# Eco Motion Control for Connected and Automated Vehicles in Smart Cities

by

Yeojun Kim

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Francesco Borrelli, Chair
Professor Scott Moura
Professor Roberto Horowitz
Professor Anil Aswani

Fall 2021

**Eco Motion Control for Connected and Automated Vehicles in Smart Cities**

# Abstract

Eco Motion Control for Connected and Automated Vehicles in Smart Cities

by

Yeojun Kim

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Chair

Connected and Automated Vehicles (CAVs) improve driving automation by exploiting evolving connectivity technologies which enable communication among vehicles (V2V), communication between vehicles and the road infrastructure such as traffic lights (V2I), and communication between vehicles and the cloud (V2C). The goal of this dissertation is to present novel control algorithms for CAVs which harness the untapped CAV potential of remote computations, forecasts, historical data, automation and coordination with other vehicles and infrastructure. First, we introduce a control and planning system architecture suitable for CAVs with focus on improving the energy efficiency of the vehicle. Then, for each level of the architect, we present planning and control algorithms for the longitudinal motion and powertrain control exploiting the connectivity, and provide a set of thorough simulations and experiments aimed at quantifying energy efficiency improvements for automated cars and railways. The experimental results are complemented with a theoretical and simulation analysis of the stability and feasibility of the closed loop system under nominal and perturbed conditions. By using the proposed control methodology, our experimental results show that safe and energy efficient driving in both arterial and highway settings can be achieved. In particular, we achieve up to 20% energy savings on both settings.

I lovingly dedicate this dissertation to everyone who has supported me throughout this journey. I have received great support and assistance which have been integral and indispensable for the preparation and writing of this dissertation.

I would first like to thank my supervisor, Professor Francesco Borrelli. He has guided me to formulate the research questions and methodology in a right direction. His advice, insightful feedback, and direction always helped me find ways to resolve problems I have encountered while conducting my research. Moreover, he always made sure I have the best available tools and resources to conduct research in his Model Predictive Control lab. Thanks for his strong support, I was never constrained to experiment with my ideas. I am very thankful that he is my supervisor and I had a very fortunate opportunity to work in his lab.

Also, I would like to thank my dissertation committee members, Professor Scott Moura, Professor Anil Aswani, and Professor Roberto Horowitz. I really appreciate their times and thoughts spent on reviewing my dissertation and giving me valuable feedback which has made my dissertation stronger.

I would like to acknowledge my colleagues from the Model Predictive Control lab and my internship at AV-Connect Inc for our wonderful collaboration. I was truly fortunate to have opportunities to work with them and develop my research with them. In particular, I want to thank Dr. Jacopo Guanetti for his collaboration and guidance which have led me to make fruitful research outcomes. Also, I want to thank Mr. Samuel Tay, an undergraduate and Master's student researcher, who has contributed a lot to my research.

Friends also deserve deep gratitude. My friends from soccer teams, the Korean community, church, and neighbors helped me stay encouraged and motivated all the time. They brought more fun to my life and made my stay at Berkeley more joyful. I would particularly like to thank Eric Choi, Kimin Lee, Kiwoo Shin, Kyle Sefraoui, Monimoy Bujarbaruah, and Jacopo Guanetti for our times together.

Last but not least, I would like to thank my family: my father Sanggil Kim, my mother Kyungmi Lee, my brother Yeomyung Kim, my sister-in-law Sooyoung Choi, and my girlfriend Minyoung, and her family. My father taught me to always strive to become a better and eager person and yet making sure to enjoy life, my mother always made me feel confident and proud of myself and she would be the best listener when I have any concerns. My brother and sister-in-law also listened to my concerns, encouraged me, and gave me insightful advice. My girlfriend Minyoung is the person who stayed closest to me physically and mentally during my study at Berkeley for more than five years. She helped me overcome many problems in my life by always staying on my side and being my life teammate. Thanks for her, I successfully completed my dissertation. I thank her for staying with me during this long time and made this journey more fun and meaningful and less troublesome. She, with her family, always encouraged me and I really appreciate them. I look forward to our future together. To my and her families in particular, I dedicate this dissertation.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation and Background

There are millions of passenger vehicles driving in the U.S. everyday. Despite their utility and convenience, their operation often causes problems in the society. According to the statistics from the National Highway Traffic Safety Administration (NHTSA), there were 36,096 fatalities from motor vehicle traffic crashes in the U.S. in 2019; i.e., 1.10 fatalities per 100 million vehicle miles traveled [1]. Additionally, transportation systems, in particular passenger cars and light trucks, contribute a major amount to the US energy consumption and greenhouse-gas emissions [2], deteriorating urban air quality and wasting energy and resources. Moreover, more vehicles on the road aggravates traffic congestion. It is reported that commuters wasted an average of 54 hours a year stalled in the U.S. traffic in 2019 [3].

Autonomous vehicles (AVs) pledge solutions to the aforementioned problems; i.e., reducing vehicle accidents and energy consumption and improving traffic throughput. For this reason, AVs have been the subject of great research to the academic community and the industrial sector in recent years. With continuous improvements in control theories, automation, perception technologies and computational power, autonomous driving features are being slowly introduced in everyday life. The U.S. Department of Transportation (DoT) standard uses SAE International's six levels of driving automation, ranging from no driving automation (level 0) to full driving automation (level 5) [4]. The most recent commercial deployment of the AV technology is represented by the Driving Assistance System (DAS) which aims to assist drivers in performing common driving tasks and maneuvers safely [5, 6]. Examples of the DAS systems include Adaptive Cruise Control (ACC), lane keeping and parking assistance, autonomous emergency braking, etc. These features belong to the level 2 driving automation. On the other hand, full driving automation without human intervention (level 5) is an on-going research which will require a more sophisticated and innovative technologies in sensing, automation, and possibly, connectivity.

Vehicle connectivity has also been actively developed in the past decades. While some efforts were made for providing passengers with convenience features and infotainment services, including emergency calls, toll payment, and integrated media content (audio, video, data) [7], connectivity has also emerged as a technology to improve safety and performance of AVs. For instance, the Dedicated Short Range Communication (DSRC) technology has established highly secure, high-speed direct communication between vehicles (V2V) and with road infrastructure such as traffic lights (V2I). Moreover, cloud connectivity (V2C) enables direct access to databases, forecasts, and remote computations from vehicles.

Connected and Automated Vehicles (CAVs) refer to vehicles equipped with driving automation technology which uses vehicle connectivity. CAVs exploit communication technologies to advance driving automation and create multiple opportunities to improve the efficiency of vehicles in the future. In the past few decades CAV applications have been studied and implemented. In the 1990s, the California PATH program demonstrated the cooperative driving of a group (or *platoon*) of connected vehicles on a highway to increase traffic throughput [8, 9]. The Grand Cooperative Driving Challenges in the Netherlands in 2011 and 2016 also demonstrated platoon control and other interactions among vehicles in different traffic situations [10, 11]. Other cooperative driving demonstrations also occurred in the Demo 2000 cooperative driving in Japan [12] and the SARTRE program [13]. Moreover, [14] proposed a concept of *shared perception* in which the V2V communication enables CAVs to share their location and the targets detected by their on-board perception systems and to create an augmented awareness in which the size of the occluded regions is reduced.

In all of the above works, improving energy efficiency of the vehicles is not one of the main goals. Instead the focus is on improving safety and road throughput and technical feasibility and implementation details of CAVs. This thesis focuses on designing advanced algorithms to improve energy efficiency of CAV by leveraging exchange of historical data and forecasts and remote computations. The contributions of this thesis can be broadly summarized as follows:

1. A model predictive control is presented which exploits stored data, forecasts, and remote computations offered by the connectivity (V2V, V2I, and V2C) for advanced longitudinal motion and powertrain control of CAVs, with the focus on improvement of energy efficiency.

2. Detailed results of on-road experiments are presented which demonstrate the technical feasibility and effectiveness of the proposed control algorithms.

3. Extension of our work to shared perception and connected and automated railway systems is presented.

Figure 1.1 depicts examples of the CAV opportunities in smart city (or intelligent transportation systems), including (1) longitudinal motion and powertrain control with V2C and V2V, (2) compact platooning, (3) enhanced approach and departure to signalized intersections, (4) shared perception, (5) coordinated movements, (6) eco-routing, and (7) virtual coupling for railway. Each depicted application was developed by the author of this dissertation and is presented in this thesis except for (6) eco-routing which is presented in [15].

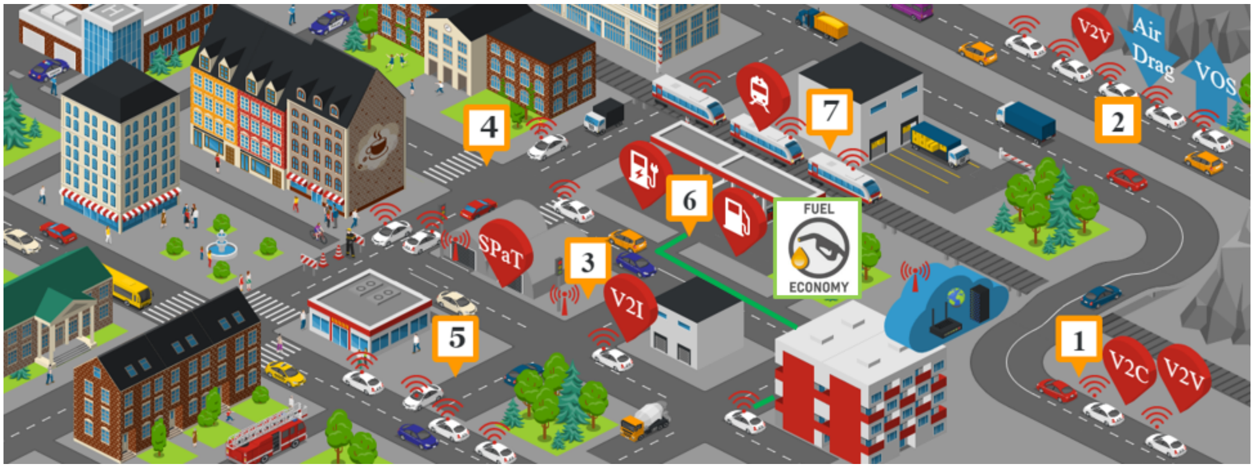


Figure 1.1: Cartoon illustrating a variety of intelligent transportation systems enabled by connected and automated vehicles (CAVs). (1) Longitudinal motion control with V2C and V2V, (2) compact platooning, (3) enhanced approach and departure to signalized intersections, (4) shared perception, (5) coordinated movements, (6) eco-routing, and (7) virtual coupling for railway (Created on `https://icograms.com`).

## 1.2 Thesis Outline

The thesis is organized as follows.

Chapter 2 details our designed and implemented software and hardware CAV experimental platform. This platform is then used by the control designs presented in the following chapters. In this chapter, we first address the CAV control and planning system architecture and describe the role of each functional block therein. Then, we describe our experimental setups for test vehicles, novel hardware-in-the-loop design, and real-world experiments.

Chapter 3 presents vehicle dynamics and energy consumption modeling. We identify the parameters for the longitudinal dynamics model of our test vehicle using collected data and characterize experimentally the relationship between inter-vehicular gap, vehicle speed, and (reduction of) energy consumption for a compact plug-in hybrid electric vehicle (our test vehicle).

Chapter 4 and 5 details the applications (1) longitudinal motion control with V2C and V2V, (2) compact platooning, and (3) enhanced approach and departure to signalized intersections in Figure 1.1. We present different energy saving longitudinal motion control designs which exploit V2V and V2I or V2C connectivity, respectively. The presented control designs are based on the vehicle dynamics and energy consumption models presented in Chapter 3 and are validated using the platform presented in Chapter 2.

In Chapter 6 and 7, we shift our focus to other CAV applications. Chapter 6 details the application (4) shared perception in Figure 1.1 and presents and experimentally demonstrates shared perception of CAVs to improve environment awareness. Chapter 7 details the application (7) virtual coupling for railway in Figure 1.1 and presents the automated railway (train) control using connectivity and shows by simulation that the proposed control design can substantially reduce headway and distance between trains and, therefore, increase railway capacity while guaranteeing safe separation between two consecutive trains.

Appendix refers to (5) coordinated movements in Figure 1.1 and presents a micro-platoon management system we developed for safe platoon demonstration on road. Our system is based on a finite-state-machine which uses a V2V communication to ensure that every member in the platoon moves in coordinated manners (ensuring every vehicle in the communication agrees to be the member of the platoon with the agreed desired speed and destination, and etc).

## 1.3 List of Publication and Video Demonstration

The results and discussions presented in this dissertation have appeared in conference proceedings and journal publications co-authored by the author of the dissertation, in particular:

- Chapter 2 is based on the following publications:
  - Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. "Control of connected and automated vehicles: State of the art and future challenges". In: *Annual Reviews in Control* 45C (2018), pp. 18–40
  - Yeojun Kim et al. "Hardware-In-the-Loop for Connected Automated Vehicles Testing in Real Traffic". In: *14th International Symposium on Advanced Vehicle Control*. 2018
- Chapter 3 is based on the following publications:
  - Yeojun Kim, Jacopo Guanetti, and Francesco Borrelli. "Compact cooperative adaptive cruise control for energy saving: Air drag modelling and simulation". In: *IEEE Transactions on Vehicular Technology* 70.10 (2021), pp. 9838–9848
- Chapter 4 is based on the following publications:
  - Yeojun Kim, Jacopo Guanetti, and Francesco Borrelli. "Robust Eco Adaptive Cruise Control for Cooperative Vehicles". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 1214–1219
  - Valerio Turri et al. "A model predictive controller for non-cooperative eco-platooning". In: *American Control Conference (ACC), 2017*. IEEE. 2017, pp. 2309–2314
- Chapter 5 is based on the following publications:
  - Yeojun Kim et al. "Eco-driving with learning model predictive control". In: *14th International Symposium on Advanced Vehicle Control*. 2018

  - Sangjae Bae et al. "Design and implementation of ecological adaptive cruise control for autonomous driving with communication to traffic lights". In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 4628–4634
  - Sangjae Bae et al. "Real-time ecological velocity planning for plug-in hybrid vehicles with partial communication to traffic lights". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 1279–1285

- Chapter 6 is based on the following publications:
  - Yeojun Kim et al. "Shared Perception for Connected and Automated Vehicles". In: *The 31st IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020

- Chapter 7 is based on the following publication:
  - Jesus Felez, Yeojun Kim, and Francesco Borrelli. "A model predictive control approach for virtual coupling in railways". In: *IEEE Transactions on Intelligent Transportation Systems* 20.7 (2019), pp. 2728–2739

- Appendix is based on the following publication:
  - Stanley W Smith et al. "Improving Urban Traffic Throughput With Vehicle Platooning: Theory and Experiments". In: *IEEE Access* 8 (2020), pp. 141208–141223

The works presented in this thesis have been appeared in the following demonstrations [27, 28, 29, 30, 31, 32].

# Chapter 2

# Experimental Platform for Connected and Automated Vehicles

This dissertation mainly studies the control strategies for safe and energy efficient control of Connected and Automated Vehicles (CAVs). The proposed control designs are then analyzed and shown effective using the numerical results from not only pure software-based simulation with data-driven models but also hardware-in-the-loop (HIL) simulation and real-world experiments. This section discusses the experimental platforms used in the dissertation, i.e., the CAV control and planning architecture, the experimental vehicle, the novel hardware-in-the-loop simulation setup for CAV testing, and real-world experiment setups.

## 2.1 Introduction

The development of the CAV technologies is matured by the recent advances in sensing and connectivity technologies. In particular, CAVs have the potential to improve vehicle safety, energy efficiency, and traffic flow [16]. In order to objectively validate safety and performance improvements, CAVs require i) fully instrumented CAVs with proper control and perception systems and ii) extensive testing under complex traffic scenarios as closely as possible to the real-world driving conditions. However, meeting these two requirements can be challenging, expensive, unsafe, and are often impossible.

Designing fully instrumented CAVs is challenging as it requires additional connectivity technology in conjunction with what is already required by building automated vehicles. [6] and [33] present the architecture of the automated vehicles (AVs) with hardware requirements. Figure 2.1 depicts the instance of the system architecture for AVs. As seen, the AV system architecture exploits the data from GPS, Radar, Lidar, and Camera to build the perception system. Using this data, the perception system creates awareness and prediction of the surrounding environment and send this information to the real-time control and planing system which performs all the real-time computations for vehicle motion planning and control and powertrain control and interfaces to the vehicle powertrain and steering actuators.

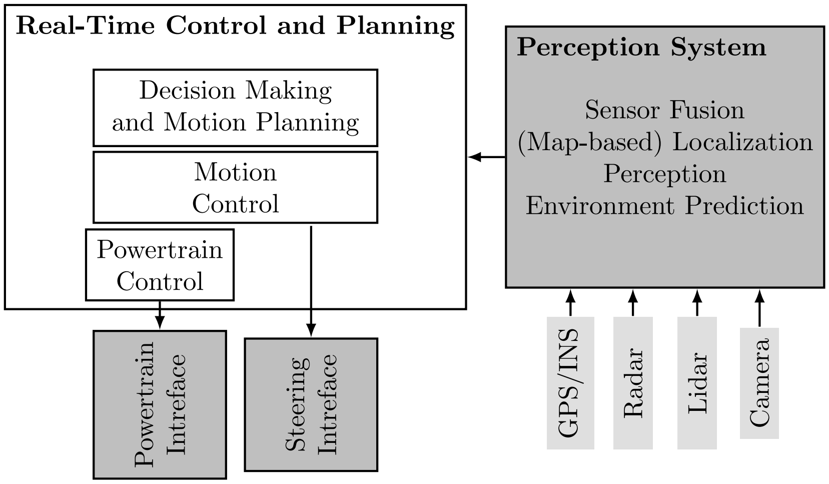


Figure 2.1: Control and Planning Architecture for Automated Vehicles

CAVs add the additional communication module and associated functionality blocks to this AV system to advance the real-time control and planning block.

Even with the fully instrumented CAV, extensive testing under complex traffic scenarios as closely as possible real-world driving conditions is required to validate the safety and energy efficiency of the developed CAV technologies. To mitigate the issues and risks with extensive testing, different levels of experimental platforms can be utilized, ranging from pure software-based simulation, HIL simulation, and real-world experiments.

In this chapter, the control and planning architecture for CAVs is introduced. Then, we describe the experimental CAV used throughout the dissertation and explain how our CAV instrumentation fulfills the performance requirements of the functionality blocks in the CAV architecture. Then, the novel HIL setup, suitable for CAV control and planning algorithms testing and validation in complex environments, is introduced. Lastly, the real-world experiment setup is discussed.

## 2.2 CAV Control and Planning Architecture for More Energy Efficient and Safer Operation

Figure 2.1 shows the control and planning architecture for AVs. Adding the communication capability, CAVs can achieve more energy efficient and safer vehicle operation than AVs without communication. This section describes how our CAV system architecture differs from the AV system architecture and details the tasks of each functionality block therein.

Figure 2.2 depicts a control architecture for CAVs focused on safe and energy efficient operation. In addition to what is already in the AV stack in Figure 2.1, the presented CAV architecture includes an additional communication module such as “DSRC” and a “Remote

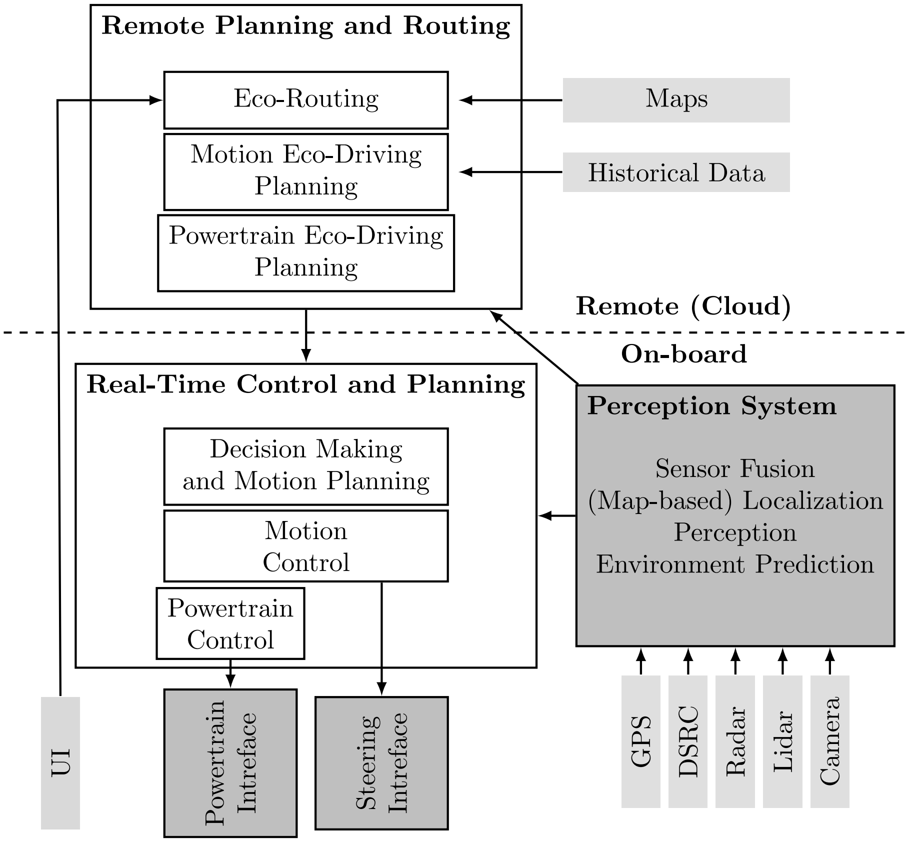


Figure 2.2: Control and planning architecture for connected and automated vehicles

Planning and Routing" functional blocks. Each functionality block in our CAV architecture is described below:

- The "Perception + Localization module combines data received from Radar/Camera/Lidar sensors, GPS, and a communication module such as DSRC to perceive and even predict the driving situation (e.g. position of surrounding/communicating vehicles, road geometry, approaching traffic light signal phase and timing).

- The "Remote Planning and Routing" block collects data from Maps, Historical Data, and the "Perception System" (for GPS) and perform three tasks, *Eco-Routing*, *Motion Eco-Driving Planning*, and *Powertrain Eco-Driving Planning*. The *Eco-Routing* computes the most energy efficient route given the current location. The *Motion Eco-Driving Planning*, given the current location and the upcoming route, computes the most energy efficient velocity reference which is then used by the motion planning and control in the "Real-Time Control and Planning". The *Powertrain Eco-Driving Planning*, given the current location and the upcoming route, computes the optimal powertrain control policy learned from historical data of the same route [34]. This policy is then used by the powertrain controller in the "Real-Time Control and Planning".

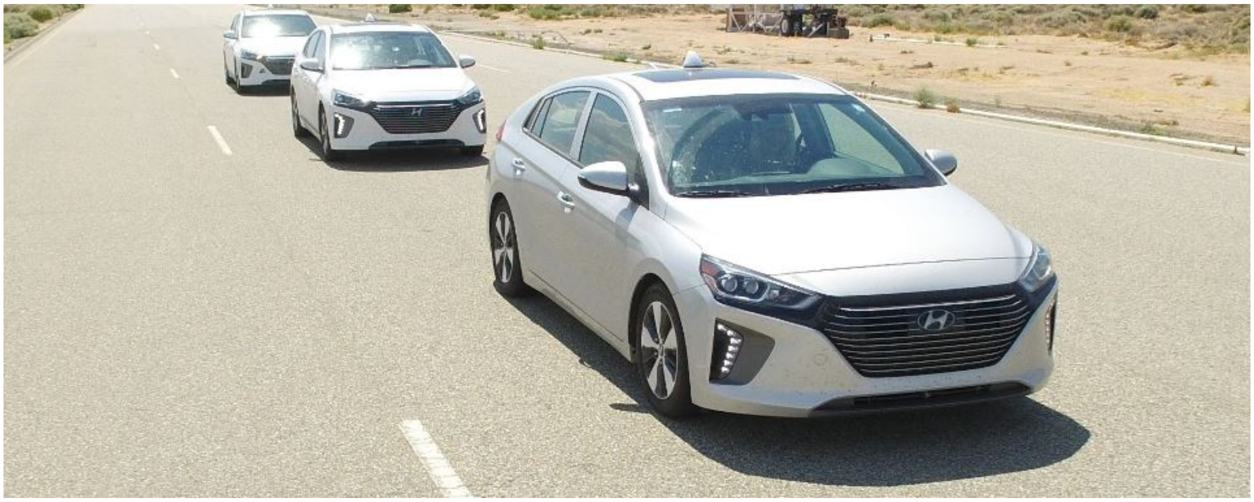


Figure 2.3: A picture of Ioniqs

- The “Real-Time Control and Planning” block collects data from “Perception System” and “Remtoe Planning and Routing” and performs three tasks, *Decision Making and Motion Planning*, *Motion Control*, and *Powertrain Control*. The *Decision Making and Motion Planning* block includes maneuver planning (e.g. decision to stay in a lane or change), path planning, and trajectory planning. The *Motion Control* block regulates the longitudinal and lateral motion of the vehicle by computing the total demand wheel torque and steering angle, and is interfaced to the powertrain control and the steering system of the vehicle. It ensures that the vehicle executes the motion plan with improved safety and, possibly, energy efficiency. The *Powertrain Control* receives the total demand wheel torque and regenerative and hydraulic braking torques from the motion control block, and safely distributes them to the actuators (e.g. internal combustion engine, electric motor/generator, hydraulic brake actuator).

## 2.3 Experimental Vehicle

The main test vehicle platform used in this dissertation is a 2017 Hyundai Ioniq depicted in Figure 2.3. Its powertrain is plug-in hybrid electric with a pre-transmission parallel architecture [35] as depicted in Figure 2.4. It consists of an internal combustion engine (E), electric motor (M), high voltage battery (B), clutch unit (C), transmission unit (T) and load from longitudinal dynamics (W). Thick lines between components represent electrical connections whereas dotted connections represent mechanical connections. The details about longitudinal vehicle dynamics and energy consumption modeling are discussed in Chapter 3.

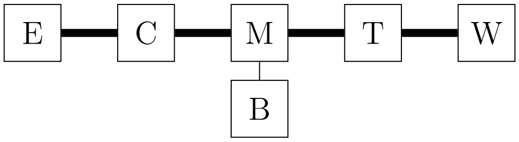


Figure 2.4: Architecture of a pre-transmission (or single-shaft) parallel hybrid electric powertrain

### 2.3.1 Experimental Vehicle Hardware Setup

CAVs require extensive vehicle instrumentation in order to i) exploit the communication technology and ii) execute motion planning and control using the real-time data from not only the on-board sensors but also the communication module, as illustrated in Figure 2.2. Moreover, this dissertation focuses on energy saving aspect of CAV technologies. For these reasons, it is critical to properly equip the test vehicle with high precision energy measurements and the high performance Electronic Computation Unit (ECU) which will allow real-time execution of motion planning and control with both sensor and communication data.

Figure 2.5 depicts our test CAV hardware instrumentation. For environment sensing, our vehicle has a front facing Mando camera and a front facing Delphi ESR radar. Moreover, to communicate with other CAVs and infrastructure using an On-board Unit (OBU) and Roadside Unit (RSU), respectively, our vehicle uses a Cohda MK5, which supports Dedicated Short Range Communication (DSRC) and GPS information (GNSS 2.5 meters accuracy [36]). The MK5 Unit allow us to exchange DSRC messages according to the SAE J2735 Message Set Dictionary [37], with some customization in order to communicate target information in the Target table. For high precision measurement, our vehicle is instrumented with a high accuracy fuel flow meter, and high accuracy current sensors (ETAS DAQ) to measure the high voltage battery current, the motor current, the starter/generator current, and the current for auxiliaries.

CAV control and planning algorithms exploit a large amount of data, such as sensor data, communication data, and historical data. Moreover, some algorithms, especially the ones developed in this dissertation, are based on optimal control principles, such as Model Predictive Control, which can demand high computational power. To mitigate these issues, our vehicle is also instrumented with two computation units to accept data inputs from various sources and to distribute the computation load. The two computation units are a dSpace MicroAutobox, which consists of an IBM PowerPC processor capable of running at 900 MHz, and a Matrix embedded PC-Adlink (MXC-6101D/M4G with Intel Core i7-620LE 2.0 GHz processor). Moreover, our Matrix embedded PC-Adlink has a WiFi Internet connection to enable the communication with cloud.

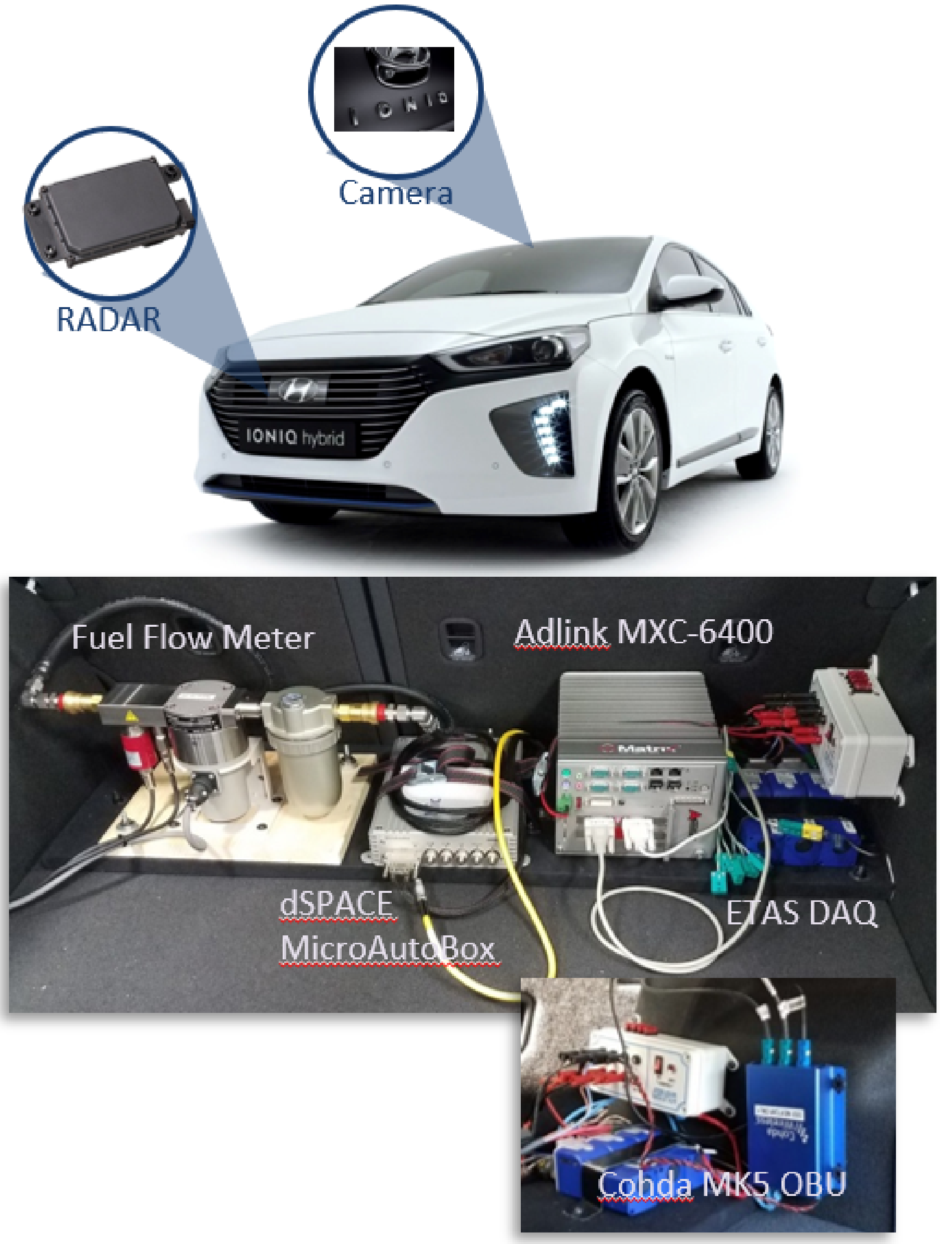


Figure 2.5: Hardware instrumentation for connected and automated vehicles

## 2.4 Hardware-In-the-Loop Setup for CAVs

We present a hardware-in-the-loop (HIL) simulation setup for repeatable testing of Connected Automated Vehicles (CAVs) in dynamic, real-world scenarios. The main benefits of our HIL setup are threefold. First, we exploit advanced simulations in CAVs by using complex traffic simulations modeled with experimental and collected data. Second, our HIL design is capable of Vehicle-to-Everything (V2X) communication and cloud connectivity. Lastly, our real-world test vehicle and other simulated vehicles in the traffic network are mutually interactive in the simulation environment.

### 2.4.1 HIL Hardware Setup

In general, HIL includes different hardware components depending on the availability of their high-fidelity models. In our HIL setup, we replace environment and, sometimes, a test vehicle with simulators while the on-board ECUs with connectivity (OBU and RSU) are the same as in the real-world testing, i.e., the hardware setup described in section 2.3. Figure 2.6 depicts our novel HIL hardware setup which mainly consists of a desktop computer, a dedicated vehicle dynamic and powertrain simulator (or a test vehicle on a chassis dyno if the lateral vehicle motion can be neglected in testing), and on-board ECUs.

The desktop computer, which has an Intel(R) Core(TM) i7-7700KK CPU @ 4.20Hz with NVIDIA GeForce GTX 1080, runs the real-time environment simulation software and mirrors a test vehicle from the real-time vehicle data from a dedicated vehicle dynamic and powertrain simulator/a test vehicle on a dyno. Note that communications among the desktop computer, the on-board ECUs, and a vehicle dynamic/powertrain simulator are via CAN bus. In order to link a desktop computer to the CAN bus, a Vector CAN interface is used.

For a dedicated vehicle dynamic and powertrain simulator, there are three different options. First, the vehicle on a dyno can simply be used. This is the most accurate method to obtain the vehicle dynamics and powertrain measurements but using a dyno can be costly and sometimes impossible. Second option, which is less demanding, is to use use an ETAS LABCAR RTPC (ES5100.1 with Intel Core i7-4770S@3.1GHz and 4 CAN Bus interface), allowing for the high-fidelity vehicle dynamics and powertrain models in our HIL simulation. In both first and second options, the interfaces with the on-board ECUs are established via the CAN bus. Third option, which is the most cost efficient yet least accurate method to obtain the vehicle dynamics and measurements, is to use the Simulink models of vehicle dynamics/powertrain. In fact, even OBU and RSU can be replaced by the DSRC modules offered by PrreScan. Figure 2.7 depicts the simplified HIL hardware setup.

### 2.4.2 HIL Software Setup

High fidelity environment and/or vehicle simulators (or vehicle on a dyno) are employed to represent various complex real-world scenarios and test vehicle dynamics. Figure 2.8 depicts a schematic of the HIL software, which mainly consists of the PreScan and Vissim simulators,

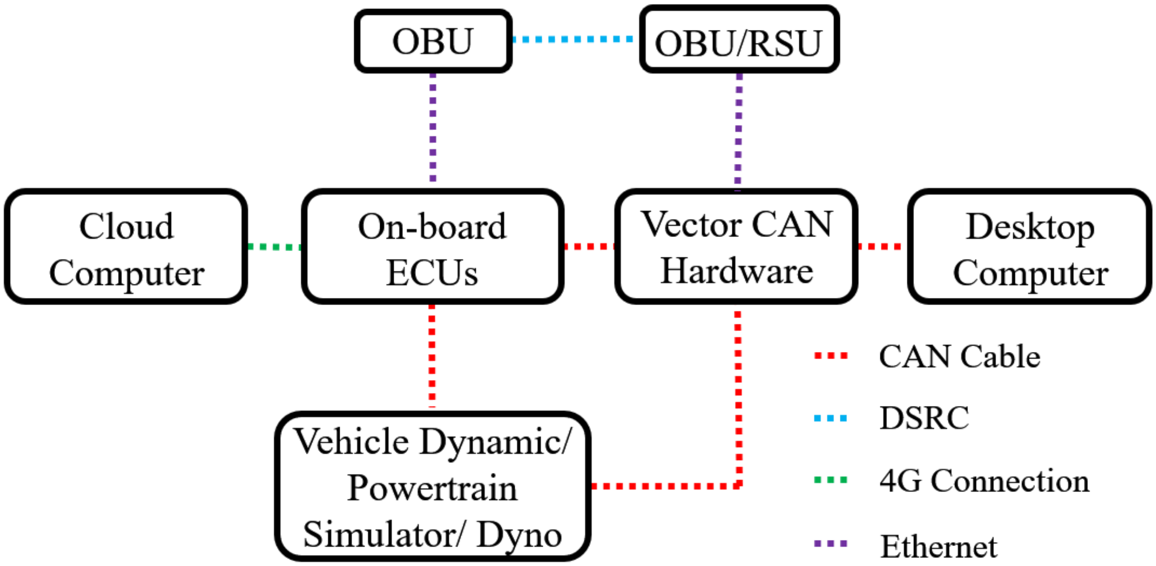


Figure 2.6: HIL hardware setup schematic

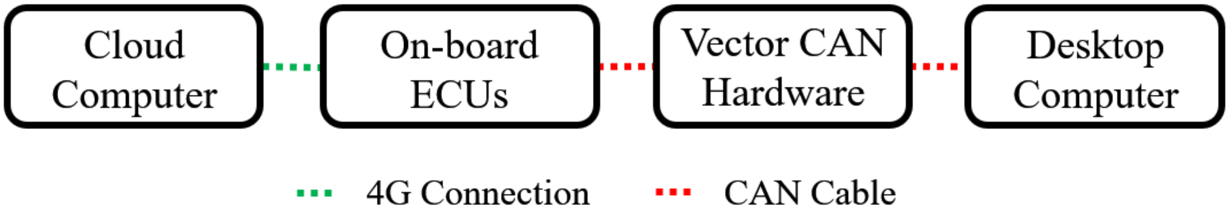


Figure 2.7: Simplified HIL hardware setup schematic

ECU software, Amazon AWS, and ETAS/Simulink/Vehicle on a dyno. Only PreScan and Vissim simulators run in the desktop computer while others are operating on their own designated hardware from Figure 2.6. Next we describe how each software plays a role in building environment, vehicle and powertrain dynamics, and controller.

### Environment

The environment simulation includes the roads, traffic infrastructure, vehicles in the traffic network, and other external influences from the environment. The PreScan software serves as the main software to build the environment and also as an interface among all other software via its interface with Matlab Simulink [38]. It also offers the models for the perception sensors such as Radar and Camera sensors. Moreover, PreScan also has a Vissim

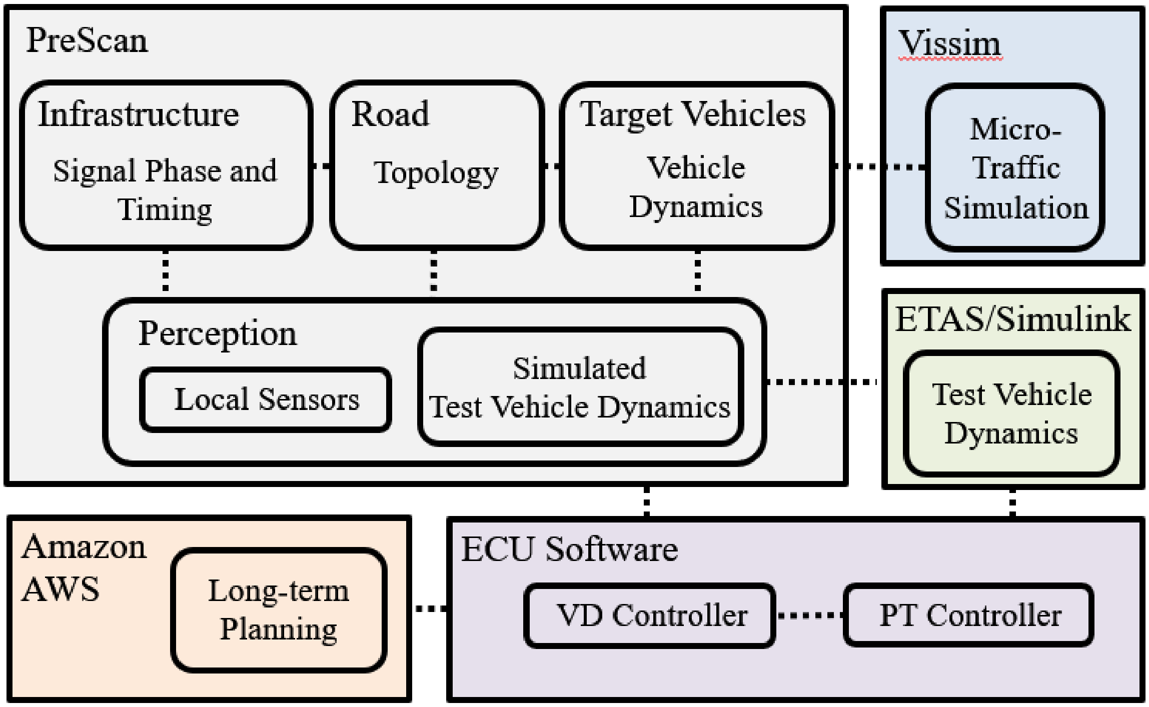


Figure 2.8: HIL software setup schematic

integration plug-in that enables microscopic traffic simulation within the PreScan simulation environment by mirroring vehicles generated by Vissim in Prescan and vice-versa [39]. Experimentally collected traffic data, which provide aggregate traffic volume and turn counts at each intersection, as well as raw traffic signal data is used to model traffic flow and level. Within Vissim, vehicles are injected into the road network and make turns at each intersection according to a probabilistic turn policy based on this data while these injected vehicles get mirrored in the Prescan environment as well.

### Vehicle and Powertrain Dynamics

We use either Dyno or ETAS software or Simulink to model the test vehicle and powertrain dynamics. Note that the test vehicle dynamics is controlled by the control input received from the ECUs. Then, the test vehicle dynamics is mirrored inside the Prescan environment. As a result, in the PreScan virtual environment, the other simulated traffic vehicles react to the mirrored test vehicle. In return, the perception sensors, equipped on the mirrored test vehicle in the PreScan virtual environment, offer the real-time sensor data to our ECUs.

**Controller**

Our controller implementation is the same as what is used for the real-world testing (see section 2.3). In our HIL setup, instead of using the actual sensor data, the controller receive, in real time, the vehicle and perception sensor CAN data from vehicle simulator (or vehicle on a dyno) and PreScan, respectively. Using these real-time data, our controller computes the vehicle dynamic/powertrain control action and send it to the vehicle simulator in ETAS software.

## 2.5 Real-World Setup for CAVs

This dissertation also presents results collected from the real-world experiment. Real-world experiments were conducted when either to demonstrate the practical implementation and effectiveness of the proposed controller in a real-world setting or to collect data using a real, full-scale test vehicle for vehicle dynamics and energy consumption modeling.

### 2.5.1 Closed Track Setup

Closed track experiments were conducted at the Hyundai-Kia California Proving Ground (CPG). In particular, the track used is a 10 km long, oval test track with the curvature always less than $500\,\mathrm{m}^{-1}$. The facility also includes an apparatus to measure the wind speed and direction.

### 2.5.2 Public Road Setup

Public road experiments were conducted on either a highway or an urban arterial road. In particular, the urban arterial road used in this dissertation is an approximately 2.5km urban corridor (the Live Oak Avenue) which consists of 8 adaptive traffic lights (phase duration is not fixed but adapt in real time to the current demand), located in the city of Arcadia illustrated in Figure 2.9. Each 8 traffic light is instrumented with an RSU which sends to our test vehicle the Vehicle-to-Infrastructure V2I information. The RSU installment in the real world is depicted in Figure 2.10. The V2I information includes the Signal Phase and Timing (SPaT) information which describes the current light color and the estimated remaining until the next change of color and the MAP information which includes the lane-level geography of the intersection. Note that using the GPS information of the vehicle, we could obtain the distance to the traffic light.

## 2.6 Conclusion

In this chapter, we discussed the experimental platforms used in the dissertation. First, we proposed the CAV planning and control system architecture and described the experimental

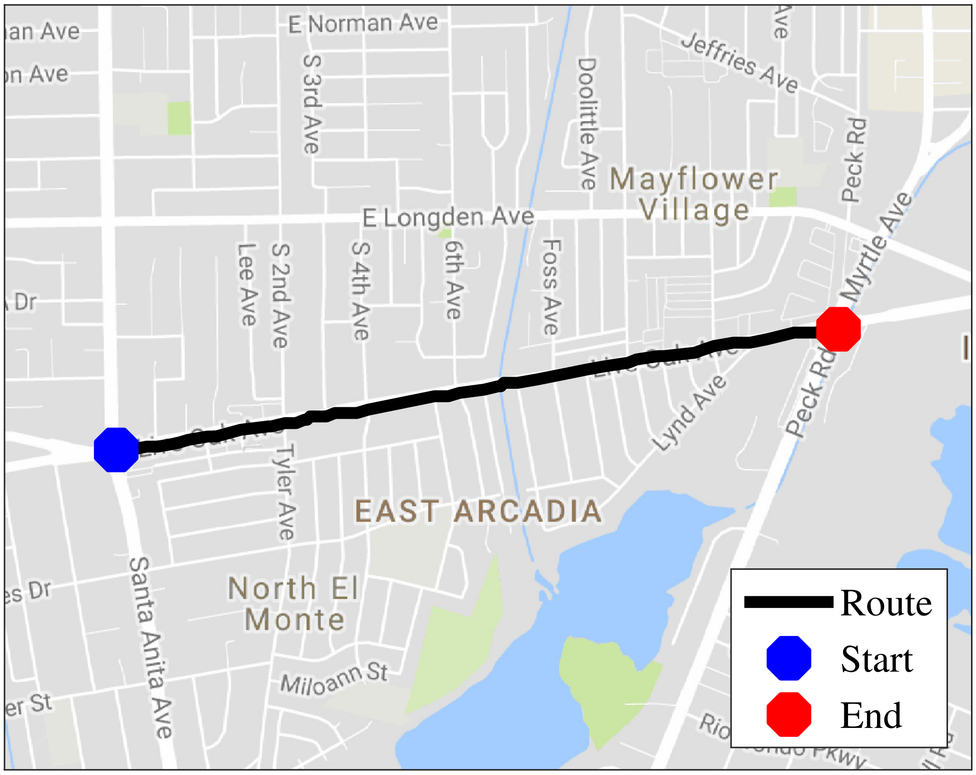


Figure 2.9: Route in urban road in Arcadia

vehicle setup to satisfy the requirements for the proposed CAV system architecture. Then, we discussed the novel hardware-in-the-loop simulation setup we have developed for our CAV testing. Finally, we shortly described the real-world experiment setups.

