UNIVERSITY OF CALIFORNIA,
IRVINE

Integrated Corridor Management for Connected Vehicles and Park and Ride Structures

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical and Computer Engineering

by

Tyler Zhang

Thesis Committee:
Professor Mohammad Al Faruque, Chair
Assistant Professor Salma Elmalaki
Assistant Professor Yasser Shoukry

2023

DEDICATION

In loving memory of my grandparents and the others who lost their lives during the pandemic.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This study was made possible with funding received by the University of California Institute of Transportation Studies from the State of California through the Road Repair and Accountability Act of 2017 (Senate Bill 1). I would like to thank the State of California for its support of university-based research, and especially for the funding received for this project.

In addition, I would like to thank Professor Mohammad Al Faruque for his insight and support throughout this project.

I want to thank my colleagues at the AICPS lab for providing such a wonderful research environment to learn from. I am especially grateful to Mohanad Odema and Mo Fakih for their guidance and assistance with this project.

I also want to thank my committee, Assistant Professor Salma Elmalaki and Assistant Professor Yasser Shoukry, for taking the time to review my work.

ABSTRACT OF THE THESIS

Integrated Corridor Management for Connected Vehicles and Park and Ride Structures

By

Tyler Zhang

Master of Science in Electrical and Computer Engineering

University of California, Irvine, 2023

Professor Mohammad Al Faruque, Chair

The forthcoming Connected Vehicles (CV) technology promises to substantially aid in managing traffic congestion and improving users' mobility along transportation corridors. Through CV technology, a global estimate of the corridor's traffic flow state can be obtained through analyzing data from its constituent components, which can enable globally-optimized traffic management strategies that efficiently utilize existing transportation infrastructure resources. In this paper, we propose a novel integrated corridor management (ICM) methodology that incorporates the underutilized infrastructure of park and ride facilities into its global optimization strategy. Firstly, we discuss how vehicle-to-infrastructure (V2I) communication protocols like basic safety messages (BSM) and traveler information messages (TIM) can be tailored to collect the state of downstream traffic and advertise park and ride advisories to upstream traffic respectively. Then, we model the system in terms of potential delays that can be experienced by vehicles traversing the corridor, and accordingly, we implement a novel centralized deep reinforcement learning (DRL) solution to control how and when such messages are advertised, with the aim of maximizing throughput and minimizing carbon emissions and travel time. We simulate our ICM strategy on a realistic model of Interstate 5 using the Veins simulation software, and the DRL agent converges to a strategy that provides marginal improvement to throughput, freeway travel time, and carbon emissions but at the cost of added travel delay for those using park and ride services.

Chapter 1

Introduction

Considerable advancements have been made in traffic management strategies over the past few decades to enhance user mobility along freeways. Still, the continuous growth of metropolitan regions and the increasing mobility needs tend to impede such progress, giving rise to new congestion bottlenecks and curbing traffic flow along freeways. Pre-pandemic, the recent Urban Mobility Report [2] showed that the total cost of traffic delay in the top urban areas in the US has grown by almost 48% over the past decade. In many of these regions (such as in California), freeways experience a great deal of traffic congestion [3], arising at bottlenecks at which high-volume, free-flowing traffic transforms into tightly-packed clusters of low-speed vehicles.

As such, recent development efforts have given special attention to the notion of *integrated corridor management (ICM)* [4], which encourages the adoption of global traffic management strategies. In practice, this is possible by consolidating the various traffic components deployed along the corridor (e.g., ramp meter controllers) into a single interconnected system with a global view of the traffic state along the entire corridor, allowing upstream traffic components to be tuned to relieve downstream bottlenecks and congestion. In other words,

an ICM strategy can coordinate various traffic control units to meet the system-wide optimization objectives the entire freeway, rather than operate in isolation based on pre-specified settings or in a localized adaptive fashion [5].

Moreover, the emerging connected vehicles (CV) technology is expected to substantially benefit ICM enterprises, where through the support of Dedicated Short Range Communications (DSRC), vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) standards [6], vehicles could communicate with each other and the surrounding transportation infrastructure. Projected benefits from incorporating such connectivity are manifold. For instance, Road Side Units (RSUs) could transmit traveler information messages (TIM) to inform commuters on the state of traffic [7], heavy occupancy vehicles (HOV) could transmit signal request messages (SRM) [7] to request priority at key corridor components, and vehicles could communicate with nearby vehicles and RSUs for travel planning and safety features such as platooning and adaptive cruise control [6]. In fact, the potential of CVs has been recognized by the National Highway Traffic Safety Administration (NHTSA), who recently proposed a new Federal Motor Vehicle Safety Standard mandating the installation of DSRC-based communication equipment for new vehicles [8, 9].

The combined potential of ICM strategies and CVs could be even greater when underutilized resources are brought into the global optimization strategy through the power of wireless connections. Amongst these underutilized resources are the park and ride (PAR) facilities developed to enable cheap, accessible public transport from specialized parking lots to service commuters and enhance traffic flow by reducing the number of vehicles on the road [10]. Recent reports have shown that most of the 302 park and ride facilities in California do not come close to their maximum capacity, even at peak hours; they are only 65% full on average and stakeholder agencies lose money on parking services [11, 12].

From here, our focus in this paper is to analyze and demonstrate how the effective accommodation of PAR supply and demand dynamics within ICM optimization strategies can

benefit the overall traffic flow along the corridor. At the heart of our methodology is a centralized learning-based approach that leverages traffic information from the corridor and the parking availability from relevant PAR facilities, and uses them to devise a corridor-wide advertisement strategy in which PAR advisor messages are broadcast from the various RSUs deployed along the corridor. Through this integration, we envision increased usage with PAR structures, enabling users to become more acquainted with PAR services and ultimately enhancing PAR efficiency, corridor flow, and commuters' experiences. Our key contributions can be summarized as follows:

- A realistic traffic model for a congested highway corridor is developed in chapter 3.
- Proof of concept hardware tests demonstrating useful V2I data propagation paths for our ICM strategy are developed in chapter 4.
- A corridor-level simulation implementing our centralized ICM strategy. The approach is developed in chapter 5 and chapter 6, and the remaining chapters describe the simulation and evaluation of its performance.

Chapter 2

Background

2.1 DSRC Technology

Dedicated short range communications (DSRC) is a wireless standard proposed by the Federal Communication Commission that reserves 75 MHz of bandwidth in the 5.9 GHz frequency band for vehicle-to-everything (V2X) communications [13]. Typical DSRC architecture adopts the IEEE 802.11p protocol to implement its PHY and MAC layers, while the upper layers of the network stack are implemented by IEEE 1609.1, 1609.2, 1609.3, and 1609.4 [13]. Finally, the SAE J2735 and SAE J2945.1 protocols define messages for V2X and V2V scenarios respectively [7, 14]. The DSRC network layers and their corresponding protocols are summarized in table 2.1.

Typical DSRC systems consist of on board units (OBU) and roadside units (RSU) [13]. Our work focuses on leveraging V2I communications between OBUs and RSUs for our ICM strategy.

Network Layer	Protocol
Application	IEEE 1609.1
Messaging Sublayer	SAE J2735 (V2X), SAE J2945.1 (V2V)
Security	IEEE 1609.2
Network and Transport	IEEE 1609.3
Upper MAC	IEEE 1609.4
Lower MAC	IEEE 802.11p
Physical	IEEE 802.11p

Table 2.1: Network layers and their corresponding protocols

2.2 VEINS Simulator

To simulate V2I communications, we used an open source vehicular network simulation framework called Veins (Vehicles in Network Simulation), which is built atop two popular simulators: OMNeT++, an event-based network simulator, and SUMO, a microscopic traffic simulator [15]. SUMO simulates the road geometry, routes, vehicle flows, and driving behaviors that can all be controlled using the Traffic Control Interface (TraCI) [16]. OMNeT++ simulations are built from modules that communicate by exchanging messages; users can define network topologies and model communication protocols at multiple levels of granularity [17].

Veins is an OMNeT++ project that defines a dynamic network topology from moving SUMO vehicles, models the DSRC communication stack in OMNeT++, and provides an API to control and read values from the underlying SUMO traffic simulation via TraCI [15]. Any new simulation module can utilize these functionalities, which provides a high degree of flexibility when developing V2X simulations. We chose the Veins framework primarily because it is open-source and for its flexibility.

2.3 Related Work

2.3.1 Connected Vehicle Platoons

One common line of study in CV research is to use V2V coordination to organize traveling vehicles into groups or platoons [18, 19], but these tend to focus on local algorithms and lack system-wide awareness. One such study explored the benefits of utilizing three mobility improvements called cooperative adaptive cruise control, speed harmonization, and queue warning in a highway scenario [19]. The study concluded that individual vehicle safety was improved but at the cost of sacrificing overall highway throughput.

2.3.2 Ramp Metering

A popular system-wide ICM approach is to adjust ramp metering rates to control the flow of vehicles entering the highway [20, 21, 22, 23]. Several ramp metering algorithms such as ALINEA and SWARM have existed and have been in use since the early 2000s [20]. SWARM in particular can be found in Orange County; it calculates one ramp metering rate based on local density and another based on a predetermined global volume reduction, picking the most restrictive metering rate from the two [20].

Many advancements in ramp-metering approaches have been proposed since then. In 2014, Fares and Gomaa implemented a deep reinforcement algorithm to adjust ramp metering timings based on vehicle density that increased highway flow and reduced total travel time [21]. In another study in 2016, Hashemi and Abdelghany proposed a system that dynamically searches for an optimal traffic control scheme involving intersection timing plans, diversion messages, ramp metering, and/or dynamic shoulder lanes [22]. They demonstrated improvements to total travel time when more control options are available. In a follow-up work,

Hashemi and Abdelghany trained a deep convolutional neural network to recognize the highway state and pick an optimal traffic control scheme [23]. These strategies do not explicitly mention CV technology, but the state vectors from Fares and Goma [21] and Hashemi and Abdelghany [23] could potentially be obtained through RSUs sampling data from nearby CVs.

2.3.3 Dynamic Rerouting

Ramp metering seems to be a well-explored approach, so the scope of our work focuses on improving corridor traffic using dynamic rerouting to underutilized PAR structures. Liu et al. [24] set out to establish a framework for evaluating ICM methods and conducted a study on diverting upcoming traffic to side roads using variable message signs on the freeway, using throughput and travel time as evaluation criteria. They demonstrate that diverting traffic reduces travel delay for freeway vehicles but degrades side road performance, so careful evaluation of trade-offs is required. Our ICM approach is similar to this study, so it makes sense to use highway throughput and vehicle travel time as our evaluation criteria as well.

A study conducted by Ortega et al. [25] demonstrated via simulation that use of a park and ride is expected to increase the total trip time compared to not using the resource. This indicates that we need to include evaluation criteria other than travel time in our evaluation metrics, such as emissions or highway throughput.

2.3.4 Deep Reinforcement Learning in ICM Strategies

The studies conducted by Fares and Goma [21] and Hashemi and Abdelghany [23] utilize deep reinforcement learning to estimate the state of the network with relative success. In addition, Liu et al. [26] established a Markov decision process (MDP) framework for a

freeway scenario. These examples serve as a foundation for establishing a MDP for our scenario and using deep learning to optimize the ICM strategy. However, our deep learning agent will differ from these studies in its action space of diverting traffic to PAR systems.

Chapter 3

Scenario

3.1 Interstate 5

We conducted our study on a section of Interstate 5 in LA County, a freeway that consistently ranks among the most congested corridors in California [27] and even the United States [28, 29]. The corridor of interest is the northbound 10 mile stretch between Interstate 605 and California State Road 60. We used SUMO's provided OSMWebWizard tool [16] to convert an OpenStreetMap rendering of I-5 into a SUMO road network. The resulting highway network is shown in figure 3.1. For this study, we considered vehicles travelling northbound from the source I-605 to the sink SR 60.

