds. *GTFS Data for the City of Dallas*. https://transitfeeds.com/p/dart/26. Accessed: 2023-11-21. 2023.

[148] Transport Simulation Software. *Aimsun Next 22*. 2022.

[149] U.S. Department of Transportation Federal Highway Administration. *Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software 2019 Update to the 2004 Version*. Apr. 2019. URL: https://ops.fhwa.dot.gov/publications/fhwahop18036/index.htm (visited on 07/14/2022).

[150] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. "Congested traffic states in empirical observations and microscopic simulations". In: *Phys. Rev. E* 62 (2 Aug. 2000), pp. 1805–1824. DOI: 10.1103/PhysRevE.62.1805. URL: https://link.aps.org/doi/10.1103/PhysRevE.62.1805.

[151] Martin Treiber and Arne Kesting. "Traffic Flow Dynamics". In: (2013). DOI: 10.1007/978-3-642-32460-4. URL: http://link.springer.com/10.1007/978-3-642-32460-4.

[152] Theodore Tsekeris and Antony Stathopoulos. "Measuring Variability in Urban Traffic Flow by Use of Principal Component Analysis". eng. In: *Journal of transportation and statistics* 9.1 (2006), pp. 49–62. ISSN: 1094-8848.

[153] Valerio Turri et al. "A model predictive controller for non-cooperative eco-platooning". In: *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 2309–2314.

[154] U.S. Department of Transportation Federal Highway Administration. *Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling*. May 2020. URL: https://ops.fhwa.dot.gov/publications/fhwahop13015/sec2.htm.

[155] William S. Vickrey. "Congestion Theory and Transport Investment". In: *The American economic review* 59.2 (1969), pp. 251–260.

[156] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[157] Paul Waddell. "UrbanSim: Modeling urban development for land use, transportation, and environmental planning". In: *Journal of the American planning association* 68.3 (2002), pp. 297–314.

[158] J. Wang et al. "Controllability Analysis and Optimal Control of Mixed Traffic Flow With Human-Driven and Autonomous Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–15. DOI: 10.1109/TITS.2020.3002965.

[159] Jiawei Wang et al. "Leading cruise control in mixed traffic flow". In: *arXiv* (2020). arXiv: 2007.11753.

[160] L. Wang and B. K. P. Horn. "On the Chain Stability of Bilateral Control Model". In: *IEEE Transactions on Automatic Control* 65.8 (2020), pp. 3397–3408. DOI: 10.1109/TAC.2019.2945877.

[161] John Glen. Wardrop. *Some theoretical aspects of road traffic research*. eng. Road paper ; no. 36. London: [Institution of Civil Engineers], 1952.

[162] Michael Wegener. "Overview of Land Use Transport Models". In: Aug. 2004, pp. 127–146. DOI: 10.1108/9781615832538-009. URL: http://www.emeraldinsight.com/doi/10.1108/9781615832538-009.

[163] Willem van Willigen et al. "Safety in the Face of Uncertainty: Critical Headway Estimation in Cooperative Adaptive Cruise Control". In: *International Journal of Intelligent Transportation Systems Research* 13 (2015), pp. 95–106.

[164] C. Wu, A. M. Bayen, and A. Mehta. "Stabilizing Traffic with Autonomous Vehicles". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. May 2018, pp. 6012–6018. DOI: 10.1109/ICRA.2018.8460567.

[165] C. Wu et al. "Framework for control and deep reinforcement learning in traffic". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2017, pp. 1–8. DOI: 10.1109/ITSC.2017.8317694.

[166] Cathy Wu et al. "Flow: Architecture and benchmarking for reinforcement learning in traffic control". In: *arXiv preprint arXiv:1710.05465* 10 (2017).

[167] Fangyu Wu et al. "Tracking vehicle trajectories and fuel rates in phantom traffic jams: Methodology and data". In: *Transportation Research Part C: Emerging Technologies* 99.July 2017 (2019), pp. 82–109. ISSN: 0968090X. DOI: 10.1016/j.trc.2018.12.012. URL: https://doi.org/10.1016/j.trc.2018.12.012.

[168] Jianjun Wu, Ziyou Gao, and Huijun Sun. "Simulation of traffic congestion with SIR model". eng. In: *Modern physics letters. B, Condensed matter physics, statistical physics, applied physics* 18.30 (2004), pp. 1537–1542. ISSN: 0217-9849.

[169] D. Xie, X. Zhao, and Z. He. "Heterogeneous Traffic Mixing Regular and Connected Vehicles: Modeling and Stabilization". In: *IEEE Transactions on Intelligent Transportation Systems* 20.6 (2019), pp. 2060–2071. DOI: 10.1109/TITS.2018.2857465.

[170] Yanyan Xu, Riccardo Di Clemente, and Marta C González. "Understanding vehicular routing behavior with location-based service data". In: *EPJ Data Science* 10.1 (2021), pp. 1–17.

[171] Yun Xu and Royston Goodacre. "On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning". In: *Journal of analysis and testing* 2.3 (2018), pp. 249–262.

[172] Diana Yanakiev and Ioannis Kanellakopoulos. "Speed tracking and vehicle follower control design for heavy-duty vehicles". In: *Vehicle System Dynamics* 25.4 (1996), pp. 251–276.

[173] Weichuan Yin and Yingqun Zhang. "Identification Method for Optimal Urban Bus Corridor Location". In: *Sustainability* 12.17 (2020). ISSN: 2071-1050.

[174] Sunhyung Yoo and Jinwoo (Brian) Lee. "Revising bus routes to improve access for the transport disadvantaged: A reinforcement learning approach". In: *Journal of Public Transportation* 25 (2023), p. 100041. ISSN: 1077-291X.

[175] Hyejin Youn, Michael T. Gastner, and Hawoong Jeong. "Price of Anarchy in Transportation Networks: Efficiency and Optimality Control". In: *Phys. Rev. Lett.* 101 (12 Sept. 2008), p. 128701.

[176] Hong Zheng et al. "A primer for agent-based simulation and modeling in transportation applications". In: (2013).

[177] Yang Zheng, Jiawei Wang, and Keqiang Li. "Smoothing traffic flow via control of autonomous vehicles". In: (2018). arXiv: arXiv:1812.09544v1.