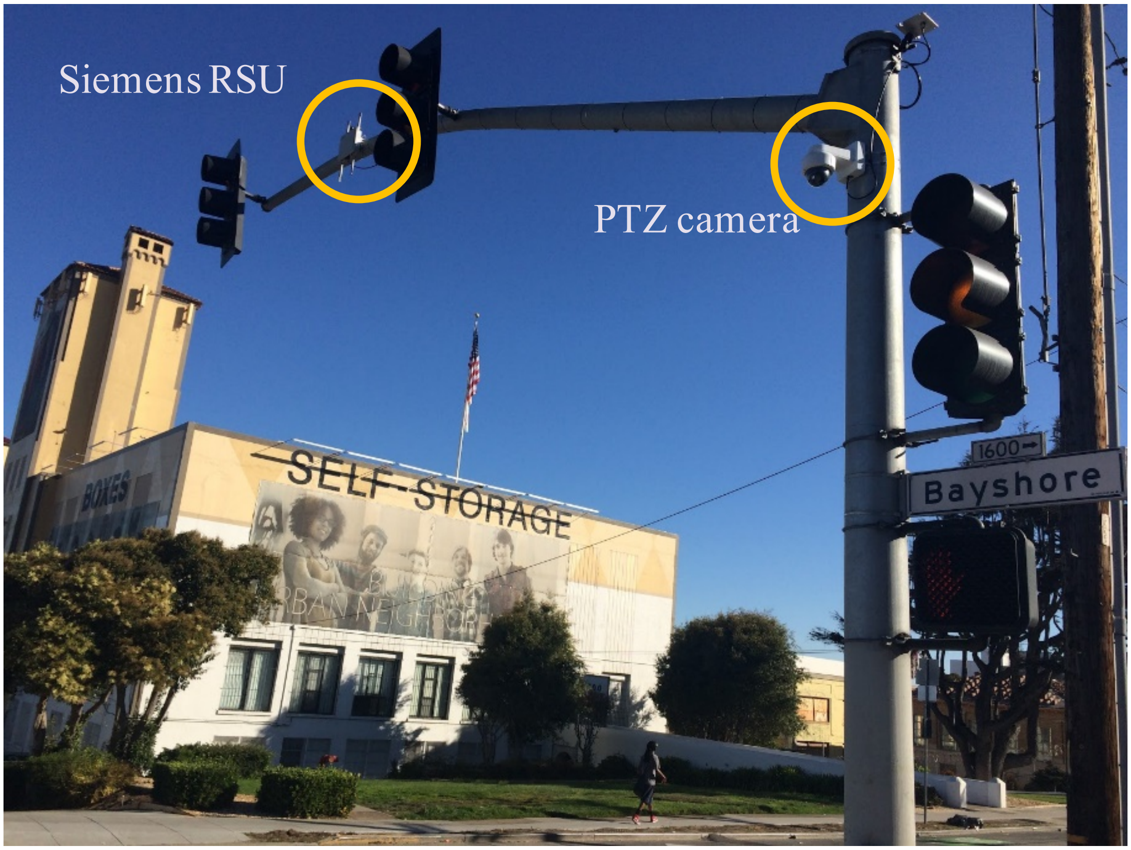


Figure 2.10: Traffic sensors installed (by Sensys Network) on Egbert Ave. & Bayshore Blvd, San Francisco, USA. The vehicle communicates with the RSU which sends the real-time V2I information. The same sensors are installed at intersections across the testing route in Arcadia.

# Chapter 3

# Vehicle Dynamics and Energy Consumption Modeling

Having accurate vehicle dynamics is critical in the development of the Connected and Automated Vehicles (CAVs) motion planning and control for the following two reasons. First, the control design exploited in this dissertation for the CAV motion control is based on Model Predictive Control (MPC) of which the performance depends on the accuracy of the modeling of the vehicle and the environment. Second, having accurate vehicle models in our Hardware-In-the-Loop (HIL) simulation allows us to validate the energy saving performance of the developed CAV motion control as well as the safety performance.

This chapter focuses on modeling longitudinal vehicle dynamics and energy consumption modeling associated with the longitudinal vehicle dynamics. We carefully examine how the aerodynamic drag and energy consumption are affected by vehicle speed and inter-vehicular gap. To limit to the scope of this dissertation, we skip the lateral vehicle dynamics because we are particularly interested in energy saving aspects of motion control which are largely affected by the longitudinal motion rather than the lateral motion.

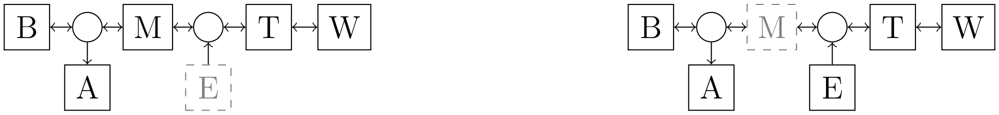
## 3.1 Longitudinal Vehicle Dynamics Model

The longitudinal vehicle dynamics model relates the wheel force (torque) as an input to the longitudinal velocity of the vehicle as a state. It is influenced by longitudinal wheel torques, air drag forces, rolling resistance forces, and gravitational forces.

We model the dynamics of the longitudinal vehicle speed as [35]:

$$M\frac{dv(t)}{dt} = F_w(t) - F_b(t) - F_R(\vartheta(t), d(t), v(t)) \tag{3.1}$$

where $F_w$ is the drivetrain traction force at the wheels, $F_b$ is the braking force at the wheels, $M$ is the vehicle mass, $\vartheta$ is the road slope, and $F_R(\vartheta(t), d(t), v(t))$ is the resistance force



(a) Full Electric (FE) configuration

(b) Full Combustion engine (FC) configuration

Figure 3.1: two configurations of a pre-transmission hybrid electric powertrain used for our data collection campaign

which is expressed as:

$$F_R(\vartheta(t), d(t), v(t)) = Mg(\sin\vartheta(t) + C_r \cos\vartheta(t)) + C_v v(t) + \frac{1}{2}\rho A C_x(d(t))v(t)^2 \tag{3.2}$$

where $A$ is the front area, $\rho$ is the air density, $C_r$ is the rolling coefficient, $C_v$ is the viscous friction coefficient, $C_x(d)$ is the air drag coefficient which is a function of inter-vehicular distance $d$. We also denote the wheel radius as $R_w$, the wheel speed as $\omega_w = v/R_w$, the wheel torque as $T_w = F_w R_w$, the braking torque as $T_b = F_w R_w$.

We designed and executed a set of experiments aimed at identifying the parameters in (3.1) and the effects of inter-vehicular distance on energy consumption of our test vehicle.

## 3.2 Experimental Setup

In this subsection, we provide specific descriptions for the experiments conducted to identify longitudinal vehicle dynamics model parameters and characterize the effect of short inter-vehicular distance in $C_x$ and energy. General details about the experimental platform such as the test vehicles and the closed track are already provided in Chapter 2. The experiment involves a set of our two identical Hyundai Ioniqs, which are the Plug-in Hybrid Electric Vehicles (PHEVs). The weight of the two cars is matched before every experimental session. Because production adaptive cruise control only allows a minimum time gap of about 1 s, we implemented our dedicated longitudinal control which allows smaller time gap.

Our test platform does not include a direct measurement of the wheel power, but has accurate measurements of fuel and electric energy consumption through high accuracy sensors. Therefore, the energy performance *after* the powertrain can be accurately characterized in terms of total electric battery and/or fuel consumption. To minimize the effect of powertrain operation on the data collected in our experiments, we implemented a (very simple) dedicated powertrain controller.

The dedicated powertrain controller uses only one prime mover (engine or motor) at a time to drive the vehicle; i.e., the vehicle either operates in purely electric (or full electric, FE) mode or in purely thermal (or full combustion engine, FC) mode. Figure 3.1 depicts these two configurations of the PHEV powertrain. We also denote by $\omega_w$ the wheel speed, by $r_g$ the gear ratio between the wheels and the powertrain axle (which is the same for the motor and the engine), by $P_a$ the electric power absorbed by auxiliaries such as air conditioning,

Table 3.1: Number of test laps for each powertrain mode (FE/FC), time gap (columns), and velocity (rows).

| | $\infty$ | 2 s | 1 s | 0.5 s | 0.3 s |
|---|---|---|---|---|---|
| 24.6 m/s<br>(55 mph) | FE: 1<br>FC: 2 | FE: 0<br>FC: 2 | FE: 1<br>FC: 2 | FE: 1<br>FC: 2 | FE: 1<br>FC: 2 |
| 29.1 m/s<br>(65 mph) | FE: 1<br>FC: 2 | FE: 0<br>FC: 2 | FE: 1<br>FC: 2 | FE: 1<br>FC: 2 | FE: 1<br>FC: 2 |
| 33.5 m/s<br>(75 mph) | FE: 1<br>FC: 2 | FE: 0<br>FC: 2 | FE: 1<br>FC: 2 | FE: 1<br>FC: 2 | FE: 1<br>FC: 2 |

and by $\mu \in \{\mathrm{FE}, \mathrm{FC}\}$ the operating mode of the powertrain. Finally, the battery and fuel power are modeled as

$$\begin{aligned} P_b &= \begin{cases} \dfrac{T_w \omega_w}{\eta_m(T_w/r_g, \omega_w r_g)} + P_a, & \text{if } \mu = FE, \\ 0, & \text{if } \mu = FC, \end{cases} \\ P_f &= \begin{cases} P_a, & \text{if } \mu = FE, \\ \dfrac{T_w \omega_w}{\eta_e(T_w/r_g, \omega_w r_g)}, & \text{if } \mu = FC, \end{cases} \end{aligned} \tag{3.3}$$

where $\eta_m$ and $\eta_e$ are the efficiencies of the electric motor and internal combustion engine, respectively; both are assumed to be known static functions of the motor and engine torque and speed.

The test facility used for these experiments is the Hyundai-Kia California Proving Ground oval track (see Section 2.5 for more details). Each experiment is performed on a single lane at a fixed combination of speed, time gap, and powertrain mode. During each experiment, while the longitudinal motion of the vehicles is controlled by the dedicated longitudinal controller, the lateral motion was controlled by the human drivers who tried to maintain the vehicles at the center of the lane throughout the experiments. Moreover, each experiment is conducted in conditions of low wind speed (average and maximum wind speed always less than $3.18\,\mathrm{m\,s^{-1}}$ and $3.95\,\mathrm{m\,s^{-1}}$, respectively, which is below the limits set in the SAE J1263 standard [40]). Table 3.1 summarizes the amount of data collected (in terms of number of laps around a 10 km long oval test track) for each experiment. All experiments are performed in a time window of 4 days.

## 3.3 Energy Consumption Results

Figure 3.2 shows the energy consumption results from the tests listed in Table 3.1. Note that these results are obtained with a homogeneous two-vehicle platoon composed of Hyundai

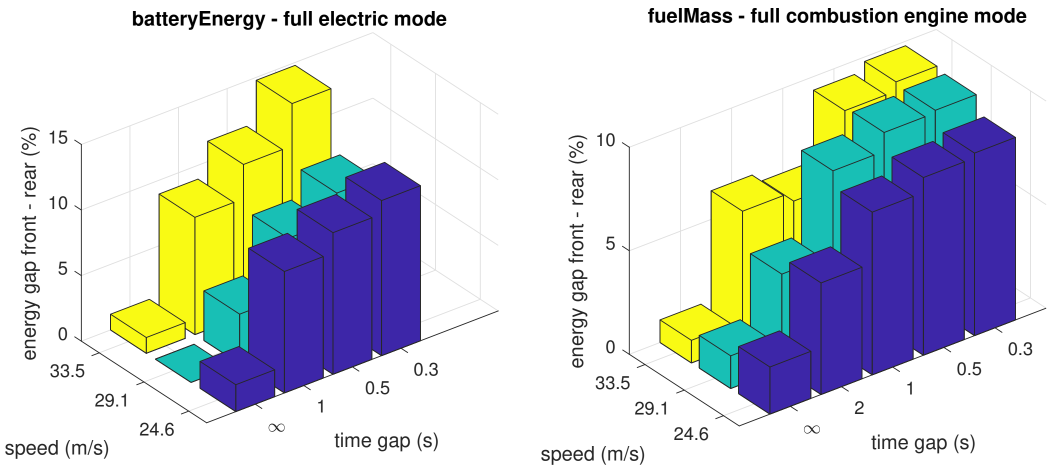


Figure 3.2: Percentage variation of electric energy consumed (rear vs front) in full electric mode (left, $\mu = FE$) and fuel consumed (rear vs front) in full combustion engine mode (right, $\mu = FC$), as a function of speed and time gap.

Ioniq PHEVs. Clearly, results will be different for other vehicles or the heterogeneous platoon. The left chart refers to data collected in full electric mode ($\mu = FE$), while the right chart refers to data collected in full combustion engine mode ($\mu = FC$). Each chart shows time gap (in seconds) and vehicle speed (in meters per second) on the horizontal axes, and the percentage variation of energy consumption between front and rear vehicle on the vertical axis. The percentage variation is computed on the integral of battery power $P_b$ for full electric data, and on the integral of fuel power $P_f$ for full combustion engine data.

In Figure 3.2, the time gap is labeled as $\infty$ for experiments performed on isolated vehicles; these tests were performed to gather baseline data, as well as for comparing the performance of the two vehicles. Both figures show that, in isolated conditions and at different speeds, the fuel/energy consumption of the two test vehicles differ by at most 3 %. Such difference can be attributed to small variability between the two vehicles.

The collected data enable direct evaluation of the energy performance variation (improvement) after the powertrain. In full combustion engine mode, the fuel consumption is reduced between 4 % and 10 %. In full electric mode, the electric energy consumption is reduced between 4 % and 15 %. In both powertrain modes, the air drag appears to consistently reduce with time gap, for a fixed speed except one instance. At the speed of $33.5\,\mathrm{m\,s^{-1}}$, the fuel energy consumption was slightly reduced (less than 0.5%) when the time gap increased from 1 s to 2 s. We believe that this is due to imperfect lateral alignment between two vehicles. All the other parameters such as wind, weight, auxiliary load are either corrected for or kept within small bounds.

## 3.4 Interpretation of the Results

The data presented above are purely based on our measurements, in the listed operating conditions. We now provide an interpretation of the same data in terms of the vehicle longitudinal model (3.1) and powertrain model (3.3). In essence, we propose a simple model which generalizes the results above to any operating condition. To this aim, we use an existing model of the powertrain components (to connect the measurements of the energy consumption to the estimated wheel torque) and we fit a longitudinal vehicle model.

First, we show that our estimated wheel torque and our powertrain model explain well the energy consumption values that we found in the experiments. For all the data points in our measurements, we compute the predicted consumption of battery energy $\hat{P}_b$ and fuel energy $\hat{P}_f$ according to model (3.3), using the measured wheel torque $T_w$, wheel speed $\omega_w$, auxiliary power $P_a$, and powertrain mode $\mu$ as inputs. Figure 3.3 shows the histograms of the normalized residuals

$$e_b = \frac{P_b - \hat{P}_b}{RMS(P_b - \hat{P}_b)}, \quad e_f = \frac{P_f - \hat{P}_f}{RMS(P_f - \hat{P}_f)}.$$

Now, we fit the longitudinal model (3.1). The model parameters $C_r, C_v, C_x$ are known to depend on a number of factors, including gear ratio, vehicle speed, tire pressure, road surface, and inter-vehicle distance [35]. We assume that:

- the rolling coefficient $C_r$ is constant;
- the viscous coefficient $C_v$ is constant;
- the air drag coefficient is a function of inter-vehicle distance, $C_x = C_x(d)$.

To fit the model with mode- and distance-dependent parameters, we first define the cost function

$$\begin{aligned}
J(\bar{d}, \bar{\mu}) = \\
&\|F_w(k) - ma(k) - mgC_r - C_v v(k) \\
&- \frac{1}{2}\rho A C_x(\bar{d}) v(k)^2\|, \\
&\forall k \in \mathcal{K}(\bar{d}, \bar{\mu}) := \left\{k | d(k) = \bar{d}, \mu(k) = \bar{\mu}\right\},
\end{aligned}$$

where $k$ is a datapoint index, $\bar{d}$ and $\bar{\mu}$ are the values of (target) $d$ and $\mu$ allowed by Table 3.1 (in symbols $\bar{d} \in \mathcal{D}$ and $\bar{\mu} \in \mathcal{M}$), and $\mathcal{K}$ is the cluster of all datapoint indexes corresponding to $\bar{d}$ and $\bar{\mu}$. We formulate the following optimization problem:

$$\begin{aligned}
\min_{C_r, C_v, C_x(\bar{d}), \forall \bar{d} \in \mathcal{D}, \forall \bar{\mu} \in \mathcal{M}} \sum_{\bar{d} \in \mathcal{D}, \bar{\mu} \in \mathcal{M}} & J(\bar{d}, \bar{\mu}) \\
\text{s.t.} & C_r \geq 0, \\
& C_v \geq 0, \\
& C_x(\bar{d}) \geq 0, \forall \bar{d} \in \mathcal{D}
\end{aligned} \tag{3.4}$$

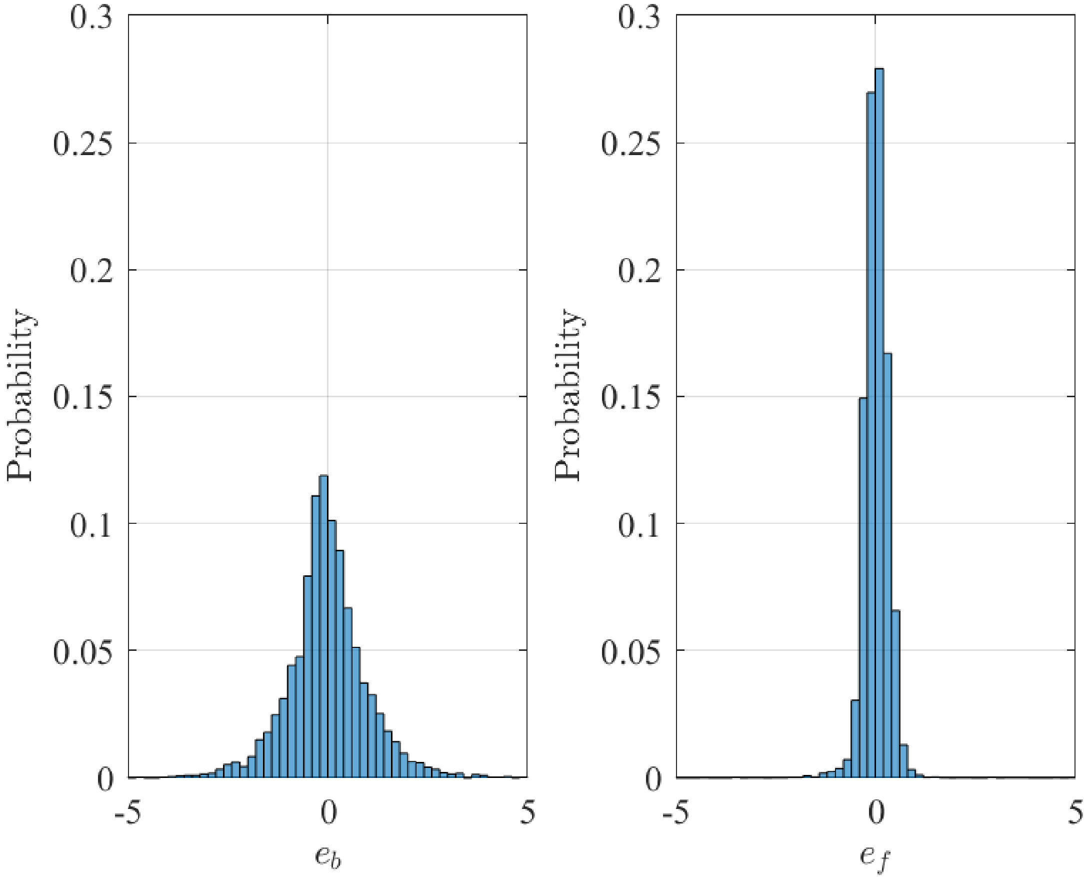


Figure 3.3: Histograms of the normalized residuals $e_b$ and $e_f$.

where $C_x(\bar{d})$ is defined using the vehicular air drag reduction model from [41] as

$$C_x(d) = C_{x,0}\left(1 - \frac{C_{x,1}}{d + C_{x,2}}\right). \tag{3.5}$$

We parse the problem with YALMIP [42], solve it with IPOPT [43] on 80 % of the data, and find the parameters summarized in Table 3.2.

We validate the resulting model on the remaining 20 % of the data. Figure 3.4 shows the histogram of the residuals for the validation dataset. For all the validation datapoints, we compute the predicted wheel torque $\hat{T}_w$ according to model (3.1) and the parameters in Table 3.2, using the measured speed $v$ and inter-vehicle distance $d$ as inputs ($\vartheta = 0$ throughout our experiments). Figure 3.3 shows the histograms of the normalized residuals

$$e_w = \frac{T_w - \hat{T}_w}{RMS(T_w - \hat{T}_w)}.$$

Table 3.2: Known (top) and fitted (bottom) parameters for the rear vehicle.

| Model parameter | Value |
|---|---|
| $M$ | $1844\,\mathrm{kg}$ |
| $R_w$ | $0.288\,\mathrm{m}$ |
| $\rho$ | $1.206\,\mathrm{kg\,m^{-3}}$ |
| $A$ | $2.629\,\mathrm{m^2}$ |
| $C_r$ | $0.0093$ |
| $C_v$ | $0$ |
| $C_x(d)$ | $0.3350\left(1 - \dfrac{68.3193}{d + 142.4522}\right)$ |

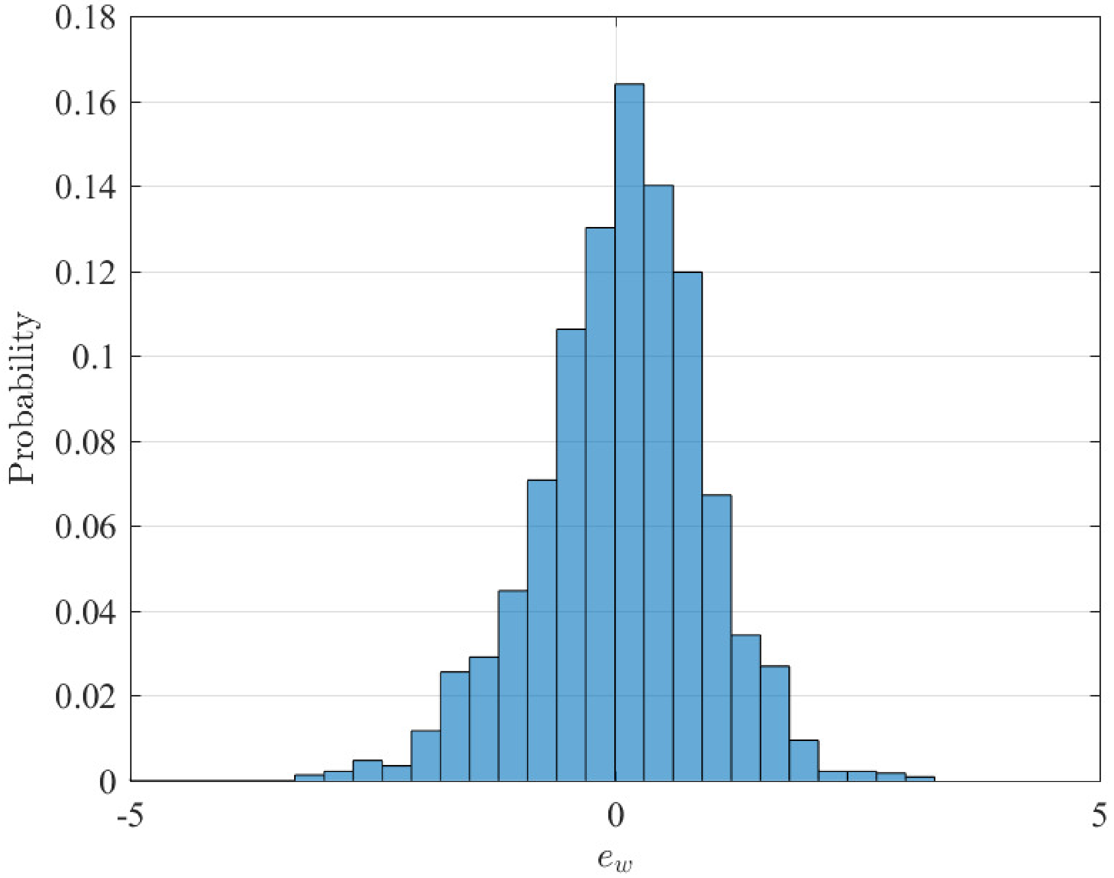


Figure 3.4: Histogram of the normalized residuals $e_w$.

## 3.5 Conclusion

This chapter discusses a longitudinal vehicle dynamics model used throughout the dissertation. We conducted dedicated experiments to gather data to identify parameters in the longitudinal vehicle dynamics model and to study the effects of inter-vehicular distance on energy consumption.

Having a high fidelity longitudinal vehicle dynamics model allows us to validate the energy saving of the CAVs *before* the powertrain, i.e., during the *wheel-to-meters* conversion step (the conversion of mechanical energy at the wheel into kinetic and potential energy, which causes the vehicle to move [44]). However, in this dissertation, we also compare energy saving *after* the powertrain, i.e., during both the *wheel-to-meters* conversion step and the *tank-to-wheel* conversion step (the conversion of energy stored in fuel, battery packs, etc. into mechanical energy at the wheel [44]) such as total fuel/battery consumption, miles per gallon of gasoline-equivalent (MPGe) [45]. To do this, we also use a powertrain dynamics and powertrain controller model, which takes the demand engine, motor, and hydraulic braking torques as inputs and predicts gasoline and battery consumption of the vehicle. While we omitted discussions on powertrain dynamics modeling in this dissertation to limit the scope of the work, readers are encouraged to see e.g. [34] for details on powertrain dynamics and powertrain controller modeling and its validation for our test vehicle.

# Chapter 4

# Connected and Automated Vehicle Motion Control Using Vehicle-to-Vehicle Communication

## 4.1 Introduction

This chapter discusses a Connected and Automated Vehicle (CAV) motion control which exploits benefits from communication between vehicles (V2V). The CAV motion control is performed by the Real-Time Control and Planning module in our CAV architecture in Figure 2.2. In particular, we focus on the design of a Longitudinal Motion Control (LMC) rather than a lateral motion control because we are interested in CAV energy improvement characterization which is affected primarily by the longitudinal motion rather than the lateral motion.

There has been extensive study on different designs of the CAV longitudinal motion control for energy efficient driving [46, 47, 16]. In particular, model predictive control (MPC) designs have been explored and tested due to its ability to exploit additional yet valuable information from the CAV communication and cloud connectivity and to optimize the vehicle motion accordingly. [48, 20, 49, 50, 16]. However, these studies still lack more focused (if not any) investigation on the following aspects: energy saving analysis of for compact platooned passenger vehicles with Cooperative Adaptive Cruise Control (CACC) backed by the experimental data, the energy saving LMC for hybrid electric vehicles exploiting the regenerative braking dynamics, the CACC design which can guarantee the persistence of both the minimum and the maximum distance between the vehicles. The goal of this chapter is to fill these knowledge gaps by presenting different possible control designs and the corresponding result analysis tacking each point addressed above.

Using V2V communication, CAVs can wirelessly exchange their states, and consequently improve situation awareness in the face of the limitations of on-board perception systems. CACC, an enhancement of adaptive cruise control (ACC) enabled by V2V communication, is an instance of the LMC with V2V communication. Previous research on CACC has focused on the *string stability* of vehicle stream to maximize the road throughput [51, 52, 53]. String stability is defined as the uniform boundedness of all the states of the interconnected system at all times, if the initial states of the interconnected system are uniformly bounded [51]. In platoon control, string stability implies that the leading vehicle's velocity and position perturbation does not cause a larger amplification in the following vehicle's velocity and position [54].

Some works on CACC which pursue energy saving are focused on the minimization of the inter-vehicular distance, both to maximize the road throughput and to reduce the vehicle aerodynamic drag (and consequently its energy consumption) [55, 56, 57]. The relationship between inter-vehicular distance, aerodynamic drag, and vehicle energy consumption is often cited in the CACC literature; however, its experimental validation is limited. The effect on energy consumption can be substantial in heavy-duty vehicles, which have large frontal areas. An experimental characterization of such relationship is found in [41] for a heavy-duty vehicle. Passenger vehicles can also benefit from a reduced inter-vehicular distance: in [58], one-eighth scale models of minivan vehicles were tested in a wind tunnel, and it was shown that, even for such vehicles, a reduced inter-vehicular distance translates directly to reduced aerodynamic drag and energy consumption. It is reported that the vehicles driving at very short gaps can reduce their consumption up to 30% on highways and up to 10% on urban roads. However, to the best of our knowledge, the literature lacks an experimental characterization of the aerodynamic drag and the vehicle energy consumption as a function of the inter-vehicular distance, *on full scale passenger vehicles.* Section 4.3 addresses this issue. In particular, we characterize experimentally the aerodynamic drag and vehicle energy consumption as a function of vehicle speed and inter-vehicular gap, for a compact passenger vehicle. Then, we present, through simulation study, a provably safe CACC based on model predictive control and investigate how leveraging V2V communication can lead to the energy saving which is quantified by using the experimental performance curves obtained at the previous point.

Another approach to reduce energy consumption in LMC is to receive predictions of each vehicle's trajectory via V2V communication, and accordingly adapt the velocity trajectory in a *non-cooperative* manner, avoiding energy wasteful behaviors and achieving *eco-driving*, which refers to changing driving styles from aggressive style to a more refined driving style to improve energy efficiency [59]. Hereafter, we name a non-cooperative energy saving LMC as *eco-LMC*.

Velocity smoothing is the key idea behind the eco-LMC. Model predictive control (MPC) is widely adopted for the eco-LMC design because of its capability to optimize over the predicted future behavior as well as the current state of the vehicle and to handle multiple constraints of the system. Robust model predictive control design is used in MPC designs for cruise control [60, 19]. However, these are only applicable for cruise control, not adaptive cruise control which takes into account of the front vehicles for safety. [61] presents a robust MPC for adaptive cruise control but is only dedicated for building a personalized driving which can learn from human demonstrations, not necessarily on saving energy. In [48], the authors proposed a MPC design for an adaptive cruise control system which seeks to improve performance such as tracking capability, fuel economy, and driver desired response. The authors designed a quadratic cost function to penalize detrimental behavior for each performance and the proposed controller optimizes over a combination of these objective functions. However, the proposed controller lacks robust safety guarantee against uncertain front vehicle velocity prediction and is short-sighted (only considering over a limited horizon). A long term forecast of the front vehicle velocity can be used to construct a terminal set [20] (which is our initial work on eco-LMC) and terminal cost [50] of the MPC optimization problem to prevent vehicles from excessive accelerating. In particular, [20] demonstrated its effectiveness with experimental results using the internal combustion engine vehicle on a closed track setup described in Section 2.5.1. However, these designs assumed that the front vehicle prediction is perfect, i.e., has no errors.

In Section 4.4, we propose an eco-LMC suitable for a hybrid electric vehicle of which the powertrain has a pre-transmission parallel architecture as depicted in Figure 2.4. The proposed controller seeks to minimize wasteful energy behavior by reducing unnecessary braking (especially, hydraulic braking) while robustly satisfying safety constraints such as speed limits and safety distance with the front vehicle. The virtue of the proposed controller is that it exploits available information about the front vehicle velocity forecast while it can still guarantee safety regardless of the forecast accuracy. We use the Learning Based Model Predictive Control (LBMPC) framework to enhance performance based on the V2V forecast, while preserving safety guarantees. The key idea of LBMPC is to use (i) the system model subject to the disturbance to robustly satisfy the constraints and (ii) the predicted model to improve performance [62].

The eco-LMC design presented in Section 4.4 and the references mentioned in above only consider the minimum distance constraint without regard to the maximum distance. This is more problematic when the vehicles want to maintain a platoon. Without an appropriate tuning and design for the cost function, the vehicles in the platoon may exceed the maximum distance and cause the platoon to split. We refer to this as *loss of contact*[1].

In Section 4.5, we propose another interesting design of an eco-LMC, namely *eco-CACC* which exploits V2V forecasts to improve energy saving performance, while formally guaranteeing robust safety and preventing loss of contact regardless of accuracy of the front vehicle velocity forecast in the V2V communication. Note that this section does not discuss the management of the platoon system; Appendix A presents the micro-platoon management we developed for our CAV platoon. The proposed eco-CACC optimizes the longitudinal motion of a single vehicle in the platoon to minimize control effort and jerk while ensuring that the controlled vehicle maintains contact with its preceding vehicle in the platoon at all times. Compared to existing works [63, 20] or the control designs presented in the previous sections, we concurrently guarantee both the minimum distance constraint to front collisions and the maximum distance constraint to avoid loss of contact. This is the reason we call our

[1] Loss of contact can happen because a third vehicle cuts in between the ego vehicle and the preceding vehicle, or because the communication link is lost due to a large inter-vehicular distance.

controller cooperative.

## 4.2 LMC Using Model Predictive Control

In this section we present a simplified Model Predictive Control (MPC) design for a LMC. We use a robust MPC framework to guarantee that the vehicle always avoids collisions with the front vehicle and satisfies the velocity limits despite the uncertain forecast of the front vehicle velocity. Building on this concept, we subsequently demonstrate further advanced control strategies to seek higher energy efficiency in the remainder of this chapter.

### 4.2.1 Control Oriented Vehicle Dynamics Model

In LMC, we consider the problem of controlling a single vehicle (hereafter named *ego* vehicle) and its interaction with the vehicle directly in front of the ego vehicle (hereafter named *front* vehicle). Therefore, the state variables are the distance to the front vehicle and the velocity of the ego vehicle, denoted as $d$ and $v$, respectively, and the input variables are the wheel traction force and the braking force, denoted as $F_w$ and $F_b$, respectively. The front vehicle velocity is an uncertain variable, denoted as $v_f$.

Finally, we model the longitudinal dynamics as in (3.1) and apply Euler discretization with step $t_s$, obtaining at time $t$

$$d(k+1|t) = d(k|t) + t_s(v_f(k|t) - v(k|t)), \tag{4.1a}$$

$$v(k+1|t) = v(k|t) + \frac{t_s}{M}(F_w(k|t) - F_b(k|t) - F_R(\vartheta(k|t), d(k|t), v(k|t))) \tag{4.1b}$$

where $F_R(\cdot)$ is obtained from the model (3.1) and (3.2). However, in (3.2), the aerodynamic drag coefficient $C_x$ is a non-linear function of the distance to the front vehicle $d$, as seen in (3.5), while the road grade $\theta$ depends on the vehicle absolute position. Using such a model in an MPC framework requires a strategy to solve the resulting highly nonlinear optimization problem in real time which can be cumbersome in computation. Moreover, the solver may converge to local minima. To resolve these issues, we assume that (i) the ego vehicle is traveling on a flat road and (ii) $C_x$ always takes the maximum value. With these assumptions, we introduce a control oriented model, in which we approximate $F_R(\cdot)$ as

$$F_R(\vartheta(k|t), d(k|t), v(k|t)) \approx \tilde{F}_R(v(k|t)) := Mg\bar{C}_r - C_v v(k|t) + \frac{1}{2}\rho A\bar{C}_x v(k|t)^2 \tag{4.2}$$

where $\bar{C}_r = C_r$ and $\bar{C}_x = C_{x,0}$.

In the remainder of the paper, the dynamic model (4.1) is denoted by

$$x(k+1|t) = f(x(k|t), u(k|t), v_f(k|t)), \tag{4.3}$$

where $x(k|t) = [d(k|t),\ v(k|t)]^\top$ and $u(k|t) = [F_w(k|t),\ F_b(k|t)]^\top$.

### 4.2.2 State and Input Constraints and Uncertainty Bounds

State constraints are enforced to prevent collision with the front vehicle and violation of speed limits. Input constraints are enforced to account for the physical limitations of the actuators of the ego vehicle. The state and input constraints are compactly expressed in the form

$$x(k|t) \in \mathbb{X} := \{(d, v) : d_{\text{min}} \leq d,\ 0 \leq v \leq v_{\text{max}}\}, \tag{4.4a}$$

$$u(k|t) \in \mathbb{U} := \{(F_w, F_b) : 0 \leq F_w \leq F^w_{\text{max}},\ 0 \leq F_b \leq F^b_{\text{max}}\} \tag{4.4b}$$

where $d_{\text{min}}$ denotes the minimum safety distance and $v_{\text{max}}$ is the maximum velocity. where $\underaccent{\bar}{C}_r$ and $\bar{C}_r$ are the minimum and the maximum values of $(\sin(\vartheta) + C_r \cos(\vartheta))$ and $\bar{C}_x$ is the maximum values of $C_x(\cdot)$.

The front vehicle velocity has an acceleration, $a^f$ that is an uncertain, bounded variable:

$$a^f(k|t) \in \mathbb{A}^f(v_f) := \{a^f(k|t) : a^f_{\text{min}}(v_f) \leq a^f(k|t) \leq a^f_{\text{max}}(v_f)\}. \tag{4.5}$$

where the uncertainty bounds $a^f_{\text{min}}(v_f) := \mathbb{1}_{v_f > 0} a^f_{\text{min}}$ and $a^f_{\text{max}}(v_f) := \mathbb{1}_{v_f < v_{\text{max}}} a^f_{\text{max}}$ where $a^f_{\text{min}}, a^f_{\text{max}}$ can be fixed and assumed *a priori* by the ego vehicle, or can be communicated by the front vehicle and adapted to the current operating condition, if an appropriate V2V communication protocol has been established. In the latter case, the uncertainty set $\mathbb{A}^f$ can expand or shrink, based on the maximum braking and acceleration bounds transmitted by the front vehicle via V2V communication. Note that we bounds the velocity by 0 and $v_{\text{max}}$ because we assume that the front vehicle cannot not drive backward nor drive faster than the maximum velocity.

### 4.2.3 Robust Model Predictive Control Formulation

We formulate the LMC problem as the following constrained finite horizon optimal control problem at time $t$.

$$\min_{u(\cdot|t)} \sum_{k=t+1}^{t+N_p-1} g(x(k|t), u(k|t)) \tag{4.6a}$$

subject to

$$x(k+1|t) = f(x(k|t), u(k|t), v_f(k|t)), \tag{4.6b}$$

$$v_f(k+1|t) = v_f(k|t) + t_s \tilde{a}^f(k|t), \tag{4.6c}$$

$$x(k|t) \in \mathbb{X},\ u(k|t) \in \mathbb{U}, \tag{4.6d}$$

$$\forall k \in [t, ..., t + N_p - 1]$$

$$[x(t+N_p|t)^\top, v_f(t+N_p|t)]^\top \in \mathbb{C}, \tag{4.6e}$$

$$x(t|t) = \bar{x}(t|t),\ v_f(t|t) = \bar{v}_f(t|t) \tag{4.6f}$$

where $N_p$ is the prediction horizon; (4.6b) is the system update function based on (4.1); (4.6c) is a front vehicle velocity update with an Euler disretization and an uncertain front vehicle acceleration prediction $\tilde{a}^f \in \mathbb{A}^f(v_f(k|t))$; (4.6d) are constraints defined by (4.4); $\mathbb{C}$ in (4.6e) is a robust invariant set which is explained in the next subsection; (4.6f) is the state and front vehicle velocity measurements at time $t$ using the available information at time $t$.; the cost function $g(x, u)$ in (4.6a) is a trade off between the control effort, the velocity tracking error, and the distance tracking error. The reference velocity or distance can be either user-defined or defined by the Motion Eco-Driving Planning in Remote Planning and Routing module in Figure 2.2.

While $\tilde{a}^f$ can be any value in the set $\mathbb{A}^f(v_f(k|t))$, the constraints (4.6d) and (4.6e) must be satisfied for all $a^f \in \mathbb{A}^f(v_f)$ unless true values of $a^f$ is provided by the communication. It is generally accepted that finding a sequence of input actions cannot contain the spread of predicted states caused by the disturbance. Hence, the standard robust MPC solves its optimization problem by finding a sub-optimal state feedback policy rather than finding a sequence of input actions [64]. However, this method is generally computationally intractable for a non-linear system. In our optimization problem (4.6) with the nonlinear system dynamics (4.6b), we resolve this issue by finding a sequence of input actions rather than a input policy while assuming that the front vehicle fully decelerates until a complete stop, i.e., $a^f(k|t) = a^f_{\min}(v_f(k|t))$. Due to the simple nature of our dynamics (4.6b) and constraints (4.6d), robust satisfaction of constraints can be achieved assuming that the front vehicle fully decelerates until a complete stop.

Solving (4.6), we have the following optimal inputs and states:

$$u^*(t) = [u^*(t|t), u^*(t+1|t), ..., u^*(t+N_p-1|t)]. \tag{4.7a}$$

$$x^*(t) = [x^*(t|t), x^*(t+1|t), ..., x^*(t+N_p|t)], \tag{4.7b}$$

The first input $u^*(t|t)$ is applied to the system during the time interval $[t, t+1)$. At the next time step $t+1$, a new optimal control problem in the form of (4.6), based on new measurements of the state, is solved over a shifted horizon, yielding a *moving* or *receding* horizon control strategy

$$u(t) = u^*(t|t). \tag{4.8}$$

and the closed loop system is written as

$$x(t+1) = f(x(t), u^*(t|t), v_f(t)). \tag{4.9}$$

### 4.2.4 Recursive Feasibility

MPC only predicts for a short horizon $N_p$ and, therefore, is, without a proper design, subject to having an infeasibile solution and violating constraints in the future time steps. In this subsection, we show how our controller systemically ensures state and input constraint satisfaction at all times, the property known as *recursively feasibility.* Here we formally define the notion of recursive feasibility for the MPC controller (4.6)-(4.9).

**Definition 4.2.1** (Recursive feasibility)**.** *The MPC controller* (4.6)-(4.9) *is said to be recursively feasible, if the existence of a solution* $u^*(\cdot|t)$ *for* (4.6) *at time* $t = 0$ *implies feasibility of the MPC problem* (4.6) *for all time* $t > 0$*.*

In robust MPC, the recursive feasibility can be systemically achieved by enforcing the state at the terminal horizon to be in the robust control invariant set. A general definition of robust control invariant set is given in [65]; for our problem it can be formalized as follows.

**Definition 4.2.2** (Robust control invariant set)**.** *A set* $\mathbb{C} \subseteq \mathbb{X}$ *is said to be a robust control invariant set for the system* (4.1) *subject to constraints* (4.4)*, if*

$$\begin{aligned}&[x^\top(0), v_f(0)]^\top \in \mathbb{C} \;\Rightarrow \\ &\qquad \exists\, u(t) \in \mathbb{U}\, \textit{such that}\, f(x(t), u(t), v_f(t)) \in \mathbb{C},\; \forall a^f(t) \in \mathbb{A}^f(v_f(t)),\; \forall t \in \mathbb{N}_+.\end{aligned} \tag{4.10}$$

To compute a robust control invariant set for our system (4.1) subject to constraints (4.4), we build on the method from [61], which is suited to linear systems. First we introduce the uncertain linear system

$$x^l(k+1) = \begin{bmatrix} d^l(k+1) \\ v^l(k+1) \end{bmatrix} = \begin{bmatrix} d^l(k) + t_s(v^{f,l}(k) - v^l(k)) \\ v^l(k) + \frac{t_s}{m}\left(\frac{u^l(k)}{R_w}\right) \end{bmatrix} \tag{4.11}$$

and the state/input constraints and the uncertainty set are compactly written as

$$x^l(k) \in \mathbb{X},\, u^l(k) \in \mathbb{U}^{\text{tr}},\, a^f(k) \in \mathbb{A}^f(v_f(k)), \tag{4.12}$$

where $\mathbb{X}$ and $\mathbb{A}^f(\cdot)$ are defined in (4.4a) and (4.5), respectively, and

$$\mathbb{U}^{\text{tr}} := \{u : -(F^w_{\max} - (mg\bar{C}_r + \frac{1}{2}\rho A\bar{C}_x v^2_{\max})) \leq u \leq F^b_{\max} - mg\bar{C}_r, \}$$

Using the method from [61], we can obtain a polytopic representation of a robust control invariant set for the system (4.11) with the constraints (4.12). Then, we are in place to state the following theorem.

**Theorem 1.** *A set* $\mathbb{C}$ *is a robust control invariant set for the system* (4.1)-(4.5) *if* $\mathbb{C}$ *is also a robust control invariant set for the linear system* (4.11)-(4.12)*.*

*Proof.* Proof for Theorem 1 is straightforward from the definition of a robust control invariant set [66]. A robust control invariant set $\mathbb{C}$ for the system (4.11) with constraints (4.12) means that if $x^l(0) \in \mathbb{C}$, there exists a control law $u^l(t) = \Phi(x^l(t)) \in \mathbb{U}^l$ such that $x^l(t) \in \mathbb{C}$, for all $a^f(t) \in \mathbb{A}^f(v_f(t))$.