3.1.1 Realistic Traffic Modeling

This corridor includes junctions to local roads and a couple of busier freeways, which provides a diverse set of on ramps and exits in terms of traffic volume. Ramp volumes and average daily traffic data was collected from the Caltrans Traffic Census Program [30] to generate

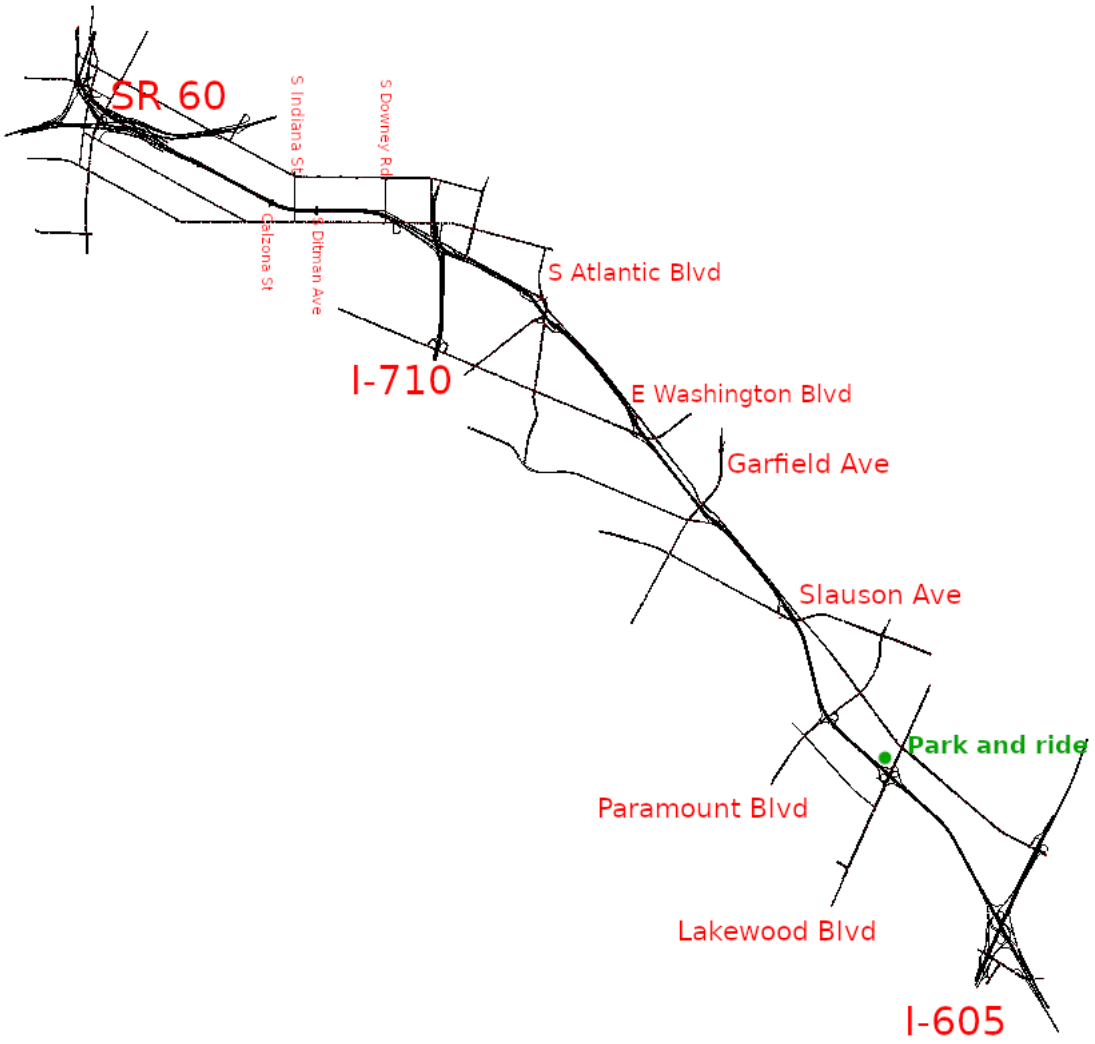


Figure 3.1: Interstate 5 model

realistic traffic flows. Average daily traffic volume was taken from the 2015 survey due to the completeness of the data for the junctions in this corridor.

To model the on ramp traffic, we spawned vehicles at each on ramp as a Poisson process. Since the ramp volumes are recorded as daily averages [30], we converted them into a Poisson rate with units $\frac{vehicles}{second}$ according to equation 3.1.

$$\lambda_i = \frac{V_i \frac{veh}{day}}{24 * 60 * 60 \frac{sec}{day}} = \frac{V_i}{86400} veh/sec \tag{3.1}$$

where λ_i is the Poisson rate for on ramp i and V_i is its average daily volume.

To model the exit behavior, each vehicle is assigned a destination following a fixed probability distribution when it spawns (referred to as a route distribution [16]). We computed the probability of taking an exit according to equation 3.2.

$$\mathbb{P}_i = \frac{L_i}{\sum_{n=1}^{i} V_n} \tag{3.2}$$

where $\mathbb{P}_i$ is the probability of taking exit i, L_i is the average daily volume of exit i, and V_n is the average daily volumes for on ramps before exit i. Each on ramp has a unique route distribution table populated with $\mathbb{P}_i$ for the exits that are accessible downstream. If none of the exits are taken, the vehicle will route to the sink. For example, a vehicle at the on ramp of S Downey Rd (see figure 3.1) will have:

1. Chance to exit at Ditman Ave = $\mathbb{P}_{DitmanAve}$
2. Chance to exit at Calzona St = $\mathbb{P}_{CalzonaSt}$
3. Chance to exit at the sink = $1.0 - (\mathbb{P}_{DitmanAve} + \mathbb{P}_{CalzonaSt}) = \mathbb{P}_{sink}$

NOTE: Indiana St does not have an exit, so it is omitted.

3.1.2 Park and Ride

Along this corridor is the North Lakewood Park and Ride structure just off of Lakewood Blvd. For this study, we assumed that there exists a bus schedule that takes a passenger to the sink of the highway at fixed intervals. Thus, buses spawn at a fixed interval throughout the simulation. These buses do not get rerouted to any other exits and proceed directly to the sink.

3.2 Preliminary Exploration

We observe situations in which diverting traffic to the North Lakewood Park and Ride structure yields improvements to mainline flow, CO2 emissions, and overall speed of traffic at the cost of total travel time for the exiting vehicles. To begin, using the traffic scaling parameter provided by SUMO, we scale the spawn rates of the vehicles until congestion becomes apparent, which occurs when the spawn rates are doubled. When this is visualized, we observe that the congestion is heaviest at the Interstate 605 and Interstate 710 junctions due to the merges.

In this scenario, we divert vehicles to the park and ride using three different exits at a fixed probability that we call the compliance parameter. If the vehicle is compliant with being rerouted, it has an equal chance to take any of the three exits. These three exits are I610, Lakewood Boulevard, and Paramount Boulevard and for this scenario we set the park and ride to be just off of the Lakewood exit as shown in figure 3.1.

After running the simulation for a fixed amount of time, we evaluate the average travel time, average carbon emissions, and number of vehicles that reach the sink for the following vehicles:

- cars that intend to travel from source to sink without exiting
- cars that intend to travel from source to sink that are diverted
- buses leaving park and ride structure that drive to sink

Since diverted vehicles are removed from the simulation when they exit, we estimate the cost of the detour using Google Maps data and the average bus travel time and add these values to the vehicle's simulation travel time. If their estimated arrival time is still within the simulation time, we count these vehicles toward the overall throughput. We obtain emissions

values for each vehicle using SUMO's default HBEFA3 emissions model [16]. Finally, to evaluate the overall congestion of the scenario, we record the average speed of all vehicles in the simulation.

We run the SUMO simulation for one hour of simulated time using compliance parameters of 0%, 10%, 25%, and 50%. Table 3.1 compiles the results of these experiments. We

Table 3.1: Fixed compliance experimental results

Compliance	Travel Time (s)	CO2 (kg)	Speed ($\frac{m}{s}$)	Count ($\frac{veh}{hr}$)
0%	1063.18	5.33	15.77	1793
10%	1040.60	4.96	16.15	1899
25%	1034.60	4.37	16.53	1992
50%	969.07	3.51	17.48	2188

can see that increasing the rerouting compliance yields improvements in all categories in terms of average travel time, average CO2 emissions, overall simulation vehicle speed, and throughput. One limitation of this preliminary study is that it cannot distinguish between buses with passengers and buses without passenger, which may inflate the overall throughput and emissions values. Additionally, we are always diverting vehicles in this scenario when in practice we would prefer to divert vehicles in response to downstream congestion.

These results suggest that for this congestion scenario, diverting traffic to a PAR structure may yield benefits. A well-designed reinforcement learning agent with more fine-grained control may be able to find more optimizations.

Chapter 4

V2I Communication Proof of Concept

We created a hardware test bed inspired by [1] to demonstrate the V2I communication flow between a smart parking structure, cloud server, RSU, and an OBU for the ICM strategy. To perform wireless DSRC communications, we used the USRP B210 board from Ettus Research, which is a software-defined radio that can receive and transmit data at any frequency between 70 MHz - 6GHz [31]. A diagram of the setup is shown in figure 4.1.

A Raspberry Pi 3 representing a smart parking structure is connected via Ethernet cable to the desktop, which is running a Python script representing a cloud server. These two communicate over TCP. Additionally, in a separate process on the desktop, another Python script is running and represents an RSU. The RSU script communicates to the USRP B210, which will broadcast or listen on the DSRC 5.9GHz band. A picture of this half of the setup is shown in figure 4.2a. The other half of the setup is a laptop connected to another USRP B210, which can also broadcast or listen on the DSRC 5.9GHz band. A picture of this half of the setup is shown in figure 4.2b.

To interface with the USRP B210, we used a WiFi tranceiver module created by the open source Wime Project [32], which provides a complete physical layer implementation of

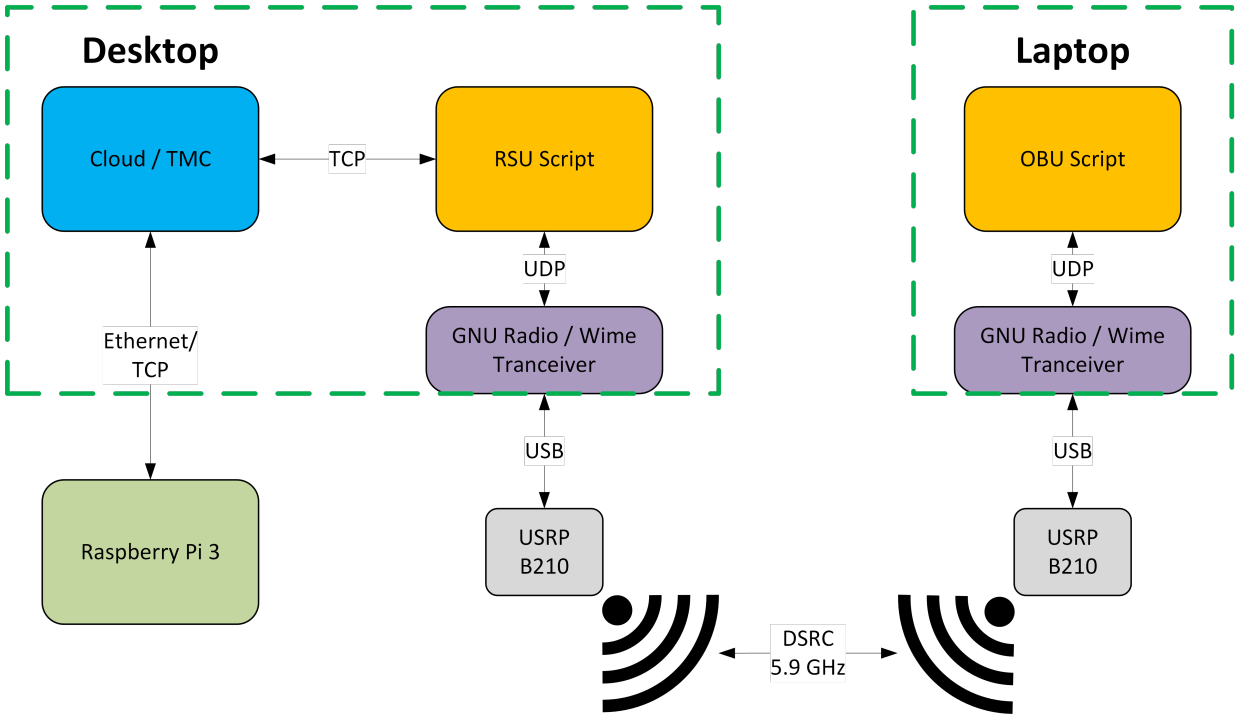
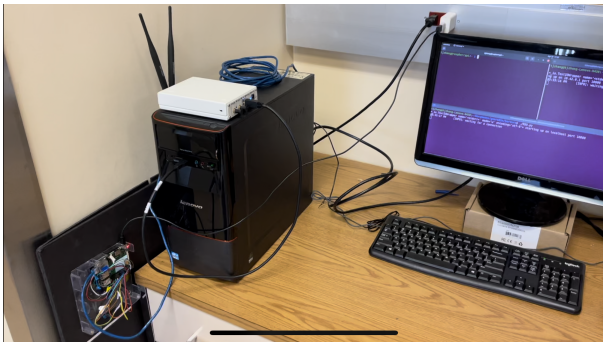


Figure 4.1: Block diagram of the hardware test bed

802.11p in GNU Radio. This was modified to send and receive UDP packets from the local machine as shown in figure 4.3. To send information to the USRP B210 for wireless transmission, the Python script simply writes to a UDP socket at *localhost:52001*. To read information received from the USRP B210, the Python script simply reads from a UDP socket at *localhost:52002*.

4.1 DSRC Tests

The first test is a forward propagation of data from the Raspberry Pi to the laptop which represents broadcasting park and ride information to a connected vehicle. The Raspberry Pi sends a dictionary of parking space info to the cloud server, which forwards the data to the RSU. The RSU wirelessly broadcasts the message to the awaiting laptop using DSRC. Screenshots demonstrating this data propagation are shown in figures 4.4a and 4.4b.



(a) A RPI 3 connected to the desktop via ethernet, and the B210 plugged into the desktop via USB

(b) The B210 plugged into the laptop via USB

Figure 4.2: Hardware setup for testing DSRC communications

The second test is a backward propagation of data from the laptop to the cloud server which represents collecting state information from a connected vehicle. The laptop broadcasts some basic vehicle information to the awaiting RSU using DSRC. The RSU then forwards the message to the cloud server. Screenshots demonstrating this data propagation are shown in figures 4.5a and 4.5b.

4.2 Practical Implementation

This hardware proof of concept clarifies the ICM mechanisms needed to read traffic state and alert drivers to reroute to a PAR structure. In these tests, we send data directly from one script to another, but in practice, standard DSRC message types are needed in both cases. The SAE J2735 standard [7] defines many message types for V2X communications; none are designed specifically for our use case, but a couple message types are flexible enough to be adapted. For an RSU broadcasting an advisory message to reroute to a PAR, one could use the traveler information message (TIM) [7], which is used to broadcast various advisory or road sign info messages. The TIM can be configured to be active on a minute by minute basis and even has limited support for custom strings, which can be useful for informing drivers

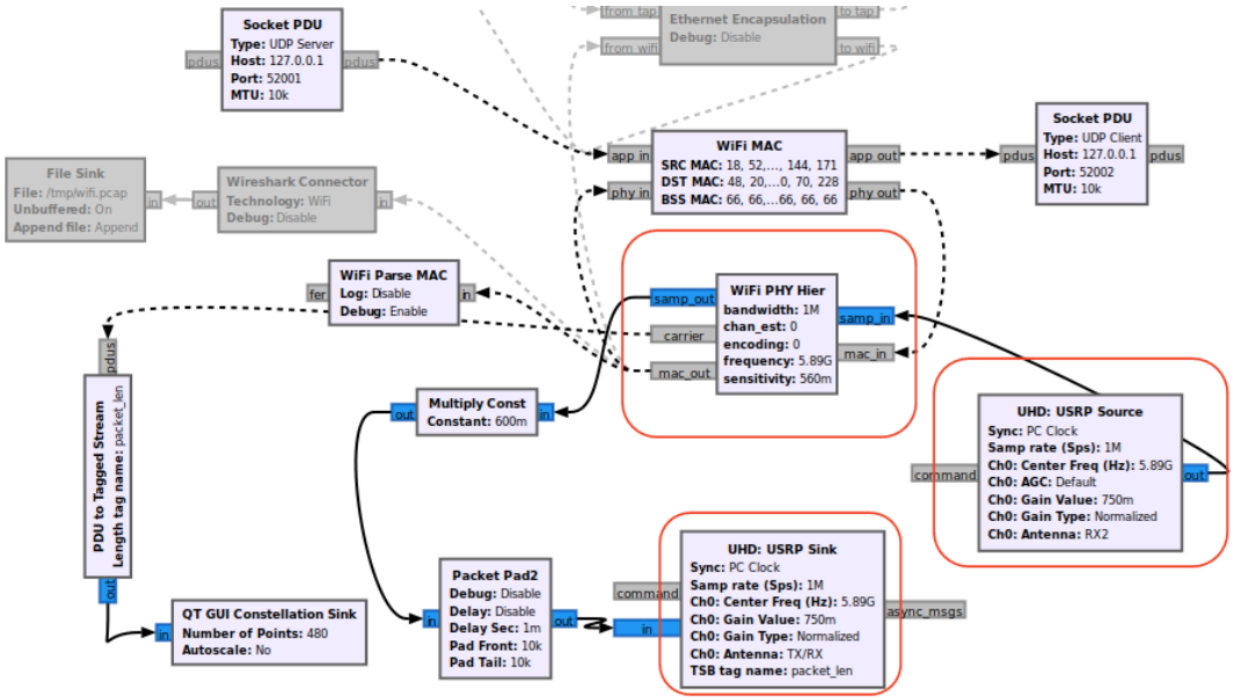
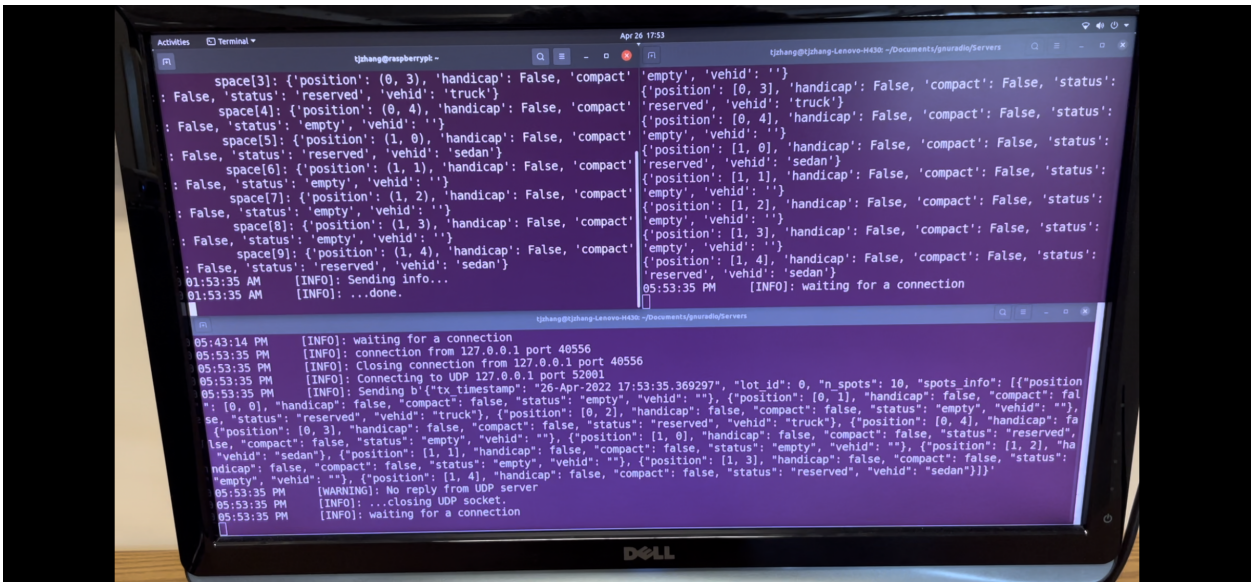


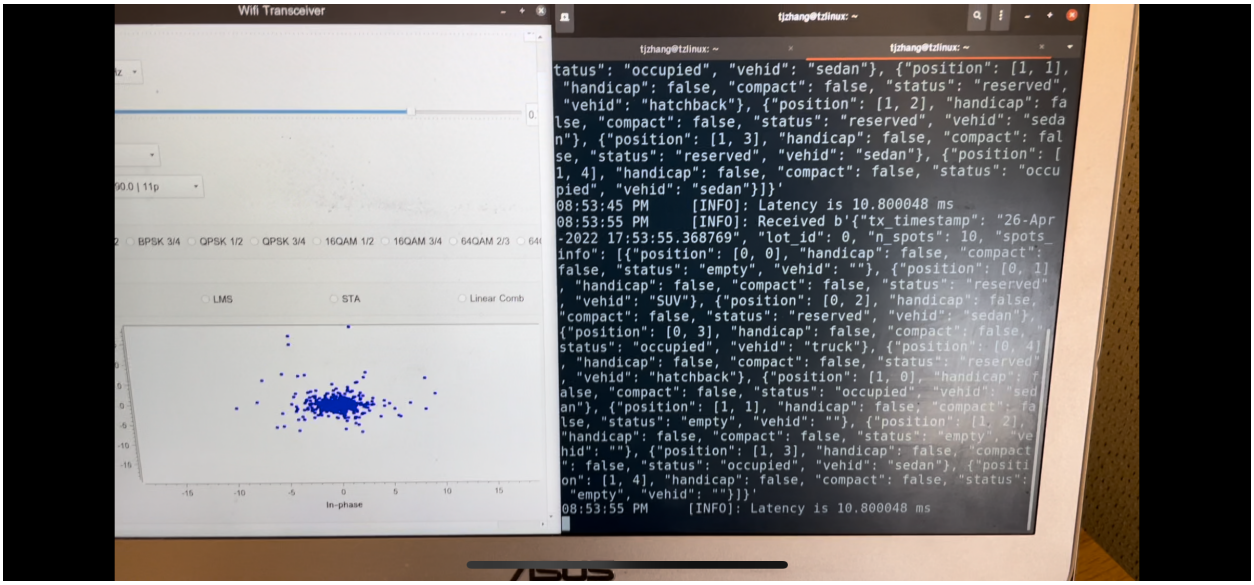
Figure 4.3: Hardware blocks for 802.11p in GNU Radio, the same interface from [1]

about the nearest PAR. For collecting state information, an RSU could collect a basic safety messages (BSM) [7] from vehicles nearby and aggregate the data. Part 1 of the BSM frame is mandatory and reports the vehicle's position and velocity. Part 2 of the BSM frame is optional but could be customized with additional information that is of importance to the RSU or ICM strategy, such as emissions information or vehicle type.

The cloud server functions similarly to a Transportation Management Center (TMC) [33] but with the additional role of a park and ride management system. Communicating to the TMC from an RSU or a smart parking structure can be done over LTE; there are examples of RSUs being equipped with bidirectional LTE radios such as one developed by Siemens [34].

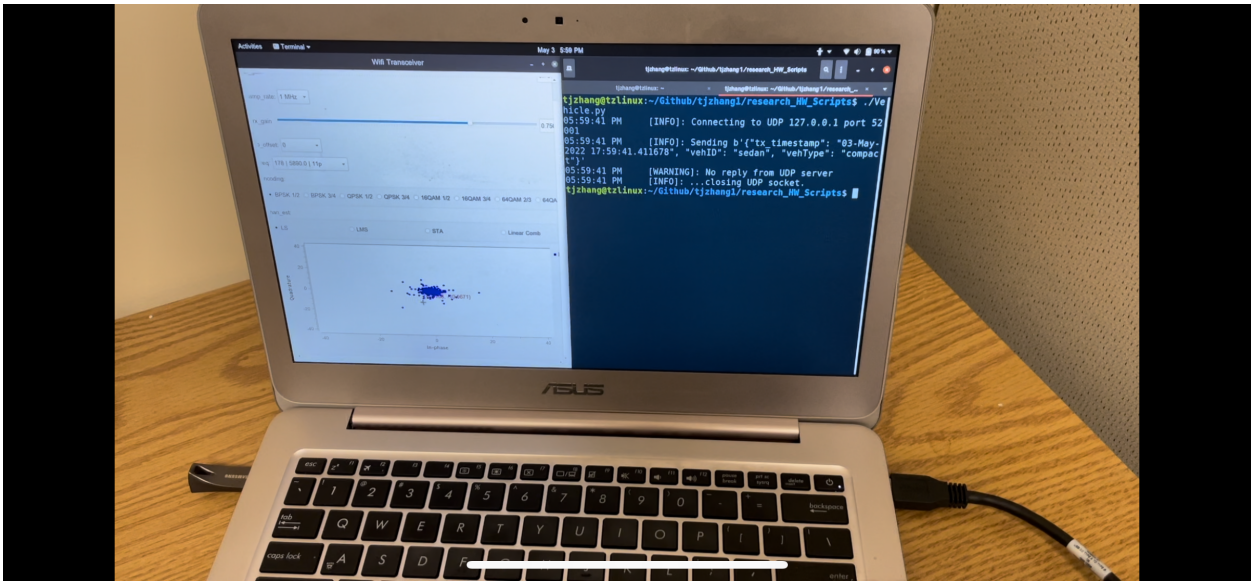


(a) Desktop screenshot showing interface for RPI (top-left), cloud (top-right), and RSU (bottom). Parking info is generated by RPI and propagates to RSU for broadcasting.

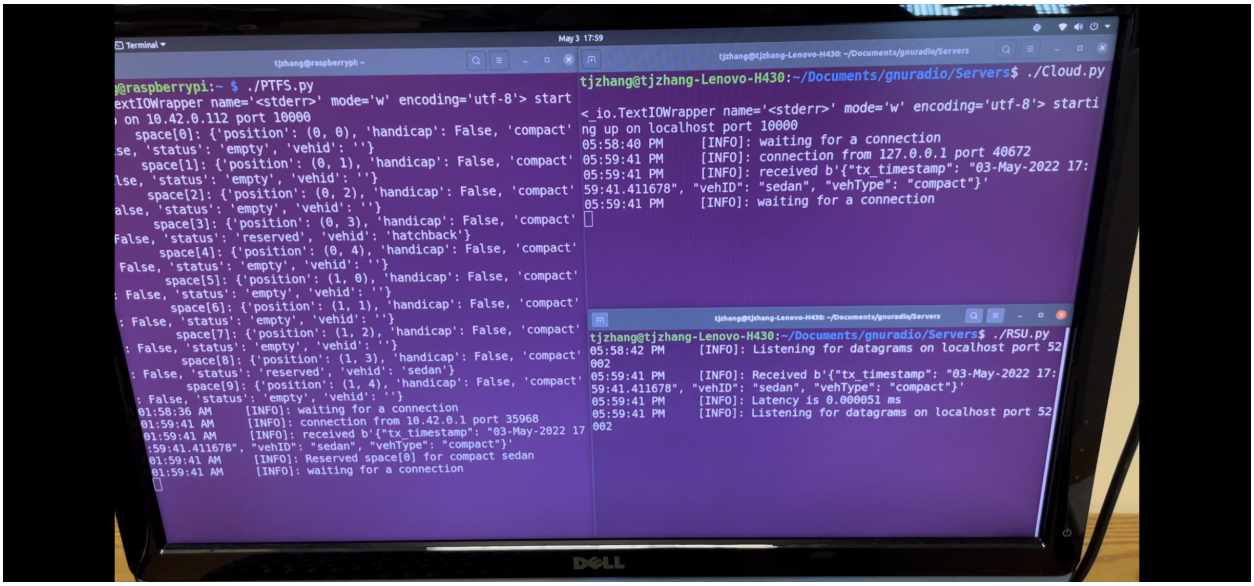


(b) Laptop screenshot showing interface for OBU (right). The OBU receives and displays the parking info from the broadcast.