Now, observe that the control policy $u(t) = \Phi(x(t)) + mg\bar{C}_r + \frac{1}{2}\rho A\bar{C}_x v(t)^2$ makes the set $\mathbb{C}$ also a robust control invariant set for our original system (4.1)-(4.5) as it cancels out the nonlinear terms in (4.1), $mg\bar{C}_r + \frac{1}{2}\rho A\bar{C}_x v(t)^2$. Moreover, it is easy to see that the control policy $u(t)$ always satisfies the input constraints (4.4b). □

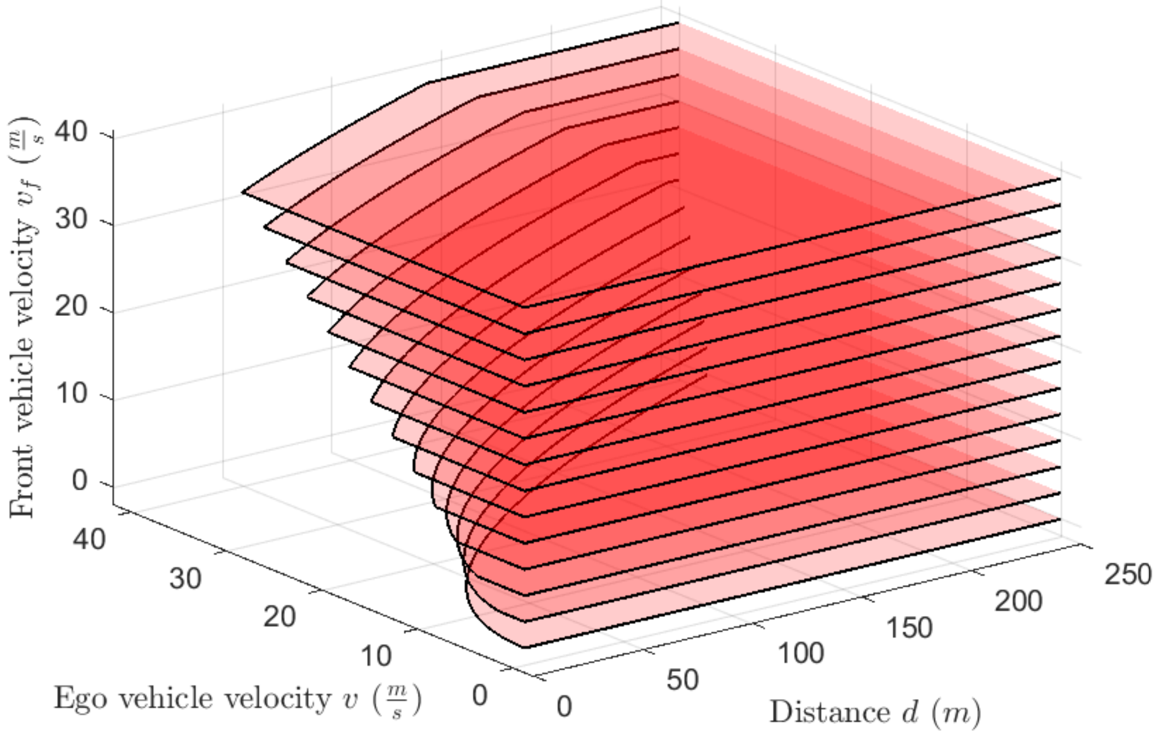


Figure 4.1: Robust control invariant set at different front vehicle velocities.

The obtained robust control invariant set is depicted in Figure 4.1.

Finally, we are in place to state the following theorem.

**Theorem 2.** *Consider the LMC* (4.6)-(4.9) *for the ego vehicle with* $N_p \geq 1$*. Assume that (i) the LMC is feasible at* $t = 0$ *(ii) there is no model mismatch and the measurements are accurate, i.e.,* $\bar{x}(t|t) = \bar{x}(t)$ *and* $\bar{v})_f(t|t) = \bar{v}_f(t)$*. If the set* $\mathbb{C}$ *is a robust control invariant set according to Definition 4.2.2 for the system* (4.1)-(4.5)*, then the LMC* (4.6)-(4.9) *is persistently feasible.*

*Proof.* We prove Theorem 2 by induction. Assume at time $t$, the optimization problem (4.6) is feasible, resulting in the optimal state and input trajectories (4.7). As the first optimal input (4.8) is applied to the ego vehicle during the time interval $[t, t+1)$, we have a new state $x^*(t+1|t) \in \mathbb{X}$ at $t+1$. Due to the constraint (4.6e) and the properties of the robust control invariant set, there exists at least one input trajectory which can keep the states of the ego vehicle inside the same set for any admissible acceleration (4.5) of the front vehicle for infinite time. Because $\mathbb{C} \subseteq \mathbb{X}$, the optimization problem (4.6) at $t+1$ is feasible for the ego vehicle. Therefore, we conclude by induction that the LMC (4.6)-(4.8) is persistently feasible. □

This section introduced a simple LMC design using the robust MPC framework and showed how the proposed LMC (4.6)-(4.9) guarantees that the ego vehicle satisfy the safety distance and velocity constraints at all times regardless of the front vehicle's unknown acceleration. The rest of this chapter (the following sections) is focused on how the vehicle communication can transform the LMC controller (4.6)-(4.9) into an energy saving LMC. Section 4.3 presents the LMC design which seeks to achieve energy saving by minimizing the inter-vehicular distance. Section 4.4 and 4.5 present the energy saving LMC designs which seek to achieve energy saving by driving at energy efficient velocity profiles and minimizing braking and jerk.

## 4.3 CACC for Compact Platooning using V2V

This section is focused on the value of V2V communication in longitudinal motion control of CAVs. In particular, we propose a provably safe CACC which exploits a V2V message, which includes the current and predicted states, and show the controller performance in reducing the inter-vehicular distance and the energy consumption. Finally, we quantify the energy saving using the experimental performance curves from Figure 3.2 in Chapter 2.

### 4.3.1 V2V Communication Structure

We now detail a possible V2V message structure, which can carry the information required to positively affect the vehicle energy consumption. We assume the predecessor-following communication topology depicted in Figure 4.2, where V2V messages are only sent from a vehicle to the vehicle immediately following it. In this communication topology, each CAV sends, to the immediately following CAV, a V2V message, denoted as $m_i$ where $i$ indicates the index/identity of the vehicle.

In our setup where there are only two vehicles, the ego and the front vehicles, the ego vehicle receives, from its direct front vehicle, a V2V message $m^f$, composed as

$$m^f(t) = \{s^f(t-h),\, v_f(t-h),\, \mathbb{A}^f,\, [\tilde{a}^f(t-h|t-h), \ldots, \tilde{a}^f(t-h+N_T|t-h)]\} \tag{4.13}$$

where $h$ is a communication/measurement delay; $s^f(t-h)$ and $v_f(t-h)$ are the absolute location and the velocity of the front vehicle at time $t-h$; $\mathbb{A}^f$ is the acceleration bounds as expressed in (4.5). $N_T$ is defined as the trust horizon, which indicates the number of time steps for which the front vehicle trusts its acceleration forecast to take the same values of its true acceleration, i.e. $\tilde{a}^f(j|t-h) = \bar{a}^f(j)$, $j = t-h, ..., t-h+N_T$ where $\bar{a}^f(j)$ is the front vehicle acceleration at time $j$.

### 4.3.2 Model Predictive Control for CACC

Our CACC is based on the MPC-based LMC (4.6)-(4.8). (4.6) is the formulation of the CACC optimization problem at time $t$ where the cost function (4.6a) penalizes deviations

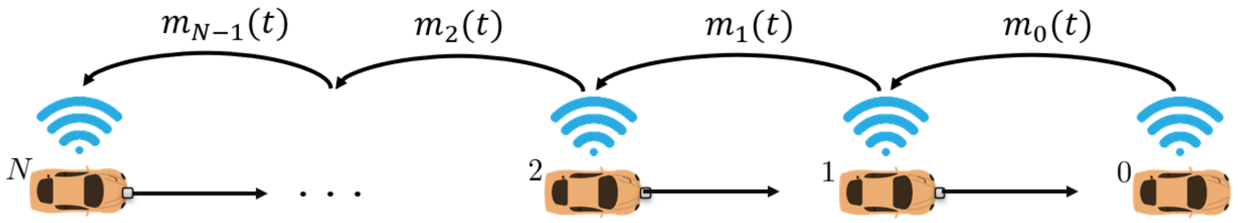


Figure 4.2: Platoon of $N + 1$ vehicles with predecessor following communication topology sending message $m$ at time t (the 0-th vehicle is the leader)

from the minimum safety distance $d_{\min}$, the actual excitation, and the jerk, with the weighting terms $Q$, $R$, and $D$, respectively. The cost function is compactly written as:

$$g(x(k|t), u(k|t)) = \|d(k|t) - d_{\min}\|_Q^2 + \|u(k|t)\|_R^2 + \|u(k|t) - u(k-1|t)\|_D^2 \tag{4.14}$$

Note that because the goal of the controller is to minimize the distance gap with the front vehicle in order to save energy, we use the minimum distance $d_{\min}$ to be the desired tracking distance. For the same purpose, $Q$ is set to be much larger than $R$ and $D$.

Moreover, it is noted that in order to guarantee string stability for the system controlled by our time-domain MPC control design, additional constraints can be imposed. These include restricting the magnitude of the ego vehicle's acceleration within that of the preceding vehicle [67] or imposing the move-suppression distance constraint [52].

The initial measurements are set in (4.6f) where $\bar{x}(t|t)$ and $\bar{v}^f(t|t)$ are the state and front vehicle velocity measurements at time $t$ using the available information at time $t$. In a vehicle without connectivity, distance to and velocity of the front vehicle are only available via on-board sensors such as radar or camera. However, these measurements are subject to delay and noise [68]. V2V communication can reduce the noise and the delay in the perception of the front vehicle down to the communication latency. In order to compensate for any perception delay (either from measurement or V2V communication latency), we use the following shifting equation:

$$\bar{x}(t-h|t) = \begin{bmatrix} \bar{d}(t-h) + n^d(t-h) \\ \bar{v}(t) \end{bmatrix}, \quad \bar{x}(k+1|t) = \begin{bmatrix} \bar{d}(k|t) + t_s(\bar{v}^f(k|t) - \bar{v}(k|t)) \\ \bar{v}(k+1) \end{bmatrix} \tag{4.15a}$$

$$\bar{v}^f(t-h|t) = (\bar{v}^f(t-h) + n^{v_f}(t-h))^+, \quad \bar{x}(k+1|t) = \left(\bar{v}^f(k|t) + t_s\tilde{a}^f(\bar{v}^f(k|t))\right)^+ \tag{4.15b}$$

$$\forall k = t-h, t-h+1, \dots, t-1,$$

where $\bar{d}(t-h)$, $\bar{v}(t-h)$, and $\hat{v}^f(t-h)$ are the true distance to the front vehicle, velocity, and the front vehicle velocity, respectively; Note that there exits a known, constant, non-negative delay $h$ either measured or communicated at time $t$; $n^d(t-h)$ and $n^{v_f}(t-h)$, such that $|n^d(t-h)| \leq n^d_{\max}$ and $|n^{v_f}(t-h)| \leq n^{v_f}_{\max}$, are the noises for distance and front vehicle velocity measurements, respectively; $\tilde{a}^f(\cdot)$ is defined as

$$\tilde{a}^f(v_f) = \begin{cases} \hat{a}^f(k|t-h), & \text{if}\, k \leq t-h+N_T, \\ a^f_{\min}(\bar{v}^f), & \text{else}, \end{cases} \tag{4.16}$$

where $\hat{a}^f(k|t-h)$ is the communicated front vehicle acceleration forecast at time $k$ obtained at time $t$; $N_T$ is the trust horizon. In the case of ACC without V2V communication, the front vehicle acceleration forecast and its bounds are not available. Hence, the front vehicle acceleration forecast $\tilde{a}^f(\cdot)$ is assumed to be the conservative (over-estimated) value of the maximum braking at all times.

### 4.3.3 Performance Analysis

In this subsection we show - through simulations - how the information and predictions shared via V2V communication can be leveraged to reduce the energy consumption in the small platoon of CAVs. We focus on the value of V2V communication for reducing the inter-vehicular gap, which is linked to the aerodynamic drag, as discussed in Section 3.1. We consider a homogeneous vehicle platoon of two Hyundai Ioniq PHEVs, the model of which is identified in Section 3.1, traveling on a level road ($\vartheta = 0$). The front vehicle is running at a constant velocity of 25 m/s; the following vehicle (the ego vehicle) has an initial velocity of 15 m/s and its motion is regulated by the controller (4.6)-(4.8),(4.14); their initial inter-vehicular distance is 50 m. The two vehicles communicate different message contents, depending on the scenario. Unless otherwise specified, environment and controller parameters are listed in Table 4.1. Note that as a default setting, we assume that the front vehicle can generate the minimum deceleration of $-6\text{m/s}^2$ while the only current measurements of its velocity and distance are available without any errors. Moreover, Table 4.2a-4.2c lists the energy consumption compared to that of the front vehicle under different assumptions about on-board sensors and V2V communication during the steady state phase; i.e., the time period after the ego vehicle reaches a constant distance gap with the front vehicle; the selected metrics are savings on wheel energy, fuel consumption in FC mode, and battery energy in FE mode. Note that each column represents the percentage of energy spent compared to that of the front vehicle in the unit of the indicated energy source *solely.* The fuel and battery energy consumption calculations are interpolated from the experimental data presented in Figure 3.2.

#### Effects of Reduced Measurement Delay

In CAVs, the states of the other CAVs are accessible through the on-board sensors and V2V communication. With smaller delay in the measurement, the current state (4.15) can be estimated less conservatively and more accurately.

Figure 4.3 depicts the trajectories of the front and the ego vehicles, in terms of their inter-vehicular distance, velocities, and wheel torque when the ego vehicle has the front vehicle

Table 4.1: Environment and controller parameters for different scenarios. Some parameters that are changed are described in each scenario.

| **Parameter** | **Description** | **Constant** |
|:---:|:---:|:---:|
| $N_p$ | MPC horizon | 20 |
| $t_s(s)$ | sampling time | 0.2 |
| $d_{\min}(\mathrm{m})$ | minimum distance | 5 |
| $v_{\max}(\mathrm{m/s})$ | maximum velocity | 40 |
| $F^w_{\max}(\mathrm{N})$ | maximum wheel force | $\frac{1083}{R_w}$ |
| $F^b_{\min}(\mathrm{N})$ | minimum wheel force | $\frac{-2500}{R_w}$ |
| $Q$ | distance tracking penalty | 10 |
| $D$ | jerk penalty | 0.03 |
| $h$ | measurement delay | 0 |
| $a^f_{\min}(\mathrm{m/s^2})$ | front vehicle minimum acceleration | -6 |
| $N_T$ | trust horizon | 0 |
| $n_{\max}(\mathrm{m, m/s})$ | measurement noise bound | 0 |

measurement with a known delay of 0 s, 0.2 s, and 0.4 s, i.e. $h = 0, 1, 2$, respectively. With smaller delay, the ego vehicle decelerates the latest and catches up with the front vehicle the most quickly and reaches a smaller constant distance gap ($t \geq 30\,\mathrm{s}$); this is clearly seen in the zoomed section in the plots. As a result of a shorter constant distance gap, the smaller the delay is, the more saving on wheel energy, fuel, or battery the ego vehicle achieves during the steady state, as seen in Table 4.2a.

### Effects of the Front Vehicle's Maximum Deceleration

When there is no communication between the two vehicles, the ego vehicle may assume that the front vehicle can perform maximum braking (at its physical limit) at any time. Unfortunately, this assumption causes a conservative behavior to the following vehicle i.e. a larger inter-vehicular distance gap. This can be mitigated if the ego vehicle receives the limit of the front vehicle's maximum braking in its V2V message as shown in equation (4.13).

This section shows how the magnitude of the front vehicle's maximum deceleration can affect the performance of the ego vehicle. First, it affects the shape of a robust control invariant set used in (4.6e). Figure 4.4 shows the robust control invariant sets obtained for different magnitudes of the front vehicle's maximum deceleration; the smaller the deceleration magnitude, the bigger the robust control invariant set. Second, with a smaller magnitude of the front vehicle's maximum deceleration, the front vehicle acceleration predictor in (4.16) is less conservative and produces a velocity trajectory with less braking. As a result of these two effects, the magnitude of the front vehicle's maximum deceleration influences the performance of the LMC (4.6)-(4.8), (4.14).

Table 4.2: Comparison of LMC controllers. Units for energy consumption are %.

| | Wheel | Fuel in FC mode | Battery in FE mode |
|---|---|---|---|
| front vehicle | 100 | 100.0 | 100.0 |
| $h = 2$ | 92.5 | 92.7 | 91.7 |
| $h = 1$ | 91.6 | 92.3 | 91.3 |
| $h = 0$ | 90.6 | 91.9 | 90.8 |

(a) Effects of measurement delay.

| | Wheel | Fuel in FC mode | Battery in FE mode |
|---|---|---|---|
| front vehicle | 100.0 | 100.0 | 100.0 |
| $a^f_{\min} = -9\text{m/s}^2$ | 93.2 | 93.4 | 92.3 |
| $a^f_{\min} = -6\text{m/s}^2$ | 90.6 | 91.9 | 90.8 |
| $a^f_{\min} = -3\text{m/s}^2$ | 87.6 | 91.2 | 89.1 |

(b) Effects of front vehicle maximum deceleration.

| | Wheel | Fuel in FC mode | Battery in FE mode |
|---|---|---|---|
| front vehicle | 100.0 | 100.0 | 100.0 |
| $N_T = 0$ | 90.6 | 91.9 | 90.8 |
| $N_T = 3$ | 87.6 | 91.4 | 89.6 |
| $N_T = 8$ | 85.2 | 90.3 | 89.5 |

(c) Effects of trust horizon.

Figure 4.5 depicts the trajectories of the front and the ego vehicles, in terms of their inter-vehicular distance, velocities, and wheel torque when the front vehicle sends to the ego vehicle its maximum braking rates of $-9\,\text{m}\,\text{s}^{-2}$, $-6\,\text{m}\,\text{s}^{-2}$, and $-3\,\text{m}\,\text{s}^{-2}$. The smaller the front vehicle's maximum deceleration magnitude, the faster the ego vehicle reaches a constant velocity and distance. Moreover, the ego vehicle's constant distance gap is smaller with a smaller front vehicle maximum deceleration magnitude. As a result of a shorter constant distance gap, the smaller the front vehicle's maximum deceleration assumption is, the more saving on wheel energy, fuel, or battery the ego vehicle achieves during the steady state, as seen in Table 4.2b. .

**Remark 1.** *Another benefit of knowing maximum acceleration and deceleration of the preceding vehicle in advance is that this information can be utilized to guarantee the string stability of our CACC controller as done in [67, Eq. (25)].*

### Effects of Trusting Acceleration Forecast

This section is devoted to the scenario where the front vehicle shares its acceleration forecasts with $N_T > 0$. Similar approach and analysis were conducted in [69] for the purpose of

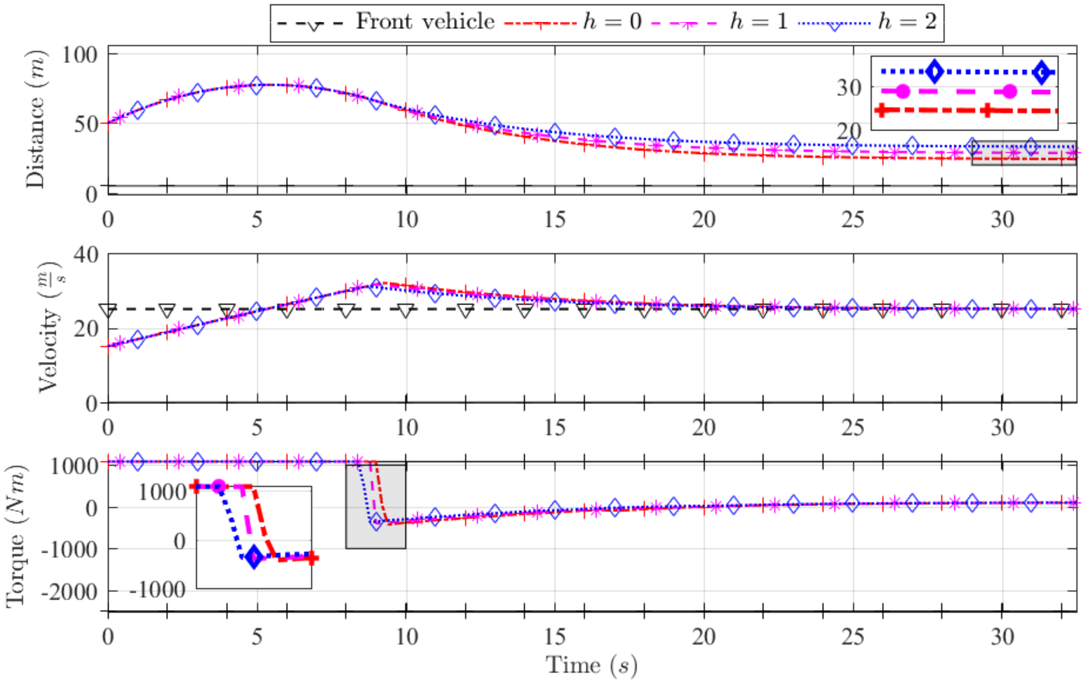


Figure 4.3: Plots of the distance between the ego and the front vehicles, their velocities, and the wheel torques of the ego vehicle at different measurement delay $h$.

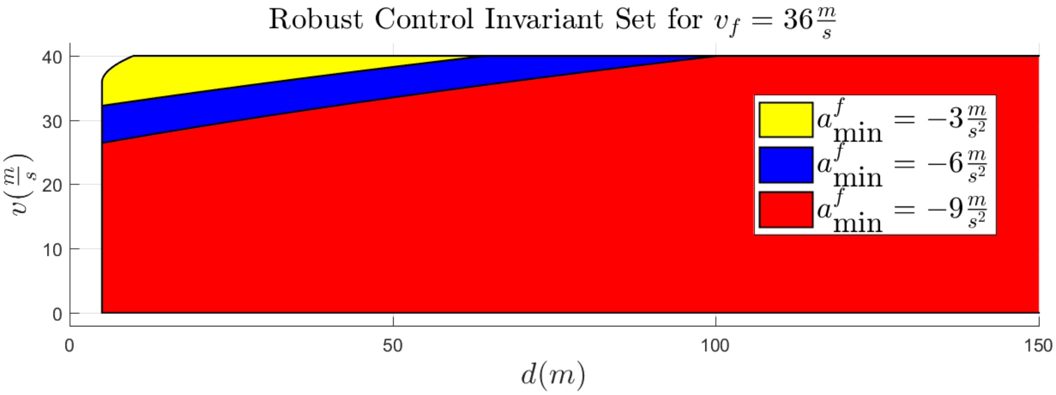


Figure 4.4: Robust control invariant set for different front vehicle's maximum deceleration rates

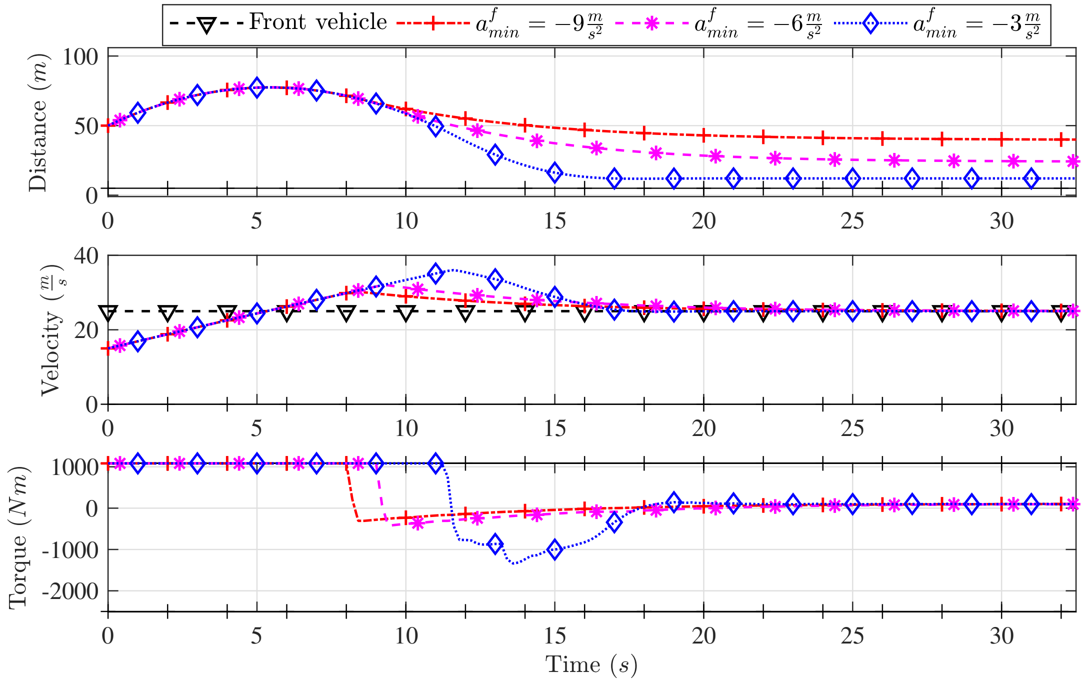


Figure 4.5: Plots of the distance between the ego and the front vehicles, their velocities, and the wheel torques of the ego vehicle at the front vehicle's different maximum braking rates $a^f_{\text{min}}$.

maximizing road throughput. It is reported in [69, 70] that the string stability is achieved when the trust horizon $N_T$ is long enough. In this simulation, we focus on the energy saving effect of a length of a trust horizon $N_T$ in the V2V message (4.13).

Figure 4.6 depicts the trajectories of the front and the ego vehicles, in terms of their inter-vehicular distance, velocities, and wheel torque when the front vehicle sends to the ego vehicle the acceleration forecast with a trust horizon of 0, 3, and 8 steps (0s, 0.6s, and 1.6s, respectively). A longer trust horizon has a similar effect to a smaller maximum deceleration magnitude of the front vehicle. In short, with a longer trust horizon, the ego vehicle reaches a constant distance gap and velocity faster and this constant gap is smaller. As seen in Table 4.2c, The longer the trust horizon is, the more energy saving on wheel energy, fuel, and battery the ego vehicle achieves during the steady state.

### Effects of the Front Vehicle's Distance/Velocity Measurement Noise Magnitude

Communication improves the accuracy of the environment perception by allowing the direct exchange of states among vehicles. With only a radar/camera, noise magnitudes can be significant [68]. In this section we study the scenario where the V2V message from the front vehicle offers measurement of the front vehicle velocity and location with lower measurement noises. To simulate the measurement noise, we generate uniformly distributed random num-

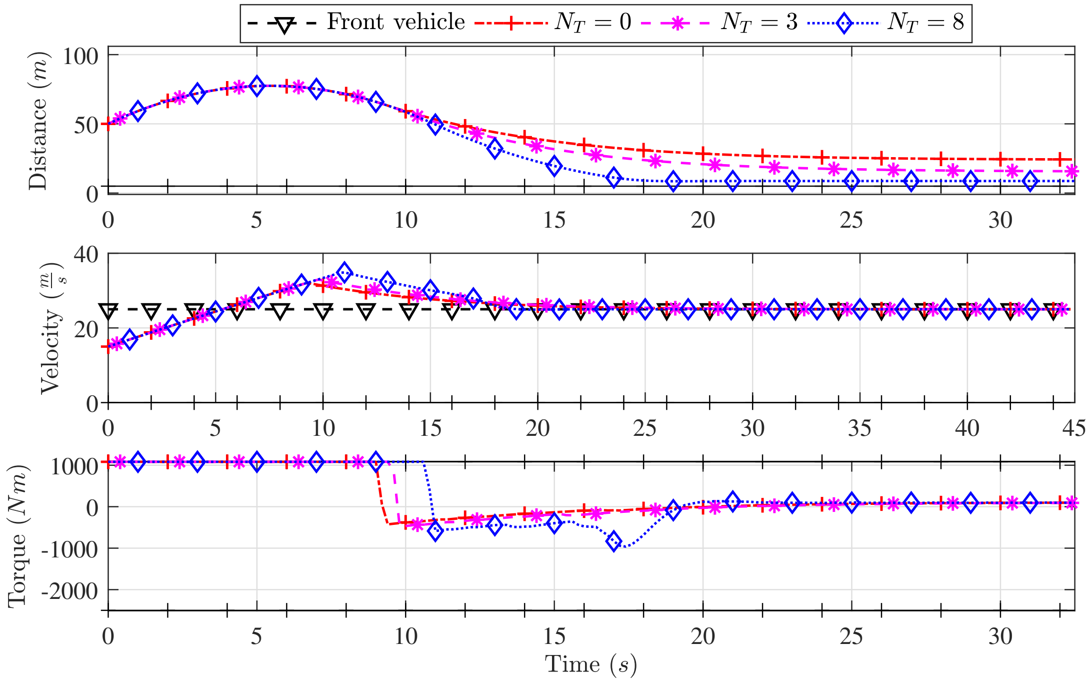


Figure 4.6: Plots of the distance between the controlled and the front vehicles, their velocities, and the wheel torques of the ego vehicle with different trust horizon $N_t$.

bers, of which magnitudes are bounded by $n_{\max}$, and add them to the distance and front vehicle velocity measurements in (4.15a).

Figure 4.7 shows the trajectories of the front and the ego vehicles, in terms of their inter-vehicular distance, velocities, and wheel torque when the ego vehicle measures the front vehicle distance with noise magnitudes bounded by 0m, 0.15m, and 0.3m and the front vehicle velocity with noise magnitudes bounded by 0m/s, 0.15m/s, and 0.3m/s. One can observe, as expected, that the bigger noise bounds result in larger fluctuations in the wheel torque trajectories.

## 4.4 Eco-LMC for Velocity Smoothing Using V2V

### 4.4.1 Abstract

This section presents a non-cooperative energy saving LMC, namely *eco-LMC*. It uses predictions of each vehicle's trajectory via V2V communication, and accordingly adapts the velocity trajectory, avoiding energy wasteful behaviors and achieving eco-driving.

The scope of our eco-LMC is to improve energy efficiency in both the *wheel-to-meters* and the *wheel-to-tank* energy conversion steps [44]. The wheel-to-meters energy conversion step is the conversion of mechanical energy at the wheel into kinetic and potential energy, which

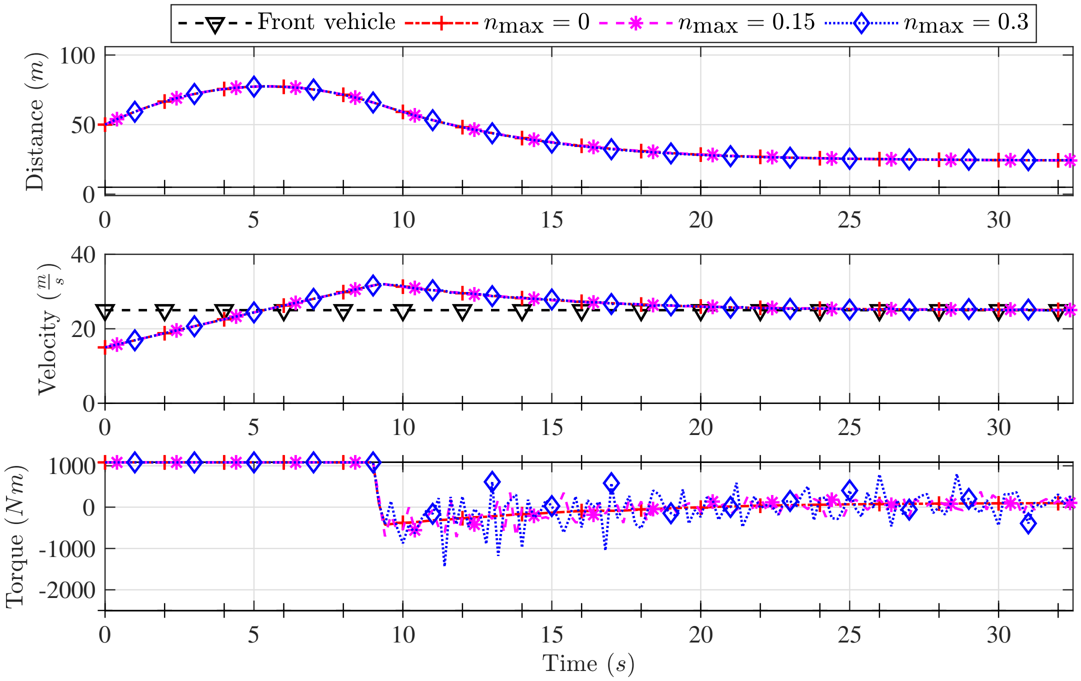


Figure 4.7: Plots of the distance between the controlled and the front vehicles, their velocities, and the wheel torques of the ego vehicle with different maximum magnitudes of the front vehicle velocity and distance $n_{\max}$.

causes the vehicle to move. The wheel-to-tank energy conversion step is the conversion of mechanical energy into the energy stored in fuel, battery packs, etc. In this work the energy usage in the *wheel-to-meters* conversion is reduced by preventing braking (especially, hydraulic braking) and inefficient acceleration torque demands while the energy saving in the *wheel-to-tank* conversion improves by maximizing regenerative braking during braking.

### 4.4.2 Longitudinal Vehicle and Regenerative Brake Dynamics Modeling

Our eco-LMC exploits the regenerative braking dynamics as well as the longitudinal vehicle dynamics which takes into account of the ego vehicles longitudinal velocity and its interaction with the front vehicle. Additionally considering the regenerative braking dynamics, we seek to maximize the usage of regenerative braking in the braking events. While the the longitudinal vehicle dynamics is the same as the one in (4.1), the first order dynamic model is used for the actuation of regenerative braking. Note that we assume that there is no delay for the traction torque and the hydraulic braking forces unlike the regenerative braking force. This assumption is based on our observation that our system (our test vehicle) does not instantly execute the regenerative braking force as requested because doing so can deteriorate the motor and battery health.

Finally, at time $t$, the discrete-time state update equations with the sampling time $t_s$ are

$$d(k+1|t) = d(k|t) + t_s(v_f(k|t) - v(k|t)), \tag{4.17a}$$

$$v(k+1|t) = v(k|t) + \frac{t_s}{m}\left(F_w(k|t) - F_b^h(k|t) - F_b^r(k|t) - F_R(\vartheta(t), d(k|t), v(k|t))\right), \tag{4.17b}$$

$$F_b^r(k+1|t) = \left(1 - \frac{t_s}{\tau}\right) F_b^r(k|t) + \frac{t_s}{\tau} F_{b,des}^r(k|t) \tag{4.17c}$$

where the state variables $d$, $v$, and $T_r^b$ denote the predicted distance, speed, and regenerative braking torque of the ego vehicle, respectively; the input variables $F_w$, $F_b^h$, and $F_{b,des}^r$ denote the predicted total accelerating torque, total braking torque, and desired regenerative braking torque, respectively; the resistance force $F^R(\bullet)$, which includes aerodynamic drag, rolling resistance, and gravitational force, is approximated by (3.2) with parameters listed in Table. 3.2.

Compactly, the dynamic model (4.17) is indicated by

$$x(k+1|t) = f_r(x(k|t),\, u(k|t),\, v_f(k|t), \vartheta(t)) \tag{4.18}$$

where

$$\begin{aligned} x(k|t) &= [d(k|t), v(k|t), F_b^r(k|t)]^\top, \\ u(k|t) &= [F_w(k|t), F_b^h(k|t), F_{b,des}^r(k|t)]^\top. \end{aligned}$$

### State/Input Constraints

In addition to the state constraints listed for the LMC in (4.4a), we impose input constraints to represent physical limitations of the motor, engine, and hydraulic brake actuators in our vehicle. They are expressed as

$$\begin{aligned} 0 &\le F_w(k|t) \le F_{\max}^w, \\ 0 &\le F_b^h(k|t) \le F_{b,\max}^h. \\ 0 &\le T_{r,des}^b \le F_{b,\max}^r(t). \end{aligned} \tag{4.19}$$

where $F_{b,\max}^r(t)$, which denotes the maximum regenerative braking capacity, is provided by the vehicle system which monitors the battery state-of-charge (SOC). Moreover, to avoid aggressive regenerative braking demands which can deteriorate the motor and battery health, we enforce

$$F_{r,des}^b(k|t) - F_{r,des}^b(k-1|t) \le \Delta F_{b,\max}^r \tag{4.20}$$

where $k \ge 1$ and $\Delta F_{b,\max}^r$ is the maximum increase in desired regenerative braking torque.

Summarizing, the input constraints are compactly expressed as

$$u(k|t) \in \mathbb{U}_r(t). \tag{4.21}$$

Moreover, for useful notations in the remainder of this section, we define coasting dynamics and regenerative braking dynamics as:

$$x_{\text{coast}}(k+1|t) = f_r(x_{\text{coast}}(k|t),\, u_{\text{coast}},\, v_f(k|t), \vartheta(t)), \tag{4.22a}$$

$$x_{\text{regen}}(k+1|t) = f_r(x_{\text{regen}}(k|t),\, u_{\text{regen}}(t),\, v_f(k|t), \vartheta(t)) \tag{4.22b}$$

where $u_{\text{coast}} = [0, 0, 0]^\top$ and $u_{\text{regen}}(t) = [0, F^r_{b,\max}(t), 0]^\top$.

### 4.4.3 Learning-Based Model Predictive Control for Eco-LMC

In order to use a front vehicle velocity forecast to improve energy efficiency while preventing front-end collisions, the MPC design is based on a LBMPC scheme from [62].

#### Stage Cost

The energy efficient driving strategies for HEVs include moderate acceleration, maintaining even driving pace, reducing wasteful braking, maximizing regenerative braking in the braking events, etc [59]. Therefore, the objectives of our eco-LMC are designed to reflect these aspects of energy-efficient driving while tracking an user-defined velocity.

#### Velocity/Distance Tracking

The principal objective of the eco-LMC is to maintain the ego vehicle close to an user-defined velocity $v_{\text{des}}$ or an user-defined distance $d_{\text{des}}$. In the MPC problem, the cost function related to velocity tracking is defined as

$$l_v(x(t)) = Q_v(v(k|t) - v_{\text{des}})^2 + Q_d(d(k|t) - v_{\text{des}})^2 \tag{4.23}$$

where $Q_v \geq 0$ is a weight constant to penalize velocity deviation from a desired velocity.

#### Brake Penalty

The eco-LMC seeks to minimize power dissipation from braking. As depicted in Figure 2.4 (b), our HEV utilizes two types for braking methods: regenerative braking and friction braking generated by the traction electric motor and the hydraulic system, respectively. Whereas hydraulic braking totally wastes braking energy through friction and heat, regenerative braking recuperates some portion of energy by using the electric motor as a generator and stores it in the on-board electric battery. However, there is still some energy loss even in regenerative braking due to emerging heat and generator inefficiency.

Accounting for this difference in power loss, our cost function introduces braking penalty using the following function:

$$l_{brk}(x(t), u(t)) = Q^b_r\Big(F^r_b(k+1|t) + F^r_b(k+1|t)^2\Big) + Q^b_h\Big(F^h_b(k|t) + F^h_b(k|t)^2\Big) \tag{4.24}$$

where $Q^b_r \geq Q^b_h \geq 0$. $Q^b_r$ and $Q^b_h$ are weight constants to penalize regenerative braking and hydraulic braking, respectively.

### Aggressive Torque Penalty

Avoiding aggressive torque is also a key principle behind eco-driving [71, 72]. Therefore, the eco-LMC also seeks to minimizing jerk

$$l_{jrk}(u(t)) = \\ Q_{jrk}\bigg(\Big(F_w(k|t) - F_b^r(k|t) - F_b^h(k|t)\Big) - \Big(F_w(k+1|t) - F_b^r(k+1|t) - F_b^h(k+1|t)\Big)\bigg)^2 \tag{4.25}$$

where $Q_{jrk} \geq 0$ denotes a weight constant.

### Terminal Cost

It is generally accepted that the MPC can achieve better performance with a longer horizon. In particular, given that the front vehicle velocity prediction is available for a long horizon, the eco-LMC can benefit from having a long MPC horizon. However, increasing the horizon results in heavier computational burden and infeasibilty for real-time computation of the MPC solution. A common approach of avoiding myopic (short-sighted) performance with a short horizon for real-time computation is to employ the terminal cost, a cost applied to the state at the final prediction horizon step. This terminal cost function should characterize the cost-to-go for the longer horizon.

In this work, we propose a design of terminal cost to approximate inevitable braking in the future and, therefore, prevent wasteful excessive acceleration in advance. The idea of penalizing future braking was proposed in our previous work [20] and shown effective experimentally in energy saving for an internal combustion engine vehicle. The approach in [20] uses terminal set constraints named *coasting set* to enforce the vehicle to coast earlier when it foresees braking. However, this approach only considers a single braking source and, therefore, when directly applying it to HEVs, treats the hydraulic and regenerative braking equally although the regenerative braking can waste less energy than the hydraulic braking. Moreover, it requires that the front vehicle velocity prediction is perfect. Our terminal cost approach in this work overcomes these limitations by penalizing the hydraulic and regenerative braking differently and allowing imperfect prediction of the front vehicle velocity by employing a learning-based MPC approach.

First we define a coasting set $\mathbb{C}_{\text{coast}}$ and a regenerative braking set $\mathbb{C}_{\text{regen}}$ as followings:

$$\begin{aligned} \mathbb{C}_{\text{coast}} :=& \big\{(d(t+N_p|t), v(t+N_p|t)) \mid \text{dynamics (4.22a)}, \\ & d(k) \geq d_{\text{safe}}\, \forall k = t+N_p, ..., t+N_c\big\}, && \text{(4.26a)} \\ \mathbb{C}_{\text{regen}} :=& \big\{(d(t+N_p|t), v(t+N_p|t)) \mid \text{dynamics (4.22b)}, \\ & d(k) \geq d_{\text{safe}}\, \forall k = t+N_p, ..., t+N_c\big\}. && \text{(4.26b)} \end{aligned}$$

$\mathbb{C}_{\text{coast}}$ and $\mathbb{C}_{\text{regen}}$ approximate the distance and velocity sets such that the ego vehicle maintains the safety distance with the front vehicle for the remaining horizon of the front vehicle velocity prediction when the vehicle is coasting and applying maximum regenerative braking, respectively.

$\mathbb{C}_{\text{coast}}$ and $\mathbb{C}_{\text{regen}}$ can be constructed a-priori to the MPC computation. First, $\mathbb{C}_{\text{coast}}$ and $\mathbb{C}_{\text{regen}}$ are the collection of the distance constraint between time $k = t + N_p$ and $k = t + N_c$ and they can be expressed as:

$$\begin{aligned}
&\forall k = t + N_p, ...t + N_c, && \text{(4.27a)}\\
&\quad d(k) \geq d_{\text{safe}}, && \text{(4.27b)}\\
&\Rightarrow d(t+N_p|t) + t_s \sum_{i=t+N_p}^{i=k} \big(v_f(i|t) - v(i|t)\big) \geq d_{\text{safe}}, && \text{(4.27c)}\\
&\Rightarrow d(t+N_p|t) \geq d_{\text{safe}} - t_s \sum_{i=t+N_p}^{i=k} \big(v_f(i|t) - v(i|t)\big) && \text{(4.27d)}
\end{aligned}$$

where $v(i|t)$ is a function of $v(t+N_p|t)$; it is the maximum of zero or the velocity at time step $i$ using the dynamics (4.22a) and (4.22b) for $\mathbb{C}_{\text{coast}}$ and $\mathbb{C}_{\text{regen}}$, respectively, with the initial velocity of $v(t+N_p|t)$. Now, it is clear and straightforward that (4.27) can be expressed as:

$$d(t+N_p|t) \geq d_{\text{safe}} - t_s \Delta v^*(v(t+N_p|t)) \tag{4.28}$$

where $\Delta v^*(v(t+N_p|t)) = \min_{j=N_p,\ldots,N_c} \Big( \sum_{i=t+N_p}^{i=t+j} \big(v_f(i|t) - v(i|t)\big) \Big)$. Then, we grid the admissible velocity space from 0 to in $v_{\max}$ in (4.4a), and for each gridded velocity as the initial velocity $v(t+N_p|t)$, we can formulate the constraint (4.28).

We simply approximate this set of constraints as a set of linear constraints; i.e.,

$$d(t+N_p|t) \geq m(l)v(t+N_p|t) - b(l) \quad \forall\, l = 1, ..., N_{\text{grid}} - 1 \tag{4.29}$$

where

$$\begin{aligned}
&v^*(l) := \text{l-th velocity in the gridded velocity space}, && \text{(4.30a)}\\
&d^*(l) = d_{\text{safe}} - t_s \Delta v^*(v^*(l)), && \text{(4.30b)}\\
&m(l) = \frac{d^*(l+1) - d^*(l)}{v^*(l+1) - v^*(l)}, && \text{(4.30c)}\\
&b(l) = -d^*(l) + v^*(l)m(l), && \text{(4.30d)}\\
&N_{\text{grid}} := \text{the number of gridded velocity}. && \text{(4.30e)}
\end{aligned}$$

Figure 4.8 depicts the examples of these two sets when the front vehicle is running at a constant velocity of 20m/s. Finally, our eco-LMC MPC approximates the terminal cost $V_f(x(N_p|t))$ as how much the state at the final prediction horizon step violates these linear

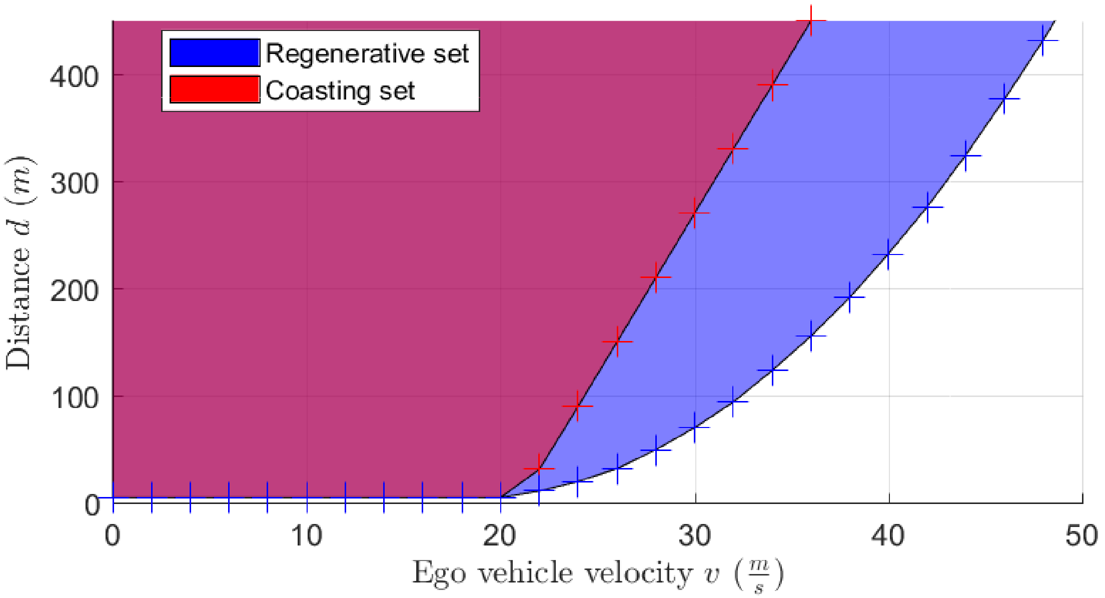


Figure 4.8: Coasting (red) and Regenerative (blue) sets. Marker indicates the minimum distance at each gridded velocity.

constraints; i.e.

$$\begin{aligned} V_f(x(t+N_p|t)) &= Q_{\text{coast}} z_{\text{coast}}^2 + Q_{\text{regen}} z_{\text{regen}}^2 \\ &\text{where} \\ &\quad d(t+N_p|t) + z_{\text{coast}} \geq m_{\text{coast}}(k) v(t+N_p|t) - b_{\text{coast}}(k), \\ &\quad d(t+N_p|t) + z_{\text{regen}} \geq m_{\text{regen}}(k) v(t+N_p|t) - b_{\text{regen}}(k), \\ &\quad \forall\, k = 1, ..., N_{\text{grid}} - 1. \end{aligned}$$

where $Q_{\text{coast}}$ and $Q_{\text{regen}}$ are non-negative weight constants.

Summarizing, the total cost function of our MPC problem is compactly expressed as

$$\begin{aligned} J(\boldsymbol{x(t)}, \boldsymbol{u(t)}) = \sum_{k=t}^{t+N_p-1} \Big( l_v(x(k|t)) + \zeta^k l_{brk}(x(k|t), u(k|t)) + l_{jrk}(u(k|t)) + \epsilon l_{exc}(u(k|t)) \Big) \\ + V_f(x(t+N_p|t)) \end{aligned} \tag{4.31}$$

where $\boldsymbol{x(t)} := [x(t|t), x(t+1|t), \ldots, x(t+N_p|t)]^\top$ and $\boldsymbol{u(t)} := [u(t|t), u(t+1|t), \ldots, u(t+N_p-1|t)]^\top$ and $\zeta \in [0, 1]$ is a parameter to penalize earlier braking than braking later and $l_{exc}(u(k|t)) := u(k|t)^\top u(k|t)$ is to penalize input excitation and $\epsilon$ is a small positive number.

**Finite-Time Constrained Optimization Problem**

The proposed eco-LMC is based on a learning based MPC framework which allows the system to use a front vehicle velocity forecast for performance improvement (4.31) while using a robust velocity forecast to enforce safety constraints (4.4a) and (4.21) for infinite time. At time $t$, our eco-LMC solves the following finite-time constrained optimization problem;

$$\min_{u(\cdot|t)} \; J(\widetilde{\boldsymbol{x}}(\boldsymbol{t}), \boldsymbol{u}(\boldsymbol{t})) \tag{4.32a}$$

subject to

$$\hat{x}(t|t) = \bar{x}(t), \quad \widetilde{x}(t|t) = \bar{x}(t), \quad \hat{v}_f(t|t) = \bar{v}_f(t), \quad \widetilde{v}_f(t|t) = \bar{v}_f(t), \tag{4.32b}$$

$$\hat{x}(k+1|t) = f_r(\hat{x}(k|t), u(k|t), \hat{v}_f(k|t), \bar{\vartheta}(t)), \tag{4.32c}$$

$$\widetilde{x}(k+1|t) = f_r(\widetilde{x}(k|t), u(k|t), \widetilde{v}_f(k|t), \bar{\vartheta}(t)), \tag{4.32d}$$

$$\hat{v}_f(k+1|t) = \hat{v}_f(k|t) + t_s a^f_{\min}(\hat{v}_f(k|t)), \quad \widetilde{v}_f(k+1|t) = \widetilde{v}_f(k|t) + t_s \widetilde{a}^f(k|t) \tag{4.32e}$$

$$[\hat{d}(t+N_p|t),\, \hat{v}(t+N_p|t)] \in \mathbb{X},\, u(k|t) \in \mathbb{U}_r(t), \tag{4.32f}$$

$$\forall k \in [0, ..., N_p - 1],$$

$$[\hat{d}(t+N_p|t),\, \hat{v}(t+N_p|t),\, \hat{v}_f(t+N_p|t)] \in \mathbb{C}, \tag{4.32g}$$

$$\tag{4.32h}$$

where $\bar{x}(t)$ and $\bar{x}(t)$ are the state and the front vehicle velocity measured at time step $t$; $a^f_{\min}(v_f(k|t)) := \mathbb{1}_{v_f>0} a^f_{\min}$ is the worst case (full braking) front vehicle deceleration prediction at $k$ time steps ahead given the information at time step $t$ and $\widetilde{a}_f(k|t)$ $\hat{x}_f(k|t)$ is an acceleration forecast at $k$ time steps ahead given the information at time $t$; $\bar{\vartheta}(t)$ is the road slope estimated at time $t$.

At time step $t$, the first input from the solution of (4.32) is applied to the system during the time interval $[t, t+1)$. At the next time step $t+1$, a new optimal control problem in the form of (4.32), based on new measurements of the state, is solved over a shifted horizon, yielding a *moving* or *receding* horizon control strategy with control law:

$$u(t) = u^*(t|t). \tag{4.33}$$

where $u^*(t|t)$ is the first element of the solution of (4.32).

**Remark 2.** *Assuming that the road slope estimation is perfect, the proposed eco-LMC* (4.32)-(4.33) *is persistently feasible and, therefore, the ego vehicle is safe at all times, i.e.,* $x(t) \in \mathbb{X}$ *and* $u(t) \in \mathbb{U}_r(t)$ *for all* $t$.

### 4.4.4 Simulation Results

Pure software-based simulation are conducted to explore the energy saving potential in different driving scenarios. The vehicle longitudinal dynamics model in the simulation environment is identified in (3.1) in Chapter 3 while the powertrain model is borrowed from

Table 4.3: Environment and controller parameters for different scenarios.

| **Parameter (unit)** | **Description** | **Value** |
|:---:|:---:|:---:|
| $N_p$ | MPC horizon | 20 |
| $N_c$ | front vehicle acceleration forecast horizon | 150 |
| $t_s(s)$ | sampling time | 0.2 |
| $d_{\min}(\mathrm{m})$ | minimum distance | 5 |
| $v_{\max}(\mathrm{m/s})$ | maximum velocity | 40 |
| $v_{\mathrm{des}}(\mathrm{m/s})$ | maximum velocity | 40 |
| $F^w_{\max}(\mathrm{N})$ | maximum wheel torque | $\frac{1083}{R_w}$ |
| $F^h_{b,\max}(\mathrm{N})$ | maximum hydraulic braking force | $\frac{1083}{R_w}$ |
| $Q_v$ | velocity tracking penalty | 100 |
| $Q_d$ | distance tracking penalty | 0 |
| $Q^b_r$ | hydraulic braking penalty | 2000 |
| $Q^b_h$ | regenerative braking penalty | 800 |
| $\zeta$ | earlier braking factor | 0.9 |
| $Q_{jrk}$ | jerk penalty | 400 |
| $Q_{\mathrm{coast}}$ | terminal coast weight 1 | 100 |
| $Q_{\mathrm{regen}}$ | terminal coast weight 2 | 200 |
| $a^f_{\min}(\mathrm{m/s^2})$ | front vehicle minimum acceleration | 0 |

and validated in the work of [34]. This section presents the simulation study comparing the energy efficiency between the proposed controller with other baseline approaches. Controller parameters are fixed throughout the simulations and listed in Table 4.3.

### Simulation Setup

Our simulation environment is built in MATLAB Simulink which consists of: i) experimentally validated high fidelity vehicle dynamics and powertrain models of the PHEV test vehicle, ii) front vehicle velocity and slope variations for a recorded route, iv) an GUROBI solver for solving the optimization problem (4.32) [73].

To objectively show the efficacy of the proposed eco-LMC, we compare the efficiency improvements between our proposed controller with a perfect velocity prediction (hereafter named *Eco-LMC*), constant velocity prediction (hereafter named *CV-Eco-LMC*), or robust velocity prediction (hereafter named *Rob-Eco-LMC*) and other baseline approaches. The baseline controllers includes:

1. a front vehicle which runs a predetermined velocity trajectory

2. the eco-LMC (4.6)-(4.8) without a terminal cost $V_f(x(t+N_p|t))$ but with a perfect velocity prediction for the MPC horizon (hereafter named as *NT-LMC*).

In the next two subsections we present simulation results in two car following scenarios: a sinusoidal velocity catch-up and a car-following on highway.

### Sinusoidal Velocity Catch-Up

The first simulation demonstrates a catch-up of the front vehicle driving at a sinusoidal velocity profile on a flat road. The objective of this simulation is to exhibit and compare the force and velocity trajectories of the ego vehicle with our eco-LMC (4.6)-(4.8) and to highlight the velocity smoothing aspect.

Figure 4.9 depicts the closed loop trajectories of the ego vehicle and the front vehicle in terms of their distance, velocities, wheel torques, and regenerative braking torques. A few remarks from the plot follow.

- The distance and velocity constraints are satisfied at all times, regardless of the front vehicle prediction models and controllers.
- Velocity of the ego vehicle is smoother than that of the front vehicle.
- Rob-Eco-LMC has the most smooth velocity and torque trajectory with the least amount of braking torque. However, it tracks the front vehicle in a very conservative manner, causing a very large gap with the front vehicle ($d \geq 200m$).
- Eco-LMC with the perfect velocity prediction generate a smooth velocity trajectory with small amount of braking. Moreover, the distance tracking performance is not as sacrificed as that of Rob-Eco-LMC.
- CV-Eco-LMC and NT-LMC have similar input and state trajectories.

Table 4.4 reports the savings in terms of positive wheel energy and regenerative braking energy. First, regardless of the front vehicle velocity prediction model, the ego vehicle with the proposed Eco-LMC saves positive wheel energy at least by 50%. Among every longitudinal motion controller, the Rob-Eco-LMC consumes the least amount of positive wheel energy and regenerative braking. However, the drawback of the Rob-Eco-LMC is that it tracks the front vehicle in a conservative manner, causing a very large gap with the front vehicle ($d \geq 200m$). In real life, this large inter-vehicular gap causes frequent vehicle cut-ins in front of the ego vehicle, resulting in inefficient driving (with more trip time and unnecessary braking) and, thus, more energy consumption.

The next most efficient controller is the Eco-LMC with the perfect velocity prediction, saving 63% in positive wheel energy. Unlike the Rob-Eco-LMC, the Eco-LMC with the perfect velocity prediction makes the ego vehicle maintain a close distance with the front vehicle. In fact, one can tell that the distance becomes a little larger than those of the other controllers only when there is an upcoming braking event in the next 30s. This is because the Eco-LMC utilizes the whole 30 second predictions of the front vehicle using the terminal cost $V_f(\cdot)$, making itself far-sighted enough to optimize its input trajectory in advance.

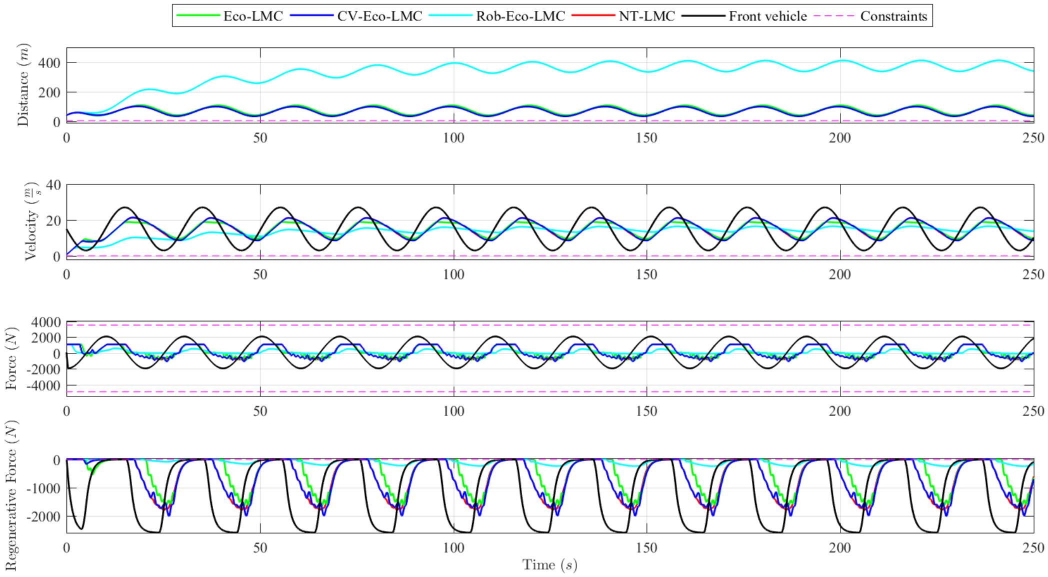


Figure 4.9: Plots of the distance to the front vehicle, velocities, and wheel demand and regenerative braking torques.

Table 4.4: Comparison of ACC controllers in sinusoidal Catch-up

| ACC strat. ($v_f$ model) | Positive work (%) | Regen. work (%) |
|---|---:|---:|
| Front vehicle | 100 | 100 |
| Eco-LMC (perfect) | 36.6 | 50.7 |
| CV-LMC (constant) | 49.6 | 75.9 |
| Rob-Eco-LMC (robust) | 15.1 | 9.1 |
| NT-LMC (perfect) | 49.8 | 75.9 |

CV-LMC and NT-LMC have similar energy saving performance. CV-LMC which imposes the terminal cost $V_f(\cdot)$ with the constant front vehicle velocity prediction does not help energy efficiency much when comparing it to the NT-LMC which does not impose any terminal cost. This result as well as the result from the Rob-Eco-LMC tells us that having a good forecast of the front vehicle velocity prediction is important in making the proposed eco-LMC more energy efficient while ensuring that the distance to the front vehicle does not become too large.

### Simulation Results: Car-Following on Highway

This simulation demonstrates the longitudinal motion of the ego vehicle following, on the same lane, the front vehicle driving a recorded human-driven highway velocity trajectory on

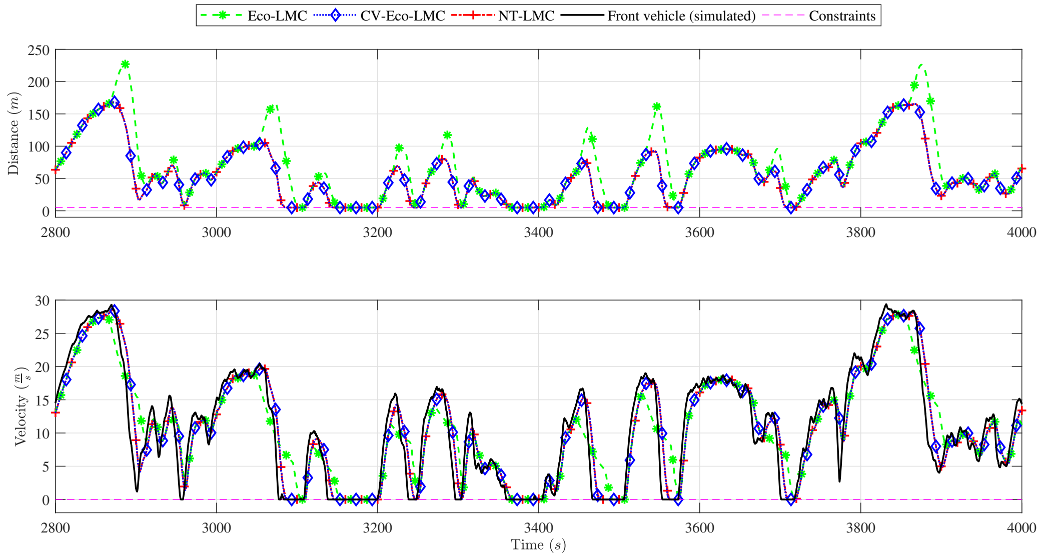


Figure 4.10: Plots of the distance to the front vehicle and velocities.

a road with changing slopes. The total length of the trip is 10km and the total trip time is 8500s. Both vehicles have an initial SOC of 95%.

Figure 4.10 shows a zoomed profiles of the distance and velocity trajectories under different controllers. As seen, Eco-LMC has the most smooth velocity trajectory. Also, Eco-LMC has distance jumps (e.g., at 2900s, 3060s, etc) to prepare for the deceleration event. Also, as seen in the sinusoidal catch-up simulation, CV-Eco-LMC has the similar velocity and distance profiles as NT-ACC. Also, it is noted that the constraints (minimum distance and velocity limits and input limits) are always satisfied for every controller.

Table 4.5 compares the MPGe and Positive work in % among different LMC controllers and the front vehicle with actual measurement and the simulated front vehicle. To measure MPGe and work of the simulated front vehicle, the wheel torque trajectory is first obtained by the recorded front vehicle velocity and slope trajectories and the vehicle dynamics model, and, then, fuel and battery consumption is calculated using the wheel torque trajectory, the powertrain model, and our model for the charge-depleting and charge-sustaining (CD-CS) strategy which is the production powertrain control strategy of our test vehicle. The MPGe and work differences between the measured front vehicle and the simulated front vehicle are within 6% and 9%, respectively. Our Eco-LMC is the most energy efficient by improving about 20% in MPGe and saving about 21% in the positive work done by the wheel when compared against the simulated front vehicle. Also, CV-LMC and NT-LMC, which have the same energy performance, also have a better energy performance than the simulated front vehicle but are not as energy efficient as ECO-LMC.

Table 4.5: Comparison of LMC controllers with the charge-depleting and charge-sustaining (CD-CS) powertrain strategy in highway car-following. MPGe calculates that 33.7 kilowatt-hours of electricity is equivalent to one gallon of gas [45].

| ACC strat. ($v_f$ model) | MPGe | Positive work (J) |
|---|---:|---:|
| Front measured vehicle | 106.0 | 109.0 |
| Front simulated vehicle | 100 | 100 |
| Eco-LMC (perfect) | 120.5 | 78.9 |
| CV-LMC (constant) | 114.5 | 84.1 |
| NT-LMC (exponential) | 114.5 | 84.1 |

### 4.4.5 Preliminary Experimental Results

Experiments are conducted on a public highway to demonstrate the practical implementation and effectiveness of the proposed controller in a real, full-scale test vehicle (see Chapter 2 for the experimental setup). We have two vehicles driving together on a highway with other traffics. The front vehicle is controlled by the human driver and the following vehicle is controlled by the CV-Eco-LMC and trying to directly follow the front vehicle. Here, we do not compare the energy consumption (MPGe and the positive work) because the two vehicles used in the experiment were in different conditions (different initial SOC, weights, tire pressure, and etc). Rather, we focus on comparing the velocity and torque profiles highlighting the smoother torque and velocity profiles of the eco-LMC compared to those of the human driver. Note that the next section (Section 5.2) will present public highway experimental results quantitatively detailing the energy saving performance of the eco-LMC and the human driver.

In order to solve the eco-LMC optimization problem (4.32), we utilized a CVXGEN solver, which generates fast custom code for solving small, quadratic program (QP) representable convex optimization problems. Because CVXGEN is only able to solve a QP, we reformulate the optimization problem (4.32) into a QP by, at every time $t$, simply converting the nonlinear dynamics (4.18) into a linear form by linearizing the model about the measured velocity $\bar{v}(t)$.

Figure 4.11 shows the distance, velocity, and wheel torque trajectories. It is seen that the minimum distance constraint is satisfied at all times. Figure 4.12 shows the zoomed section of the same experiment between $120s$ and $225s$. As seen, CV-Eco-LMC provides a smoother velocity trajectory and wheel torque trajectory. Similar trend is also shown in our second experiment depicted in Figure 4.13. As we have seen in the simulation above, the smoother velocity profile can result in improved energy efficiency. Another thing to note is that we observe a jump in the distance trajectory at about $175s$ due to the large distance gap and the vehicle cut-in. As a result, there is a sudden accelerating and braking torque. This kind of behavior can result in deteriorated energy efficiency, which is the reason that the conservative front vehicle acceleration forecast can result in lower energy efficiency.

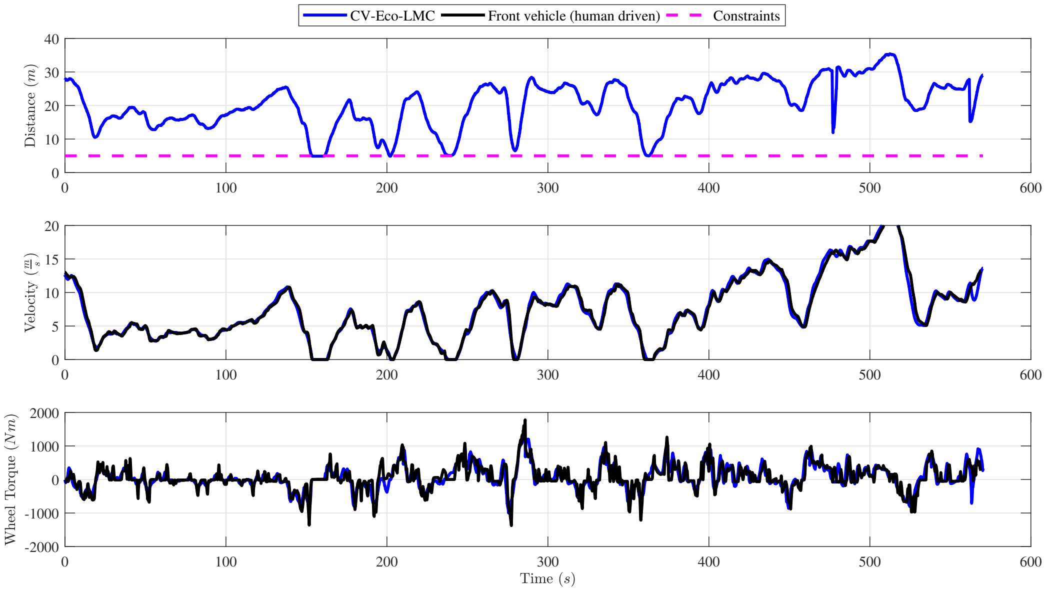


Figure 4.11: Plots of the distance to the front vehicle and velocities.

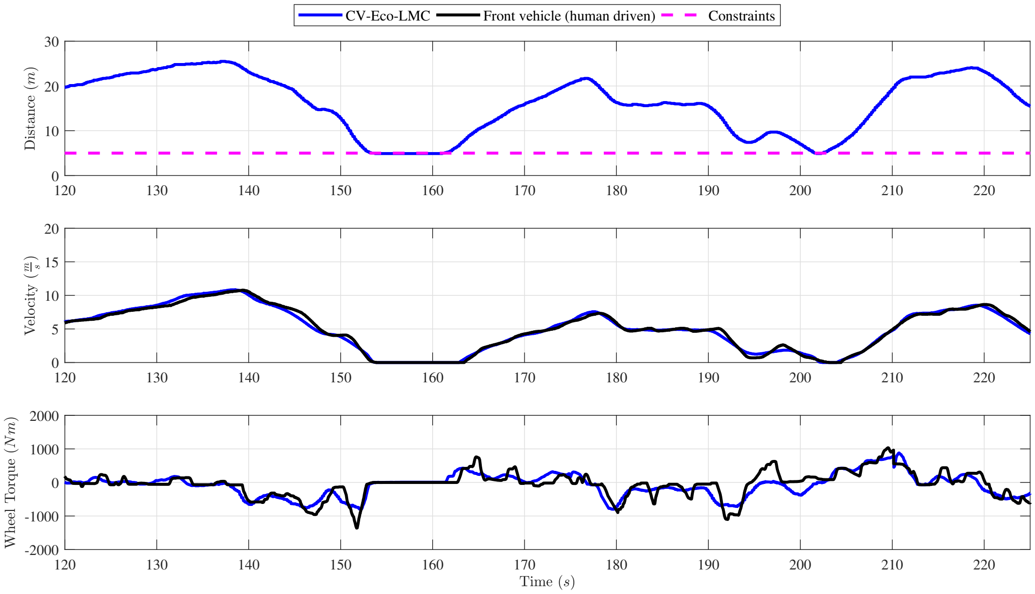


Figure 4.12: Plots of the distance to the front vehicle and velocities. Zoomed between 120$s$ and 225$s$.

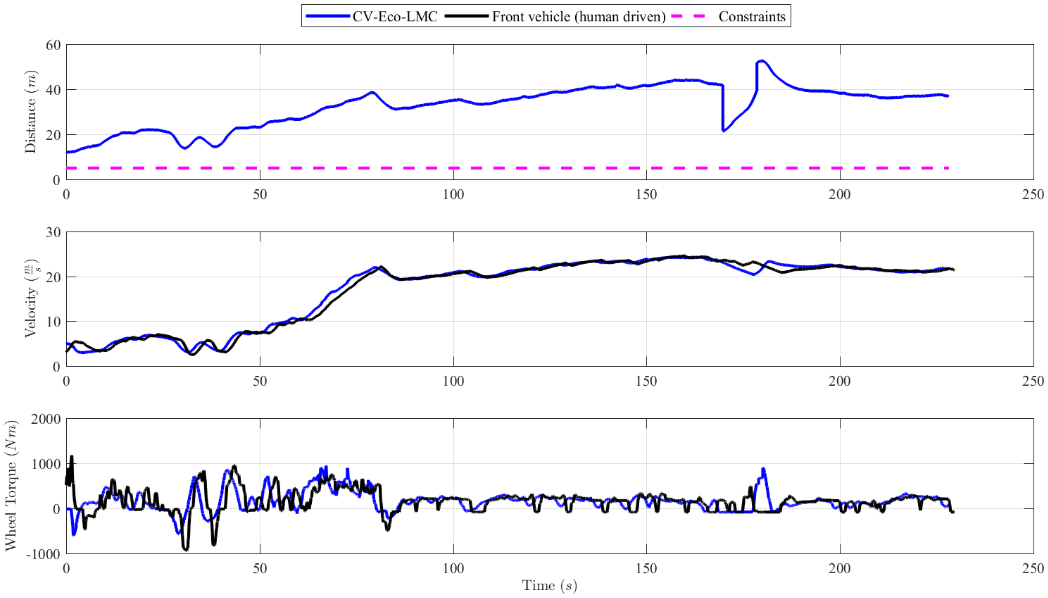


Figure 4.13: Plots of the distance to the front vehicle and velocities during short 200$s$ demonstration.

# 4.5 Eco-Cooperative Adaptive Cruise Control in Platoon

## 4.5.1 Abstract

The drawback of the eco-LMC design presented in Section 4.4 is that the controlled vehicle is, without an appropriate tuning and design for the cost function, prone to exceeding the maximum inter-vehicular distance and causing the loss of contact. Section 4.5 presents another interesting design of an eco-LMC, namely *eco-CACC*, applicable for a vehicle control in a micro-platoon. The eco-CACC exploits V2V forecasts to improve energy saving performance, while formally guaranteeing robust safety and preventing loss of contact regardless of accuracy of the acceleration forecast in the V2V communication. The objective of the eco-CACC is optimize the longitudinal motion of a single vehicle in the platoon to minimize control effort and jerk while ensuring that the controlled vehicle maintains contact with its preceding vehicle in the platoon at all times. Compared to existing works [63, 20] or the control designs presented in the previous sections, we concurrently guarantee both the minimum distance constraint to front collisions and the maximum distance constraint to avoid loss of contact.

Our eco-CACC is decentralized, i.e., each vehicle has an on-board controller which only uses data available from V2V communication and on-board range/perception sensors. Our scope is saving energy in the *wheel-to-meters* conversion step, which is directly affected by longitudinal motion control system; in general, energy usage is reduced by by preventing over-acceleration, braking, and jerk.

### 4.5.2 V2V Communication Structure

Our CAVs adopt a predecessor following communication topology as depicted in Figure 4.2. Each vehicle in the platoon sends, to the following vehicle, a V2V message containing the current value and $N_p - 1$ steps ahead prediction of the acceleration. For the $i$-th vehicle at time $t$, this message is denoted as

$$m_i(t) = \tilde{\mathbf{a}}_{\mathbf{i}}(\mathbf{t}) = [\tilde{a}_i(t|t), \tilde{a}_i(t+1|t), \ldots, \tilde{a}_i(t+N_p-1|t)], \tag{4.34}$$

where $\tilde{a}_i(k|t)$ is the predicted acceleration of the $i$-th vehicle at time $k$ using the available information at time $t$. It is assumed that message (4.34) for every vehicle in our platoon is instantly available for communication without any delay. Therefore, only the first acceleration is assumed to be the same value of the true acceleration, i.e. $\tilde{a}_i(t|t) = \bar{a}_i(t)$ where $\bar{a}_i(t)$ is the true vehicle acceleration at time $t$.

### 4.5.3 Model Predictive Control for Eco-CACC

Our eco-CACC is based on the MPC-based LMC (4.6)-(4.8) with the following modifications.

#### Simplified Vehicle Dynamics

We focus on how to design the CACC which systemically guarantees that the controlled ego vehicle avoids both collision and loss of contact with the front vehicle. For this purpose, we assume that the motion of each vehicle is governed by a simple point-mass model and, therefore, is a linear model unlike the previously identified vehicle dynamics model (3.1). Our controller takes into account of both the ego vehicle dynamics and the uncertain front vehicle dynamics. The state $x$ include the longitudinal position in a road-aligned coordinate, $s$, and the longitudinal velocity, $v$. The input is the acceleration of the controlled ego vehicle, $u$, and the uncertainty is the acceleration of the front vehicle, $a^f$. The discrete-time model update for the $i$-th controlled ego vehicle and the $(i-1)$-th front vehicle at time step $k$ given the available information at time $t$ are expressed as

$$\begin{aligned} x_i(k+1 \mid t) &= Ax_i(k \mid t) + Bu_i(k \mid t) \\ x_{i-1}(k+1 \mid t) &= Ax_{i-1}(k \mid t) + Ba^f(k \mid t) \end{aligned} \tag{4.35}$$

where $A = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} \frac{1}{2}t_s^2 \\ t_s \end{bmatrix}$, $a^f(k) \in \mathbb{A}^f(v_{i-1}(k))$ is a front vehicle acceleration and $t_s$ represents the discretization time. Note that $\mathbb{A}^f$ is defined in (4.5).

Our system is subject to state and input constraints. The input constraints represent the physical limitations of the actuator and are expressed as

$$u_i(k \mid t) \in \mathbb{U} := \{u_i \in \mathbb{R} : a_{\min} \leq u_i \leq a_{\max}\}. \tag{4.36}$$

The state constraints enforce safety and performance guarantees. First, the relative distance to the front vehicle is lower- and upper-bounded to prevent front collision and loss of contact from the platoon, respectively; this constraint takes the form

$$d_{\min} \leq s_{i-1}(k \mid t) - s_i(k \mid t) \leq d_{\max}. \tag{4.37}$$

Second, the longitudinal velocity of the vehicle is also bounded to obey speed limits; this constraint is expressed as

$$v_{\min} \leq v_i(k \mid t) \leq v_{\max}. \tag{4.38}$$

The state constraints (4.37) and (4.38) are compactly expressed as

$$[x_i(k \mid t), x_{i-1}(k \mid t)]^\top \in \mathbb{X} \tag{4.39}$$

We also introduce $\mathbb{X}_{v_{i-1}}$ as the set of states of two consecutive vehicles which don't violate the constraints (4.39), while the front vehicle velocity is $v_{i-1}$:

$$[x_i(k \mid t), x_{i-1}(k \mid t)]^\top \in \mathbb{X}_{v_{i-1}} := \{[x_i(k \mid t),\, x_{i-1}(k \mid t)]^\top \in \mathbb{X} : v_{i-1}(k \mid t) = v_{i-1}\}. \tag{4.40}$$

### Robust Control Invariant Set for Platoon

We construct a robust control invariant set which avoids both collision and loss of contact with the front vehicle. The definition of the robust control invariant set is given in Definition 4.2.2 but we re-write it here in terms of the new system (4.35)-(4.40).

**Definition 4.5.1** (Robust control invariant set)**.** *A set $\mathbb{C} \subseteq \mathbb{X}$ is said to be a robust control invariant set for the system* (4.35) *subject to constraints* (4.36)-(4.39)*, if*

$$\begin{aligned} &[x_i(0),\, x_{i-1}(0)]^\top \in \mathbb{C} \;\Rightarrow \exists\, u_i(t) \in \mathbb{U} \;\; s.t. \\ &[Ax_i(t) + Bu_i(t),\, Ax_{i-1}(t) + Ba_{i-1}(t)]^\top \in \mathbb{C}, \quad \forall a_{i-1} \in \mathbb{A}(v_{i-1}),\; \forall t \in \mathbb{N}_+ \end{aligned} \tag{4.41}$$

We now extend the work of [61] in order to compute a robust control invariant set for our system (4.35) with constraints(4.36)-(4.39). First, we need the following definition.

**Definition 4.5.2** (Precursor set)**.** *For the system* (4.35)-(4.39)*, we define the precursor set of set $\mathbb{Z}$ as*

$$\begin{aligned} \operatorname{Pre}(\mathbb{Z}, a^f) = \{[x_i,\, x_{i-1}]^\top \in \mathbb{R}^4 : \exists\, u_i \in \mathbb{U} \;\; \textit{such that} \\ [Ax_i + Bu_i,\, Ax_{i-1} + Ba^f]^\top \in \mathbb{Z}\}. \end{aligned} \tag{4.42}$$

The work in [61] only addresses how to find a robust control invariant set when there is no maximum distance constraint. First, they find the control invariant set $\underline{\mathbb{C}}_0$ assuming that the front vehicle is at a full stop. Then, they recursively define the k-steps precursor sets $\underline{\mathbb{C}}_k$ from $\underline{\mathbb{C}}_0$ under the assumption that the front vehicle is fully decelerating at $a_{\min}$:

$$\underline{\mathbb{C}}_{k+1} = \operatorname{Pre}(\underline{\mathbb{C}}_k, a_{\min}) \cap \mathbb{X}, \forall k \in \mathbb{N}_+ \,. \tag{4.43}$$

Finally, the robust control invariant set $\underline{\mathbb{C}}$ is obtained by collecting the sets $\underline{\mathbb{C}}_k$; in fact, $\underline{\mathbb{C}}_k$ is the projection of $\underline{\mathbb{C}}$ onto the distance and velocity space when the front vehicle velocity is $-a_{\min} t_s k \ \mathrm{m\,s^{-1}}$.

Because our linear system with polytopic constraints only has additive disturbance, the robust precursor set can be computed as the intersection of the precursor sets of the systems subject to the disturbances at the vertices of the uncertainty set. Therefore, in our case, at any velocity of the front vehicle, the robust precursor set can be found by computing the intersection of the precursor sets assuming that the front vehicle is accelerating/decelerating at the bounds of set $\mathbb{A}^f(v_{i-1})$.

To obtain the robust control invariant set $\mathbb{C}$ according to Definition 4.5.2, we first compute $\underline{\mathbb{C}}$ using the method described above. Similarly, we can compute the robust control invariant set $\bar{\mathbb{C}}$ which assumes that the front vehicle is only fully accelerating until the velocity reaches $v_{\max}$. Algorithm 1 outlines a method to construct $\underline{\mathbb{C}}$ and $\bar{\mathbb{C}}$.

By taking the intersection of the two sets $\underline{\mathbb{C}}$ and $\bar{\mathbb{C}}$, we obtain a robust control invariant set for our system (4.35)-(4.39). In practice, $\underline{\mathbb{C}}$ and $\bar{\mathbb{C}}$ can easily be computed for some finite number of discrete values of the front vehicle velocity, whereas the front vehicle velocity is a continuous variable in reality. To accommodate this issue, we compute a conservative approximation of the robust control invariant set for the $i$-th vehicle, when the $(i-1)$-th vehicle velocity is $v_{i-1}$:

$$\mathbb{C}_{v_{i-1}} = \underline{\mathbb{C}}_l \cap \bar{\mathbb{C}}_h \tag{4.44}$$

where

$$\begin{aligned} l = & \operatorname*{arg\,max}_{k \in \mathbb{N}_+} k \\ & \text{s.t.} \;\; v_{i-1} \geq v_{\text{cut}}(k),\, v_{\text{cut}}(0) = 0,\, v_{\text{cut}}(k+1) = v_{\text{cut}}(k) - a_{\min} t_s \,, \\ h = & \operatorname*{arg\,max}_{k \in \mathbb{N}_+} k \\ & \text{s.t.} \;\; v_{i-1} \leq v_{\text{cut}}(k),\, v_{\text{cut}}(0) = v_{\max},\, v_{\text{cut}}(k+1) = v_{\text{cut}}(k) - a_{\max} t_s \,. \end{aligned} \tag{4.45}$$

Collecting $\mathbb{C}_{v_{i-1}}$ for all possible $v_{i-1}$, we obtain the robust control invariant set $\mathbb{C}$ for our system (4.35)-(4.39). Finally, Figure 4.14 illustrates the robust control invariant set computed by (4.44) and Algorithm 1 when the front vehicle velocity are discrete values.

**Algorithm 1** Computation of $\underline{\mathbb{C}}$ and $\bar{\mathbb{C}}$

```
Input System (4.35)-(4.39), $\mathbb{X}$, $\mathbb{X}_{v_{\min}}$, $\mathbb{X}_{v_{\max}}$
Output $\underline{\mathbb{C}}$ and $\bar{\mathbb{C}}$
    $\Omega_0 \leftarrow \mathbb{X}_{v_{\min}}$, $k \leftarrow -1$
    Repeat
        $k \leftarrow k + 1$,
        $\Omega_{k+1} \leftarrow \mathrm{Pre}(\Omega_k,\, 0) \cap \Omega_k$
    Until $\Omega_{k+1} = \Omega_k$
    $\underline{\mathbb{C}}_0 \leftarrow \Omega_k$, $v_f \leftarrow v_{\min}$, $i \leftarrow -1$
    Repeat
        $i \leftarrow i + 1$,
        $v_f \leftarrow v_f - t_s a_{\min} - \frac{1}{2} a_{\min} t_s^2$
        $\underline{\mathbb{C}}_{i+1} \leftarrow \mathrm{Pre}(\underline{\mathbb{C}}_i,\, a_{\min}) \cap \mathbb{X}$
    Until $v_f \leq v_{\max}$
    $\Omega_0 \leftarrow \mathbb{X}_{v_{\max}}$, $k \leftarrow -1$
    Repeat
        $k \leftarrow k + 1$,
        $\Omega_{k+1} \leftarrow \mathrm{Pre}(\Omega_k,\, 0) \cap \Omega_k$
    Until $\Omega_{k+1} = \Omega_k$
    $\bar{\mathbb{C}}_0 \leftarrow \Omega_k$, $v_f \leftarrow v_{\max}$, $j \leftarrow -1$
    Repeat
        $j \leftarrow j + 1$,
        $v_f \leftarrow v_f - t_s a_{\max} - \frac{1}{2} a_{\max} t_s^2$
        $\bar{\mathbb{C}}_{j+1} \leftarrow \mathrm{Pre}(\bar{\mathbb{C}}_j,\, a_{\max}) \cap \mathbb{X}$
    Until $v_f \geq v_{\min}$
    $\underline{\mathbb{C}} \leftarrow [\underline{\mathbb{C}}_0, \underline{\mathbb{C}}_1, \ldots, \underline{\mathbb{C}}_{i+1}]$
    $\bar{\mathbb{C}} \leftarrow [\bar{\mathbb{C}}_0, \bar{\mathbb{C}}_1, \ldots, \bar{\mathbb{C}}_{j+1}]$
```

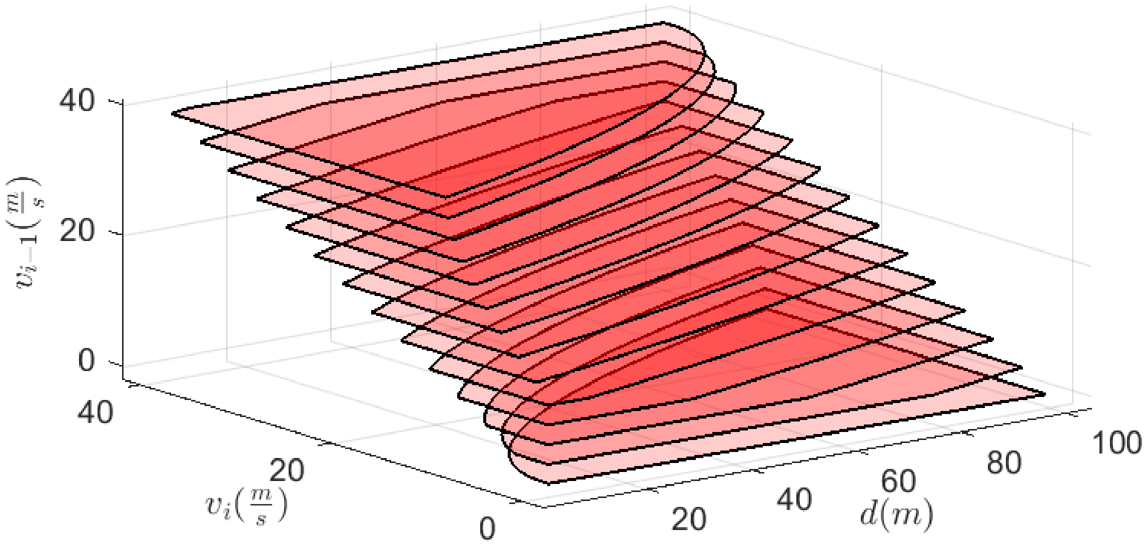


Figure 4.14: Robust control invariant set computed by (4.44) and Algorithm. 1. The front vehicle velocity takes discrete values between 0 and 40 $\frac{m}{s}$

**Cost Function**

The objectives of our MPC include minimizing input efforts and jerk. Therefore, the cost function for the $i$-th vehicle is expressed as:

$$J_i = \sum_{k=t}^{t+N_p-1} \Big[ Q_u(u_i(k \mid t))^2 \tag{4.46a}$$

$$+ Q_{\Delta u}(u_i(t-1) - u_i(k \mid t))^2 \Big] \tag{4.46b}$$

$$+ \sum_{k=t}^{t+N_p-2} \Big[ Q_{\Delta u}(u_i(k \mid t) - u_i(k+1 \mid t))^2 \Big] \tag{4.46c}$$

where $u_i(t-1)$ is the input from the last time step $t-1$. Term (4.46a) penalizes input efforts, terms (4.46b) and (4.46c) penalize the input deviation from the current acceleration and the respective previous predicted input, respectively. Because the distance to the front vehicle is upper- and lower-bounded, our cost function does not necessarily comprise the cost to track a desired velocity or a front vehicle distance. The constraints (4.39) and (4.10) will enforce the vehicles to stay within a given distance range.

**Model predictive control design**

The longitudinal movement of the $i$-th vehicle in the platoon ($i \geq 1$) is governed by the following MPC which, at time step $t$, solves the following optimization problem:

$$\begin{aligned}
\underset{u_i(\cdot|t)}{\text{minimize}} \quad & J_i && \text{(4.47a)}\\
\text{subject to} \quad & x_i(k+1|t) = Ax_i(k|t) + Bu_i(k|t)\,, && \text{(4.47b)}\\
& x_{i-1}(k+1|t) = Ax_{i-1}(k|t) + B\tilde{a}_{i-1}(k|t)\,, && \text{(4.47c)}\\
& [x_i(k|t),\, x_{i-1}(k|t)]^\top \in \mathbb{X}\,, && \text{(4.47d)}\\
& u_i(k|t) \in \mathbb{U}\,, && \text{(4.47e)}\\
& \forall k = t, ..., t+H-1, && \\
& x_i(t|t) = \bar{x}_i(t),\, x_{i-1}(t|t) = \bar{x}_{i-1}(t)\,, && \text{(4.47f)}\\
& [x_i(t+1|t),\, x_{i-1}(t+1|t)]^\top \in \mathbb{C}\,, && \text{(4.47g)}
\end{aligned}$$

where $\bar{x}_i(t)$ and $\bar{x}_i(t)$ are the state measurements at time $t$ and are assumed to be accurate; $\tilde{a}_{i-1}(k|t)$ is provided by the V2V message (4.34). Recall that only the element $\tilde{a}_{i-1}(t|t)$ can be fully trusted as the actual control input of the front vehicle, while the following elements $\tilde{a}_{i-1}(k|t),\ k > t$ are only predicted and, therefore, may be incorrect.