Figure 4.4: Screenshots demonstrating forward flow of information from parking structure to OBU



(a) Laptop screenshot showing interface for OBU (right). Vehicle info is generated and broadcast.



(b) Desktop screenshot showing interface for RPI (left), cloud (top-right), and RSU (bottom-right). The RSU receives the vehicle from the broadcast and forwards it to the cloud.

Figure 4.5: Screenshots demonstrating backward flow of information from OBU to cloud

Chapter 5

System Model

In this section, we model the possible travel times experienced by vehicles when traversing the integrated corridor, and accordingly, define the evaluation metrics and formulate the optimization problem to effectively manage traffic along the ICM supported by CV technology and PAR facilities. Table 5.1 summarizes the notation used for our model.

Let $\Pi_N = \{\pi_1, \pi_2, ..., \pi_N\}$ be the N consecutive links constituting the integrated corridor, where each $\pi_i \in \{0,1\}^d$ can be represented as a d-dimensional binary vector characterizing the number of lanes in each link, and d being the maximum number of lanes supported by any corridor link; $P_M = \{\rho_1, \rho_2, ..., \rho_M\}$ represent the M park and ride structures associated with the integrated corridor; $\Psi = [\psi_1, \psi_2, ...\psi_N]$ as a binary vector to characterize possible entry ramp positions where $\psi_i = 1$ indicates that an entry ramp exists at link π_i; $\Theta_N = [\theta_1, \theta_2, ...\theta_N]$ characterize possible exit ramp positions where $\theta_i = 1$ indicates that an exit ramp exists at section π_i.

Notation	Definition
V_K	The set of K vehicles traversing the corridor from start to the sink, i.e., $\{v_1, v_2, ..., v_K\}$
P_M	The set of M PAR and ride structures accessible for the corridor management process, i.e., $\{\rho_1, \rho_2, ..., \rho_M\}$
Π_N	The set of N consecutive partitions (links) constituting the integrated corridor's mainline, i.e., $\{\pi_1, \pi_2, ..., \pi_N\}$
Ψ_N	Vector of N binary variables indicating whether an exit ramp exists associated with link partition π_i, i.e., $[\psi_1, \psi_2, ...\psi_N]$
Θ_N	Vector of N binary variables indicating whether an entry ramp exists at every partition π_i, i.e., $[\theta_1, \theta_2, ...\theta_N]$
S_j	The number of *total* parking slots at PAR structure j such that $j < M$
$s_j^{(t)}$	The number of *occupied* parking slots in the PAR structure j at time t
λ_{k,ρ_j}	The arrival time of vehicle k at PAR ρ_j.
τ_j^{HOV}	The recurring time period at which an HOV vehicle leaves the j^{th} PAR structure
$\tau_{k,i\to j}$	Time taken by vehicle k to travel from i to j.
$\mathcal{T}_{action}$	Delay experienced by a vehicle when performing *action* defined in the subscript; $action \in \{exit, PAR, ret\}$
$x_j^{(t)}$	Binary variable indicating transmission of park and ride messages at time interval t from RSU j.
$X^{(t)}$	Binary vector of park and ride advertisements states for the L deployed RSUs at time t, i.e., $[x_1^{(t)}, x_2^{(t)}, ...x_L^{(t)}]$
y_k	binary variable expressing the manner in which the k^{th} vehicle commuters opted to traverse the corridor, i.e., directly or through PAR

Table 5.1: Table of notations for the proposed system model

5.1 Travel Delay

Assume a vehicle k enters the corridor through a link n with the final destination being the corridor's sink at link N. Given PAR is supported, the time taken by vehicle k to traverse the corridor can be given as follows:

$$T_{k,N} = \begin{cases} \sum_{i=n}^{N-1} \tau_{\pi_i \to \pi_{i+1}}, & \text{if } y_k == 0 \\ \mathcal{T}_{exit} + \mathcal{T}_{PAR}^{total} + \mathcal{T}_{rem}, & \text{if } y_k == 1 \end{cases} \tag{5.1}$$

where $y_k \in \{0, 1\}$ describes the method by which vehicle k traversed the corridor: y_k=0 indicating the conventional direct approach and y_k=1 implying opting for a PAR option. For the former, the time taken by vehicle k is estimated through the accumulation of times, $\tau_{\pi_i \to \pi_{i+1}}$[1], representing the time taken by the vehicle to traverse from one link i to the next i+1 until the final transition to the sink point, $\tau_{\pi_{N-1} \to \pi_N}$. For the latter case, we breakdown the time experienced by vehicle k commuters opting for a PAR option into three components: (i) T_{exit} is the time taken to reach an exit ramp from the corridor once y_k values has turned to 1, (ii) $\mathcal{T}_{PAR}^{total}$ representing the total PAR service time, and (iii) $\mathcal{T}_{rem}$ representing the remainder time to be traversed by the returning vehicle until the sink. These delay components are further defined as follows:

$$\mathcal{T}_{exit} = \sum_{i=n}^{n'-1} \tau_{\pi_i \to \pi_{i+1}} \text{ s.t. } \theta_{n'} == 1 \tag{5.2}$$

where $\mathcal{T}_{exit}$ is vehicle k's cumulative time to reach the exit ramp at link n' from a starting position n on the corridor.

Concerning $\mathcal{T}_{PAR}^{total}$, we first define $\mathcal{T}_{\rho}(\cdot)$ as a function characterizing the added time expenditure of vehicle k commuters when serviced by a PAR structure as follows:

$$\mathcal{T}_{\rho}(\delta, \rho_j, \pi_{n'}) = \tau_{\delta \to \rho_j} + \tau_{\rho_j}^{wait} + \tau_{\rho_j \to \pi_{n'}} \text{ s.t. } s_j < S_j \tag{5.3}$$

where δ is a generic expression for the starting position of vehicle k navigating towards the the servicing PAR structure, ρ_j. We specify δ generalized as such to account for cases in which vehicle k did not navigate directly to the servicing PAR (as shown below); ρ_j is the servicing PAR facility, and $\pi_{n'}$ represents the corridor link n' with an accessible entry ramp (i.e., $\psi_{n'}$=1) that is nearest to the ρ_j. Hence, $\tau_{\delta \to p_j}$ is the time taken by vehicle k to reach ρ_j from a starting position δ, whereas $\tau_{\rho_j \to \pi_{n'}}$ is the time expenditure by the HOV carrying

[1]We omit the subscript k here for reading convenience

vehicle k commuters from when it is released at ρ_j until it re-enters the corridor through link $\pi_{n'}$ – intrinsically capturing the travel and ramp metering times. Lastly, the waiting time endured by the arriving passengers at ρ_j till the HOV leaves is denoted by $\tau_{\rho_j}^{wait}$ defined as follows:

$$\tau_{\rho_j}^{wait} = \tau_j^{HOV} - (\lambda_{k,\rho_j} \pmod{\tau_j^{HOV}}) \tag{5.4}$$

where λ_{k,ρ_j} is the arrival time of vehicle k at its designated slot in ρ_j; τ_j^{HOV} is the time period length before a new HOV is released from ρ_j with a final destination at the corridor's sink π_N. λ_{k,ρ_j} is defined with in reference to the same time unit of τ_j^{HOV}. For example, if $\tau_j^{HOV} = 15$ minutes and λ_{k,ρ_j} is the 47^{th} minute within a certain hour of the day, then $\tau_{\rho_j}^{wait}$ translates to 13 minutes waiting time. From here, the total PAR time, $\mathcal{T}_{PAR}^{total}$, can be modeled as follows:

$$\mathcal{T}_{PAR}^{total} = \begin{cases} \mathcal{T}_\rho(\pi_n, \rho_j, \pi_{n'}), & \text{if } s_j < S_j \\ \tau_{\pi_n \to \rho_j} + \sum_{i=j}^{j'-2} \tau_{\rho_i \to \rho_{i+1}} + \mathcal{T}_\rho(\rho_{j'-1}, \rho_{j'}, \pi_{n'}), & \\ \quad \text{if } s_j == S_j \forall j < j', \text{ and } s_{j'} < S_{j'} \text{ s.t. } j' < M' & \\ \tau_{\pi_n \to \rho_j} + \sum_{i=j}^{M'-1} \tau_{\rho_i \to \rho_{i+1}} + \tau_{\rho_{M'} \to \pi_{n'}}, & \\ \qquad \text{if } s_j == S_j \forall j \leq M & \end{cases} \tag{5.5}$$

where the first case captures the conventional scenario when PAR service is provided for vehicle k at ρ_j as its occupancy s_j did not exceed the maximum S_j. Otherwise, the latter two cases capture unfavorable scenarios when the vehicle arrives at ρ_j to find its occupancy at full capacity (i.e., s_j=S_j). The vehicle's behavior afterwards is to either navigate to another facility from the M' nearby PAR structure (if any), or return to traverse the corridor

normally. The former is captured by the second case which accounts for the possibility that vehicle k may explore multiple PAR facilities until it reaches one with empty parking slots at j'. The third and last case is for when the vehicle navigates back to the corridor after failing to find a single parking spot in all M' facilities. Parking unavailability represents corner cases that need to be accounted for. In reality, only a small fraction of vehicles may experience this since space availability of PAR facilities would be updated regularly at the centralized management system for the corridor.

Lastly, $\mathcal{T}_{ret}$ represents the remainder time to be traveled along the corridor once the returning vehicle (HOV or the same vehicle k) renters the corridor at link π_n at which an entry ramp exists as follows:

$$\mathcal{T}_{ret} = \sum_{i=n}^{n'} \tau_{\pi_i \to \pi_{i+1}} \text{ s.t. } \theta_n == 1 \tag{5.6}$$

5.2 Evaluation Criteria

Let the set $V_K = \{v_1, v_2, ..., v_K\}$ be the total number of vehicles on the corridor during a time window t, where every $v_k \in V_K$ has entered the corridor from its starting link, π_n, with a final destination at the sink, π_N. Let also $V_{K'} \subset V_K$ be the subset of vehicles that reach the corridor's sink during window t such that $K' \leq K$. As such, we can define the following key metrics to evaluate the overall congestion state along the corridor:

$$\mathbb{FR} = |V_{K'}| \tag{5.7}$$

$$\mathbb{T} = \frac{\sum_{k=1}^{K'} T_{k,N}}{\mathbb{FR}} \ \ \forall \ \ v'_k \in V_{K'} \tag{5.8}$$

$$\mathbb{CE} = f(V_K, efficiency, SPEED) \tag{5.9}$$

in which $\mathbb{FR}$ represents the flow rate of the corridor during window t, evaluated as the cardinality of $V_{K'}$; $\mathbb{T}$ represents the average travel time experienced by the vehicles reaching the sink, π_N, evaluated as the sum of individual travel times over the flow rate; $\mathbb{CE}$ represents the carbon emissions exerted by the V_K set of vehicles traversing the corridor during window t. The evaluation of $\mathbb{CE}$ depends on V_K as well as their corresponding fuel efficiency and travel speeds.

5.3 Problem Formulation

From here, we can formulate our problem as follows: given an integrated corridor with RSUs that can broadcast PAR availability to CVs, our goal is to find the optimal advertisement strategy, $X^{*(t)}$, at time window t to inform upstream traffic about the state of traffic conditions and suggest PAR alternatives accordingly. Thus, we can define the global optimization objective at time t as follows:

$$X^{*(t)} = \max_{X^{(t)}} F(\mathbb{T}, \mathbb{FR}, \mathbb{CE}) \tag{5.10}$$

where F represents a global optimization function to be maximized with respect to the evaluation metrics defined in equations 5.7, 5.8, and 5.9. One straightforward implementation of F would be to employ a weighted sum formula with negative weights assigned to the metrics that need to be minimized (i.e., $\mathbb{CE}$ and $\mathbb{T}$). It should be noted that the global optimization objective can be generalized to account for other objectives, such as minimizing the costs of fuel and park and ride charges for commuters.