Solving (4.47), we have the following optimal states and inputs:

$$\begin{aligned}
x_i^*(t) &= [x_i^*(t|t), x_i^*(t+1|t), ..., x_i^*(t+H|t)], && \text{(4.48a)}\\
u_i^*(t) &= [u_i^*(t|t), u_i^*(t+1|t), ..., u_i^*(t+H-1|t)]. && \text{(4.48b)}
\end{aligned}$$

Then, the first input $u_i^*(t|t)$ is applied to the system during the time interval $[t, t+1)$;

$$u_i(t) = u_i^*(t|t). \tag{4.49}$$

### 4.5.4 Persistent Feasibility

With the invariant set stated in Definition 4.5.1, we are in place to state the following theorem.

**Theorem 3.** *Consider the eco-CACC* (4.47)-(4.49) *for the i-th vehicle with* $N_p \geq 1$. *Assume that (i) the* $(i-1)$*-th vehicle sends to the i-th vehicle its acceleration forecast* (4.34) *necessary to solve the optimization problem* (4.47) *for all* $t \geq 0$*, and (ii) the Eco-CACC for the i-th vehicle is feasible at* $t = 0$*. If the set* $\mathbb{C}$ *in* (4.47g) *is computed by Algorithm 1 and, therefore, a robust control invariant set according to Definition 4.2.2 for the system* (4.35)-(4.39)*, then the Eco CACC* (4.47)-(4.49) *for the i-th vehicle in the platoon is persistently feasible.*

*Proof.* We prove Theorem 3 by induction. Assume at time step $t$, the optimization problem (4.47) is feasible for the $i$-th vehicle, resulting in the optimal state and input trajectories (4.49). As the first optimal input (4.49) and the first element in V2V message from the

Table 4.6: Controller and model parameters

| | | | |
|---|---|---|---|
| $H$ | MPC Horizon | - | 20 |
| $t_s$ | sampling time | s | 0.2 |
| $Q_u$, $Q_{\Delta u}$ | weights | - | 5, 10 |
| $a_{\min}$ | minimum acceleration | m/s$^2$ | -4 |
| $a_{\max}$ | maximum acceleration | m/s$^2$ | 4 |
| $d_{\min}$ | minimum distance | m | 5 |
| $d_{\max}$ | maximum distance | m | 100 |
| $v_{\min}$ | minimum velocity | m s$^{-1}$ | 0 |
| $v_{\max}$ | maximum velocity | m s$^{-1}$ | 40 |

$(i-1)$-th vehicle, $\bar{a}_{i-1}(t)$, are applied to the $i$-th and the $(i-1)$-th vehicles during the time interval $[t, t+1)$, respectively, we have a new state $[x_i^*(t+1|t), x_{i-1}(t+1|t)]^\top \in \mathbb{C}$ at $t+1$. Due to the constraint (4.47g) and the properties of the robust control invariant set, there exists at least one input trajectory which can keep the states of the $i$-th vehicle inside the same set for any admissible acceleration (4.5) of the $(i-1)$-th vehicle for infinite time. Because $\mathcal{C} \subseteq \mathbb{X}$, the optimization problem (4.47) at $t+1$ is feasible for the $i$-th vehicle. Therefore, we conclude by induction that the eco-CACC (4.47)-(4.49) for the $i$-th vehicle is persistently feasible. □

**Remark 3.** *The eco-CACC* (4.47)-(4.49) *is still recursively feasible regardless of the precision of the front vehicle acceleration forecast* $\tilde{a}_{i-1}(\cdot|t)$ *as long as the first acceleration* $\tilde{a}_{i-1}(t|t)$ *is correct and* $\tilde{a}_{i-1}(k|t) \in \mathbb{A}(v_{i-1}(k|t))$.

**Remark 4.** *The motion of the i-th vehicle controlled by eco-CACC* (4.47)-(4.49) *is only dependent on admissible movements* (4.5) *of the* $i-1$ *vehicle; therefore, if (i) the motion of every vehicle in the platoon is controlled by our eco-CACC, (ii) problem* (4.47) *is feasible at* $t=0$ *for every vehicle in the platoon, (iii) every vehicle in the platoon sends to its following vehicle the solution of its optimization problem* (4.6) *as the acceleration forecast* (4.13)*, and (iv) there is no delay in computation and communication, then the whole platoon will be feasible for all* $t \geq 0$.

### 4.5.5 Simulation Results

In this section we present simulation results of our proposed eco-CACC (4.47)-(4.49). The controller and model parameters used in our simulations are listed in Table 4.6.

#### Two-vehicle platoon

In this section, a two-vehicle platoon with V2V communication is simulated. The leader follows velocity trajectory recorded in highway driving with time-varying traffic density, whereas the follower's motion is dictated by the proposed controller (4.47)-(4.49).

Table 4.7: Comparison of Average Energy and Jerk for Two-vehicle Platoon

| Vehicle | Total energy (%) | Jerk average (%) |
|---|---:|---:|
| Leader | 100.0 | 100.0 |
| Follower with BSM | 91.4 | 59.5 |
| Follower with perfect info | 91.0 | 53.2 |

In order to see the effects of sharing extended forecasts, we compare two cases of V2V messages. In the first case, the V2V message is the standard Basic Safety Message (BSM) defined by SAE J2735 [37], which only includes the current vehicle state and acceleration. In this case, our controller (4.47)-(4.49) assumes that the leader is at constant velocity after the current acceleration, i.e., at time step $t$, V2V message (4.13) from the leader is $\tilde{\mathbf{a}}_{\text{leader}}(t) = [\bar{a}_{\text{leader}}(t), 0, \ldots, 0]$. The second case is when the V2V message contains a perfect $N_p$-step ahead forecast; $\tilde{a}_{\text{leader}}(k \mid t) = \bar{a}_{\text{leader}}(k)$ for all $k = t, t+1, \ldots, t+N_p-1$. Note that both cases preserve the recursive feasibility as stated in Remark 3.

Figure 4.15 shows the distance and velocity trajectories of the leader and follower vehicles when the V2V message only has BSM, therefore, the vehicle assumes a constant velocity, and the V2V message contains a perfect acceleration forecast. As seen, in both cases, the distance and velocity satisfy their constraints at all times. Notice that, in order to optimize performance, the relative distance varies significantly, but always within the specified bounds. On the other side, the follower's velocity is more steady compared to the leader's velocity except for occasional hard acceleration/deceleration spikes to satisfy the upper- and lower-distance constraints. To avoid these spikes, we can construct a new cost function or utilize the terminal cost in our MPC optimization problem (4.47).

Table 4.7 summarizes the performance of our controller in this scenario. The metric for energy consumption is the integral of the positive traction power required to achieve the closed loop trajectories according to the longitudinal vehicle dynamics model (3.1) using the identified parameters listed in Table 3.2. The average jerk is obtained by the average of the absolute value of the change in acceleration at every time step. Compared to the leader, both followers with BSM or with perfect information have comparably large savings in total energy consumption. Moreover, both followers have large average jerk reduction. It is also noted that the follower with perfect front vehicle acceleration forecast has more jerk reduction and more energy saving than the follower with BSM and a constant front vehicle velocity forecast.

### Multi-vehicle platoon

A four-vehicle platoon is simulated in this section. The 0-th vehicle, or the leader, follows a sinusoidal velocity trajectory with an average of $14\text{m}\,\text{s}^{-1}$, an amplitude of $20\text{m}\,\text{s}^{-1}$, and a period of 20s. The 1-st vehicle receives from the 0-th vehicle perfect information about its future acceleration. Other followers, or the 2-nd and 3-rd vehicles, receive from their

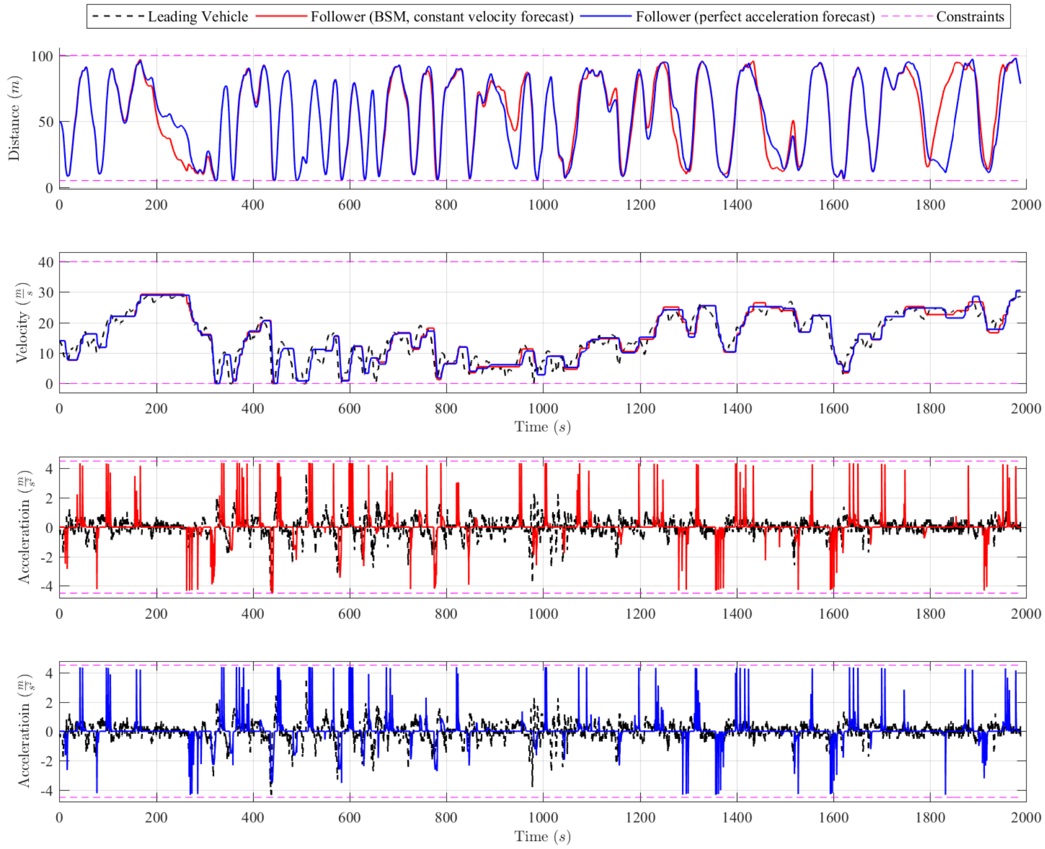


Figure 4.15: Plots of the distance between the leader and the follower and their velocities for the two-vehicle platoon. The leader's trajectory is recorded from the actual highway and the leader sends to the follower either only a standard BSM message or a perfect acceleration forecast.

respective preceding vehicles their predicted future accelerations (4.13) as the solution (4.48) of the optimization problem (4.6). It is noted that these predicted accelerations are not necessarily the same as their actual future accelerations.

Figure 4.16 shows plots of the distance, velocity, and acceleration for every vehicle in the platoon. As seen, each vehicle obeys its constraints on the distance, velocity, and acceleration at all times. Also, the velocity and acceleration trajectories of the following vehicles are more steady than those of their respective preceding vehicles. As a result, the 3-rd vehicle can safely track the front 2-nd vehicle without any effort to accelerate nor to decelerate after the velocity catch-up phase. Table. 4.8 summarizes the performance of our controller for each vehicle in the platoon, in terms of the approximated total energy consumption and the average jerk. Overall, the vehicles consume less energy and undergo less jerk as their

Table 4.8: Comparison of Average Acceleration and Jerk for Multi-vehicle Platoon

| Vehicle | Total energy (%) | Jerk average (%) |
|---|---:|---:|
| 0-th vehicle | 100.0 | 100.0 |
| 1-st vehicle | 79.9 | 98.6 |
| 2-nd vehicle | 54.7 | 73.4 |
| 3-rd vehicle | 39.8 | 45.8 |

position is more downstream in the platoon.

### Robustness to incorrect prediction

We now briefly investigate the robustness of our controller against incorrect prediction. This means that the predicted acceleration in the message transmitted to the following vehicle is different from the actual future acceleration.

In this simulation, there are two vehicles; the leader and the follower. The follower receives an incorrect acceleration forecast of the leader, i.e., constant velocity after the first step acceleration although it actually performs the maximum acceleration/braking, $a_{\text{max}}$ and $a_{\text{min}}$, respectively. Fig. 4.17 shows plots of the distance between the two vehicles and their velocities. In both cases, both distance and velocity constraints are satisfied at all time. Due to the robust control invariant set constraint (4.47g), the follower is able to brake or accelerate in time, preventing violations of state constraints.

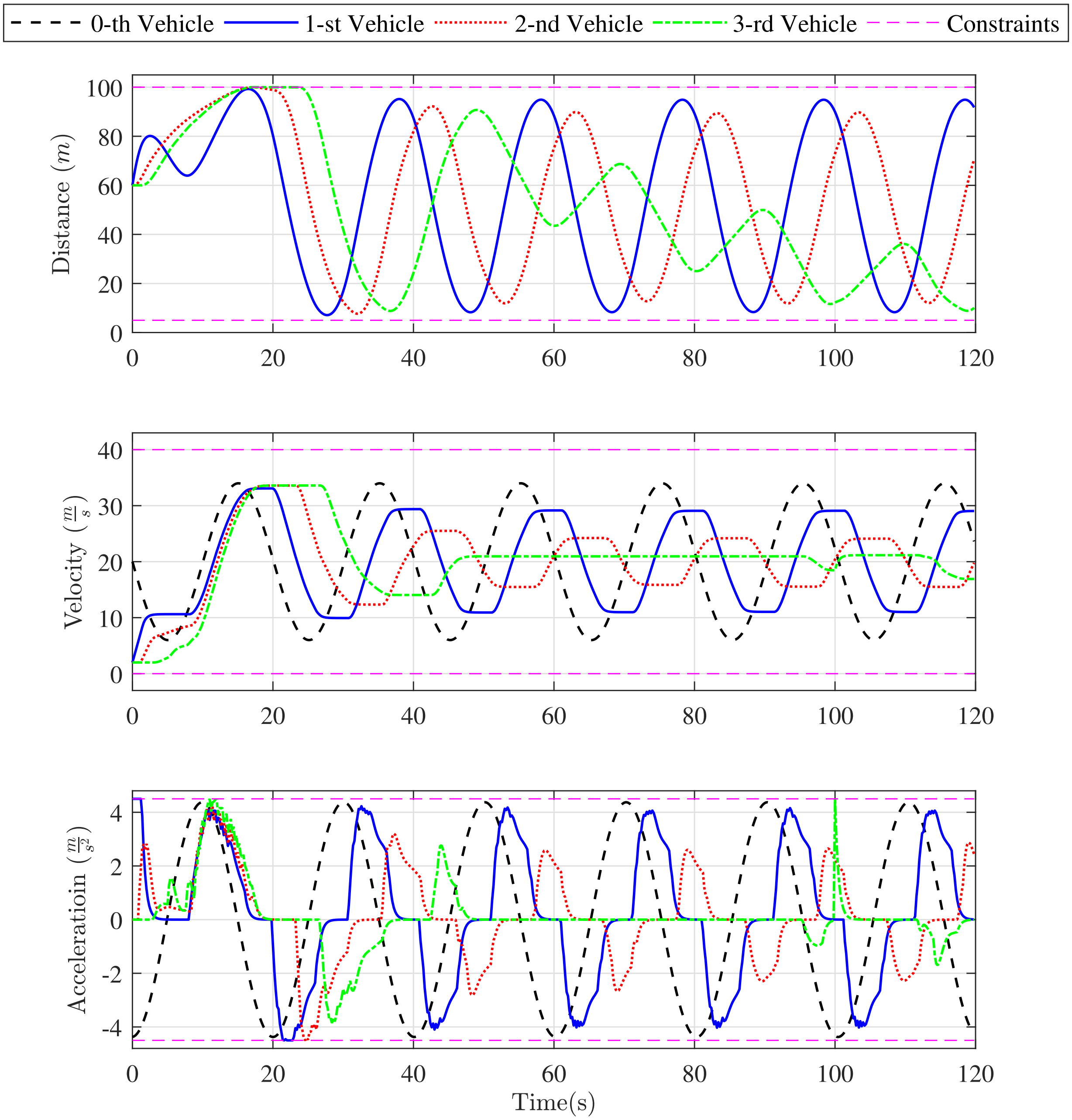


Figure 4.16: Plots of the distance, velocity, and acceleration for the platoon with 4 vehicles. The platoon is coordinated as the Figure 4.2.

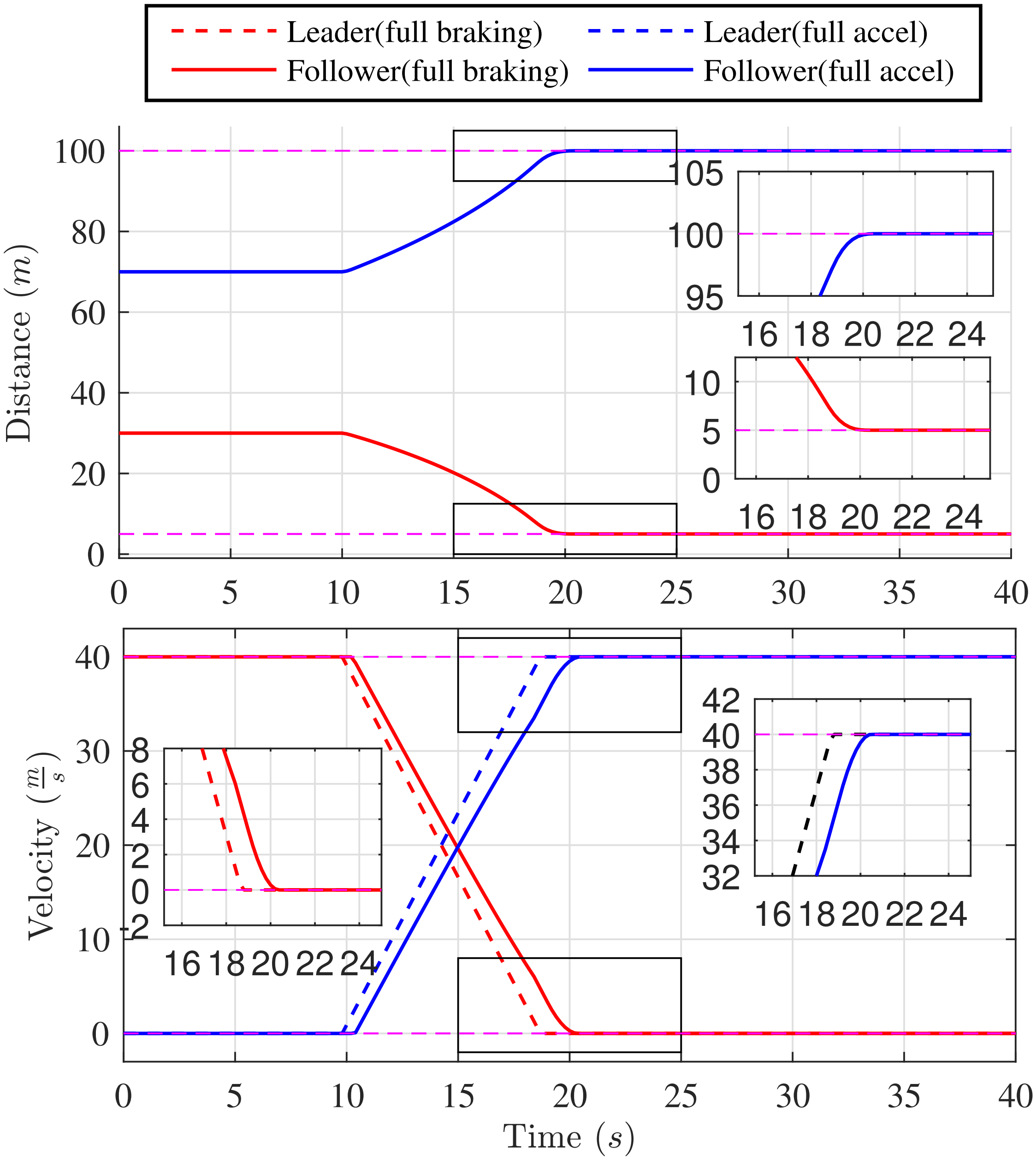


Figure 4.17: Plots of the distance between the two vehicles (up) and their velocities (down). The leading vehicle performs the maximum braking/accelerating while the follower receives an incorrect V2V message about the future acceleration of the leader.

# Chapter 5

# Connected and Automated Vehicle Motion and Powertrain Control using Vehicle-to-Cloud/-Infrastructure Communication

## 5.1 Introduction

This chapter discusses a Connected and Automated Vehicle (CAV) motion and powertrain control which exploits benefits from communication between the vehicle and the cloud (V2C) and between the vehicle and the infrastructure (V2I), in particular, traffic lights. The efficacy of the proposed control designs is demonstrated through simulation, hardware-in-the-loop (HIL) simulation, and/or real world experiments.

As already explained in Chapter 2 through our Connected and Automated Vehicle (CAV) system architecture in Figure 2.2, the powertrain controller distributes the torque demand from the vehicle dynamics controller accordingly between the engine and motor based on an energy management system (EMS). Most commercial PHEVs implement the charge depleting-charge sustaining (CD-CS) EMS; i.e., it distributes the torque demand mostly on the motor until the battery state of charge (SOC) gets close to the minimum limit, and afterwards the torque demand is distributed based on the CS strategy between the engine and motor to sustain the SOC level above the minimum limit. However, this EMS can result in a myopic behavior in energy saving aspects as this strategy can inevitably require the engine to operate even in inefficient torque/speed demands especially for a long trip which has a long CS region (low SOC); due to the low SOC, the vehicle must turn on the engine and use it to meet the torque demands to move the vehicle. Literature has extensively studied powertrain energy management strategies, especially for HEVs [74, 75]. In [76], the authors first introduced the equivalent consumption minimization strategy (ECMS) which adjusts the power split between the internal combustion engine (ICE) and the electric motor

(EV) based on the instantaneous optimization of an equivalent fuel consumption. However, the large computational burden to solve the optimization in real time is a limitation of this approach.

To mitigate such large computational effort, [77] proposes an adaptive algorithm for parallel HEV ECMS which further minimizes fuel consumption by periodically refreshes the control parameter in the ECMS optimization problem according to the current road load and the predicted information about the remaining trip. Similarly, [78] uses approximate dynamic programming (ADP) to learn the optimal EMS policy using data of vehicle speed, road grade, battery charge, power demand, and availability of charging stations. However, these previous works used synthesized trip data (not actual trip data fully) to train the optimization problem parameters and they do not present the energy saving performance paired with the longitudinal motion control of CAVs.

The eco-Charge controller tackles these issues and seeks to achieve a higher overall trip energy efficiency by exploiting the historical trip data of the same route/time. The eco-Charge controller is a route-based data driven EMS that utilizes the prediction of future usage of the ego vehicles battery based on the current and historical trip data [79]. It consists of two layers, a cloud layer (Powertrain Eco-Driving Planning) and an on-board layer (on-board powertrain controller). In the cloud layer, a global EMS optimization problem is solved to train EMS policy parameters (approximating value function), and in the on-board layer, a real-time powertrain controller distributes the torque demand by solving the optimization problem using trained EMS policy parameters offered in real time by the cloud layer through V2C communication. Section 5.2 presents the summary of the eco-Charge control design while the details of the control design can be found in [79]. Rather, this section focuses on presenting our public road experiments using the combined eco-LMC and eco-Charger on our test vehicle and demonstrate/quantify their energy savings by comparing it with that of the human driven vehicle.

On the other hand, Vehicle-to-infrastructure (V2I) communication can provide our CAV with look ahead information about traffic and signalized intersections in the downstream road. Examples of this V2I information are SPaT which includes signal phase and timing information and MAP which includes lane level geography of the intersection. This information can be very useful for a reference velocity generation from the Motion Eco-Driving Planning module and a safe longitudinal motion controller (LMC) from the Motion Control module as seen in our CAV system architecture in Figure 2.2.

The strategies to explore the V2I information for a reference velocity generation from the Motion Eco-Driving Planning module are well addressed in the literature, especially in arterial scenarios where the vehicle is driving in traffic through a series of traffic lights with fixed schedules. Finding an optimal velocity trajectory utilizing SPaT and MAP is often regarded as an approach to achieve eco-driving with traffic lights in the literature and has shown substantial energy saving. Some works only focus on improving energy efficiency of adaptive cruise control in a short time horizon (about next 3 seconds) by only accounting for one upcoming intersection [80]. This can be handy especially when the vehicle has limited computation power. However, the limited horizon comes with limited energy savings. There are other works which present a long horizon optimal velocity trajectory with multiple traffic lights in planning horizon [81, 82, 83, 44]. However, these assume that the traffic lights are in fixed schedules whereas a lot of traffic lights in real world have time-varying traffic cycle schedules to allow more vehicle throughput.

The work in [84] solves the eco-driving problem through signalized intersections with *uncertain* effective red light duration; they are uncertain due to the time-varying traffic cycle schedules. The uncertainty is addressed by formulating chance constraints on passing through the intersections during green lights based on statistical and historical signal data. However, this approach is only applicable for a vehicle travelling on a free flow traffic condition; i.e., there is no surrounding vehicles which can affect the motion of the ego vehicle.

Our control design in the CAV control architecture allows us to systemically handle this problem of both uncertain traffic lights and the existence of other vehicles. In particular, Section 5.3 presents the Eco-driving paired with a safe urban-LMC. The Eco-driving uses statistical and historical signal data as well as real time V2I data to compute a reference velocity with a long horizon; this reference velocity is an optimal velocity reference to achieve energy saving. Then, this reference velocity is used by our on-board urban-LMC, which ensures safety using the real-time perception and V2I data Finally, we present the HIL and real-world experimental results to show the efficacy of our approach.

Optimal velocity reference can be a useful information even for vehicles travelling in a long route not necessarily including traffic lights. Predictive cruise control (PCC) is a cruise control tracking a reference velocity that is generated using preview information about about the downstream road [81, 85]. This information, which includes static information such as speed limits or road grade, and dynamic information such as traffic speed or intersection delays, can be instantly accessible to CAVs via Vehicle-to-Cloud (V2C) communication.

PCC fits into our CAV system structure in Figure 2.2. The reference velocity trajectory generation can cloud-aided in the Remote Planning and Routing module (i.e. it exploits information that is generally retrieved from the cloud). This reference trajectory is based on long-term forecasts and cannot be implemented in open loop. The real-time control simply tracks the reference signal, does not exploit perception sensors or cooperation, and requires driver intervention to ensure basic safety. Nonetheless, reference generation for PCC can also be integrated with the Motion Control with our LMC controllers proposed in this thesis.

Often, the reference velocity trajectory generation in PCC is cast as a velocity trajectory optimization problem for minimum energy consumption; [85, 86] present optimization methods to calculate the optimal velocity trajectory for maximizing energy efficiency. In [81, 84] or in Section 5.3, an optimal control problem is formulated to minimize energy consumption when the vehicle is going through a series of traffic lights. However, their approaches to obtain velocity reference generation are limiting in the following senses. First, they require a priori knowledge about the environment. Second, the complexity of their approaches increases with the length of trip.

To overcome these issues, the problem of PCC can be cast using the Learning Model Predictive Control (LMPC) framework presented in [87]. This reference-free controller is an attractive approach because it is able to solve long horizon optimal control problems while ensuring global safety constraint guarantees. It has been successfully implemented in autonomous driving applications [88, 89]. However, the original work of LMPC does not address the problem of completing the task within a given time limit as it is usually formulated for minimum time problems, where this issue does not arise. As a result, without proper design of the cost of the optimal control problem, the task may result in taking more time to actually finish a task.

Section 5.4 presents the LMPC framework to PCC of a vehicle which repeatedly drives along the same route. We modify the original LMPC so that we can enforce the task (trip) to be completed within a time limit. Moreover, in every trip, our controller can learn the static environment features such as road grade and attempts to improve its performance, which includes total energy consumption and comfort. We present the simulation results to show efficacy for the proposed PCC.

## 5.2 Powertrain Eco-Driving Planning and On-board Powertrain Control using V2C

As seen in our Connected and Automated Vehicle (CAV) system architecture in Figure 2.2, Vehicle-to-cloud (V2C) communication and remote computation enables Powertrain Eco-Driving Planning, which then allows the on-board powertrain controller to operate more energy efficiently. This section first discusses the design of our Powertrain Eco-Driving Planning and the related powertrain controller; the paired Powertrain Eco-Driving Planning and on-board powertrain control are hereafter named as the *eco-Charge controller*. Finally, we present the experimental results on public highway and show the energy saving efficacy of our CAV with the eco-LMC (4.6)-(4.8) and the eco-Charge controller.

### 5.2.1 Eco-Charge Control Design

We formulate the eco-Charge problem as a finite-horizon optimal control problem of the following form in the time domain, with sampling time $t_s$ of 1 second at time t.

$$\min_{u(\cdot|t)} \sum_{k=t}^{t+N_p-1} g(x(k|t), u(k|t), w(k|t)) + l(x(t+N_p|t), t) \tag{5.1a}$$

subject to

$$x(k+1|t) = f(x(k|t), u(k|t), w(k|t)), \tag{5.1b}$$

$$0 = h(x(k|t), u(k|t), w(k|t)), \tag{5.1c}$$

$$u(k|t) \in \mathcal{U}(t),\ x(k|t) \in \mathcal{X}, \tag{5.1d}$$

$$\forall k \in [t, t+1, ..., t+N_p-1],$$

$$x(t|t) = \bar{x}(t),\ x(t+N_p|t) \in \mathcal{X}., \tag{5.1e}$$

where the state vector $x = [SOC, e_{\text{state}}]^\top$, the input vector $u = [T_e, T_e, e_{\text{switch}}]^\top$, and the forecast vector $w = [P_{\text{aux}}, T_d]^\top$ where $SOC$ is the battery internal energy state-of-charge, $e_{\text{state}}$ is the engine on/off status, $T_e$ is the engine torque, $T_m$ is the motor torque, $e_{\text{switch}}$ is the engine on/off trigger, $P_{\text{aux}}$ is the power consumption of electric auxiliaries, and $T_d$ is the demanded torque at the wheel. The forecast of $T_d$ may be offered by the vehicle dynamics controller (such as eco-LMC (4.6)-(4.8)), and the forecast of $P_a$ may be produced using weather forecasts and a model of the on-board air conditioning, assuming the latter is the main cause of power consumption.

The system update dynamics in (5.1b) is the powertrain dynamics expressed as the following.

$$x(k+t|t) = \begin{bmatrix} SOC(k+1|t) \\ e_{\text{state}}(k+1|t) \end{bmatrix} = \begin{bmatrix} SOC(k|t) - t_s \frac{V_{\text{oc}}(k|t) - \sqrt{V_{\text{oc}}^2(k|t) - 4R_b(k|t)P_b(k|t)}}{2R_b(k|t)Q_b} \\ e_{\text{switch}}(k|t) \end{bmatrix} \tag{5.2}$$

where $V_{\text{oc}}$ and $R_b$ are the battery open-circuit voltage and internal resistance, respectively; both are nonlinear functions of the SOC; $P_b$ is the battery power which is a function of the input $T_m$ [90]; $Q_b$ is the battery capacity. More details on the powertrain model can be found in [79, 91].

The algebraic constraint $h$ in (5.1c) enforces the summation of $T_m$ and $T_e$ at the transmission input shaft; i.e., The traction torque is modeled as

$$h(x(k|t), u(k|t), w(k|t)) = T_d(k|t) - g_i(k|t)\eta_t(T_m(k|t) + c_{\text{on}}\eta_c T_e(k|t)), \tag{5.3}$$

where $\eta_t$ and $\eta_c$ denote the transmission and clutch efficiency, respectively; $c_{\text{on}}$ is a boolean variable, true when the clutch is closed; $g_i$, $i = 1, ..., 6$ is the gear ratio corresponding to the $i^{th}$ gear number. Note that we assume that $\eta_t$ and $\eta_c$ are assumed to be constant, and $c_{\text{on}}$ is the same variable as $e_{\text{on}}$ (clutch is engaged if and only if the engine is turned on).

The state constraint, $\mathbf{X}$, and the input constraint, $\mathbf{U}(t)$, in (5.1d) denote the constraints for SOC limits and the actuator limits:

$$\text{SOC}^{\text{min}} \leq \text{SOC}(k|0) \leq \text{SOC}^{\text{max}}, \tag{5.4}$$

where $\text{SOC}^{\text{min}}$ and $\text{SOC}^{\text{max}}$ are the vehicle parameters.

$$T_e^{\text{min}}(t) \leq T_e(k|0) \leq T_e^{\text{max}}(t), \tag{5.5}$$

where $T_e^{\text{min}}$ and $T_e^{\text{max}}$ are the maximum and minimum engine torques which are decided by the vehicle speed and gear ratio, which are assumed to be constants in the prediction horizon.

The stage cost function $g$ in (5.1a) represents the powertrain total power consumption which consists of the fuel power $P_f$ and the battery power $P_b$,

$$g(x, u, w) = P_f(x, u, w) + P_b(x, u, w).$$

Finally, the terminal cost $l$ in (5.1a) is the approximated value function offered by the Powertrain Eco-Driving Planning in the cloud layer. The reason to include this is to avoid myopic behavior of our eco-Charge controller without having to increase the prediction horizon and, therefore, preserving computational tractability. The Powertrain Eco-Driving Planning in the cloud layer uses stored historical trip data to obtain optimal control value functions for each historical trip cycle by solving the optimization problem which is similar to (5.1) except that the horizon is until the end of the trip and that it does not have any terminal cost. Then, using the computed value functions, the function $l$, which approximate the optimal cost-to-go as a function of the feature states (such as SOC, engine on/off, and time-dependent variables such as average speed, average acceleration, estimated time left of the trip, and etc), is computed with a supervised learning algorithm. And finally, the obtained function $l$ is sent to the on-board layer via V2C connectivity, and, in the on-board layer, the real-time powertrain control problem (5.1) is solved.

Solving the problem (5.1) at time $t$, we obtain the optimal input trajectory $u^*(\cdot|t)$. The first input $u^*(t|t)$ is applied to the powertrain system, and at the next time step $t+1$ a new optimal control problem in the form of is solved using the new measurements $\bar{x}(t+1)$ and the first input is again applied to the system, yielding a *moving* or *receding* horizon control strategy.

The work in [79] presents the energy saving performance of the eco-Charge controller. In the next subsection, we present the energy saving performance of the eco-Charge controller combined with the eco-LMC and compare it with the human driven vehicle with the production powertrain controller.

### 5.2.2 Experimental Results

Experiments are conducted on a public highway in the same setup as described in Section 4.4.5. The only difference is that the two experimental vehicles now have the same initial conditions such as initial SOC, weights, tire pressure, and auxiliary load consumption throughout the experiments. This allows us to compare energy consumption more fairly.

The two vehicles, the *front vehicle* and the *eco vehicle*, travelled at the same time the round-trip route (75 miles total) in Fremont, CA, as depicted in Figure. 5.1. The two vehicles were driven on the same lane at all time while the front vehicle was always driven ahead of the eco vehicle in lane-wise longitudinal direction; the front vehicle was not always directly in front of the eco vehicle due to the cut-in vehicles. The longitudinal motion of the front vehicle was controlled by the normal human driver whereas that of the eco vehicle is controlled by the eco-LMC (4.6)-(4.8) with the constant velocity forecast about its front vehicle. Also, the powertrain sytem of the front vehicle is governed by the vehicle's own production controller with the CD-CS strategy while the eco-vehicle powertrain system is governed by our eco-Charge controller.

Prior to the experiments, we collected 16 datasets for the human-driven vehicle traveling the round-trip route (75 miles total) in Fremont, CA, as depicted in Figure. 5.1. Each dataset was taken during the rush hour (heavy traffic time) and includes GPS traces and measurements from vehicle sensors and estimators (see Chapter 2 for details about the vehicle hardware setup). Using this data, we constructed the Powertrain Eco-Driving Planning and designed the terminal cost $l$ in (5.1a). Note that in real world, more datasets in different routes can be achieved through the V2C communication.

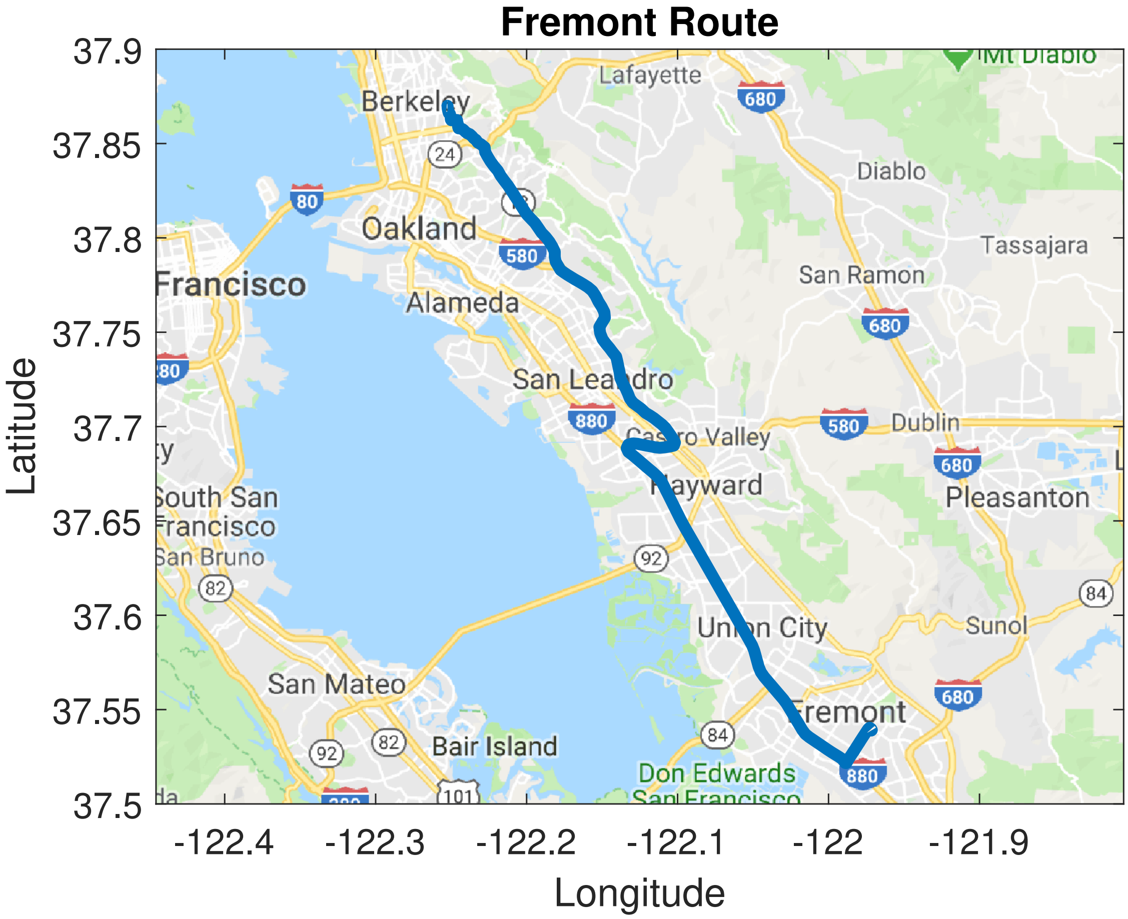


Figure 5.1: Berkeley to Fremont round trip (75 miles total)

We conducted our experiments in two different traffic conditions; med-heavy traffic and free flow traffic. Note that we swapped the test vehicles in each experiment to ensure the fair comparisons and validations of our CAV controller performance. First, Figure 5.2 and Table 5.1 show the experimental result in med-heavy traffic. Figure 5.2 depicts the closed loop trajectories of the eco vehicle and the front vehicle in terms of the velocities and SOCs and the front target distance detected by the eco vehicle. A few remarks from the plot follow.

- Velocity of the eco vehicle is smoother than that of the front vehicle.
- The eco vehicle uses SOC in a blended manner between battery and fuel, gradually exhausting SOC toward the end of the trip. The front vehicle, however, quickly exhausts its SOC with the CD-CS powertarin control strategy.

- The safety distance constraints are satisfied at all times.
- The distance plot shows the the eco vehicle sometimes has a large distance gap with the front target.

Table 5.1 lists the relative energy consumption saving in MPGe and positive energy. The eco vehicle saves about 15 percent positive work and improves about 20 percent in MPGe. Because the eco vehicle, controlled by the eco-LMC, drives at a more even pace than the front vehicle, we were able to achieve the saving in positive work; however, this often results in a large distance gap sacrificing distance tracking performance. Moreover, our eco-Charge uses the engine as well as the motor even during the high SOC regions, seeking to operate the two actuators at their optimal efficiency points. This allows the vehicle to avoid depleting the SOC in the middle of the trip and having to operate for the rest of the trip the engine actuator even when the desired torque and speed is at the in-efficient operating points. Eventually this leads to our MPGe saving.

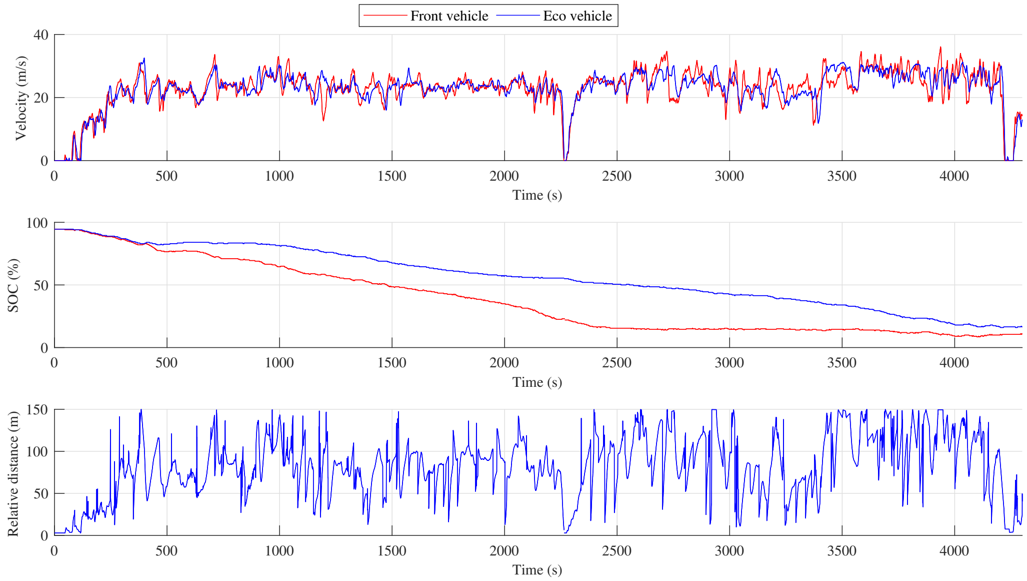


Figure 5.2: Plots of the velocities (top) and SOCs (middle) of the front vehicle and the eco vehicle, and the front target distance (down) detected by the eco vehicle. The experiment was taken during med-heavy traffic.

Next, we present our experimental result taken during low traffic. Figure 5.3 and Table 5.2 show the respective experimental result. Figure 5.3 depicts the closed loop trajectories of the eco vehicle and the front vehicle in terms of the velocities and SOCs and the front target distance detected by the eco vehicle. Compared to the med-heavy experimental results in Figure 5.2, it is seen that both front and eco vehicles drive at an even pace although the eco vehicle velocity trajectory is still less jerky than the front vehicle trajectory. Therefore,

Table 5.1: Comparison of MPGe and positive work in med-heavy traffic

| Vehicle | MPGe (%) | Positive energy (J) |
|---|---|---|
| Front vehicle | 100.0 | 100.0 |
| Eco vehicle | 120.2 | 84.6 |

Table 5.2: Comparison of MPGe and positive work in free flow traffic

| Vehicle | MPGe(%) | Positive energy (%) |
|---|---|---|
| Front vehicle | 100.0 | 100.0 |
| Eco vehicle | 101.3 | 87.0 |

as seen in Table 5.2, the eco vehicle only saves about 13 percent in positive work saving. However, we only obtain 1.3 percent saving in MPGe. The primary reason for this less saving is that our eco-Charge controller is tuned only for the trip with heavy traffic. Therefore, we lose optimality in our powertrain controller as we have a misleading terminal cost $l$ in (5.1a). To further reason this less saving, we next look at the velocity-acceleration operating point plot and the engine torque-speed operating point plot.

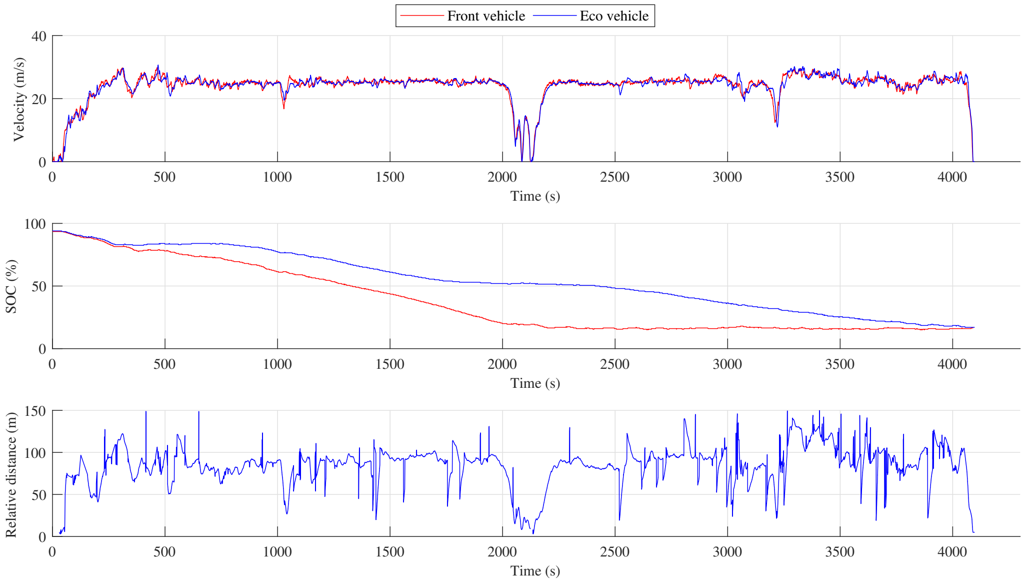


Figure 5.3: Plots of the velocities (top) and SOCs (middle) of the front vehicle and the eco vehicle, and the front vehicle distance (down) detected by the eco vehicle. The experiment was taken during low traffic (free flow).