Chapter 6

Deep Reinforcement Learning

In this section, we present our solution for ICM leveraging CV technology and existing PAR infrastructure. At the heart lies a Deep Reinforcement Learning (DRL) model deployed on the centralized server capable of pooling traffic data from various corridor partitions, as well as the PAR occupancy status. Accordingly, collected data can be analyzed by the agent in real-time to guide the optimization process of the PAR advertisement strategy across the deployed RSUs. The key components of our proposed DRL solution are detailed in the following sections.

6.1 State Space

We define the set of observation parameters that need to be collected during time window t to be provided as input signals to the DRL agent.

6.1.1 Parking Occupancy Status

Firstly, information about the current available parking slots, $\phi_j^{(t)}$, for each associated PAR_j are provided as $\{\phi_j^{(t)} = S_j - s_j^{(t)}\}_{j=1}^{M}$. In reality, occupancy status may vary in real-time (e.g., due to vehicles arriving from outside the corridor), and thus, every ϕ_j merely serves as an *indication* rather than a *guarantee* on the current occupancy status.

6.1.2 Traffic Density

At the j^{th} RSU, the traffic density, $\Gamma^{(t)}$ can be estimated through monitoring the percentage of time in which nearby linked sensors (e.g., induction loops) are occupied on the main line. Given M RSUs, the set of densities' observation can be given as $\{\Gamma_j^{(t)}\}_{j=1}^{M}$. Note that each density estimate can be multi-dimensional in d due to collecting data from multiple sensors deployed across several adjacent lanes. Additionally, since RSU coverage is limited to the range of DSRC, density may also be estimated by collecting BSMs to count the number of vehicles in range.

6.1.3 Speed

Similarly at each RSU, we can estimate the traffic flow speed to monitor the time it takes a vehicle to traverse to consecutive points along the mainline. As mentioned in chapter 4, BSMs are required to report vehicle speed, so RSUs could average this information collected over time window t to approximate flow speed. Thus, we also include the set of speeds $\{\mu_j^{(t)}\}_{j=1}^{M}$.

6.2 Action Space

For each RSU deployed along the corridor, the DRL decides whether to broadcast park and ride advisories to connected vehicles based on the collected data, allowing drivers to decide whether to continue on the corridor or reroute to one of the nearby PAR structures. Such advertisement lasts for the entire duration of the corresponding time window. Formally, the action space can be given by the following vector as $X^{(t)} = [x_1^{(t)}, x_2^{(t)}, ...x_L^{(t)}]$ where L is the index of the last RSU on the corridor.

6.3 Training and Reward

The DRL agent uses a double Q-learning (DDQN) algorithm [35] in order to learn Q-values of actions for the corresponding state estimation. We utilize an epsilon-greedy method to balance between exploration and exploitation and implement a replay buffer to avoid catastrophic forgetting.

In order to train the DRL agent, we define training episodes of length $c \cdot t$, where c is a positive integer multiplier specifying the number of steps and t represents the window of simulation time per step. An episode terminates whenever c training windows have passed for a fair assessment. The reward evaluation metrics are three-fold:

1. The throughput of passengers, computed using equation 5.7.
2. The average travel time per passenger, computed using equation 5.8.
3. The average carbon dioxide emitted per passenger, computed using equation 5.9.

These values are accumulated dynamically as vehicles exit at the sink. Not all vehicles will accumulate their rewards when reaching the sink, we concern ourselves only with mainline

vehicles, which we define as:

- Vehicles that spawn at the source and travel to the sink without rerouting to a park and ride structure
- HOVs that carry redirected passengers to sink

All other vehicles that do not fit this criteria are ignored as far as rewards are concerned.

With this in mind, our reward function is specified in equation 6.1 as a weighted sum of the three reward metrics.

$$R = \beta_1 \cdot (\mathbb{FR} - target_{FR}) + \beta_2 \cdot (\mathbb{T} - target_T) + \beta_3 \cdot \mathbb{CE} \tag{6.1}$$

where R is the reward, β_1 is the weight modifier for the throughput $\mathbb{FR}$, β_2 is the weight modifier for the travel time $\mathbb{T}$, and β_3 is the weight modifier for the CO2 emissions $\mathbb{CE}$. Since mainline vehicles will continuously be exiting the simulation, we subtract a fixed value $target_{FR}$ from throughput $\mathbb{FR}$, with $target_{FR}$ representing the expected minimum throughput. Similarly, since mainline vehicles will always take time to reach the end of the corridor, we subtract a fixed value $target_T$ from the measured travel time $\mathbb{T}$, with $target_T$ representing the minimum time a vehicle may need to drive from sink to source.

Chapter 7

Experimental Setup and Training

The basic idea of the experiment is to train the DRL model with the I5 Veins simulation in the loop until the model converges to a policy. SUMO generates randomized traffic patterns based on real-world ramp volume data as explained in chapter 3. The Veins framework lets us map SUMO vehicles to OMNeT++ vehicle applications and SUMO lane detectors to OMNeT++ RSU applications. The Veins-Gym interface [36] allows the Veins simulation to communicate with the Python DRL script. Figure 7.1 shows a high-level overview of the simulation structure and functionality.

We built the Veins simulation from the I5 corridor developed in chapter 3. For the simulation, we chose to use SUMO's default driving behavior and CO2 emissions model [16]. We also reduced the size of the scenario, setting the sink just after Garfield Ave (see figure 3.1) to reduce computation complexity. SUMO performs a vast amount of calculations and the program can only run on a single core, which bottlenecks training.

The RSU application collects data about the number and average speed of nearby vehicles from the lane detectors. In real life, RSUs could collect this data by aggregating BSMs received from vehicles via DSRC, and this behavior could be simulated in Veins. However,

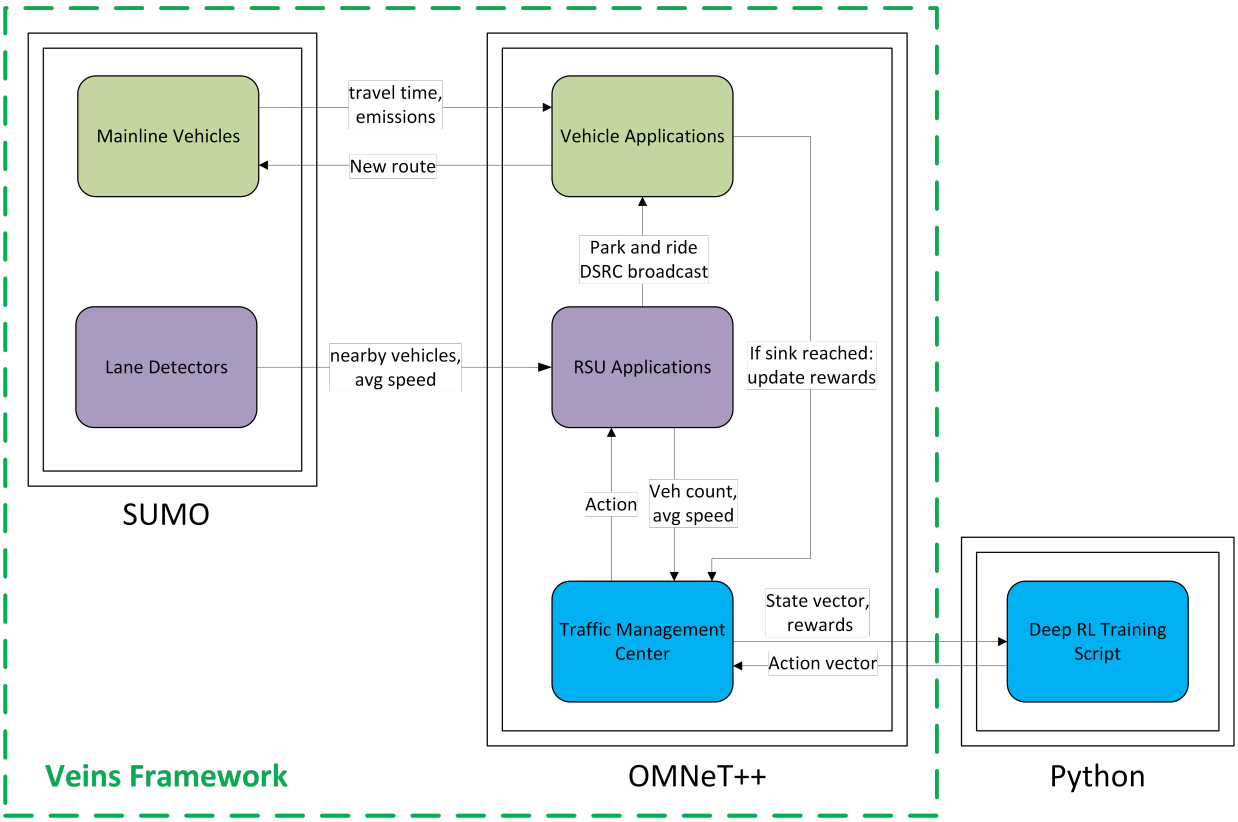


Figure 7.1: Flow chart of simulation-in-the-loop training process

to keep simulation complexity low, we simply programmed the RSU application to read the necessary measurements from its corresponding lane detectors. These RSUs send their observations to a central Transportation Management Center (TMC), which interacts with the DRL agent.

The vehicle application keeps track of its travel time and total emissions. If it reaches the sink of the simulation, it will update the rewards accumulator in the TMC for the current time window t with the vehicle's travel time and emissions, and increment the throughput by one. If the vehicle leaves the simulation to park at a PAR, and a parking space is available, its rewards will be transferred to the next HOV. If the vehicle attempts to park but no parking space is available, its rewards will be transferred to a continuing vehicle that will proceed to the sink of the simulation. HOVs and continuing vehicles that carry the rewards of a parked vehicle will update the TMC once they reach the sink, and they cannot be rerouted at any

point.

Periodically, the TMC sends a vector of RSU observations and the rewards for time window t to the Python script, and the script responds with an action vector indicating which RSUs should broadcast a PAR message. The TMC relays this information to the RSUs which will broadcast PAR messages via simulated DSRC at a fixed interval to nearby vehicle applications. Vehicle applications that receive an advisory PAR broadcast from a nearby RSU have a fixed percent chance to reroute to the park and ride by taking the next exit.

Here are some assumptions made when designing the Veins simulation:

- There is only one PAR structure, the North Lakewood Park and Ride.
- There is only one passenger per vehicle. When vehicles update the accumulated rewards, they will only increase the throughput by one. If HOVs have N passengers onboard, they will increase the throughput by N.
- Commuters rerouted to the PAR structure can take the next HOV leaving the PAR once they arrive. See section 7.2 for how rerouting delay is computed.
- Vehicles do not exit the PAR structure during the simulation, the number of available spaces depends solely on the DRL agent.
- HOVs do not encounter additional waiting times at entering ramp meters. We added a dedicated HOV lane at the on ramp nearest to the North Lakewood Park and Ride.
- RSUs are positioned prior to every exit and on ramp along the corridor. See figure 7.2 to see the full placement.
- DSRC broadcasts have no propagation delay. Simulating broadcast delay would create simulation overhead and does not add much value to the scope of this experiment.

- DSRC range is limited to 75m radius around RSU, which is within the 250-300m effective range of DSRC [13].

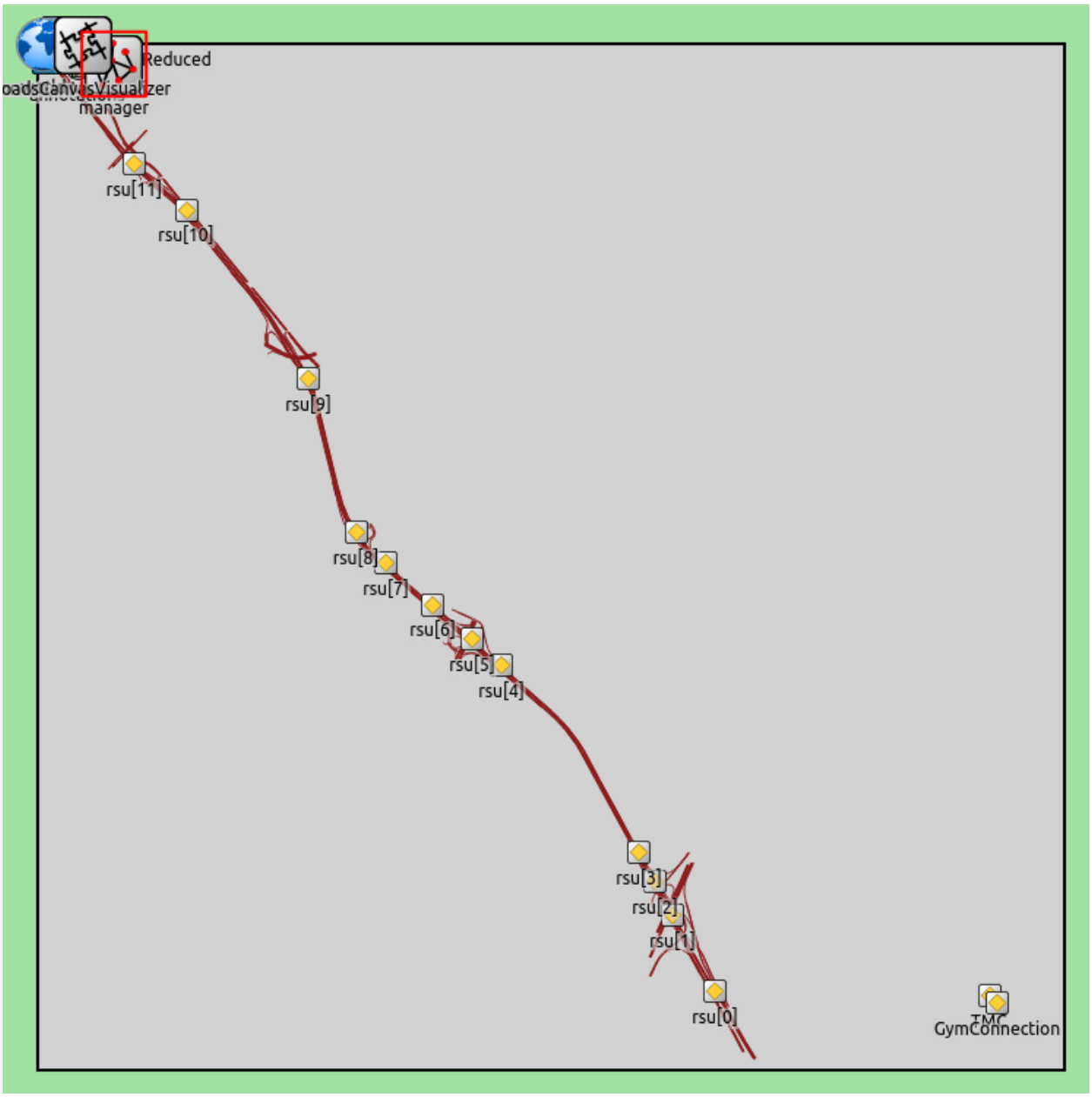


Figure 7.2: RSU placement for Veins simulation

7.1 Parameters

The following defines a list of Veins simulation parameters.

- S: traffic spawn scale factor
- λ_{HOV}: the frequency of the HOVs leaving the PAR
- t: the time window that constitutes one step in the reinforcement learning algorithm
- α: compliance probability of vehicles with PAR messaging
- γ: probability distribution function capturing additional time for the vehicle to get to and find parking at the PAR
- $f_{redirect}$: the frequency of PAR broadcasts during a time window t
- P: number of parking spaces available at PAR
- r: fixed ramp metering rate for on ramps

Training of the reinforcement learning algorithm took place with the simulation parameters described in table 7.1. The HOV frequency schedule λ_{HOV} is set spawn every 60s so passengers would not wait too long relative to the time it takes to traverse the corridor. The amount of time that constitutes a step size and the action space t is set to 600s because it

Parameter	Value
S	2.0
λ_{HOV}	$\frac{1}{60sec}$
t	600 s
α	10%
$f_{redirect}$	30 s
P	400
r	900 VPH

Table 7.1: Parameter values for Veins simulation

takes roughly 300s for a vehicle to reach the sink and contribute its rewards. The compliance parameter α is only at 10% since realistically the majority of drivers would ignore an advisory message to find the nearest PAR. The ramp metering rate is set to a fixed interval of 900 VPH or one vehicle every four seconds, which is within range of typical ramp meter rates in California [37].

7.2 Delay

As vehicles exit the simulation, the travel time cost of reaching the PAR is calculated before its reward is transferred to an HOV or continuing vehicle. This penalty is dependent on the exit that it takes. Since we used OpenStreetMap to generate our network as explained in chapter 3, the simulation distance and speed limits are based on real life values, so we can use Google Maps data to approximate the travel time for exiting vehicles. Using Google Maps, we estimated the travel time to reach the North Lakewood Park and Ride structure from each exit along the corridor. These values are summarized in table 7.2.

Exit	**Time to N Lakewood PAR (s)**
I-605	300
Lakewood Blvd	30
Paramount Blvd	180
Slauson Ave	300
Garfield Ave	360
Washington Blvd	540
Atlantic Blvd	720
Atlantic and Triggs	840
Eastern Ave	900
Ditman Ave	1080
Calzona St	1200

Table 7.2: Approximate travel times to reach the North Lakewood Park and Ride from various highway exits, taken from Google Maps

Parameter γ is simply a normal distribution that models the additional time for the vehicle to

get to and find parking within the PAR. We take the absolute value of the normal distribution to avoid negative delay values. If the vehicle instead finds that there are no more parking spots available, the delay is modeled more harshly. Thus, the total travel time penalty for a vehicle taking an exit is represented in equation 7.1.

$$delay = \begin{cases} T_{exit} + |normal(0, T_{exit}/8)| + (T_{arrival} \bmod \frac{1}{\lambda_{HOV}}) & \text{if spaces available} \\ 2 * T_{exit} + |normal(0, T_{exit}/4)| & \text{else} \end{cases} \tag{7.1}$$

where T_{exit} represents the corresponding exit time from table 7.2, and $(T_{arrival} \bmod \frac{1}{\lambda_{HOV}})$ represents the additional delay a driver would experience waiting for the next HOV to leave the station.

7.3 Deep RL Training

From table 7.2, exits to I-605, Lakewood, Paramount, and Slauson have the smallest rerouting time penalty. Thus, the action vector is restricted to RSU[0], RSU[3], RSU[4], RSU[6], and RSU[7], marked in figure 7.2. Since each RSU is represented as a bool in the action vector, this limits the action space size to $2^5 = 32$, which helps with exploration.

The DRL agent utilizes two hidden layers for its neural network. Since the state vector involves {average speed, occupancy} for 12 RSUs and the available parking spaces, the flattened input vector is $1 + 12 * 2 = 25$ floating point values. These are fed into a hidden layer of 128 perceptrons using the the ReLU activation function. The next hidden layer is 64 ReLU perceptrons. Finally, the output layer contains 32 perceptrons to approximate the Q values for each possible combination of actions for the 5 RSUs.

In terms of reward function, we programmed the TMC to perform equation 6.1 when the Veins simulation is running. The reward metric weights are:

- $target_{FR} = 600$. This is the fixed constant that is subtracted from throughput before weighting. Typically at least 600 vehicles can exit the simulation in one simulation step.

- $\beta_1 = \frac{10}{target_{FR}}$. This is the weight for throughput, making it so an additional 60 vehicles will add an extra point of reward.

- $target_T = 400$. This is the fixed constant that is subtracted from travel delay before weighting. Typically it takes at least 400s to reach the end of simulation.

- $\beta_2 = -\frac{1}{target_T}$. This is the weight for travel delay, it is negative so that it will be minimized.

- $\beta_3 = -\frac{4}{10000000}$. This is the weight for the carbon emissions, it is negative so that it will be minimized. The emissions values are in the order of magnitude of 10^6.

We trained the deep learning agent on one fixed seed of the simulation for 1000 episodes to obtain the DRL agent model. Setting the seed to a fixed value makes the traffic behavior (including route probabilities, Poisson process spawning, and driver AI) deterministic within SUMO. Then, in the next chapter, we evaluate the model's performance on other simulation seeds to test if the model generalizes to different permutations of the defined traffic behavior.

Chapter 8

Results and Discussion

Figure 8.1 shows the training results, a plot of rewards over episode number when training on simulation seed 1201 for 1000 episodes. To test if the model generalizes to different

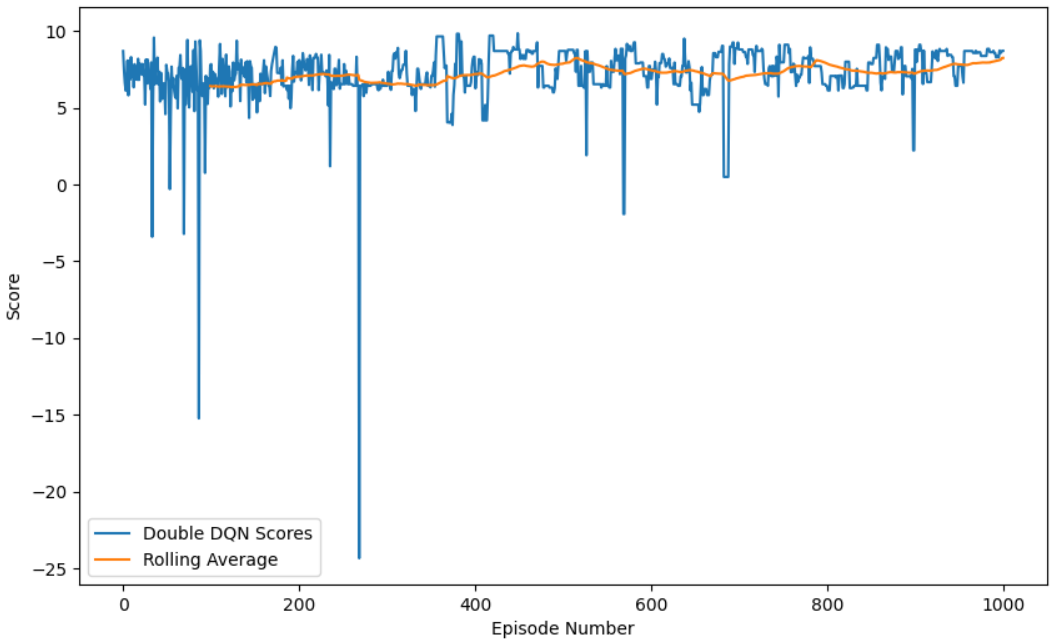
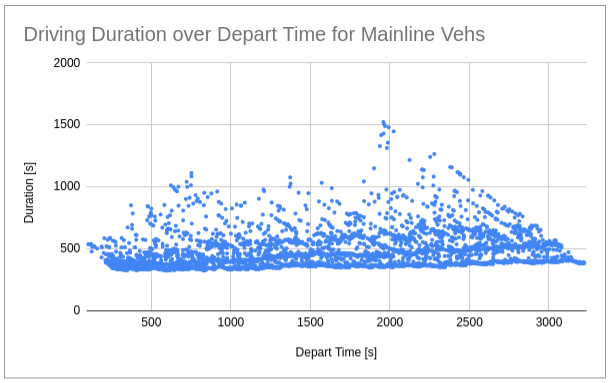


Figure 8.1: DRL agent training results on seed 1201

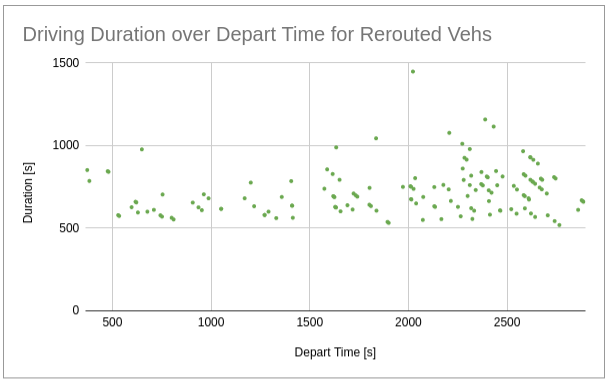
permutations of the defined traffic behavior, we evaluated the model on three different seeds of the SUMO simulation and plotted the throughput, average travel delay per passenger, and average carbon emissions per passenger for each seed. In the following sections, we compare these metrics against the same seed simulation without any ICM control agent.