The traffic dependent energy saving performance can be more clearly explained when we look at the velocity-acceleration operating points map in Figure 5.4 and the engine torque-speed operating points map in Figure 5.5 . Figure 5.4 shows that in med-heavy traffic, the front vehicle has large acceleration magnitudes even at high speeds whereas the eco vehicle mostly maintains within a moderate acceleration magnitude (less than 3 $m/s^2$). However, in free flow, both vehicles maintain within a moderate acceleration magnitude. Figure 5.5 shows that in med-heavy traffic, the front vehicle operates the engine at low efficient torque-speed regions whereas the eco vehicle mostly operates near high efficient regions. However, in free flow, both vehicles mostly operate the engine on the high efficient regions. At the end, these two plots explain that our eco vehicle achieves less energy saving because the front vehicle is already driving at a high energy efficiency during the free flow traffic. In fact, the MPGes for both front and eco vehicles are higher than the MPGe of the eco vehicle in med-heavy traffic by more than 4 percents.

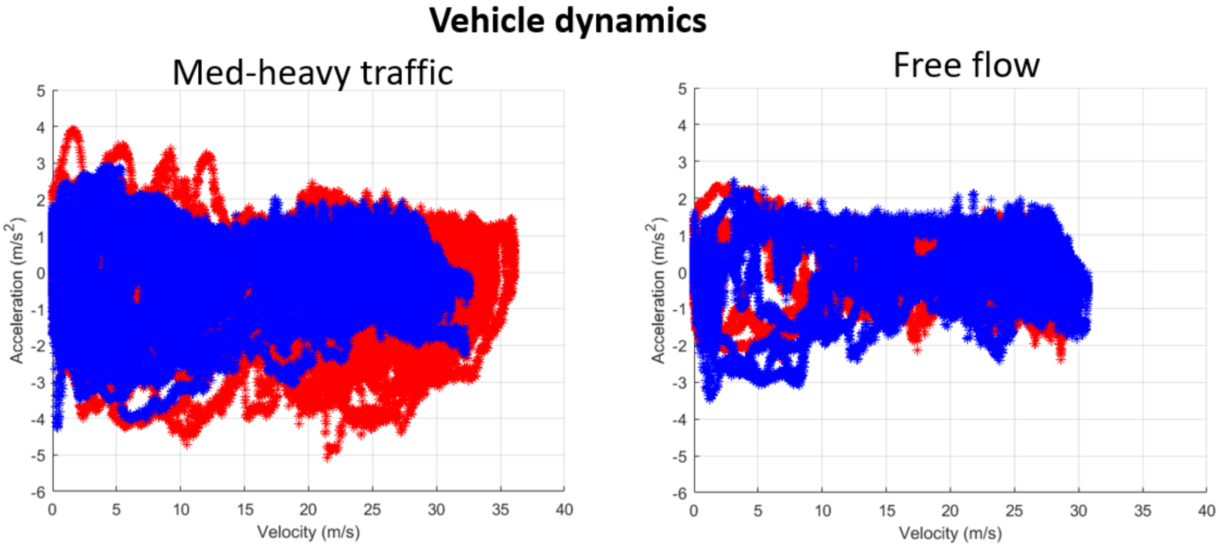


Figure 5.4: Velocity-acceleration operating points plot (map) for med-heavy traffic and free flow experiments. Blue and red marks indicate the data for the eco vehicle and the front vehicle, respectively.

## 5.3 Urban Longitudinal Motion Control with Motion Eco-Driving Planning using V2I

In this section, we present the Eco-driving paired with a safe urban-LMC. The Eco-driving uses statistical and historical signal data as well as real time V2I data to compute a reference velocity with a long horizon. Then, this reference velocity is tackled by our on-board urban-LMC, which ensures safety using the real-time perception and V2I data Finally, we present

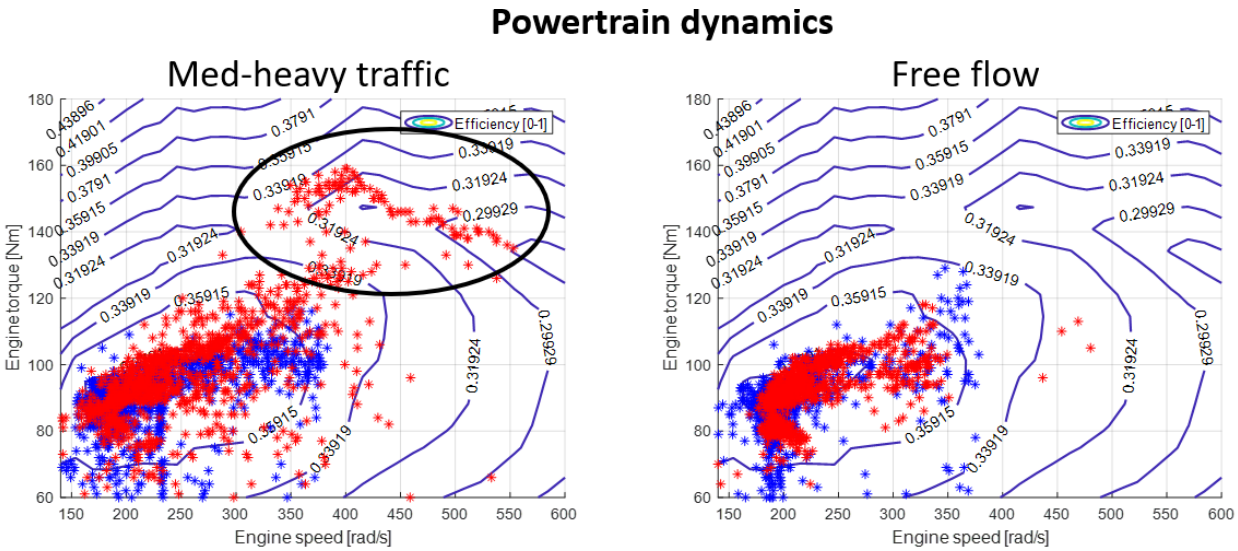


Figure 5.5: Engine torque-speed operating points plot (map) for med-heavy traffic and free flow experiments. Blue and red marks indicate the data for the eco vehicle and the front vehicle, respectively. The black circle shows that the front vehicle operated the engine at low efficient operating points.

the hardware-in-the-loop (HIL) and real-world experimental results to show the efficacy of our approach.

### 5.3.1 Motion Eco-Driving Planning

In this subsection we present our approach for a optimal velocity reference generation, or Eco-driving, in the Motion Eco-Driving Planning module. This velocity reference, computed across a long horizon in space, is obtained by solving an optimization problem which minimizes energy based on both the probabilistic real-time SPaT and its empirical statistics. In the following subsections, we formulate the optimization problem for Eco-driving.

#### Vehicle Dynamics Model

We consider the space dependent dynamics with velocity $v(k)$ and travel time $t(k)$ as states and wheel traction and braking forces as inputs at position $k\Delta s$ where $\Delta s$ is the position step size. The system dynamics can be expressed as

$$x(k+1) = \begin{bmatrix} v(k+1) \\ t(k+1) \end{bmatrix} = f^s(x(k), u(k)) = x(k) + \begin{bmatrix} \frac{a(u(k))\Delta s}{v(k)} \\ \frac{\Delta s}{v(k)+\frac{a(u(k))\Delta s}{v(k)}} \end{bmatrix}, \tag{5.6}$$

where $u(k) = [F_w(k), F_b(k)]^\top$, $a(u(k)) := \frac{dv(k)}{dt}$ is obtained from our longitudinal vehicle dynamics model in (3.1) after replacing the time dependent slope $\vartheta(t)$ with the space dependent slope $\theta(k)$ and approximating $C_x(\cdot)$ with the constant $C_{x,0}$.

### Motion Eco-Driving Planning Optimization Problem

The problem is cast in the longitudinal position domain, as a finite-horizon optimal control problem of the following form.

$$\min_{u(\cdot)} \sum_{k=0}^{N_p^s-1} g(x(k), u(k)) \tag{5.7a}$$

$$\text{subject to}$$

$$x(k+1|t) = f(x^j(k|t), u^j(k|t)), \tag{5.7b}$$

$$x(k) \in \mathcal{X},\ u(k) \in \mathbb{U}, \tag{5.7c}$$

$$\forall k \in [0, 1, ..., N_p^s - 1],$$

$$x^j(t|t) = \bar{x}^j(t), \tag{5.7d}$$

$$x(N_p^s) \in \mathcal{X}_N, \tag{5.7e}$$

where $N_p^s$ is the number of space discretization points; $g(x(k), u(k))$ approximates a trade-off between the travel time and the total energy consumption in terms of total wheel energy or the total power cost for battery and fuel consumption in case of a hybrid electric vehicle like our test vehicles; We refer to [22, 23] for how to obtain $g(x(k), u(k))$ for our test vehicle.

The convex input constraint set $\mathbb{U}$ defines the actuator limits as done in (4.4b); The terminal set $\mathcal{X}_N$ describes the time constraint of the whole trip; The state constraint set $\mathcal{X} := \mathcal{X}_1 \cap \mathcal{X}_2$ describes the surrounding environment. A convex set $\mathcal{X}_1$ models bounds on the speed and the travel time and is expressed as:

$$\mathcal{X}_1 = \{(v, t) : v_{\min} \leq v \leq v_{\max},\ t_{\min} \leq t \leq t_{\max}\} \tag{5.8}$$

where $(t_{\min}, t_{\max})$ represents the minimum and maximum travel time boundaries. $\mathcal{X}_1$ models the vehicle obeying the traffic light rules; i.e., stopping at the red light. Our optimization problem accommodates $N_{TL}$ signalized intersections, assuming they can be approximated as points along the route. If we assume exact forecasts of the red light phase duration $c_r^i$ were available throughout the route, below is how to obtain the set. Every traffic signal has an independent cycle time denoted as $c^i \in \left[0, \overline{c}^i\right]$, $i = 1, \ldots, N_{TL}$, where $c^i = 0$ denotes the beginning of the red light phase and $\overline{c}^i \in \mathcal{R}^+$ is the cycle period. We denote the red light phase duration by $c_r^i \in \left(0, \overline{c}^i\right)$, and by $t_p^i$ the time at which the ego vehicle passes through the $i$-th intersection. In the domain of the intersection cycle time $c_i$, the passing time is computed as $c_p^i = \left(c_0^i + t_p^i\right) \mod \overline{c}^i$, where $c_0^i$ is the cycle time at $s = 0$. We enforce that intersections are not crossed during red light phases by $\mathcal{X}_2 = \left\{t : c_p^i(t) \geq c_r^i,\ \forall i = 1, \ldots, N_{TL}\right\}$.

In practice, many intersections adapt their phase duration based on the time of the day and on the traffic level, making perfect forecasts unrealistic. In this case, [84] and our previous work [23, 22] proposed to replace $\mathcal{X}_2$ with the chance constraint

$$\tilde{\mathcal{X}_2} = \left\{t : \Pr\left[c_p^i(t) \geq c_r^i + \alpha^i\right] \geq 1 - \eta^i,\ \forall i = 1, \ldots, N_{TL}\right\},$$

where $\Pr[A]$ is the probability of event $A$, $\alpha^i \in \left[0, \overline{c}^i - c_r^i\right]$ models the adaptation of the red light phase, and $\eta^i \in [0,1]$ is the level of constraint enforcement.

We refer to [84, 22, 23] for an extensive analysis, implementation details, and approaches to define $\tilde{\mathcal{X}}_2$ based on historical signal timing data.

### 5.3.2 Urban Longitudinal Motion Control with Traffic Lights

Urban longitudinal motion control (urban-LMC) belongs to the Motion Control in the Real-Time Control and Planning module. The design of an urban-LMC is very similar to our the MPC-based LMC (4.6)-(4.8). The difference is that the urban-LMC takes into account of the traffic light signal information which introduces additional position-dependent constraints, to enforce that the downstream intersection is crossed during a green phase.

In this section we present our urban-LMC, of which the objectives are twofold: (i) Track the reference velocity computed by the higher level Motion Eco-Driving Planning. (ii) Enforces vehicle safety, such as collision avoidance with a leading vehicle, and non-violation of traffic light laws regardless of the front vehicle's future behavior and traffic light dynamics.

#### Longitudinal Vehicle Dynamics

The problem is to control the ego and its interaction with the vehicle and the traffic light ahead. Therefore, in addition to the control oriented longitudinal dynamics (3.1), we consider a new additional state, distance to the upcoming traffic light $d_{TL}$.

In short, at time step $t$, the system dynamics discretized with the sampling time $t_s$ is expressed as

$$x(k+1|t) = \begin{bmatrix} d_{TL}(k+1|t) \\ d(k+1|t) \\ v(k+1|t) \end{bmatrix} = \underbrace{\begin{bmatrix} d_{TL}(k|t) - t_s v(k|t) \\ d(k|t) + t_s(v_f(k|t) - v(k|t)) \\ v(k|t) + \frac{t_s}{M}(F_w(k|t) - F_b(k|t) - F_R(\vartheta(t), d(k|t), v(k|t))) \end{bmatrix}}_{f^{tl}(x(k|t), u(k|t), v_f(k|t))} \tag{5.9}$$

where $u(k|t) = [F_w(k|t), F_b(k|t)]^\top$, $F_R(\vartheta(k|t), d(k|t), v(k|t))$ is given in (3.2). Note that we use the vehicle dynamics based on time step $t$ instead of the dynamics based on position step $k$ (5.6).

#### Traffic Light Constraints

In addition to the constraints (4.4), we now have additional constraint to ensure that the vehicle stops at the red light. This can be written as

$$0 \le d_{TL}(k|t), \qquad \text{if } p^{\text{up}}(t) = \text{red}, \tag{5.10a}$$

$$0 \le d_{TL}(k|t) + \phi(t|k), \ \text{if } p^{\text{up}}(t) = \text{yellow}, \tag{5.10b}$$

where $p^{\text{up}}(t)$ denotes the phase of the upcoming traffic light and $\phi(t) \geq 0$ is a slack variable at time step $t$; The constraint (5.10a) ensures the ego vehicle to stop at the red light by keeping the distance to the traffic light larger than zero. We assume that the yellow light duration is long enough for the vehicle to come to a full stop with the maximum deceleration imposed by the actuator input constraint (4.4b). This assumption with the soft constraint in (5.10b) ensures that the vehicle is either capable of the full stop before the traffic light or passes the light when it turns to red from yellow.

### Urban-LMC Design

Our urban-LMC is based on the MPC-based LMC (4.6)-(4.8) with modest changes. At time $t$, it solves the following optimzation problem:

$$\min_{u(\cdot|t)} \sum_{k=t+1}^{t+N_p-1} g(x(k|t), u(k|t), \phi(k|t)) \tag{5.11a}$$

subject to

$$x(k+1|t) = f^{tl}(x(k|t), u(k|t), v_f(k|t)), \tag{5.11b}$$

$$v_f(k+1|t) = v_f(k|t) + t_s \tilde{a}^f(k|t), \tag{5.11c}$$

$$x(k|t) \in \mathbb{X},\ u(k|t) \in \mathbb{U}, \tag{5.11d}$$

$$\begin{cases} 0 \leq d_{TL}(\ell|t) & \text{if } p^{\text{up}}(t) = \text{red}, \\ 0 \leq d_{TL}(\ell|t) + \phi(\ell|t) & \text{if } p^{\text{up}}(t) = \text{yellow}, \end{cases} \tag{5.11e}$$

$$\forall k \in [t, ..., t + N_p - 1]$$

$$\begin{cases} [x(t+N_p|t)^\top, v_f(N_p|t)]^\top \in \mathbb{C}_{\text{TL}} & \text{if } p^{\text{up}}(t) = \text{red}, \\ [x(t+N_p|t)^\top, v_f(N_p|t)]^\top \in \mathbb{C}_f & \text{otherwise}, \end{cases} \tag{5.11f}$$

$$x(t|t) = \bar{x}(t|t),\ v_f(t|t) = \bar{v}_f(t|t) \tag{5.11g}$$

where the cost function (5.11a) includes a penalty for deviating from $v_{\text{ref}}$, a penalty on input forces, a penalty for jerk, and, finally, a penalty for violating the soft yellow light constraint in (5.10b) with the weights $W_v$, $W_u$, $W_{\Delta u}$, and $W_\phi$, respectively. The cost function is compactly written as:

$$\begin{aligned} g(x(k|t), u(k|t), &\phi(k|t)) = \\ &\|v(k|t) - v_{\text{ref}}\|_{W_v} + \|u(k|t)\|_{W_u} + \|u(k|t) - u(k-1|t)\|_{W_{\Delta u}} + \|\phi(k|t)\|_{W_\phi}. \end{aligned} \tag{5.12}$$

Finally, in (5.11f), we modify the terminal constraint set $\mathbb{C}$ to $\mathbb{C}_f$ by including the additional unconstrained state $d_{TL}$ (expanding the set dimension). To construct $\mathbb{C}_{\text{TL}}$, we simply assume there is a stopped vehicle at the distance $d_{TL}$ and obtain the invariant set using the same method to obtain $\mathbb{C}$ in Section 4.2.4.

The first input $u^*(t|t)$ from the solution of (5.11) is applied to the system during the time interval $[t, t+1)$;

$$u^i(t) = u^*(t|t). \tag{5.13}$$

At the next time step $t+1$, a new optimal control problem in the form of (5.11), based on new measurements of the state, is solved over a shifted horizon, yielding a *moving* or *receding* horizon control strategy.

### 5.3.3 Hardware-in-the-Loop and Experimental Results

This subsection demonstrates our urban-LMC (5.11)-(5.13) with the Motion Eco-Driving Planning (5.7). The HIL simulation setup and real-world public road experiment setup are provided in Section 2.4 and Section 2.5, respectively.

Our test vehicle is a Hyundai Ioniq with the PHEV powertrain controlled by the charge-depleting and charge-sustaining (CD-CS) strategy (see Section. 3 for details about the vehicle dynamics modeling and [34] for the powertrain dynamics modeling). Also note that in the CD-CS strategy, the vehicle utilizes electric-only mode (CD phase) until the battery state of charge (SOC) reaches a minimum limit. Afterwards, it consumes fuel more aggressively to maintain SOC near the minimum, and provide propulsion (CS phase) [34]. Throughout our simulation and experiments, we focused on charge-depleting mode consistently so we can compare savings in MPGe between datasets without intervention of changes in the poewrtrain control mode. Note that MPGe calculates that 33.7 kilowatt-hours of electricity is equivalent to one gallon of gas [45].

Before we run multiple simulations and analyze energy savings of our controller, we first check the sole performance of our urban-ACC controller urban-LMC (5.11)-(5.13) . We built a HIL scenario where the ego vehicle is running through a series of simulated traffic lights with other traffic vehicles. We applied our urban-ACC controller with the constant desired velocity of $11 m/s$ and Figure 5.6 depicts the outcome of the simulation. As seen, the ego vehicle can maneuver safely in the urban street which involves other traffic vehicles and traffic lights.

As our HIL simulation/testing route, we consider the Live Oak Avenue in Arcadia, CA, USA, as depicted in Figure 2.9. The route is flat and 2.6km long and has 8 signalized intersections. More details about the road setup is provided in Section 2.5.2.

#### Energy Saving Performance using HIL Simulations

We validate the energy savings of our urban-LMC (5.11)-(5.13) with the Motion Eco-Driving Planning (5.7) using our HIL setup. We ran simulations in various traffic scenarios, based on Monte Carlo simulations. At each simulation, traffic schedules (e.g., red light duration, time shift of cycle initiation at each intersection) are randomly sampled from empirical PDFs conditioned to the hour of the day. We focused on charge-depleting mode over demonstrations. To compare the performance of our controller, we compare the urban-LMC (5.11)-(5.13) with

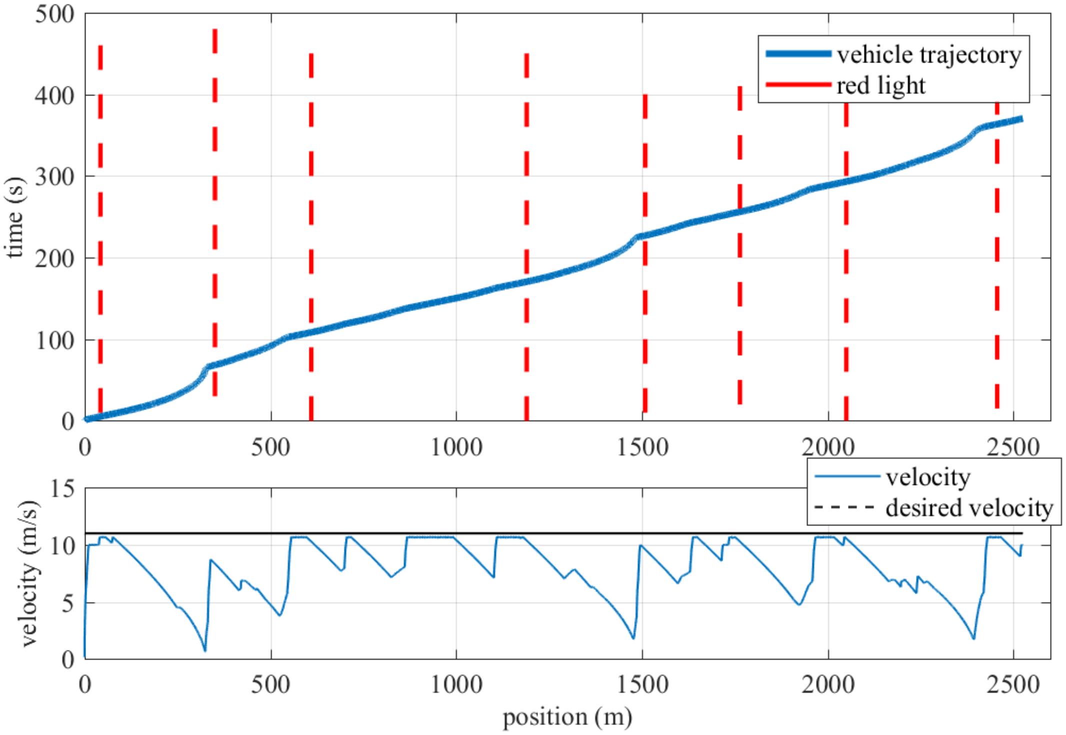


Figure 5.6: HIL simulation for a vehicle running through a series of simulated traffic lights with other traffic vehicles: first plot shows the vehicle and red light trajectories; second plot shows the closed loop trajectories of velocity.

the Motion Eco-Driving Planning (5.7) and the urban-LMC (5.11)-(5.13) with a constant velocity reference.

Figure 5.7 depicts the probability distributions of MPGe (top) and arrival times (bottom). As seen, the urban-LMC with the Motion Eco-Driving Planning saves a significant energy consumption of 27.31% compared to that of the urban-LMC with a constant velocity reference. This average saving was seen consistently in any traffic scenarios at any hour of the day. However, the Motion Eco-Driving Planning results in longer travel time (15.41%) compared to that of the urban-LMC with a constant velocity reference.

### Energy Saving Performance using Rublic road Experiments

We demonstrated our urban-LMC (5.11)-(5.13) with the Motion Eco-Driving Planning (5.7) on the public road, of which the setup is detailed in Section 2.5.2. The demonstrations were performed over different time of day where traffic flow varies from free-flow to dense. We focused on charge-depleting mode over demonstrations. Two vehicles (of the same vehicle model) drive the route simultaneously with roughly 10 seconds of time gap in start time. In one vehicle, we demonstrated the urban-LMC with a constant velocity reference and in

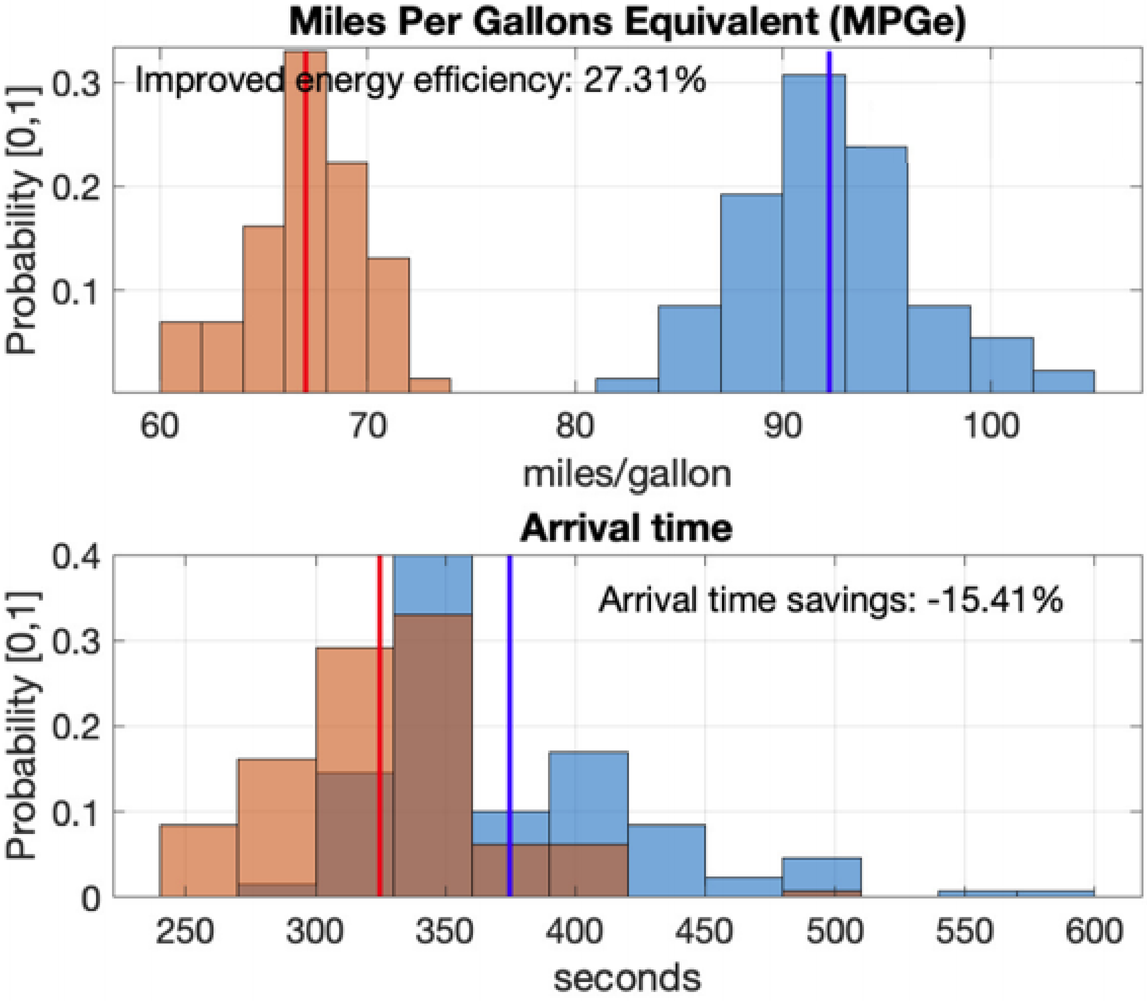


Figure 5.7: The probability distribution of MPGe (top) and arrival times (bottom) obtained by the HIL simulations. The distributions consist of a total of 130 traffic scenarios in each ECO-ACC and ACC-Only cases. The blue bar and line represent the probability bin and mean value, respectively, for the urban-LMC with the Motion Eco-Driving Planning and the red bar and line represent the probability bin and mean value, respectively, for the urban-LMC with a constant velocity reference.

the other one, we demonstrated the urban-LMC with the Motion Eco-Driving Planning. In total, we collected 17 sets of a complete run through the eight signals.

Figure 5.8 shows one instance of the velocity profiles during our experiments. As seen, the urban-LMC with the Motion Eco-Driving Planning has smoother acceleration and deceleration profiles than the urban-LMC with a constant velocity reference.

Figure 5.9 depicts the probability distributions of MPGe (top) and arrival times (bottom). Consistent with the HIL simulations shown in Figure 5.7, the urban-LMC with the Motion Eco-Driving Planning yields 30.98% of improvement in MPGe at the cost of 8.51% of additional travel time. We also noticed that there exists a mismatch of magnitudes both in MPGe between the HIL simulation and the public road experiments. We believe that this difference is due to the mismatch in powertrain dynamics modeling which fails to capture the modeling of occasional engine switch-ons during the charge-depleting mode. Also, because each run only lasts for a short time period of about 6 minutes, even a small mistmatch in the number of engine switch-ons can result in significant mistmatch for MPGe. Nevertheless, we observed the significant energy savings in our experiments consistent with the results from our HIL simultion.

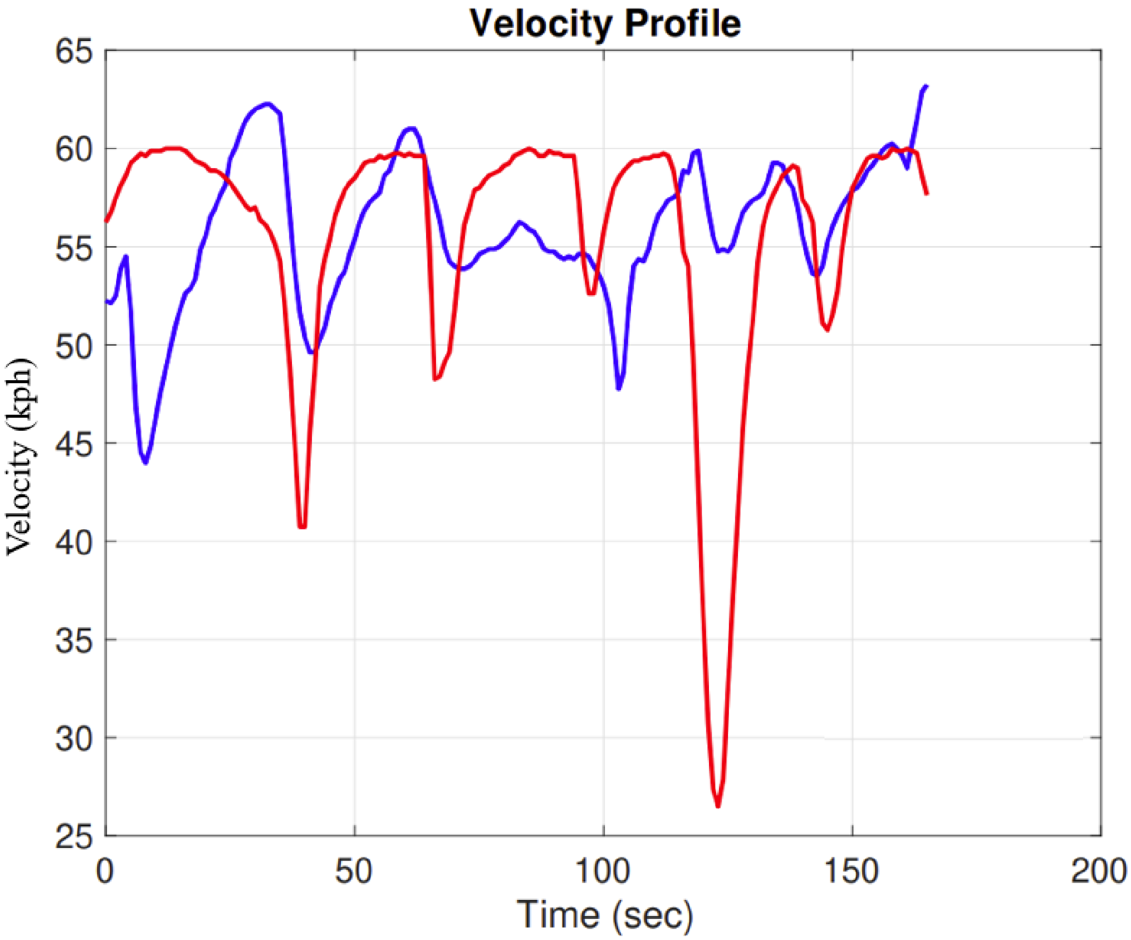


Figure 5.8: Velocity profiles from the public road experiment. The blue line represents the urban-LMC with the Motion Eco-Driving Planning and the red line represents the urban-LMC with a constant velocity reference.

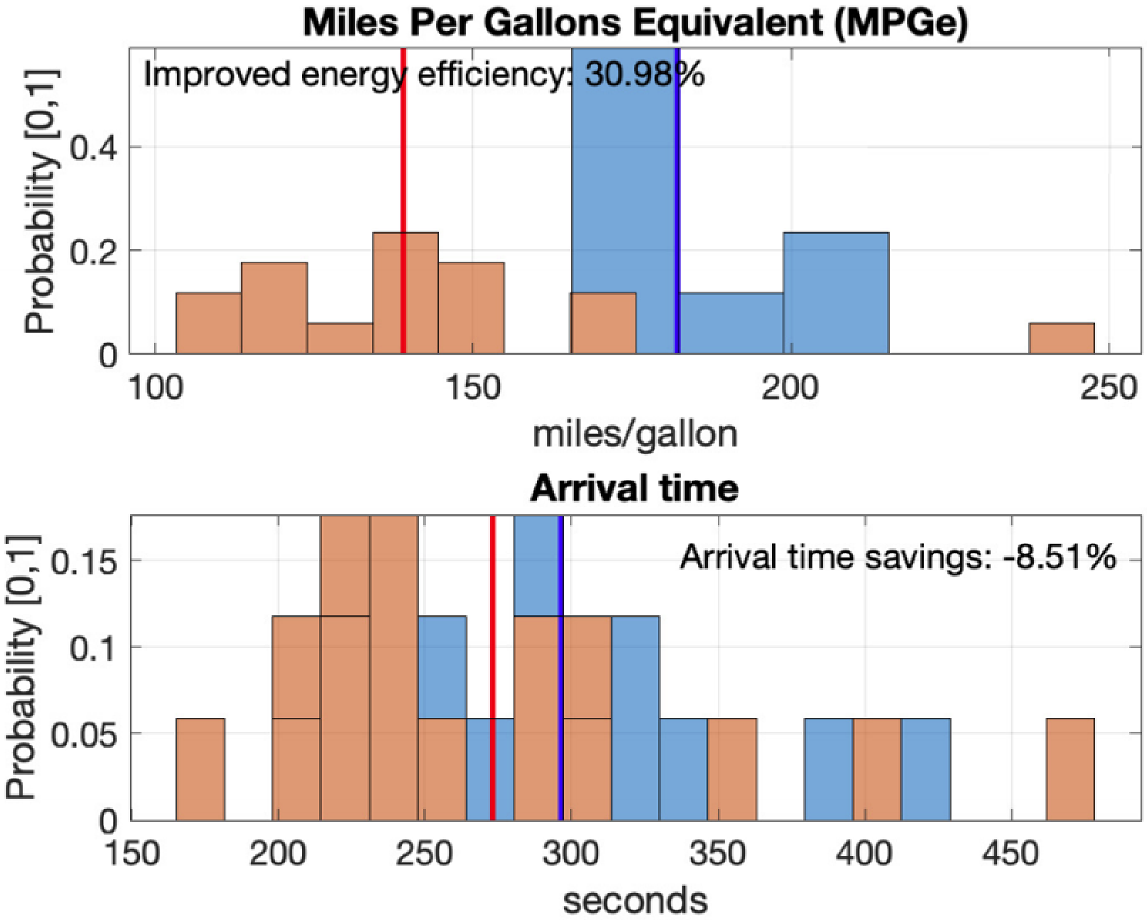


Figure 5.9: The probability distribution of MPGe (top) and arrival times (bottom) obtained by the HIL simulations. The distributions consist of a total of 17 traffic scenarios in each ECO-ACC and ACC-Only cases. The blue bar and line represent the probability bin and mean value, respectively, for the urban-LMC with the Motion Eco-Driving Planning and the red bar and line represent the probability bin and mean value, respectively, for the urban-LMC with a constant velocity reference.

## 5.4 Predictive Cruise Control with Learning Model Predictive Control using V2C

We apply the Learning Model Predictive Control (LMPC) framework to Predictive Cruise Control (PCC) of a vehicle which repeatedly drives along the same route. We modify the original LMPC so that we can enforce the task (trip) to be completed within a time limit. Moreover, in every trip, our controller can learn the static environment features such as road grade and attempts to improve its performance, which includes total energy consumption and comfort. We present the simulation results to show efficacy for the proposed PCC.

### 5.4.1 PCC Design

We aim to build PCC of a CAV performing repetitive trips along a fixed route, subject to a position-dependent road grade and a total completion time constraint. We refer to each successful trip of the route as an *iteration.* Each iteration has the same boundary conditions (initial and final position, speed, and force), and has to be completed within the desired time limit $N_f$; $x_0 = x_s$ and $x_{N_f} = x_f$. At each iteration, the controller finds a velocity trajectory that maintains or improves the specified vehicle performance objective (such as fuel economy or comfort).

#### Vehicle Dynamics

The problem is to control a single vehicle (referred to as the ego vehicle) and its interaction with the road ahead in the trip. We focus on two main dynamics, (i) the lumped longitudinal dynamics of the ego vehicle similar to (3.1), and (ii) its actuator dynamics as a first order dynamic model. A first order dynamic model is used for the actuation of traction and braking forces through engine and brakes.

The states of our model include the distance travelled along the route $s$, the velocity $v$, and the force at the wheel $F$; $x(t) = \begin{bmatrix} s(t), & v(t), & F(t) \end{bmatrix}^\top$ at time $t$. The inputs to the model are $F_w$ and $F_b$, which represent the wheel-level desired traction and braking forces, respectively; $u(t) = \begin{bmatrix} F_w(t), & F_b(t) \end{bmatrix}^\top$ at time $t$. The system dynamics are written as

$$x(t+1) = \begin{bmatrix} s(t) + t_s v(t) \\ v(t) + \frac{t_s}{m}(F(t) + mg(c_r \cos(\vartheta) + \sin(\vartheta)) + \frac{1}{2}\rho A C_d v(t)^2) \\ \left(1 - \frac{t_s}{\tau}\right) F(t) + \frac{t_s}{\tau}(F_w(t) + F_b(t)) \end{bmatrix} \tag{5.14}$$

where $t_s$ is the sampling time; $m$ and $\tau$ are the mass and the time constant of force actuation, respectively; $\vartheta$ is the road pitch angle; $g$ and $c_r$ are the gravity and rolling coefficients, respectively; $\rho$, $A$, and $C_d$ are the air density, frontal area, and drag coefficient constants, respectively. Moreover, the road angle $\vartheta$ is approximated as $\hat{\vartheta}$ using a quadratic function of the distance

$$\hat{\vartheta}(s(k)) = a_0(k) + a_1(k)s(k) + a_2(k)s(k)^2 \,, \tag{5.15}$$

where $(a_0(k), a_1(k), a_2(k))$ are parameters that are computed as follows. In the initial iteration (trip), they are initialized as zeros. During each iteration of the trip, we store velocity and force values at each position along the route; we then introduce $\bar{\vartheta}$ to estimate the angle by inverting the dynamics (5.14). At each time step $k$ of the $j$-th iteration ($j > 1$), given $s(k)^j$ the parameters $(a_0^j(k), a_1^j(k), a_2^j(k))$ are estimated on-line solving the following least mean squares problem,

$$\underset{(a_0^j(k),a_1^j(k),a_2^j(k))}{\arg\min} \sum_{(p,l)\in\mathcal{G}(s(k)^j)} \left\| \begin{bmatrix} 1 & s^l(p) & (s^l(p))^2 \end{bmatrix} \begin{bmatrix} a_0^j(k) \\ a_1^j(k) \\ a_2^j(k) \end{bmatrix} - \bar{\vartheta}_p^l \right\| \tag{5.16}$$

where $\mathcal{G}(s(k)^j)$ is the set of indices and the iteration numbers with the following property

$$\mathcal{G}(s(k)^j) = \Big\{(p,l) : s^j(k) \leq s^l(p) \leq s^j(k) + d_{\mathrm{f}}\Big\}. \tag{5.17}$$

where $d_f$ is look-ahead distance which is considered a tuning parameter. A similar method is adopted for system identification of road curvature in [88].

In the remainder of this paper, the approximated vehicle dynamics model (5.14)-(5.16) is compactly rewritten as

$$x(k+1) = f(x(k), u(k)). \tag{5.18}$$

Also, we consider state and input constraints in the form

$$x \in \mathbb{X} := \{(s, v, F) : 0 \leq v \leq v_{\max},\ F_{\min} \leq F \leq F_{\max}\}, \tag{5.19a}$$

$$u \in \mathbb{U} := \{(F_w, F_b) : 0 \leq F_w \leq F_{\max},\ F_{\min} \leq F_b \leq 0\}. \tag{5.19b}$$

We are interested in designing a predictive cruise controller which tries to improve its energy consumption performance as it repeats the same route. Therefore, the goal of our controller is to minimize this estimate of the energy consumption, $h(x, u)$.

### Predictive Cruise Control Problem

For the $j$-th iteration of trip, we can formulate the predictive cruise control problem as the following constrained finite horizon optimal control problem.

$$\min_{u^j(\cdot)} \sum_{k=0}^{N_f-1} h(x^j(k), u^j(k)) \tag{5.20a}$$

subject to

$$x^j(0) = x_s, \quad x^j(N_f) = x_f \tag{5.20b}$$

$$x^j(k+1) = f(x^j(k), u^j(k)), \quad \forall k \in [0, ..., N_f - 1], \tag{5.20c}$$

$$x^j(k) \in \mathbb{X},\ u^j(k) \in \mathbb{U}, \quad \forall k \in [0, ..., N_f - 1], \tag{5.20d}$$

where $j$ is the iteration number; (5.20b) represents the boundary conditions of the trip.

### 5.4.2 Learning Model Predictive Control with Time Constraint

In this section, a formulation of LMPC with time constraint is proposed. Solving a finite time constrained optimal control problem such as (5.20) in real time can be difficult, especially when $N_f$ is large. Therefore, we design LMPC which tries to solve the problem (5.20) and can be implemented in real time. In previous works, LMPC was introduced for repetitive and iterative tasks [87]. LMPC leverages past data to progressively improve performance while ensuring recursive feasibility, asymptotic stability, and non-increasing cost at every iteration. In this work, we extend the LMPC framework with a constraint on the time required to complete the task. In other words, we guarantee that each iteration or repetition does not exceed a total time limit.

**Remark 5.** *In the original work of LMPC in [87], the optimal control problem is defined on infinite horizon. In our problem, we focus on the finite time formulation* (5.20)*.*

#### Time Sampled Safety Set

We denote the input sequence applied to the dynamics (5.14) and the corresponding closed loop state trajectory at j-th iteration as

$$\boldsymbol{u^j} = [u^j(0), u^j(1), ..., u^j(N_f - 1)], \tag{5.21a}$$

$$\boldsymbol{x^j} = [x^j(0), x^j(1), ..., x^j(N_f)] \tag{5.21b}$$

where $u^j(t)$ and $x^j(t)$ are the input and the state at time $t$ of the j-th iteration, respectively.

The main contribution of LMPC with time constraint is the modification of the safety set in [87] to the *Time Sampled Safety Set*. We define the time sampled safety set $\mathcal{SS}^j_{\text{time}}$ at j-th iteration as

$$\mathcal{SS}^j_{\text{time}}(t) = \cup_{i=1}^{j} \cup_{k=t}^{N_f} \, x^i(k) \tag{5.22}$$

where $N_f$ is the time limit for each iteration. The difference from the original definition of the safety set is that it only includes the states visited during the remaining time, $N_f - t$. Note that $\mathcal{SS}^j_{\text{time}}(N_f)$ is only $x_f$ since each trip must finish within the time constraint $N_f$.

#### Preliminaries

In this section, we introduce some terminology used for the LMPC problem with time constraint.

At time $t$ of the j-th iteration, we define the cost-to-go associated with the input sequence (5.21a) and the corresponding state trajectory (5.21b) as

$$J^j_{t \to N_f}(x(t), t) = \sum_{k=t}^{N_f} h(x^j(k), u^j(k)) \tag{5.23}$$

where $h(\cdot)$ is the stage cost function. We have the following assumption about the stage cost $h(x, u)$.

**Assumption 1.** *$h(\cdot,\cdot)$ is a continuous function which has the following property:*

$$h(x_f, u) = 0 \text{ and } h(x,u) \geq 0 \ \forall x \in \mathbb{R}^{n_x},\ u \in \mathbb{R}^{n_u}.$$

*where $n_x$ and $n_u$ are the dimensions of $x$ and $u$, respectively.*

For any $x \in \mathcal{SS}^j_{\text{time}}(t)$, we can define the minimum cost-to-go function $Q^j(x,t)$ as

$$Q^j(x,t) = \begin{cases} \min\limits_{(i,l)\in\mathcal{F}^j(x,t)} \ J^i_{l\rightarrow N_f}(x,l) \text{ if } x \in \mathcal{SS}^j_{\text{time}}(t) \\ +\infty \quad \text{else} \end{cases} \tag{5.24}$$

where $\mathcal{F}^j(x,t)$ is defined as

$$\mathcal{F}^j(x,t) = \{(i,l) : i \in [1,j],\ l \geq t,\ x = x^i(l) \text{ such that } x^i(l) \in \mathcal{SS}^i_{\text{time}}(l)\}.$$

Note that the definition of the function $Q(\cdot,\cdot)$ is modified from the original definition in [87] because we use the new time sampled safety set $\mathcal{SS}^i_{\text{time}}(\cdot)$.