8.1 Experiment 1: Seed 64643

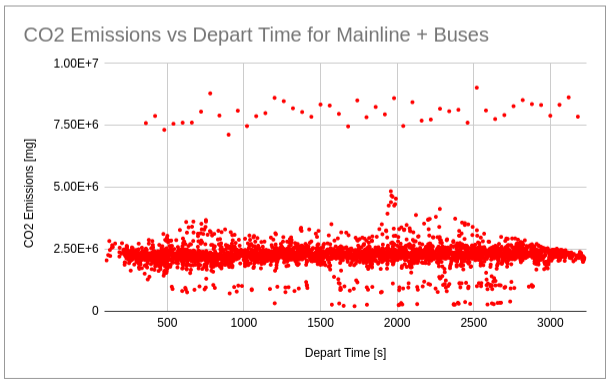
Figure 8.2 shows the results from the optimal policy applied to seed 65643. Figure 8.3 shows the results for seed 64643 without ICM control. Our DRL agent shows a 3.99%



Average Travel Time: 475.5795308 s



Avg Travel Time for Rerouted Vehs 720.1020408 s
Avg Travel Time for Nonrerouted Vehs 465.9480171 s

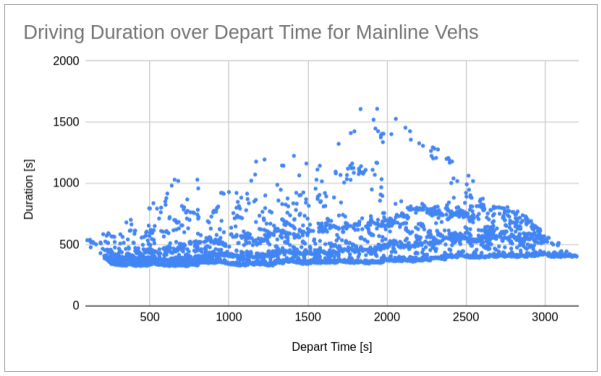


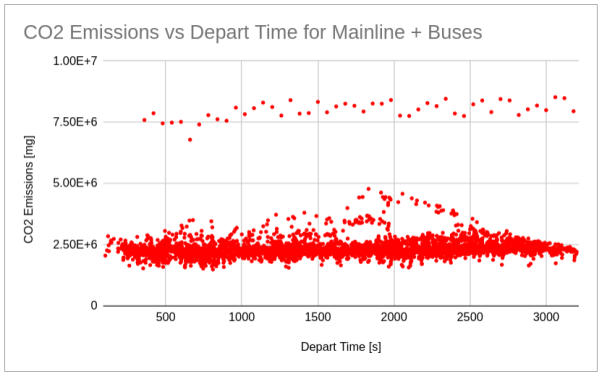
Average CO2 Emissions: 2370019.755 mg

Summary	
Rerouted Vehicle Count	147
Nonrerouted Vehicle Count	3732
Total Throughput	3879

Figure 8.2: Results for running the DRL agent on SUMO seed 65643

increase in throughput, 2.69% reduction in average travel time, and 2.97% reduction in average emissions over the no control scenario. Additionally, the 3732 drivers on the freeway





Average Travel Time: 488.7252011 s

Average CO2 Emissions: 2442638.161 mg

Total Throughput	3730

Figure 8.3: Results for running SUMO seed 65643 without DRL agent

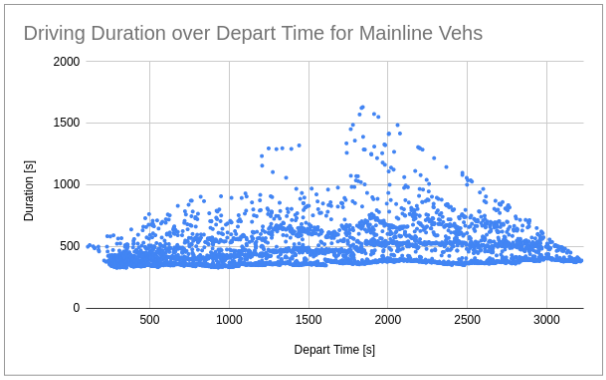
experience a 4.67% reduction in travel time while the 147 drivers taking HOVs experience a 47.34% increase in travel time over the no control scenario.

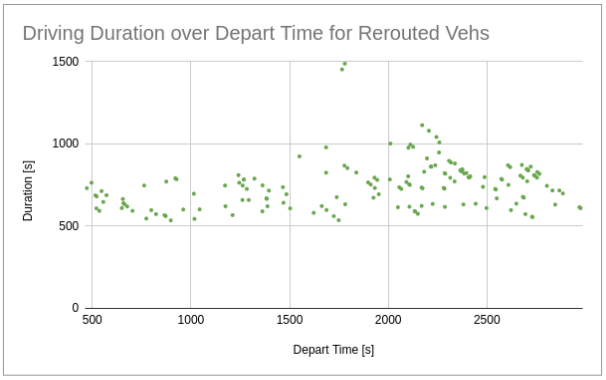
8.2 Experiment 2: Seed 44435

Figure 8.4 shows the results from the optimal policy applied to seed 44435. Figure 8.5 shows the results for seed 44435 without ICM control. Our DRL agent shows a 2.95% increase in throughput, negligible difference in average travel time, and 3.00% reduction in average emissions over the no control scenario. Additionally, the 3597 drivers on the freeway experience a 2.55% reduction in travel time while the 165 drivers taking HOVs experience a 52.56% increase in travel time over the no control scenario.

8.3 Experiment 3: Seed 27438

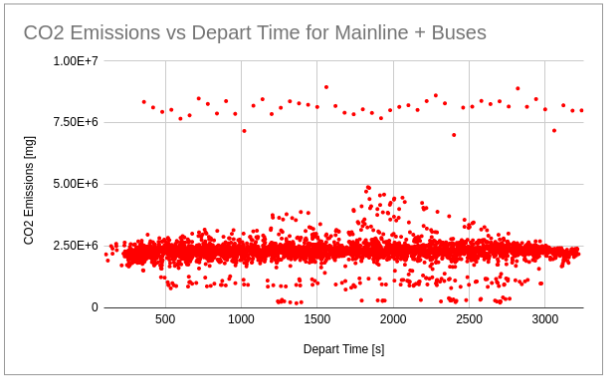
Figure 8.6 shows the results from the optimal policy applied to seed 27438. Figure 8.7 shows the results for seed 27438 without ICM control. Our DRL agent shows a negligible





Average Travel Time: 489.6887294 s

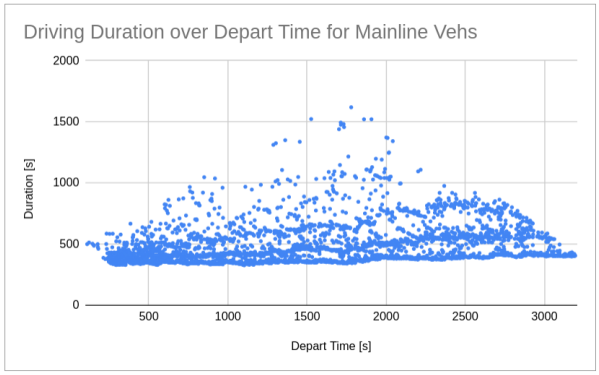
Avg Travel Time for Rerouted Vehs 748.0787879 s
Avg Travel Time for Nonrerouted Vehs 477.8359744 s

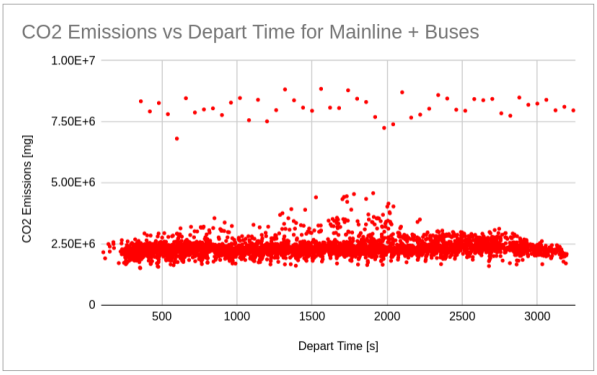


Summary	
Rerouted Vehicle Count	165
Nonrerouted Vehicle Count	3597
Total Throughput	3762

Average CO2 Emissions: 2379751.469 mg

Figure 8.4: Results for running the DRL agent on SUMO seed 44435





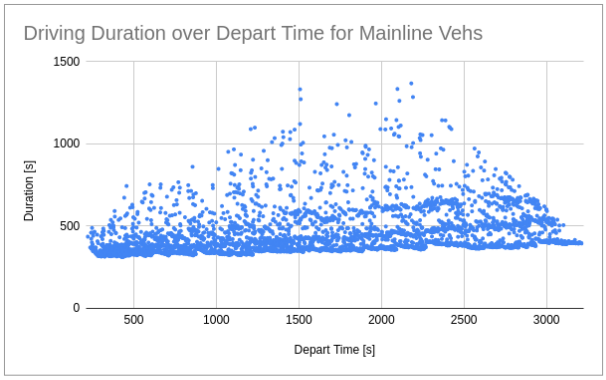
Average Travel Time: 490.3497537 s

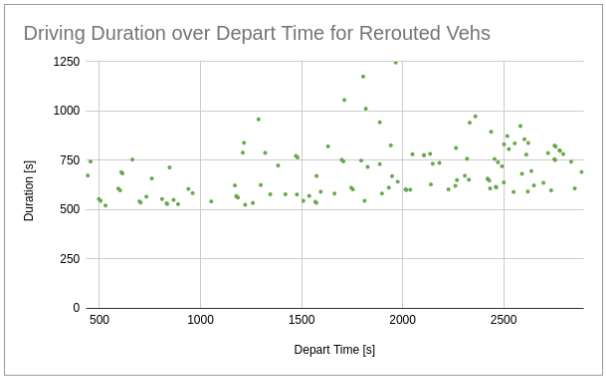
Average CO2 Emissions: 2453332.113 mg

Total Throughput	3654

Figure 8.5: Results for running SUMO seed 44435 without DRL agent

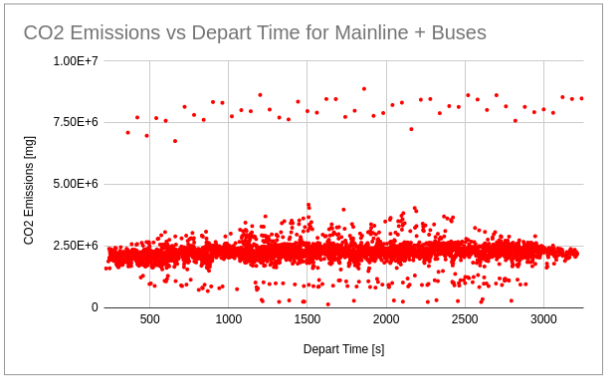
difference in throughput, 1.6% reduction in average travel time, and 3.09% reduction in average emissions over the no control scenario. Additionally, the 3639 drivers on the freeway experience a 3.26% reduction in travel time while the 123 drivers taking HOVs experience a





Average Travel Time: 461.5810739 s

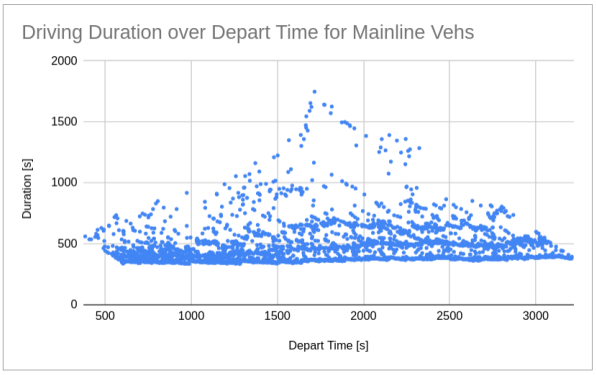
Avg Travel Time for Rerouted Vehs 696.1788618 s
Avg Travel Time for Nonrerouted Vehs 453.6515526 s

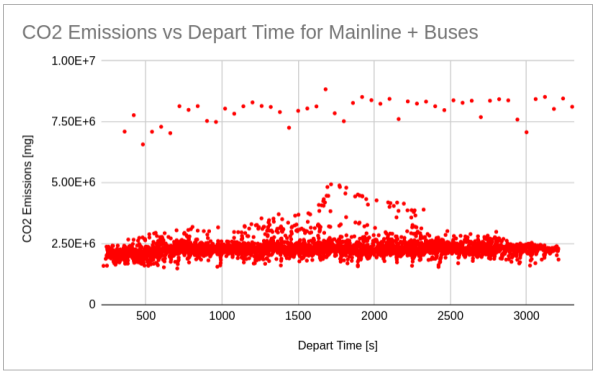


Summary	
Rerouted Vehicle Count	123
Nonrerouted Vehicle Count	3639
Total Throughput	3762

Average CO2 Emissions: 2346687.115 mg

Figure 8.6: Results for running the DRL agent on SUMO seed 27438





Average Travel Time: 468.9262427 s

Average CO2 Emissions: 2421470.953 mg

Total Throughput	3742

Figure 8.7: Results for running SUMO seed 27438 without DRL agent

48.46% increase in travel time over the no control scenario.

	DRL Agent vs No Control				
Seed	**Count**	**Avg Delay**	**Diverted Delay**	**Freeway Delay**	**Avg CO2**
64643	+3.99%	-2.69%	+47.34%	-4.67%	-2.97%
44435	+2.95%	-0.13%	+52.56%	-2.55%	-3.00%
27438	+0.53%	-1.6%	+48.46%	-3.26%	-3.09%

Table 8.1: Summary of ICM performance against no control

8.4 Discussion

We observe that our ICM approach provides marginal improvements to highway throughput, average delay, and average CO2 emissions. The findings are summarized in table 8.1. Further investigation into the optimal policy learned by the DRL agent shows that the agent primarily redirects traffic to the PAR structure from Interstate 610 and Lakewood Blvd. This makes sense because taking these two exits result in the least amount of travel time to reach the PAR (see table 7.2). Additionally, rerouting at I-610 may be especially beneficial because as observed in chapter 3, the merges at the I-610 junction are a heavy source of congestion.

We also find that average delay per passenger does not tell the whole story. Although the travel time for drivers on the corridor is improved, the travel time for drivers using the PAR services is drastically increased in all cases. In other words, by letting a few drivers suffer significantly greater travel delay by taking public transport, the other drivers on the freeway are allowed to get to the destination slightly faster. This is a significant trade-off to keep in mind for this ICM strategy, one which is consistent with the results of Ortega et al.'s study [25], who found that travellers utilizing PAR systems experience significantly increased travel time.