### LMPC with Time Constraint Formulation

At time $t$ of iteration $j > 1$, our LMPC with time constraint solves the following optimization problem:

$$\min_{u(\cdot|t)} \sum_{k=t}^{t+N_p-1} h(x^j(k|t), u^j(k|t)) + Q^{j-1}(x(N|t), t) \tag{5.26a}$$

subject to

$$x^j(k+1|t) = f(x^j(k|t), u^j(k|t)), \tag{5.26b}$$

$$x^j(k|t) \in \mathbb{X},\ u^j(k|t) \in \mathbb{U}, \tag{5.26c}$$

$$\mathbb{1}_A(k) x^j(k|t) = \mathbb{1}_A(k) x_f, \tag{5.26d}$$

$$\forall k \in [t, ..., t+N_p-1],$$

$$x^j(t|t) = \bar{x}^j(t), \tag{5.26e}$$

$$x^j(t+N_p|t) \in \mathcal{SS}^{j-1}_{\text{time}}(t+N_p), \tag{5.26f}$$

where $N_p$ is the MPC prediction horizon; (5.26d) is the constraint which forces the system to stay at $x_f$ at $t \geq N_f$; $\mathbb{1}_A(\cdot)$ is the indicator function of the set $A$ defined as $A = \{t \in \mathbb{R} : t \geq N_f\}$; $\bar{x}^j(t)$ in (5.26e) is the perfect state measurement at time $t$; (5.26f) is the terminal constraint which imposes the system to be driven into the safe set sampled from last iteration. $Q^1(\cdot,\cdot)$ and $\mathcal{SS}^1_{\text{time}}(\cdot)$ are defined by the initial successful trip.

The resulting optimal states and inputs of (5.26) are denoted as

$$\boldsymbol{x^{*,i}(t)} = [x^{*,i}(t|t), x^{*,i}(t+1|t), ..., x^{*,i}(t+N_p|t)], \tag{5.27a}$$

$$\boldsymbol{u^{*,i}(t)} = [u^{*,i}(t|t), u^{*,i}(t+1|t), ..., u^{*,i}(t+N_p-1|t)]. \tag{5.27b}$$

Then, the first input $u^{*,i}(t|t)$ is applied to the system during the time interval $[t, t+1)$;

$$u^i(t) = u^{*,i}(t|t). \tag{5.28}$$

At the next time step $t+1$, a new optimal control problem in the form of (5.26), based on new measurements of the state, is solved over a shifted horizon, yielding a *moving* or *receding* horizon control strategy with control law.

It is noted that with the assumptions 1, LMPC with time constraint (5.26)-(5.28) is recursively feasible and the cost of each iteration (trip) monotonically decreases. The proof is similar to the original work of LMPC in [87]. The key difference between the two LMPC frameworks is that the time sampled safety set shrinks in the course of time whereas in [87], the safety set is time independent; however, this doesn't affect the proof because the time sampled safety set always includes at least one point (state) which guarantees the existence of the feasible input and the cost decreases (or stays the same) at the next time step; therefore, iteration cost decreases (or stays the same) as the iteration progresses.

**Remark 6.** *LMPC with time constraint* (5.26)-(5.28) *can be reformulated as a robust control and take into account of dynamic environment with minor modifications [92].*

### 5.4.3 LMPC Relaxation for PCC

In this section we apply the LMPC with time constraint (5.26)-(5.28) to a predictive cruise controller subject to repetitions of the same commute with a total time limit. Solving the optimization problem (5.26) in real time is computationally challenging since the safety set $\mathcal{SS}^{j-1}_{\text{time}}(\cdot)$ is a set of discrete and (5.26) is a Mixed Integer Programming (MIP) problem. Therefore, we use an approximation method for (5.26): we introduce approximation functions for the time sampled safety set, $\mathcal{SS}_{\text{time}}(\cdot)$, and the terminal set, $Q(\cdot)$.

Because we restrict the vehicle velocity to be positive semi-definite, the distance travelled $s$ always monotonically increases with time $t$. Therefore, we use $s$ to shrink the time sampled set at each time. At time $t$ of j-th iteration, we approximate $\mathcal{SS}_{\text{time}}(\cdot)$ with

$$\hat{\mathcal{SS}}_{\text{time}}(t) = \Big\{(s, v, F) : s \geq s^{j-1}(t), \begin{bmatrix} v \\ F \end{bmatrix} = \Lambda \begin{bmatrix} 1 \\ s \\ s^2 \end{bmatrix} \Big\} \tag{5.29}$$

where $\Lambda \in \mathbb{R}^{2\times 3}$ is the solution of the following least mean square optimization problem

$$\arg\min_{\Lambda} \sum_{k\in\mathcal{T}(s^j(t))} \left\| \begin{bmatrix} v^{j-1}(k) \\ F^{j-1}(k) \end{bmatrix} - \Lambda \begin{bmatrix} 1 \\ s^{j-1}(k) \\ (s^{j-1}(k))^2 \end{bmatrix} \right\| \tag{5.30}$$

where $\mathcal{T}(s^j(t))$ defines the time steps in which the distance travelled during the previous $j-1$-th iteration of trip is between the current distance and a far enough distance forward, $d_f$:

$$\mathcal{T}(s^j(t)) = \Big\{k : s^j(t) \leq s^{j-1}(k) \leq s^j(t) + d_{\text{f}}\Big\}. \tag{5.31}$$

It's noted that $d_\mathrm{f}$ is a tuning parameter decided by the control designer.

In order to approximate the cost-to-go function $Q(\cdot,\cdot)$, we introduce the third-order polynomial function $\mathcal{C}(\cdot)$

$$\mathcal{C}(s) = \begin{bmatrix}1 & s & s^2 & s^3\end{bmatrix} \Delta \tag{5.32}$$

where $\Delta \in \mathbb{R}^4$ is the solution of the following least mean square optimization problem

$$\arg\min_{\Delta} \sum_{k \in \mathcal{T}(s^j(t))} \left\| J^{j-1}_{k\rightarrow N_f} - \begin{bmatrix}1 & s^j(k) & (s^j(k))^2 & (s^j(k))^3\end{bmatrix} \Delta \right\| \tag{5.33}$$

where $J^{j-1}_{k\rightarrow N_f}$ is the cost-to-go function defined in (5.23).

Finally, we approximate $Q(\cdot,\cdot)$ with

$$\hat{Q}^j(x,t) = \begin{cases} \mathcal{C}(s) \text{ if } x \in \hat{\mathcal{SS}}^j_{\text{time}}(t) \\ +\infty \quad \text{else} \end{cases} \tag{5.34}$$

where $x = \begin{bmatrix}s, & v, & F\end{bmatrix}^T$.

### LMPC with Time Constraint Relaxation

With the approximation functions (5.29)-(5.34), we can reformulate LMPC with time constraint (5.26)-(5.28) as the following optimal control problem:

$$\min_{u(\cdot|t)} \sum_{k=t}^{t+N_p-1} h(x^j(k|t), u^j(k|t)) + \hat{Q}^{j-1}(x(N_p|t), t) \tag{5.35a}$$

subject to

$$x^j(k+1|t) = f(x^j(k|t), u^j(k|t)), \tag{5.35b}$$

$$x^j(k|t) \in \mathbb{X},\ u^j(k|t) \in \mathbb{U}, \tag{5.35c}$$

$$\mathbb{1}_A(k)x^j_{k|t} = \mathbb{1}_A(k)x_f, \tag{5.35d}$$

$$\forall k \in [0, ..., N_p - 1],$$

$$x^j(t|t) = \bar{x}^j(t), \tag{5.35e}$$

$$x^j(N_p|t) \in \hat{\mathcal{SS}}^{j-1}_{\text{time}}(t+N_p), \tag{5.35f}$$

The resulting optimal states and inputs of (5.35) are denoted as

$$\hat{\boldsymbol{x}}^{*,i}(\boldsymbol{t}) = [x^{*,i}(t|t), x^{*,i}(t+1|t), ..., x^{*,i}(t+N_p|t)], \tag{5.36a}$$

$$\hat{\boldsymbol{u}}^{*,i}(\boldsymbol{t}) = [u^{*,i}(t|t), u^{*,i}(t+1|t), ..., u^{*,i}(t+N_p-1|t)]. \tag{5.36b}$$

Then, the first input $u^{*,i}_{0|t}$ is applied to the system during the time interval $[t, t+1)$;

$$u^i(t) = u^{*,i}(t|t). \tag{5.37}$$

At the next time step $t+1$, a new optimal control problem (5.35) with new measurements of the state, is solved over a shifted horizon.

### 5.4.4 Simulation Results

In this section we validate the proposed LMPC controller (5.35)-(5.37) with simulation results. Unlike other simulation and experimental results presented in this thesis, we apply the proposed PCC to an internal combustion engine vehicle of which the sole energy source is fuel; this way, we can simply express the total energy consumption as the total fuel consumption which can approximated by adopting a polynomial approximation method [93] such as the following:

$$f_{\text{fuel}}(v, F_w) = f_{\text{cruise}}(v) + f_{\text{accel}}(v, F_w)\,, \tag{5.38}$$

where

$$f_{\text{cruise}}(v) = (b_0 v + b_1 v^2 + b_2 v^3)\,,\ f_{\text{accel}}(v, F_w) = F_w(c_0 + c_1 v + c_2 v^2)\,,$$

and $(b_0, b_1, b_2, c_0, c_1, c_2)$ are parameters identified by least mean squares fitting of the experimental fuel rate data. The goal of our controller is to minimize this estimate of the fuel consumption, $h(x, u) = f_{\text{fuel}}(v, F_w)$.

Our CAV is repeating the same trip from $s_s = 0$m to $s_f = 5000$m in the Berkeley hills area, depicted in Figure 5.10. This route is subject to position-dependent slope as shown in the top plot in Figure 5.11. Each trip is initialized with $x_s = [s_s, 0, 0]$ and ends with $x_f = [s_f, 0, 0]$. We initialize the trip with a simple velocity tracking controller with a constant velocity reference.

Figure 5.11 depicts the closed loop trajectories for the first iteration of the trip and the 8-th iteration of the same trip. In every iteration of the trip, the arrival times does not exceed the terminal time. Also, as the iteration progresses, the velocity becomes higher in downhill sections and lower in uphill sections. This trend helps decrease the total fuel consumption of the trip as it uses the downhill regions to speed up and the uphill regions to slow down, leading to reduced acceleration. This behavior is also seen in force trajectories. Over the course of iterations, only in uphills, our controller maintains positive wheel force whereas in downhills, it tends to apply less braking (except near the end of trip where the vehicle must come to a full stop); therefore, it wastes less amount of energy.

Figure 5.12 depicts the normalized total cost (fuel consumption) for each iteration of the complete trips when the first trip is completed with a constant velocity tracking controller. As seen, the total cost generally decreases as the iteration number increases. There is about 4.5% reduction in fuel consumption only after 8-th trip compared to the 1st trip. It is also noted that the learning rate decreases with the iterations, as the total cost converges. This result is analogous to those in other applications of LMPC [88, 89].

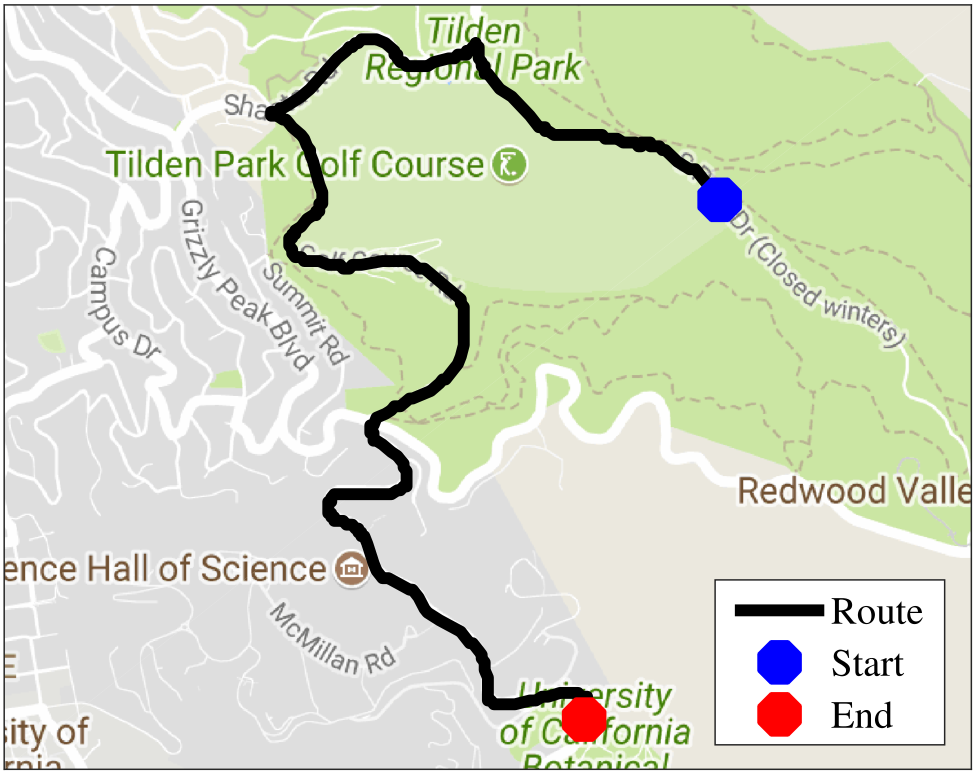


Figure 5.10: Fixed route from an origin A and a destination B in the Berkeley hill area

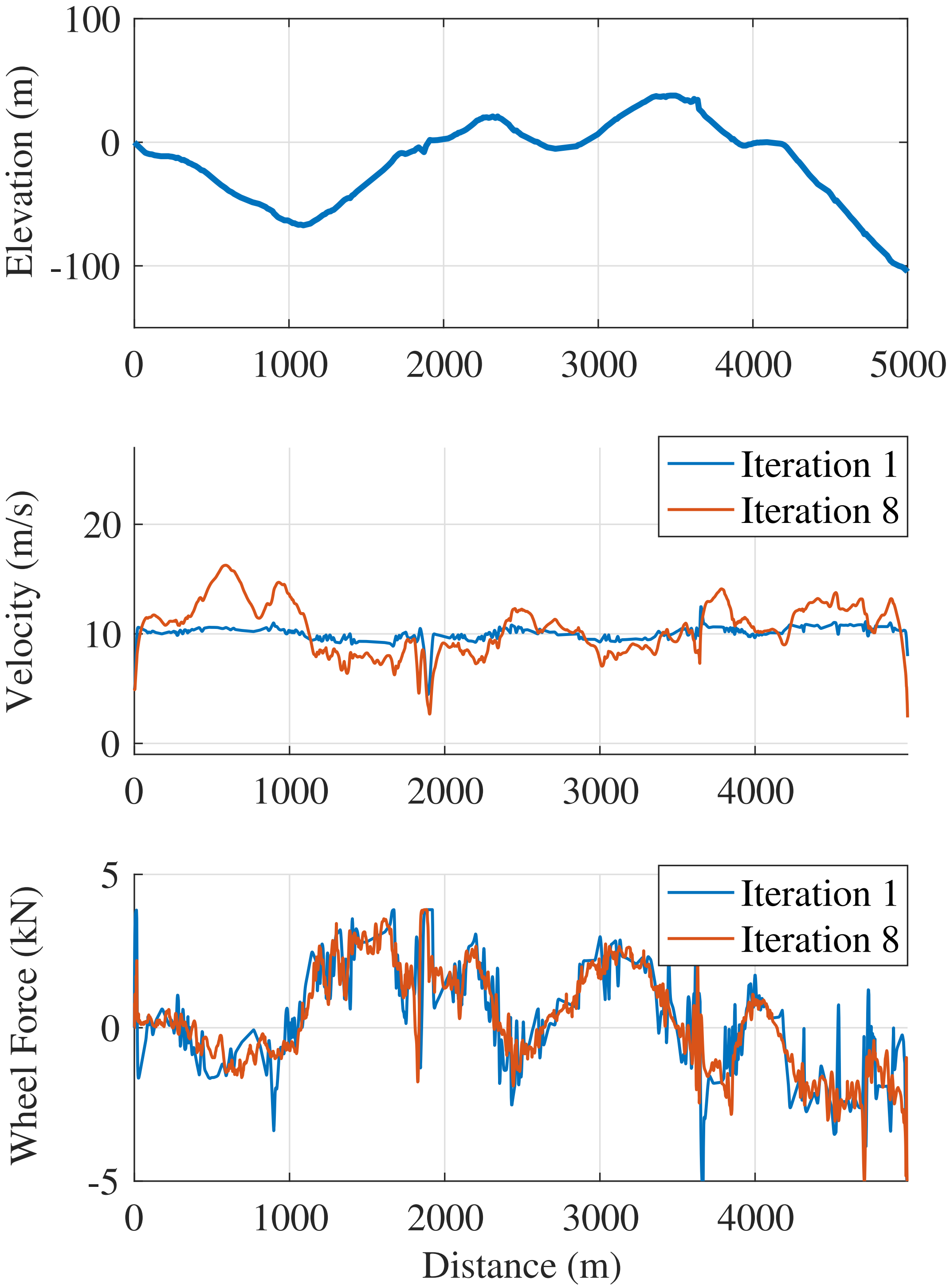


Figure 5.11: Plots of slope (top), closed loop velocity trajectories (middle), and closed loop wheel force trajectories (bottom)

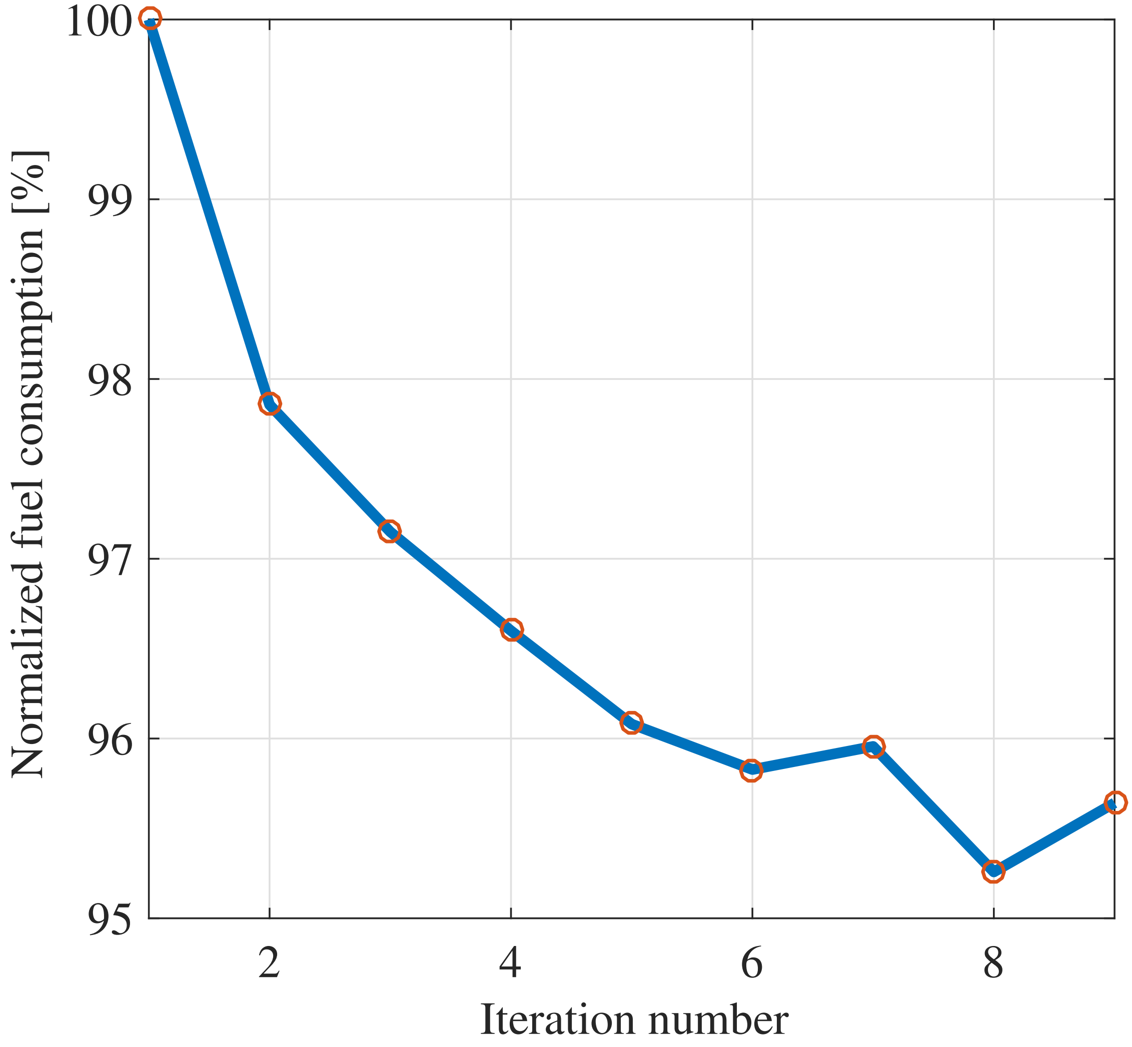


Figure 5.12: Plot of fuel consumption per iteration

# Chapter 6

# Shared Perception for Connected and Automated Vehicles

## 6.1 Introduction

In the previous chapters, we focused on the motion control of connected and automated vehicles (CAVs). In this chapter, we present another application of the CAV technology, *shared perception*. Perception systems are essential in the development and operation of an autonomous vehicle (AV) and Advanced Driver Assistance System (ADAS). AV/ADAS perception systems generally utilize a set of on-board sensors such as camera, radar, and lidar. In recent years, there have been growing efforts from both the research community and the industrial sector to advance on-board sensor technologies and perception algorithms [94]. However, perception systems that rely solely on on-board sensors are inherently limited by the detection range and field of view of their components. These limitations increase uncertainty in the motion planning and control algorithms, which in turn sacrifice performance (passenger comfort, road throughput, energy efficiency) in order to guarantee safety. Communication and connectivity in CAVs can mitigate these shortcomings by enabling real-time control and planning with increased awareness, routing with micro-scale traffic information, coordinated platooning using traffic signals information, and eco-mobility on demand with guaranteed parking enabling rapid cooperative movements [16].

Connected Vehicle (CV) technology is well-established in the automotive community. Dedicated Short Range Communication (DSRC) established highly secure, high-speed direct communication between vehicles and nearby infrastructure [37]. This enables a Vehicular Ad-hoc Network (VANET) which provides Vehicle-to-vehicle (V2V) communication through on-board units (OBU) and vehicle-to-infrastructure (V2I) communication through road-side units (RSU) [95]. DSRC was developed with a primary focus on safety applications, and is based on relatively small packets that are transmitted with low latency; bandwidth limitations would emerge when trying to exchange larger amounts of data. Recently, the fifth generation mobile technology standard (5G) has been proposed as an alternative technology for V2V communication, which would also allow to exchange larger quantities of data [96].

V2V communication can improve vehicle safety by enabling all CVs to share their location and the targets detected by their perception systems. We refer to this concept, which to our best knowledge was first investigated in [97, 14], as *shared perception.* A shared perception system presents each CV with an augmented version of its on board perception system, in which the size of the occluded regions is reduced and the detection range and field of view are increased. Different approaches to shared perception through V2V communication have been explored in the literature. In [98], CVs share and merge their occupancy grid maps using a so-called coordination transformation method. However, the performance of this map merging method is limited at scale, as the large amount of shared data increases the communication delay uncertainty. In [99], the authors propose an object-based approach to shared perception, where the raw data of each detected obstacle is first processed and filtered locally, and subsequently transmitted to the nearby vehicles. The processed obstacle data is abstracted as an *object* with states such as location, velocity, and heading angle measurements and the corresponding covariance matrices, reference frame, etc. This object-based shared perception approach provides main two benefits. First, it can reduce the communication burden by avoiding transmitting raw sensor data. Second, it improves system modularity, i.e. existing on-board sensors can be replaced or new hardware can be added without restrictions on specific hardware, as long as its raw perception data can be represented in a way that fully defines an *object* class defined above. For these reasons, the shared perception system proposed in this paper also implements an object-based shared perception concept.

Other implementations of object-based shared perception are also explored in literature. In [99, 100, 101], inter-vehicle object association is presented by applying point matching algorithms after taking care of temporal and spatial alignments of CVs and their perceived and shared objects. In [102], driver warnings were demonstrated to alert drivers of potentially dangerous situations using objects perceived by CVs. To the best of the authors' knowledge, the existing literature does not include any demonstration of vehicle motion control based on shared perception with exception of the work in [103] where wireless communications are used only for high level coordination and an on-board lidar and GPS are used for perception and localization.

This chapter proposes an object-based shared perception system in order to reduce occluded regions on the road and improve safety. This is accomplished by expanding the standard DSRC message set ([37]) with custom V2V messages and by a set of algorithms for object fusion. Finally, we experimentally demonstrate a longitudinal motion controller for CAVs that leverages the shared perception system to safely adjust the vehicle speed even before the undetected vehicle suddenly appears in direct line-of-sight.

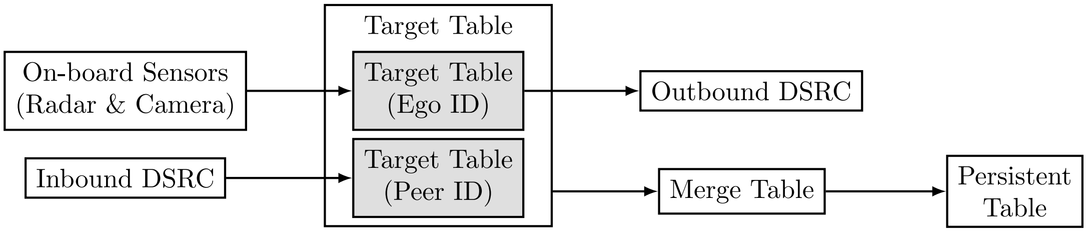


Figure 6.1: The shared perception system structure of the ego vehicle

## 6.2 Shared Perception Structure and Algorithms

In this section we introduce the proposed object-based shared perception system. We utilize on-board camera and radar sensors and communicate object-level information about their detected targets through V2V OBUs. Clearly any set of available sensors can be used in our approach.

### 6.2.1 CAV System Structure

Our CAV uses an on-board perception and control architecture shown in Figure 2.2. The CAV perception system receives real-time data from GPS, DSRC, radar, and camera devices and reconstructs the driving situation (e.g. position and speed of surrounding/communicating vehicles, road geometry, traffic lights, etc). This information is then used by the Real-Time Control and Planning system (upper-left block) to compute a safe vehicle trajectory, and the corresponding control inputs are requested to the vehicle actuators through the powertrain and steering interfaces.

### 6.2.2 Shared Perception System Structure and Algorithms

Figure 6.1 illustrates the shared perception structure of the ego vehicle, which refers to the vehicle being controlled and communicating with the other vehicles, next denoted as "peer vehicles". As illustrated in the figure, perception data from on-board sensors and V2V data from DSRC undergo a set of filtering and object-association steps to construct a set of tables, namely *target table*, *merge table*, and finally *persistent table*. Targets in "Persistent Table" are interpreted as obstacles in motion control of our CAV. A similar approach is proposed in [101].

Next we explain the steps and algorithms that constitute our proposed shared perception system. For simplicity, we focus on the simplest case with only two communicating vehicles (the ego and the peer), although the approach generalizes to cases with more communicating vehicles.

First, assume a set of targets is locally perceived by the on-board radar and camera in the ego vehicle. Each target, denoted as $z$, carries information about its validity (a flag to indicate whether the target is persistently detected and has reasonable measurements), relative position, and relative velocity; Note that this set of information is directly available via communication with the on-board radar and camera modules. We denote the list of radar targets as $\mathbf{z_r}$ and the list of camera targets as $\mathbf{z_c}$. Algorithm 2 details the steps to construct the Target Table for the ego vehicle, denoted as $\mathcal{T}^{\text{ego}}_{\text{target}}$, using $\mathbf{z_r}$, $\mathbf{z_c}$, GPS information of the two vehicles, and the V2V message such as the target table from the peer ($\mathcal{T}^{\text{peerID}}_{\text{target}}$). At each iteration with time intervals of duration $t_s$, each target from $\mathbf{z_r}$ and $\mathbf{z_c}$, if it is valid, is associated with the ego ID and sorted in an ascending order of relative longitudinal distance ($\Delta y$) from the ego vehicle. Finally, targets with ego ID in $\mathcal{T}^{\text{ego}}_{\text{target}}$ are transmitted to other communicating vehicles along with the GPS information of the ego vehicle.

Upon receiving its target table information from the peer vehicle, denoted as $\mathcal{T}^{\text{peerID}}_{\text{target}}$, each target in $\mathcal{T}^{\text{peerID}}_{\text{target}}$ is transformed into the ego vehicle's reference frame using the ego vehicle's GPS and the peer vehicle's GPS (see [99] for example). This step is denoted as changeCoordinate() in Algorithm 2 and can be improved by additionally using local radar data. Finally, the transformed targets are appended to $\mathcal{T}^{\text{peerID}}_{\text{target}}$.

**Algorithm 2** Target Table Update for ego Vehicle

```
**Input** $lat_{\text{ego}}$, $lon_{\text{ego}}$, $v_{\text{ego}}$, $\theta_{\text{ego}}$, $lat_{\text{peer}}$, $lon_{\text{peer}}$, $v_{\text{peer}}$, $\theta_{\text{peer}}$, $\mathbf{z_R}$, $\mathbf{z_C}$, $\mathcal{T}^{\text{peerID}}_{\text{target}}$
**Output** $\mathcal{T}^{\text{ego}}_{\text{target}}$
**for** each iteration, **do**
    **for** every target $z$ in $\mathbf{z_R}$ and $\mathbf{z_C}$, **do**
        **if** $z$ is valid,
            $z$.ego = egoID
            append $z$ to $\mathcal{T}^{\text{ego}}_{\text{target}}$
    sort($\mathcal{T}^{\text{ego}}_{\text{target}}$) according to $\Delta y$
    $\mathcal{T}^{\text{egoID}}_{\text{target}}$ = [ $q$ in $\mathcal{T}^{\text{ego}}_{\text{target}}$ if $q$.ego = egoID]
    Transmit $\mathcal{T}^{\text{egoID}}_{\text{target}}$ with $lat_{\text{ego}}$, $lon_{\text{ego}}$, $v_{\text{ego}}$, $\theta_{\text{ego}}$

    **for** every target $q$ in $\mathcal{T}^{\text{peerID}}_{\text{target}}$, **do**
        changeCoordinates($q, lat_{\text{ego}}, lon_{\text{ego}}, v_{\text{ego}}, \theta_{\text{ego}}, lat_{\text{peer}}, lon_{\text{peer}}, v_{\text{peer}}, \theta_{\text{peer}}$)
        append $q$ to $\mathcal{T}^{\text{ego}}_{\text{target}}$
```

The next step is to construct the Merge table and the Persistent table, denoted as $\mathcal{T}_{\text{merge}}$ and $\mathcal{T}_{\text{persist}}$, respectively. Algorithm 3 details the steps to construct them using $\mathcal{T}^{\text{ego}}_{\text{target}}$. At each time stamp, we obtain $\mathcal{T}_{\text{merge}}$ by scanning all the targets and removing all (objects that are identified as) duplicates in $\mathcal{T}^{\text{ego}}_{\text{target}}$. This step of finding and removing duplicate targets is denoted as isNeighbor() in the algorithm. In this work we simply use a distance-based method to find the target duplicates (i.e. whether the two targets are within a close proximity, $d_c$) and remove them. Existing literature is full of alternative and more sophisticated object association methods [104, 101, 100, 105] which can be applied to our framework as well.

Finally, we update $\mathcal{T}_{\text{persist}}$ by propagating the target states using a Kalman filter with a 2-D space point-mass model assuming constant velocity and using new measurements in the updated $\mathcal{T}_{\text{merge}}$. In this step, we remove targets that are older than (i.e, no new measurements for more than) $n_a$ time steps. Note that the parameters in our proposed algorithms such as $n_a$, kalman gains, and $d_c$ are tuning parameters which can differ from each vehicle due to its different sensor specifications.

**Algorithm 3** Merge Table and Persistent Table Updates

```
Input $\mathcal{T}_{\text{target}}^{\text{ego}}$
Output $\mathcal{T}_{\text{merge}}$, $\mathcal{T}_{\text{persist}}$
initialize $\mathcal{T}_{\text{persist}} = []$
for each iteration, do
    initialize $\mathcal{T}_{\text{merge}} =[]$
    for every target $z$ in $\mathcal{T}_{\text{target}}^{\text{ego}}$, do
        if isNeighbor($z$, Ego), remove $z$ from $\mathcal{T}_{\text{target}}^{\text{ego}}$
        else if isNeighbor($z$, Peer), remove $z$ from $\mathcal{T}_{\text{target}}^{\text{ego}}$
        else append $z$ to $\mathcal{T}_{\text{merge}}$
        for every target $q$ in $\mathcal{T}_{\text{target}}^{\text{ego}}$, do
            if isNeighbor($z$, $q$), remove $q$ from $\mathcal{T}_{\text{target}}^{\text{ego}}$
    for $p$ in $\mathcal{T}_{\text{persist}}$, do
        for $m$ in $\mathcal{T}_{\text{merge}}$, do
            if isNeighbor($m$, $p$),
                kalmanUpdate($p$,$m$)
                $p.age = 0$
                remove $m$ from $\mathcal{T}_{\text{merge}}$
                break
        if $p.age > n_a$, remove $p$ from $\mathcal{T}_{\text{persist}}$
        constantVelocityUpdate($p$)
        $p.age + +$
    for $m$ in $\mathcal{T}_{\text{merge}}$, do
        $m.age = 0$
        append $m$ to $\mathcal{T}_{\text{persist}}$
```

## 6.3 Experimental Demonstration with Longitudinal Motion Controller

In this section we demonstrate the proposed perception system with a closed loop longitudinal motion controller. We will consider an intersection with large areas of occlusion to demonstrate the advantage of our shared perception system when the presence of obstacles is uncertain, as illustrated in Figure 6.2. The ego vehicle, of which the longitudinal motion is controlled, communicates with the peer vehicle. The non-communicating vehicle (blue) is entering the intersection and cutting in while occluded to the ego vehicle but seen by the peer vehicle.

Sensor and communication setups for the ego and the peer vehicles are the same as explained in Section 2.3. Moreover, to execute the shared perception algorithms and the longitudinal motion control in real-time, the ego vehicle is also equipped with a Matrix embedded PC-Adlink (MXC-6101D/M4G with Intel Core i7-620LE 2.0 GHz processor) with Robot Operating System (ROS) and dSpace MicroAutoBox (IBM PowerPC 750FX processor, 800 MHz).

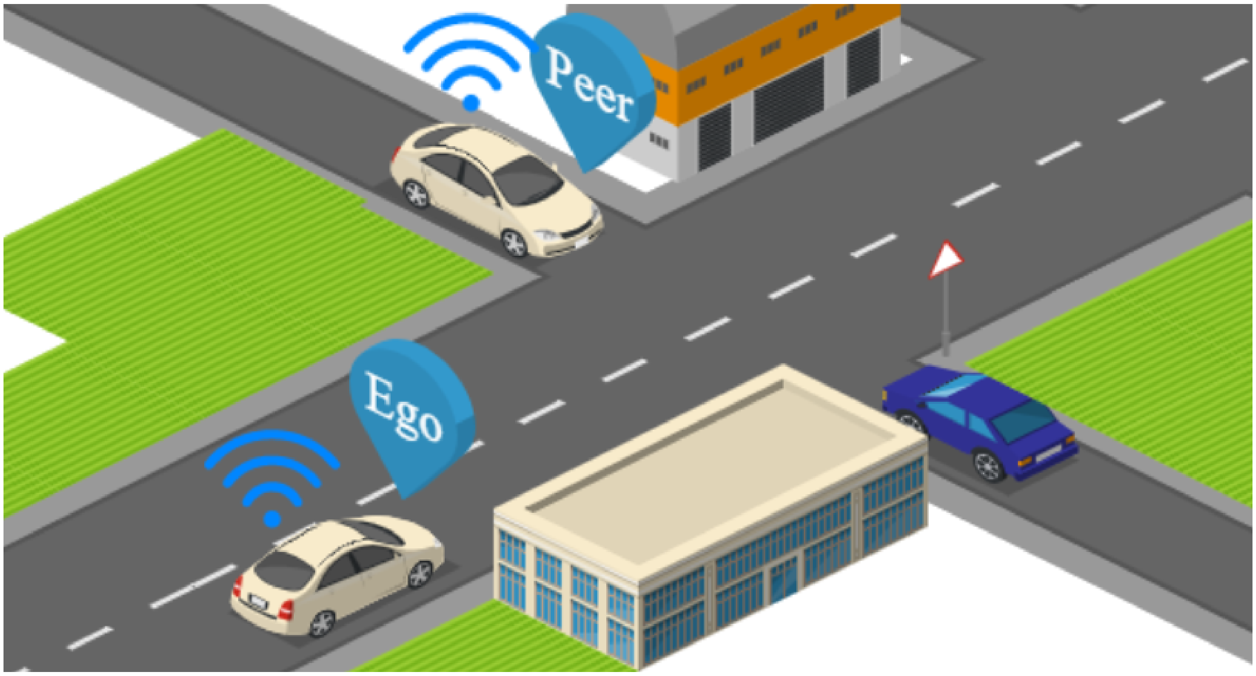


Figure 6.2: Illustration of the Experimental Setup

### 6.3.1 Longitudinal Motion Controller Design

As shown in Figure 2.2, our CAV executes the longitudinal motion control by exploiting real-time perception information from the DSRC, as well as the on-board sensors such as radar and camera. Our longitudinal motion controller, which commands the reference wheel torque, is based on the model predictive control based adaptive cruise control design proposed in Chapter 4.2. The only difference is that the distance to the front vehicle is obtained by taking the minimum distance of the targets that are in the Persistent table and entering the ego vehicle's path. To determine whether a target will enter the ego vehicle's path, we check if the following conditions are met.

$$[x_{\text{ego}}, x_{\text{ego}} + N_p v_{x,\text{ego}}] \cap [x_{\text{target}}, x_{\text{target}} + N_p v_{x,\text{target}}], \tag{6.1a}$$

$$[y_{\text{ego}}, y_{\text{ego}} + N_p v_{y,\text{ego}}] \cap [y_{\text{target}}, y_{\text{target}} + N_p v_{y,\text{target}}] \tag{6.1b}$$

where $N_p$ is the planning horizon time.

## 6.4 Results and Discussion

Figure 6.3 shows the trajectories of the ego vehicle in terms of the relative distance to the vehicle on the ego vehicle's path, velocity, and reference torque input. Note that our experiment was conducted at a low speed (less than 5 $[m/s]$). In this scenario, our share perception system can feed the relative distance to the front vehicle about 3.5 seconds earlier than when the sensor fusion of the on-board sensors (camera and radar). This allows the longitudinal motion controller to react to the possible vehicle cut-in early enough to maintain a safe distance. It is also noted that our shared perception momentarily loses a target at time instant 5 causing aggressive torque input. Designing more robust objection association (isNeighbor()) rather than a simple distance-based method in our shared perception algorithms to avoid this kind of situation remains to be our future work.

Video demonstration for various applications of the proposed shared perception system as well as our experiment can be found at [29]. Moreover, our shared perception system allowed us to safely execute a compact platoon demonstration on public urban road [31].

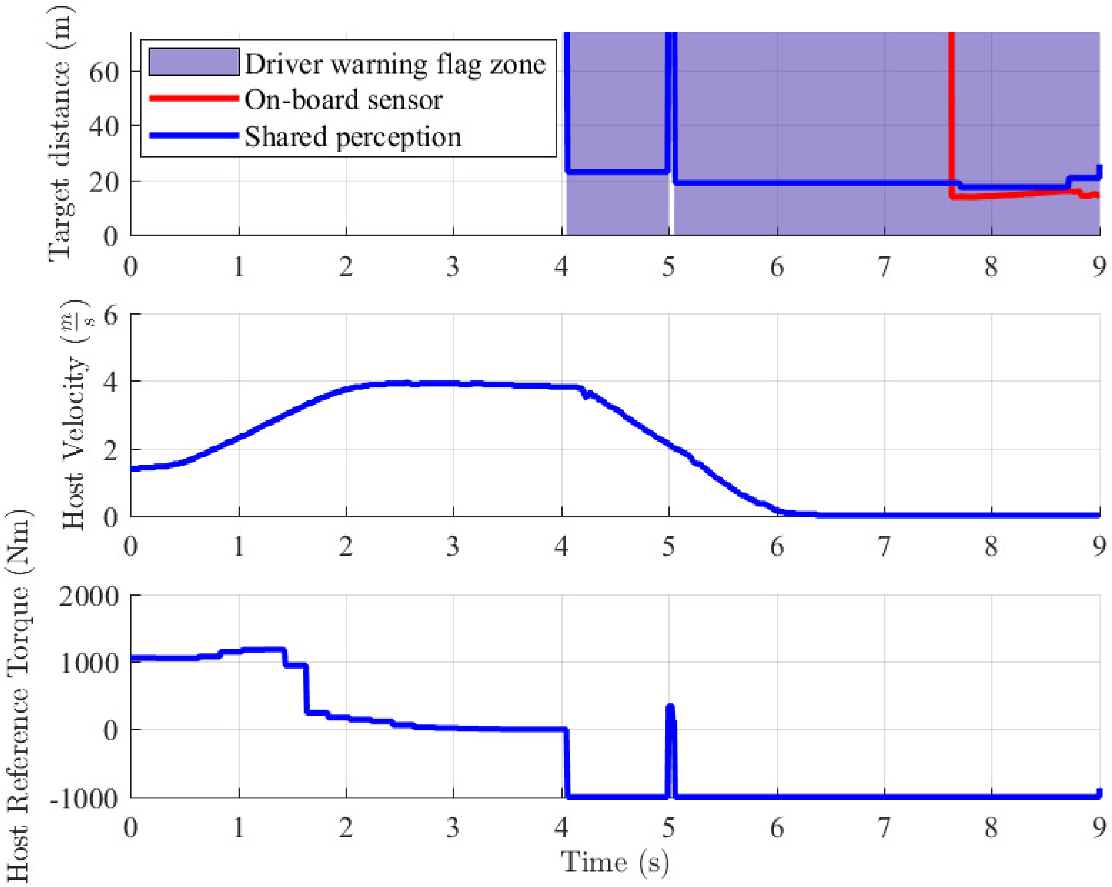


Figure 6.3: Plots of relative distance, velocity, and reference torque.

# Chapter 7

# Connected and Automated Railway Control

## 7.1 Introduction

In the previous chapters, we studied application of communication technology for the automotive vehicles. This chapter studies how the similar communication technology can benefit the railway (train) control system, particularly to increase the capacity of the railway lines.

*Virtual coupling* is an advanced approach to increase the line capacity and satisfy the increasing rail transport demand in train control systems. In the concept of virtual coupling, trains are virtually-coupled via train-to-train communication, sharing information with neighbors and receiving the reference signals coming from the infrastructure. On the basis of this information received from trains within the convoy, the on-board system is responsible for the safe tracking of speed profiles, while respecting the inter-train spacing policy, allowing the follower to pursue the leader train in a safe way and guaranteeing good transient dynamics. The virtual coupling concept presents several advantages. First, it increases the line capacity. In addition, in virtual coupling, trains can be coupled and decoupled dynamically, according to the service needs and respecting the safety requirements

Virtual train operation initially appeared in [106, 107], which describe the methods to design and develop the concept for railway operation with a platoon of trains which are linked by a wireless informational interconnection instead of a conventional mechanical connection. However, they do not provide many implementation details. A new impulse to this concept arose with the European research initiative Shift2Rail [108, 109]. It was primarily motivated by the need of increasing the flexibility, interoperability and capacity of the passenger lines, and by the achievements in the communications technology field. [110] analyzes the feasibility of virtual coupling implementation which will be developed within the Shift2Rail initiative. In summary, the state of the art shows that the concept of virtual coupling has been extensively studied and the elements that would be technically necessary have been identified. However, no theoretical or simulation studies have been carried out that analyze and design the more suitable control system for the virtual coupling concept. This chapter tries to cover this, carrying out a study on the control system to be used for virtual coupling.

The goal of this chapter is to propose an innovative solution in the field of railways for train virtual coupling following the ideas coming from the developments in the platooning of autonomous vehicles, based on the Model Predictive Control (MPC) framework.

## 7.2 Train Model

We use the discretized longitudinal train dynamics (LTD) which considers the train as a point mass with 1 degree of freedom, the driving/braking system, the rolling and bearing resistances, the air input, the aerodynamic drag, and the grade and curving resistances. The dynamic equation are:

$$\begin{aligned} x(k+1|t) &= f(x(k),\, u(k)) \\ &= x(k) + t_s \begin{bmatrix} v(k) \\ \frac{-1}{M}(A + Bv(k) + T_f C v(k)^2 + F_e(s(k)) + \frac{u(k)}{M} \\ \frac{u(k)-F(k)}{\tau} \end{bmatrix} \end{aligned} \tag{7.1}$$

where $x = [s,\, v,\, F]^\top$; $s$, $v$, and $F$ denote the train position, velocity, and driving/braking force; $u$ is the desired driving or braking force; $F_e$ is the external force due to the track; $\tau$ is the intertial lag of longitudinal dynamics; $M$, $A$, $B$, and $C$ are parameters of the train characteristics where $M$ is the mass; $A$ is the rolling resistance plus the bearing resistance; $B$ is a coefficient related with the air input; $C$ is the aerodynamic coefficient; $T_f$ is the tunnel factor.

The external force $F_e(s)$ includes two different components, $F_g(s)$ and $F_R(s)$. They are expressed as:

$$F_e(s) := F_g(s) + F_R(s), \tag{7.2a}$$

$$F_g(s) := -M * g * slope(s), \tag{7.2b}$$

$$F_R(s) := -M * 6 * 10^6 * \frac{1}{R(s)}, \tag{7.2c}$$

where $F_g$ is the gravity force; *slope* is the track grade: $g$ is the gravitational acceleration; $F_R$ is the curving resistance; $R$ is the radius of the curve. *slope* and $R$ are dependent on the line profile and the train location on the line. Figure 7.1 depicts the value of *slope* and $R$ used in the simulation in this chapter; the values are obtained from [111].