Our DRL agent most consistently improves average CO2 emissions per passenger. This is likely due to the fact that HOVs are not utilized in the uncontrolled scenario but still contribute to the overall emissions of the corridor, wasting resources. When passengers ride on an HOV, the CO2 cost for the vehicle per passenger decreases at a rate of $\frac{1}{N}$, rapidly

increasing the efficiency of the HOV. Modifying some simulation parameters could result in even greater HOV emission savings. For instance, a less frequent HOV schedule $\lambda_{HOV,j}$ would reflect more realistic public transportation schedules and allow more passengers to accumulate at the PAR before the next HOV leaves, but this would further sacrifice the travelers' delay during the extended waiting period. Another possibility is to set a higher reroute compliance probability α to reroute more vehicles to the PAR, but too high a value would be unrealistic; not many drivers are willing to reroute to a PAR. However, a higher α could be made a realistic assumption if drivers are piloting autonomous vehicles instead of just connected vehicles.

Chapter 9

Conclusion and Future Work

This work proposes a novel ICM strategy that redirects vehicles to underutilized park and ride structures to maximize freeway throughput and minimize CO2 emissions and travel time. This approach leverages the V2I capabilities of RSUs and OBUs to observe the state of connected vehicles on the freeway and to broadcast advisory messages to drivers to redirect them to the nearest park and ride structure. A centralized cloud server hosted at a Transporation Management Center communicates with the RSUs and uses deep reinforcement learning to process the observed congestion state of the corridor and choose where to broadcast PAR advisory messages.

We created a realistic corridor simulation based on Interstate 5 in the Los Angeles area to test the ICM strategy. The deep reinforcement learning agent converges to a strategy that redirects vehicles to the I-605 and Lakewood Blvd junctions, which can achieve marginal improvements in throughput, average travel time, and average emissions at the cost of significant travel delay for the few drivers taking an HOV. Specifically, we observe up to 3.99% increase in throughput, 4.67% reduction in freeway travel time, and 3.09% savings in CO2 emission savings, but with the cost of up to 52.56% additional delay for diverted drivers.

9.1 Future Work

In section 8.4, we recommend some parameter adjustments, such as decreasing the HOV schedule frequency and increasing driver compliance with PAR messages, to see if there are scenarios where rerouting vehicles to the PAR structure can create even more emissions savings. These changes could be translated into a study involving autonomous vehicles, where the system could have more control over the behavior of autonomous passenger vehicles and the timing of autonomous HOVs.

More studies are needed to see how this approach can be scaled up. The difficulty in this will be in observing and assigning rewards to the ICM actions; with our current experimental setup, a longer freeway means more time will pass before a passenger reaches the sink in an HOV, i.e. more time will pass before rewards reflect the new action. This increases simulation complexity and overall training time. One alternative could be to deploy multiple DRL agents for small sections of the freeway and coordinate them into a larger system.

Additionally, as with any system that aggregates data to make decisions, there are security concerns. An automotive security survey conducted in 2019 [38] explains that V2X communications opens up multiple new attack surfaces for vehicles in addition to the preexisting vulnerabilities in automotive electronic components. In particular, the authors find data spoofing to be a common attack method on V2I-based systems such as our ICM approach, resulting in increased traffic congestion. A 2022 study [39] develops an attack modeling methodology for a V2X Advisory Speed Limit traffic control scenario and establishes various metrics to assess the impact of an attack. A future research direction could be to develop a similar attack methodology and evaluation metrics for our V2I-based ICM approach and to test its resilience.

One could also adopt a more general security framework to analyze our ICM approach. In a previous study [40], the authors present a security analysis framework for cyber-physical

systems (CPS). By modeling cyber domain information, such as device firmware and application data, and physical domain information, such as RF signals and other side channels, as information flows, the authors show that applying data-driven algorithms can improve understanding of the cyber-physical domain relationships and reveal new vulnerabilities in the system. Moreover, our ICM approach can be classified as a networked control system (NCS) that can be modeled as generalized mathematical formula as demonstrated in [41]. Once the vulnerabilities and attacks of our ICM strategy are understood, they can be modeled and fed as input to the NCS model to study the system response to attacks over time.

Any proposed security solutions should focus on extensibility [42], i.e. solutions that are easily adapted to new use cases to support the still-developing automotive technology scene. Several previous works describe extensible security solutions for V2X communications. For instance, the authors in [43] and [44] propose novel methods for physical layer key generation that result in faster key generation time and reduced code size respectively; reduced computational resources means these cryptography methods can be implemented in more devices. In another study [45], the authors propose a blockchain-based architecture to validate a connected vehicle's location in V2I contexts, preventing position spoofing. These solutions can be incorporated into our ICM strategy.

Bibliography

[1] M. A. Al Faruque, M. Odema, and L. Chen, "Software and hardware systems for autonomous smart parking accommodating both traditional and autonomous vehicles," Pacific Southwest Region University Transportation Center (UTC), Tech. Rep. PSR-19-30, Apr. 2021. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/56844

[2] D. Schrank, B. Eisele, T. Lomax *et al.*, "Urban mobility report 2019," 2019.

[3] INRIX, "2021 global traffic scorecard," 2021. [Online]. Available: https://inrix.com/scorecard/

[4] F. H. Administration, "Integrated corridor management (icm) program: Major achievements, key findings, and outlook," https://ops.fhwa.dot.gov/publications/fhwahop19016/chapter3.htm, 2020.

[5] ——, "Integrated corridor management, transit, and mobility on demand," https://ops.fhwa.dot.gov/publications/fhwahop16036/ch1.htm, 2020.

[6] T. McGuckin, J. Lambert, D. Newton, A. Pearmine, and E. Hubbard, "Leveraging the promise of connected and autonomous vehicles to improve integrated corridor management and operations: A primer," https://ops.fhwa.dot.gov/publications/fhwahop17001/index.htm, US D.O.T Federal Highway Administration, Tech. Rep. FHWA HOP-17-001, Jan. 2017.

[7] V. C. T. Committee, *V2X Communications Message Set Dictionary*, jul 2020. [Online]. Available: https://doi.org/10.4271/J2735_202007

[8] K. Hall-Geisler, "All new cars could have v2v tech by 2023," 2017. [Online]. Available: https://techcrunch.com/2017/02/02/all-new-cars-could-have-v2v-tech-by-2023/

[9] N. H. T. S. Administration *et al.*, "Federal motor vehicle safety standards; v2v communications," *Federal Register*, vol. 82, no. 8, pp. 3854–4019, 2017.

[10] Caltrans, "Park & ride," 2023. [Online]. Available: https://dot.ca.gov/programs/traffic-operations/park-ride

[11] Transportation Research Board and National Academies of Sciences, Engineering, and Medicine, *Transit Supportive Parking Policies and Programs*, L. Jacobson and R. R. Weinberger, Eds. Washington DC: The National Academies

Press, 2016. [Online]. Available: https://nap.nationalacademies.org/catalog/23493/transit-supportive-parking-policies-and-programs

[12] A. Schmitt, "Park & rides lose money and waste land - but agencies keep building them," Jul 2016. [Online]. Available: https://usa.streetsblog.org/2016/07/05/park-rides-lose-money-and-waste-land-but-agencies-keep-building-them/

[13] F. Arena, G. Pau, and A. Severino, "A review on ieee 802.11p for intelligent transportation systems," *Journal of Sensor and Actuator Networks*, vol. 9, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2224-2708/9/2/22

[14] D. T. Committee, *On-Board System Requirements for V2V Safety Communications*, apr 2020.

[15] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, January 2011.

[16] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[17] A. Varga, *OMNeT++*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35–59. [Online]. Available: https://doi.org/10.1007/978-3-642-12331-3_3

[18] J. Hu, P. Bhowmick, F. Arvin, A. Lanzon, and B. Lennox, "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 977–984, April 2020.

[19] J. Shelton, J. Wagner, S. Samant, G. Goodin, T. Lomax, and E. Seymour, "Impacts of connected vehicles in a complex, congested urban freeway setting using multi-resolution modeling methods," *International Journal of Transportation Science and Technology*, vol. 8, no. 1, pp. 25–34, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2046043018300352

[20] W. Jin and M. Zhang, "Evaluation of On-ramp Control Algorithms," Institute of Transportation Studies, UC Berkeley, Institute of Transportation Studies, Research Reports, Working Papers, Proceedings qt1gz7w0wm, Apr. 2001. [Online]. Available: https://ideas.repec.org/p/cdl/itsrrp/qt1gz7w0wm.html

[21] A. Fares and W. Gomaa, "Freeway ramp-metering control based on reinforcement learning," *IEEE International Conference on Control and Automation, ICCA*, 06 2014.

[22] H. Hashemi and K. F. Abdelghany, "Real-time traffic network state estimation and prediction with decision support capabilities: Application to integrated corridor management," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 128–146, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X16302054

[23] H. Hashemi and K. Abdelghany, "End-to-end deep learning methodology for real-time traffic network management," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 10, pp. 849–863, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12376

[24] X. C. Liu, G. Zhang, C. Kwan, Y. Wang, and B. K. Kemper, "Simulation-based, scenario-driven integrated corridor management strategy analysis," *Transportation Research Record*, vol. 2396, no. 1, pp. 38–44, 2013. [Online]. Available: https://doi.org/10.3141/2396-05

[25] J. Ortega, J. Hamadneh, D. Esztergár-Kiss, and J. Tóth, "Simulation of the daily activity plans of travelers using the park-and-ride system and autonomous vehicles: Work and shopping trip purposes," *Applied Sciences*, vol. 10, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/8/2912

[26] X. Liu, N. Masoud, Q. Zhu, and A. Khojandi, "A markov decision process framework to incorporate network-level data in motion planning for connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 136, p. 103550, 2022.

[27] "L.A. County's I-5 named 'most congested freeway' in California," 2014. [Online]. Available: https://www.dailynews.com/2014/02/13/la-countys-i-5-named-most-congested-freeway-in-california/

[28] S. Carpenter, "Interstate 5 from Euclid Avenue to 605 is busiest corridor in the U.S." *Spectrum News 1*, 2021. [Online]. Available: https://spectrumnews1.com/ca/la-west/traffic/2021/12/07/interstate-5-from-euclid-avenue-to-interstate-605-is-busiest-corridor-in-the-u-s-

[29] "Los Angeles Has Top Two Worst Traffic Corridors in the Nation, Study Says ," 2020. [Online]. Available: https://www.nbclosangeles.com/news/local/los-angeles-has-top-two-worst-traffic-corridors-in-the-nation-study-says/2325974/

[30] "Traffic Census Program." [Online]. Available: https://dot.ca.gov/programs/traffic-operations/census

[31] E. Research, "Usrp hardware driver and usrp manual," https://files.ettus.com/manual/page_usrp_b200.html, 2023.

[32] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "Performance Assessment of IEEE 802.11p with an Open Source SDR-based Prototype," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1162–1175, May 2018.

[33] F. H. Administration, "Transportation management centers," https://ops.fhwa.dot.gov/freewaymgmt/trans_mgmnt.htm, 2020.

[34] Siemens, "Connected vehicle roadside unit (rsu)," https://www.mobotrex.com/product/siemens-connected-vehicle-roadside-unit/, 2018.

[35] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.06461

[36] M. Schettler, D. S. Buse, A. Zubow, and F. Dressler, "How to Train your ITS? Integrating Machine Learning with Vehicular Network Simulation," in *12th IEEE Vehicular Networking Conference (VNC 2020)*. Virtual Conference: IEEE, 12 2020.

[37] D. of Traffic Operations, "Ramp metering design manual," https://dot.ca.gov/programs/traffic-operations/ramp-metering, California Department of Transportation, Tech. Rep., 2022.

[38] A. Lopez, A. V. Malawade, M. A. Al Faruque, S. Boddupalli, and S. Ray, "Security of emergent automotive systems: A tutorial introduction and perspectives on practice," *IEEE Design & Test*, vol. 36, no. 6, pp. 10–38, 2019.

[39] A. B. Lopez, W.-L. Jin, and M. A. A. Faruque, "Attack modeling methodology and taxonomy for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13 255–13 264, 2022.

[40] S. R. Chhetri, J. Wan, and M. A. Al Faruque, "Cross-domain security of cyber-physical systems," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2017, pp. 200–205.

[41] A. Sargolzaei, A. Abbaspour, M. A. Al Faruque, A. Salah Eddin, and K. Yen, *Security Challenges of Networked Control Systems*. Cham: Springer International Publishing, 2018, pp. 77–95. [Online]. Available: https://doi.org/10.1007/978-3-319-74412-4_6

[42] S. Ray, W. Chen, J. Bhadra, and M. A. Al Faruque, "Extensibility in automotive security: Current practice and challenges: Invited," in *Proceedings of the 54th Annual Design Automation Conference 2017*, ser. DAC '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3061639.3072952

[43] S. Ribouh, K. Phan, A. V. Malawade, Y. Elhillali, A. Rivenq, and M. A. A. Faruque, "Channel state information-based cryptographic key generation for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7496–7507, 2021.

[44] J. Wan, A. Lopez, and M. A. A. Faruque, "Physical layer key generation: Securing wireless communication in automotive cyber-physical systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 2, oct 2018. [Online]. Available: https://doi.org/10.1145/3140257

[45] A. Didouh, A. B. Lopez, Y. E. Hillali, A. Rivenq, and M. A. A. Faruque, "Eve, you shall not get access! a cyber-physical blockchain architecture for electronic toll collection security," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7.