The input $u$ is subject to the following constraints:

$$-Ma_{br} \leq u \leq Ma_{dr}, \tag{7.3a}$$

$$-P_{br} \leq uv \leq P_{dr}, \tag{7.3b}$$

where $P_{dr}$ and $P_{br}$ are the maximum driving power and the maximum braking power, respectively; $a_{dr}$ is the maximum driving acceleration; $a_{br}$ is the maximum braking deceleration.

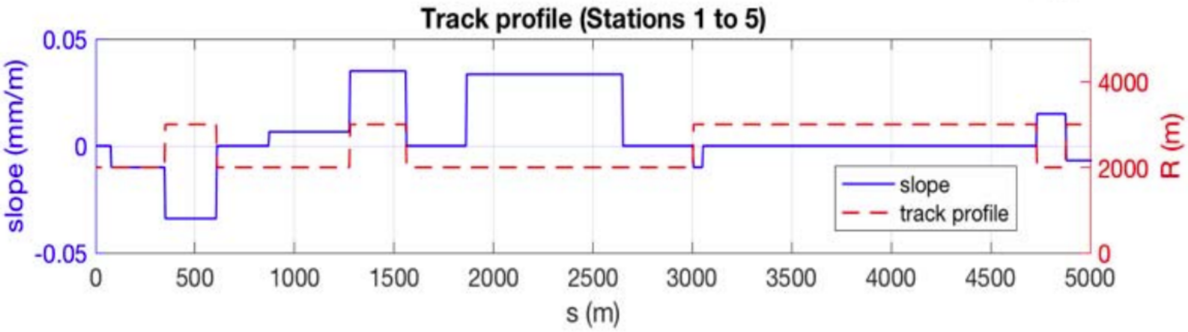


Figure 7.1: Slope and curve radius considered in the simulations for the line, showing the first five stations

# 7.3 Virtual Coupling Model Predictive Control

## 7.3.1 Problem Definition

A convoy formed by $n$ trains, as shown in Figure 7.2 is considered. The superscript denotes the train identity; e.g., $l$ denotes the leader vehicle and $f$ denotes the follower vehicle in the convoy. The distance between the two trains is then calculated by

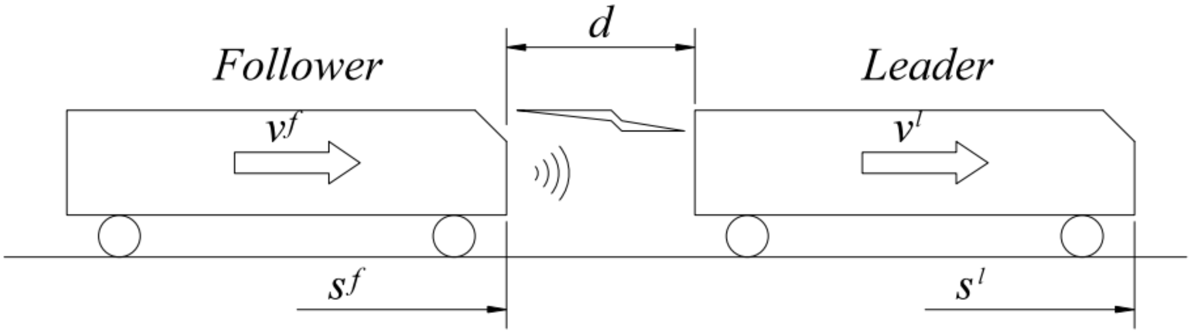


Figure 7.2: A train convoy

$$d = s^l - s^f - L \tag{7.4}$$

where $L$ is the length of each train. We also assume the presence of a communication link between both trains, which allow for the instant exchange of train state including position, velocity and acceleration.

The goal of the virtual coupling controller is to move the train convoy as fast as possible while maintaining a safer inter-train distance $d_{\text{des}}$, greater than the minimum safe distance $d_{\min}$.

### 7.3.2 Control Architecture for the Virtual Coupling

We consider a decentralized virtual coupling control with two different controllers, one for the leader and the other for the followers. The control objective of the leader is to move the train as fast as possible while satisfying the state and input constraints. The control objective of the leader is to maintain a desired distance $d_{\text{des}}$ while satisfying the minimum distance $d_{\text{min}}$ as well as the state and input constraints.

In the case of a multiple trains convoy, the block corresponding to the follower will be repeated for each following train.

The control architecture for the virtual coupling control is similar to our Connected and Automated Vehicle system architecture as depicted in Figure 2.2. In a upper-level such as the motion Eco-Driving planning in the Remote planning and Routing module, we use a Dynamic Programming (DP) approach to pre-compute an optimal speed profile that finds the maximum velocity permitted by the speed limitations imposed by the line operation, i.e. satisfying the speed constraint at all time In a lower-level such as the Motion control in the Real-Time Control and Planning module, we use a MPC design to safely track the optimal velocity reference.

### 7.3.3 Dynamic Programming for the Velocity Reference Generation

We pre-compute optimal speed profiles for both the leader and the following using DP with the goal of traveling at the fastest allowed speed while satisfying speed constraints at any time. The space-dependent velocity limit denoted as $v_{\text{lim}}(s)$ is depicted in Figure 7.3 for the first five stations. The different stations are represented in the horizontal axis. $v_{\text{lim}}(s)$ is used as a constraint by the DP to guarantee that the train obeys the speed limit at all times.

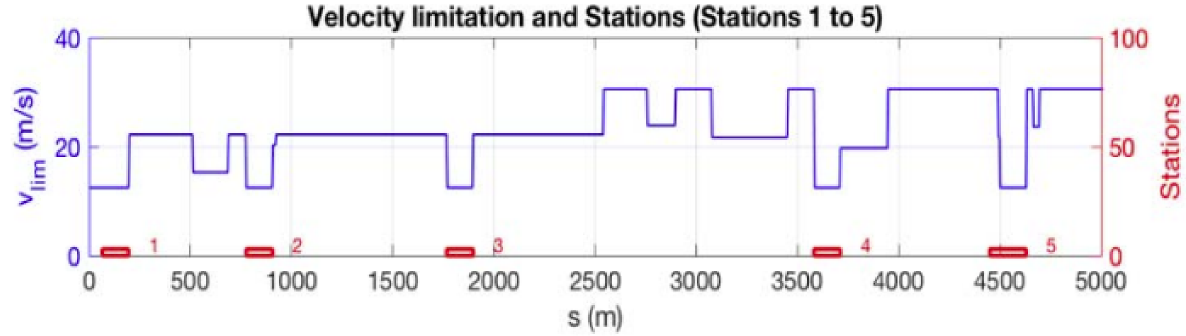


Figure 7.3: Velocity limitation $v_{\text{lim}}(s)$ for the first five stations

Because $v_{\text{lim}}(s)$ is space-dependent (i.e. dependent position on the track), a parametrization in space is used to make the position of the train $s$ an independent variable. The track is sampled with $\Delta s$ and $s(i)$ denotes the position of the train at space step $i$; i.e., $S = [s(0), \cdots, s(N_s)]$ is the discretized track and $N_s$ is the number of discretization points.

For each train, the optimization problem can be formulated as:

$$J^*_{i \to N_s}(v(i)) = \min_{u(i)} q(v^{(}i), u(i)) + J^*_{i+1 \to N_s}(v(i+1)) \tag{7.5a}$$

subject to

$$v(i+1) = \sqrt{v(i)^2 + 2\Delta s(-A - Bv(i) - Cv(i)^2 - F_e(i) - u(i)}, \tag{7.5b}$$

$$-Ma_{br} \leq u(i) \leq Ma_{dr}, \tag{7.5c}$$

$$-P_{br} \leq u(i)v(i) \leq P_{dr}, \tag{7.5d}$$

$$0 \leq v(i) \leq v_{\lim}(i\Delta s), \tag{7.5e}$$

$$\forall i \in [N_s - 1, \cdots, 1]$$

$$v(N_s) = v_f \tag{7.5f}$$

$$J^*_{N_s \to N_s}(v(N_s)) = \begin{cases} 0 & \text{if } v(N_s) \leq v_{\lim}(s(N_s)), \\ \infty & \text{otherwise}, \end{cases} \tag{7.5g}$$

where the stage cost $q(v(i), u(i))$ is defined as:

$$q(v(i), u(i)) = \|v(i) - v_{\lim}(s(i))\|_{K_{V_{DP}}} \tag{7.6}$$

where $K_{V_{DP}}$ represents the weight penalizing the output deviation from the maximum allowed speed; (7.5b) is the space discretization model of the train dyanmics (7.1) using the trapezoidal rule; (7.5f) is the terminal constraint; (7.5g) represents the initial condition for the cost function (7.5a). The resulting velocity trajectory from the solution of (7.5) is denoted as $v^*_{DP}$. Figure 7.4 depicts an instance of the speed profile $v^*_{DP}$. Finally, $v^*_{DP}$, which is computed offline for the leader and the follower, is used in the following motion control designs to ensure that the trains in the convoy obey the speed limits at all times while moving as fast as possible.

### 7.3.4 Model Predictive Control for Real-Time Train Motion Control

A Model Predictive Control (MPC) approach, similar to our longitudinal motion control for vehicle (4.6)-(4.8), is proposed for the real time control of the trains.

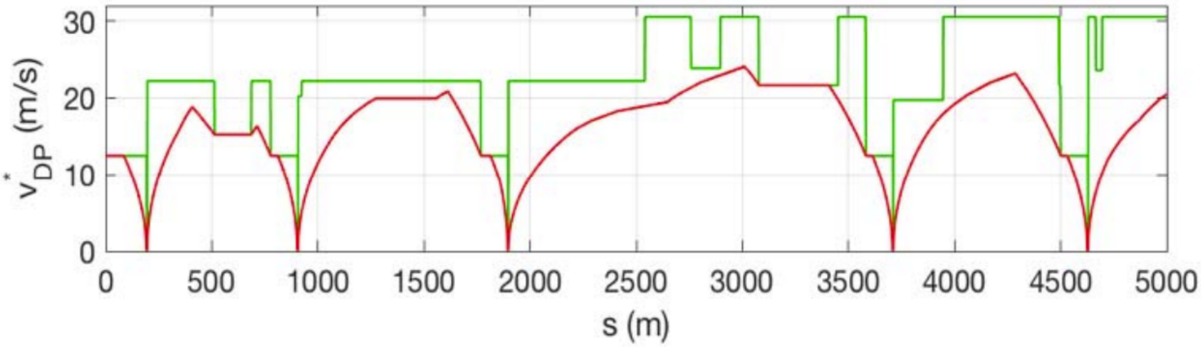


Figure 7.4: Speed profile $v^*_{DP}$ for the leader and follower during the first 5 stations. Red line represents $v^*_{DP}$, and the green line represents the speed limit.

### Motion control for the leader

At time $t$, the motion controller for the leader solves the following optimization problem:

$$\min_{u^l(\cdot|t)} J = \sum_{k=t}^{t+N_p} \left\| 1 - \frac{v^l(k|t)}{v_{\max}} \right\|_{K_V^l} + \sum_{k=t}^{t+N_p-1} \left\| \frac{j^l(k|t)}{j_{\max}} \right\|_{K_J^l} \tag{7.7a}$$

subject to

$$x^l(k+1|t) = f(x^l(k|t), u^l(k|t)), \tag{7.7b}$$

$$-Ma_{br} \leq u^l(k|t) \leq Ma_{dr}, \tag{7.7c}$$

$$-P_{br} \leq u^l(k|t)v^l(k|t) \leq P_{dr}, \tag{7.7d}$$

$$-j_{\max} \leq j^l(k|t) \leq j_{\max}, \tag{7.7e}$$

$$0 \leq v^l(k|t) \leq v_{\lim}(s^l(k|t)), \tag{7.7f}$$

$$\forall k \in [t, \cdots, t+N_p-1]$$

$$x^l(t|t) = \bar{x}^l(t) \tag{7.7g}$$

$$0 \leq v^l(t+N_p|t) \leq v_{DP}(s^l(t+N_p|t)), \tag{7.7h}$$

where

$$j^l(k|t) = \frac{(F^l(k+1|t) - F^l(k|t))M}{\Delta t}; \tag{7.8}$$

$K_V^l$ and $K_J^l$ denote the weight penalizing the output deviation from the maximum allowed speed and the weight penalizing the input jerk, respectively; $\bar{x}^l(t)$ in (7.7g) is the perfect state measurement at time $t$; (7.7h) is the terminal constraint which tries to make that the train must obey the speed limit for all times even after the MPC horizon $t + N_p$.

The first input of the solution of (7.7), denoted as $u^{l,*}(t|t)$, is applied to the system (7.1) during the time interval $[t, t+1)$.

At the next time step $t+1$, a new optimal problem in the form (7.7), based on new measurement of the state, is solved over a shifted horizon, yielding a moving receding horizon control strategy with control law,

$$u^l(t) = u^{l,*}(t|t). \tag{7.9}$$

**Remark 7.** (7.7b), (7.7f), *and* (7.7h) *are space-dependent constraints. In order to prevent numerical issues and lessen the computational load, they are a-priori estimated by employing the shifted solution of* (7.7) *from the previous time step* $t-1$*.*

### Motion control for the follower

At time $t$, the motion controller for the follower solves the following optimization problem:

$$\min_{u^f(\cdot|t)} J = \sum_{k=t}^{t+N_p} \left\| 1 - \frac{d^f(k|t)}{d_{\text{des}}} \right\|_{K_D} + \sum_{k=t}^{t+N_p} \left\| \frac{(\tilde{v}^l(k|t) - v^f(k|t))}{v_{\max}} \right\|_{K_V^f} + \sum_{k=t}^{t+N_p-1} \left\| \frac{j^f(k|t)}{j_{\max}} \right\|_{K_J^f} \tag{7.10a}$$

subject to

$$x^f(k+1|t) = f(x^f(k|t), u^f(k|t)), \tag{7.10b}$$

$$-Ma_{br} \leq u^f(k|t) \leq Ma_{dr}, \tag{7.10c}$$

$$-P_{br} \leq u^f(k|t)v^f(k|t) \leq P_{dr}, \tag{7.10d}$$

$$-j_{\max} \leq j^f(k|t) \leq j_{\max}, \tag{7.10e}$$

$$0 \leq v^f(k|t) \leq v_{\lim}(s^f(k|t)), \tag{7.10f}$$

$$\forall k \in [t, \cdots, t+N_p-1]$$

$$x^f(t|t) = \bar{x}^f(t) \tag{7.10g}$$

$$0 \leq v^l(t+N_p|t) \leq v_{DP}(s^l(t+N_p|t)), \tag{7.10h}$$

$$d_{\min} \leq d^f(t+N_p|t), \tag{7.10i}$$

$$d_{\min} \leq d^f(t+N_p|t) + \frac{(\tilde{v}^l(t+N_p|t))^2}{2a^l} - \frac{(v^f(t+N_p|t))^2}{2a^f}, \tag{7.10j}$$

where

$$d(k|t) = \tilde{s}^l(k|t) - s^f(k|t) - L, \tag{7.11a}$$

$$j^f(k|t) = \frac{(F^f(k+1|t) - F^f(k|t))M}{\Delta t}; \tag{7.11b}$$

$K_D$, $K_V^f$ and $K_J^f$ in (7.10a) denote the weight penalizing the output deviation from the desired distance, the weight penalizing the deviation from the predicted preceding vehicle velocity $\tilde{v}^l$, and the weight penalizing the input jerk, respectively; $\widetilde{d}^l(k|t)$ and $\tilde{v}^l(k|t)$ are the predicted preceding vehicle distance and velocity at time step $k$ using the available information at time $t$; $\bar{x}^f(t)$ in (7.10g) is the perfect state measurement at time $t$; $a^f$ and $a^l$ are the maximum decelerations of the follower and the leader, respectively; (7.10h) - (7.10j) are the terminal constraints which try to make that the train must not only obey the speed limit but also maintain the safety distance with the preceding vehicle for all times even after the MPC horizon $t + N_p$. Readers are encouraged to see [86] for more details about this approach.

The first input of the solution of (7.10), denoted as $u^{f,*}(t|t)$, is applied to the system (7.1) during the time interval $[t, t+1)$.

At the next time step $t + 1$, a new optimal problem in the form (7.10), based on new measurement of the state, is solved over a shifted horizon, yielding a moving receding horizon control strategy with control law,

$$u^f(t) = u^{f,*}(t|t). \tag{7.12}$$

## 7.4 Numerical Analysis

### 7.4.1 Simulation Setup

A real metro line simulation is analyzed in this section. The railway line considered corresponds to the Line 1 of Metro of Ho Chi Minh in Vietnam [111]. The parameters for the trains are listed in Table 7.1.

We consider the train convoy formed by two trains. Readers are encouraged to see [25] for simulations with the train convoy with more-than-two trains. Also, the simulations only consider a single line without crossing between different lines and switches.

All the simulations start with an initial condition in which the leader is leaving the first station, and the follower is leaving its stopping point inside the station, placed at a distance $d_{\min}$ from the leader. Both trains are enforced to stop in each station for 30 seconds. Moreover, our simulation assumes ideal conditions where there is no uncertainty in train models, measurements, and line profile (slope and radius) and no time delay in communication.

### 7.4.2 Simulation Result

**Effects of the Preceding Train Predictor**

The first simulation analyzes the controller performances when the follower vehicle has different predictors about the preceding vehicle. Note that these predictors are used in the motion controller (7.10)-(7.12) to obtain $\tilde{s}^l$ and $\tilde{v}^l$.

Mainly there are three different controller strategies: the short horizon strategy, denoted as SH, the no prediction strategy, denoted as NP, and the moving block strategy, denoted as MB. In the SH strategy, the motion controller (7.10)-(7.12) assumes that the information of the preceding train can be predicted without any error in a short horizon $N_p$. The NP strategy corresponds, to a degraded situation where the follower loses the communication link; therefore, only the current velocity and the location of the preceding train are available via on-board sensors. In this case, for the rest of the prediction horizon $N_p$, the motion

| Parameter | Unit | Value |
|---|---|---|
| $M$ | $kg$ | $1.608e^5$ |
| $L$ | $m$ | 68 |
| $A$ | $N$ | 9155 |
| $B$ | $Ns/m$ | 633.6 |
| $C$ | $Ns^2/m^2$ | 46.84 |
| $\tau$ | - | 0.25 |
| $P_{dr}$ | $W$ | $1.20e^6$ |
| $P_{br}$ | $W$ | $2.20e^6$ |
| $a_{dr}$ | $m/s^2$ | 0.9 |
| $a_{br}$ | $m/s^2$ | 0.92 |
| $j_{\max}$ | $m/s^3$ | 0.98 |
| $a^l$ | $m/s^2$ | 1.25 |
| $a^f$ | $m/s^2$ | $0.5 - 1.0$ |

(a) Model Parameters

| Parameter | Unit | Value |
|---|---|---|
| $\Delta t$ | $s$ | 0.1 |
| $d_{\text{des}}$ | $m$ | 10 |
| $d_{\min}$ | $m$ | 4 |
| $N_p$ | - | 40 |
| $K_V^l$ | - | 100 |
| $K_J^l$ | - | 0.5 |
| $K_D$ | - | $\{100, 25, 0\}$ |
| $K_V^f$ | - | 100 |
| $K_J^f$ | - | 0.5 |
| $K_{V_{DP}}$ | - | 1 |
| $K_{L_{DP}}$ | - | $1e^5$ |
| $\Delta s$ | m | $0.25 - 5$ |

(b) Controller Parameters

Table 7.1: Parameters

controller (7.10)-(7.12) assumes that the preceding train applies the worst conditions with maximum service braking. Finally, in the MB strategy, the motion controller (7.10)-(7.12) assumes that the position for the preceding train remains as the current measured position and its velocity is zero during the prediction horizon $[t, t + N_p)$.

Figure 7.5 shows the differences of behavior in terms of velocity, distance, and force between the three control strategies with $a^f = 1\frac{m}{s^2}$. As seen, the SH maintains a much shorter distance between trains (more than six times) than the MB. As a result, the follower with the MB strategy arrives significantly late to the stations with respect to the leader, disregarding the idea of a train convoy because trains are not in the station at the same time. This does not appear for the follower with the SH strategy, where the follower arrives almost at the same time to the station as the leader, successfully maintaining the convoy formation at all times. For the NP strategy, results are between SH and MB, but much closer to SH than MB.

The next set of simulations corresponds also to the same trip assuming in this case that $a^f = 0.5\frac{m}{s^2}$. The same three strategies, SH, NH, and MB, have been simulated, and Figure 7.6 shows the obtained results. As seen, similar to the case with $a^f = 1\frac{m}{s^2}$, the best results are obtained for SH while MP is not able to maintain the convoy integrity in the stations.

### Effects of the Controller Parameters

The second set of simulations analyzes the effect of considering different settings for the followers controller for the SH strategy. Here, we fix $K_D = 100$ and $K_V^l = 100$, and change values of $K_V^f \in \{100, 25, 0\}$.

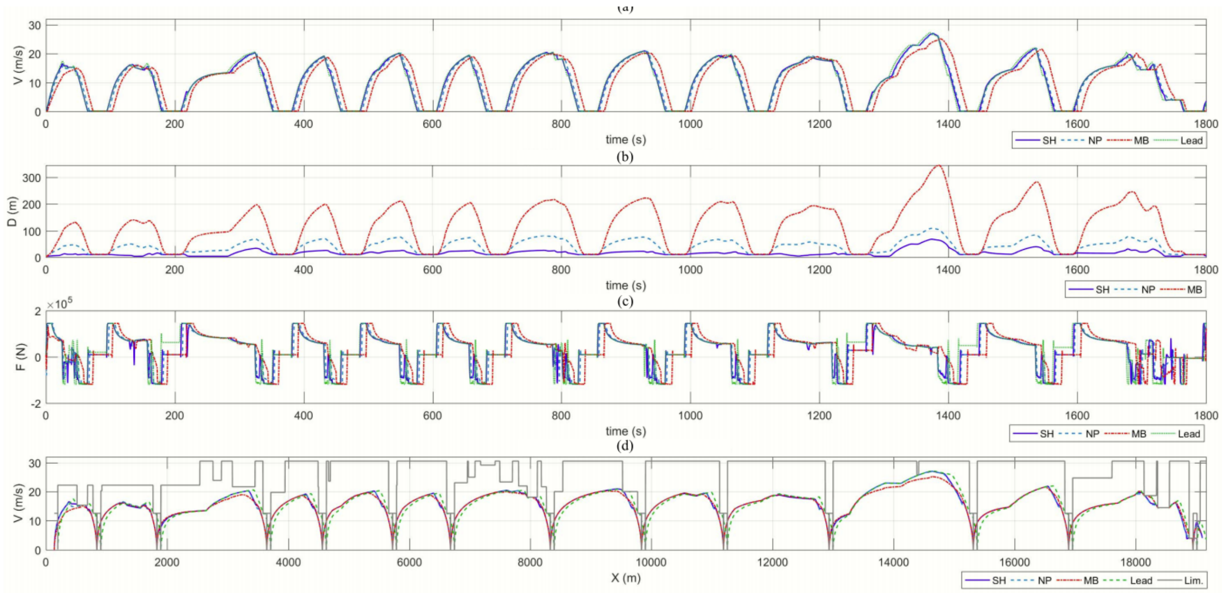


Figure 7.5: Comparison between the different control strategies, with $a^f = 1\frac{m}{s^2}$. (a) Velocity vs. time. (b) Distance between trains vs. time. (c) Force vs. time. (d) Velocity vs. position on the track.

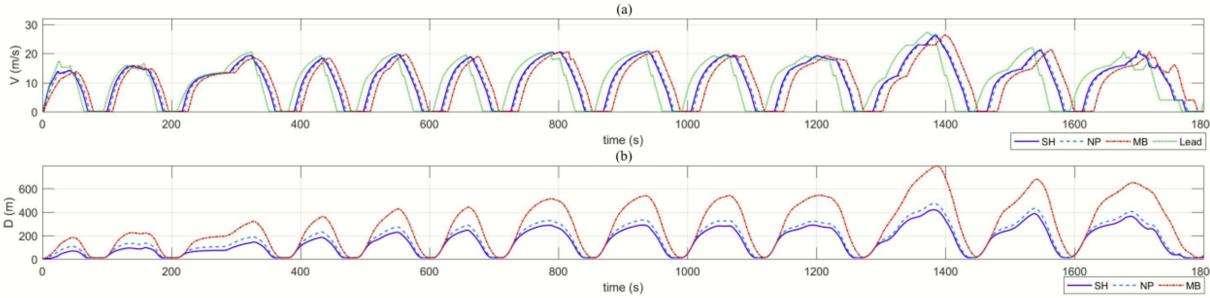


Figure 7.6: Comparison between the different control strategies, with $a^f = 0.5\frac{m}{s^2}$. (a) Velocity vs. time. (b) Distance between trains vs. time.

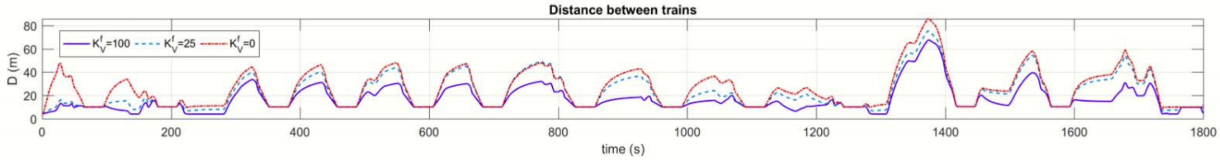


Figure 7.7: Distance vs. time with different controller parameters.

Figure 7.7 shows that the distance trajectories. As seen, the distance is higher at $K_V^f = 0$ than at $K_V^f = 100$. $K_V^f = 0$ allows the follower to only focuses on tracking the desired distance $d_{\text{des}}$ with respect to the leader whereas when $K_V^f = 100$ also causes the follower to track the maximum possible speed $v_{\max}$, reducing the distance between both trains. As a result, a more aggressive behavior is produced when $K_V^f = 100$. Figure 7.8 shows the detailed distance trajectories between stations 4 and 5 (a) and between stations 3 and 4 (b). As seen, at stations 4 and 5, where the track conditions do not vary (as seen in Figure 7.1), the distance is reduced with a higher value of $K_v^f$ while the desired distance $d_{\text{des}}$ of $10m$ is maintained in stations regardless of values of $K_v^f$. However, at stations 3 and 4, where the track conditions vary (as seen in Figure 7.1), the controller with a higher value of $K_v^f$ is more aggressive, transitorily making the distance less than $d_{\text{des}}$ of $10m$.

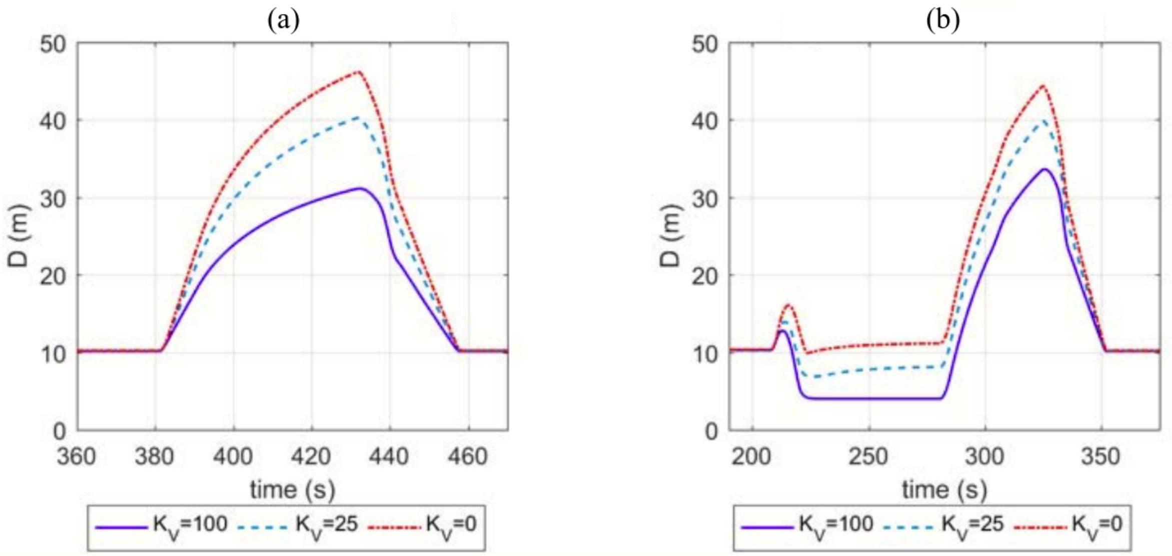


Figure 7.8: Distance vs. time with different controller parameters. (a) Distance: Stations 4-5. (b) Distance: Station 3-4.

# Appendix A

# Micro-Platoon Management

In order to safely manage a micro-platoon which consists of a small group of vehicles with V2V communication, we have designed a platoon management. Our platoon management is based on a Finite State machine (FSM) which acts as a mechanism for safely forming and maintaining a platoon.

The FSM diagram is depicted in Figure A.1 There are four primary states in our FSM: Ready, Plan Proposed, Plan Active, and Plan Cancel. Each platooning vehicle is initialized in the Ready state and communicates its state at all times. The platoon formation process is initiated when the leader moves to the Plan Proposed state by proposing to the follower vehicles the plan, including a plan ID and details about the coordinated maneuver such as desired speed/gap between vehicles in the platoon, destination, etc. As soon as the plan is received by the other communicating vehicles (the followers), the states of the followers transition to the Plan Proposed state. In the Plan Proposed state, each vehicle acknowledges that the plan is valid by checking if it would like to join the plan. For example, each vehicle can confirm that the driver agrees to join the platoon and that the proposed Plan is safe to follow. Moreover, the leader can manually cancel the plan while in the Plan Proposed state, forcing a transition to the Plan Cancel state. When the leader receives an acknowledgement from every vehicle in the Plan, it moves to the Plan Active state while also informing the followers so that all vehicles move to the Plan Active state together. To ensure safety, while in the Plan Active state every vehicle in the platoon continuously monitors the surrounding conditions to decide if the Plan must stop. For instance, some conditions which can result in canceling the plan include 1) message timeout (a particular message has not been received for a period of time longer than a specified threshold), 2) any driver intervention such as taps the gas / brake pedal, 3) Driver's decision to stop. When one or more of these conditions are detected by any vehicle participating in the Plan, the detecting vehicle informs the other vehicles in the platoon and they move together to the Plan Cancel state. After some threshold time, each vehicle transitions from the Plan Cancel state to the Ready state and the platoon can be restarted as needed. This platoon management was used in the closed track demonstration [30] and the public road demonstration [31]. More details about this demonstration can be found in [69, 26]

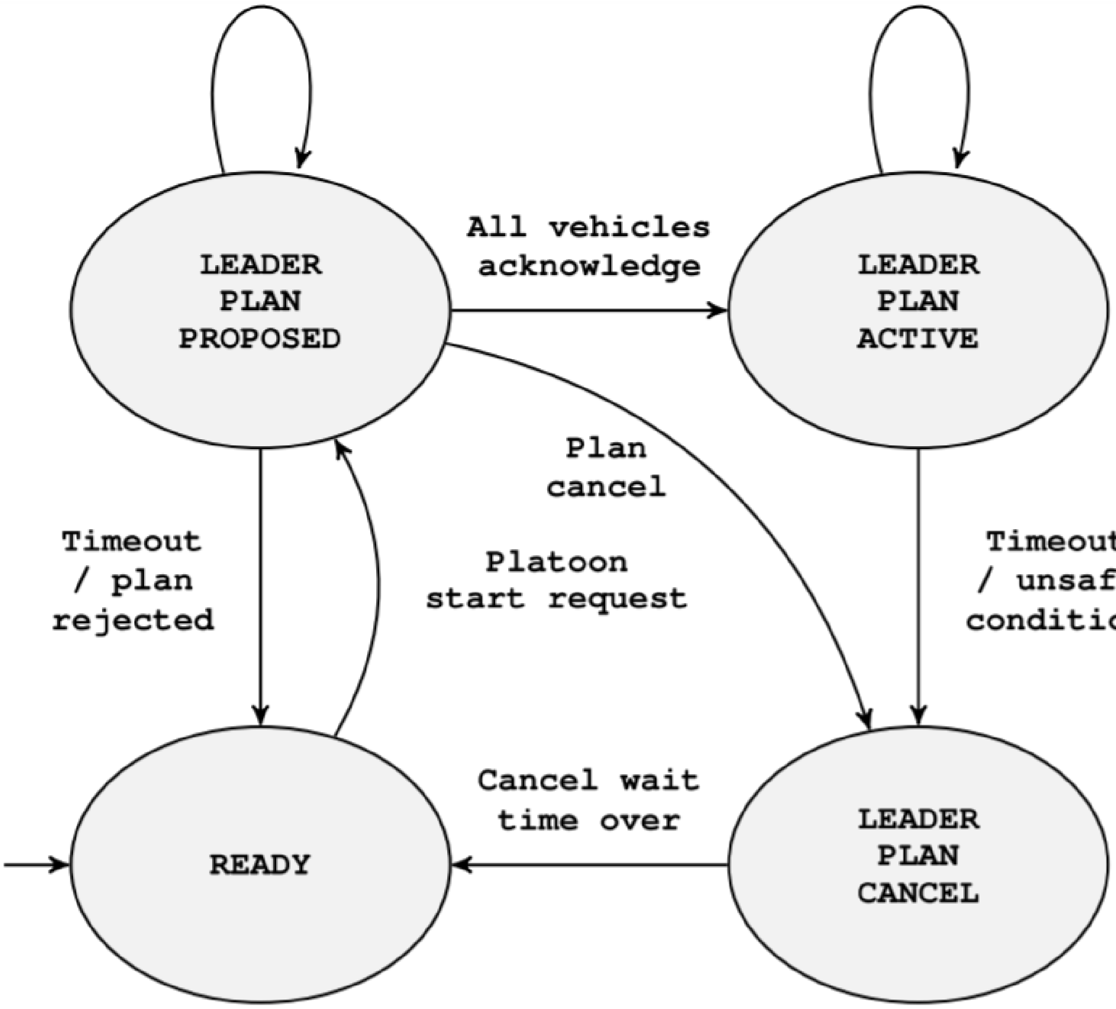


Figure A.1: A diagram of the transitions in our finite state machine, shown here for the leader vehicle for simplicity

# Bibliography

[1] The U.S. Department of Transportation's National Highway Traffic Safety Administration. *NHTSA Releases 2019 Crash Fatality Data, Dec. 18, 2020.* URL: `https://www.nhtsa.gov/press-releases/roadway-fatalities-2019-fars`.

[2] Michael J Sprung et al. "National Transportation Statistics 2018". In: (2018).

[3] Phil Lasley. "2019 URBAN MOBILITY REPORT". In: (2019).

[4] SAE On-Road Automated Vehicle Standards Committee et al. "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems". In: *SAE Standard J* 3016 (2014), pp. 1–16.

[5] Peter Knoll. "Driving assistance systems". In: *Brakes, Brake Control and Driver Assistance Systems.* Springer, 2014, pp. 180–185.

[6] Ashwin Carvalho et al. "Automated driving: The role of forecasts and uncertainty: A control perspective". In: *European Journal of Control* 24 (2015), pp. 14–32.

[7] Ken Erickson et al. *Automotive Infotainment Trends & Opportunities Driven By Integrated Media Content Availability.* Tech. rep. SAE Technical Paper, 2006.

[8] Steven E Shladover et al. "Automated vehicle control developments in the PATH program". In: *IEEE Transactions on Vehicular Technology* 40.1 (1991), pp. 114–130.

[9] Steven E Shladover. "PATH at 20 History and major milestones". In: *IEEE Transactions on Intelligent Transportation Systems* 8.4 (2007), pp. 584–592.

[10] Roozbeh Kianfar et al. "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge". In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (2012), pp. 994–1007.

[11] Cristofer Englund et al. "The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles". In: *IEEE Wireless Communications* 23.4 (2016), pp. 146–152.

[12] Shin Kato et al. "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications". In: *IEEE Transactions on Intelligent Transportation Systems* 3.3 (2002), pp. 155–161.

[13] Carl Bergenhem et al. "Challenges of platooning on public motorways". In: *17th World Congress on Intelligent Transport Systems.* 2010, pp. 1–12.

[14] Florian Ahlers and Christian Stimming. “Cooperative laserscanner pre-data-fusion”. In: *2008 IEEE Intelligent Vehicles Symposium*. IEEE. 2008, pp. 1187–1190.

[15] Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. “Eco-routing of connected plug-in hybrid electric vehicles”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 2245–2250.

[16] Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. “Control of connected and automated vehicles: State of the art and future challenges”. In: *Annual Reviews in Control* 45C (2018), pp. 18–40.

[17] Yeojun Kim et al. “Hardware-In-the-Loop for Connected Automated Vehicles Testing in Real Traffic”. In: *14th International Symposium on Advanced Vehicle Control*. 2018.

[18] Yeojun Kim, Jacopo Guanetti, and Francesco Borrelli. “Compact cooperative adaptive cruise control for energy saving: Air drag modelling and simulation”. In: *IEEE Transactions on Vehicular Technology* 70.10 (2021), pp. 9838–9848.

[19] Yeojun Kim, Jacopo Guanetti, and Francesco Borrelli. “Robust Eco Adaptive Cruise Control for Cooperative Vehicles”. In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 1214–1219.

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

[21] Yeojun Kim et al. “Eco-driving with learning model predictive control”. In: *14th International Symposium on Advanced Vehicle Control*. 2018.

[22] Sangjae Bae et al. “Design and implementation of ecological adaptive cruise control for autonomous driving with communication to traffic lights”. In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 4628–4634.

[23] Sangjae Bae et al. “Real-time ecological velocity planning for plug-in hybrid vehicles with partial communication to traffic lights”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 1279–1285.

[24] Yeojun Kim et al. “Shared Perception for Connected and Automated Vehicles”. In: *The 31st IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020.

[25] Jesus Felez, Yeojun Kim, and Francesco Borrelli. “A model predictive control approach for virtual coupling in railways”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.7 (2019), pp. 2728–2739.

[26] Stanley W Smith et al. “Improving Urban Traffic Throughput With Vehicle Platooning: Theory and Experiments”. In: *IEEE Access* 8 (2020), pp. 141208–141223.

[27] Yeojun Kim. *Hardware-in-the-loop for connected automated vehicles testing in real traffic*. `https://www.youtube.com/watch?v=68Q6xZyjpag&t=2s`. 2020. URL: `https://www.youtube.com/watch?v=68Q6xZyjpag&t=2s`.

[28] Yeojun Kim. *urban ACC testing*. `https://www.youtube.com/watch?v=O-nS9FZotfk`. 2020. URL: `https://www.youtube.com/watch?v=O-nS9FZotfk`.

[29] Yeojun Kim and Samuel Tay. *Shared Perception for Connected and Automated Vehicles*. www.youtube.com/watch?v=iBY3RQlSGDU. 2020. URL: www.youtube.com/watch?v=iBY3RQlSGDU.

[30] *UC Berkeley Vehicle Platooning at California Proving Grounds*. https://www.youtube.com/watch?v=U-O9iUZElR8&feature=youtu.be. 2020. URL: https://www.youtube.com/watch?v=U-O9iUZElR8&feature=youtu.be.

[31] *Platoon field test on Live Oak Ave. in Arcadia, CA, 02/28/2020*. https://www.youtube.com/watch?v=xPYR_xP3FuY&feature=youtu.be. 2020. URL: https://www.youtube.com/watch?v=xPYR_xP3FuY&feature=youtu.be.

[32] Yeojun Kim. *Platoon Coordination Demo*. https://www.youtube.com/watch?v=OLLmEILbOVQ. 2020. URL: https://www.youtube.com/watch?v=OLLmEILbOVQ.

[33] Claudine Badue et al. "Self-driving cars: A survey". In: *Expert Systems with Applications* (2020), p. 113816.

[34] Yongkeun Choi et al. "Data-driven Energy Management Strategy for Plug-in Hybrid Electric Vehicles with Real-World Trip Information". In: *arXiv preprint arXiv:2006.03704* (2020).

[35] Lino Guzzella and Antonio Sciarretta. *Vehicle Propulsion Systems*. Springer, 2013. ISBN: 9783642359125.

[36] Cohda. *Cohda Wireless MK5 OBU product brief sheet*. 2019. URL: https://www.cohdawireless.com/wp-content/uploads/2018/08/CW_Product-Brief-sheet-MK5-OBU.pdf.

[37] John B Kenney. "Dedicated short-range communications (DSRC) standards in the United States". In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182.

[38] *PreScan, ver:8.1.0 [Computer software]*. https://tass.plm.automation.siemens.com/prescan.

[39] *Vissim, ver:08 [Computer software], (2015)*. http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim.

[40] *Road Load Measurement and Dynamometer Simulation Using Coastdown Techniques*. Mar. 2010.

[41] Wolf-Heinrich Hucho. *Aerodynamics of road vehicles: from fluid mechanics to vehicle engineering*. Elsevier, 2013.

[42] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.

[43] Andreas Wächter and L Biegler. *IPOPT-an interior point OPTimizer*. 2009.

[44] Antonio Sciarretta, Giovanni De Nunzio, and Luis Leon Ojeda. "Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem". In: *IEEE Control Systems* 35.5 (2015), pp. 71–90.

[45] Peter D Friedman and Phil Grossweiler. “An analysis of US federal mileage ratings for plug-in hybrid electric vehicles”. In: *Energy Policy* 74 (2014), pp. 697–702.

[46] Ziran Wang et al. “A survey on cooperative longitudinal motion control of multiple connected and automated vehicles”. In: *IEEE Intelligent Transportation Systems Magazine* 12.1 (2019), pp. 4–24.

[47] Ardalan Vahidi and Antonio Sciarretta. “Energy saving potentials of connected and automated vehicles”. In: *Transportation Research Part C: Emerging Technologies* 95 (2018), pp. 822–843.

[48] Shengbo Li et al. “Model predictive multi-objective vehicular adaptive cruise control”. In: *IEEE Transactions on Control Systems Technology* 19.3 (2011), pp. 556–566.

[49] Roman Schmied et al. “Nonlinear MPC for emission efficient cooperative adaptive cruise control”. In: *IFAC-PapersOnLine* 48.23 (2015), pp. 160–165.

[50] Lorenzo Bertoni et al. “An adaptive cruise control for connected energy-saving electric vehicles”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 2359–2364.

[51] D Swaroop and J Karl Hedrick. “String stability of interconnected systems”. In: *IEEE Transactions on Automatic Control* 41.3 (1996), pp. 349–357.

[52] William B Dunbar and Derek S Caveney. “Distributed receding horizon control of vehicle platoons: Stability and string stability”. In: *IEEE Transactions on Automatic Control* 57.3 (2012), pp. 620–633.

[53] Victor S Dolk, Jeroen Ploeg, and W. P. Maurice H Heemels. “Event-triggered control for string-stable vehicle platooning”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.12 (2017), pp. 3486–3500.

[54] Jing Zhou and Huei Peng. “Range policy of adaptive cruise control vehicles for improved flow stability and string stability”. In: *IEEE Transactions on Intelligent Transportation Systems* 6.2 (2005), pp. 229–237.

[55] Sadayuki Tsugawa, Shin Kato, and Keiji Aoki. “An automated truck platoon for energy saving”. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE. 2011, pp. 4109–4114.

[56] Robin Andersson. *Online estimation of rolling resistance and air drag for heavy duty vehicles*. 2012.

[57] Michael P Lammert et al. “Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass”. In: *SAE International Journal of Commercial Vehicles* 7.2014-01-2438 (2014), pp. 626–639.

[58] Michael Zabat et al. “The aerodynamic performance of platoons: A final report”. In: (1995).

[59] Jack N Barkenbus. “Eco-driving: An overlooked climate change initiative”. In: *Energy Policy* 38.2 (2010), pp. 762–769.

[60] Valerio Turri. “Fuel-efficient and safe heavy-duty vehicle platooning through look-ahead control”. PhD thesis. KTH Royal Institute of Technology, 2015.

[61] Stéphanie Lefevre, Ashwin Carvalho, and Francesco Borrelli. “A learning-based framework for velocity control in autonomous driving”. In: *IEEE Transactions on Automation Science and Engineering* 13.1 (2016), pp. 32–42.

[62] Anil Aswani et al. “Provably safe and robust learning-based model predictive control”. In: *Automatica* 49.5 (2013), pp. 1216–1226.

[63] Ather Gattami et al. “Establishing safety for heavy duty vehicle platooning: A game theoretical approach”. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 3818–3823.

[64] David Q Mayne, María M Seron, and SV Raković. “Robust model predictive control of constrained linear systems with bounded disturbances”. In: *Automatica* 41.2 (2005), pp. 219–224.

[65] Francesco Borrelli. *Constrained optimal control of linear and hybrid systems.* Vol. 290. Springer, 2003.

[66] Franco Blanchini. “Set invariance in control”. In: *Automatica* 35.11 (1999), pp. 1747–1767.

[67] Roozbeh Kianfar, Paolo Falcone, and Jonas Fredriksson. “A receding horizon approach to string stable cooperative adaptive cruise control”. In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC).* IEEE. 2011, pp. 734–739.

[68] Nikolaos Floudas, Aris Polychronopoulos, and Angelos Amditis. “A survey of filtering techniques for vehicle tracking by radar equipped automotive platforms”. In: *2005 7th International Conference on Information Fusion.* Vol. 2. IEEE. 2005, 8–pp.

[69] Stanley W Smith et al. “Balancing Safety and Traffic Throughput in Cooperative Vehicle Platooning”. In: *arXiv preprint arXiv:1904.08557* (2019).

[70] Roozbeh Kianfar, Paolo Falcone, and Jonas Fredriksson. “A distributed model predictive control approach to active steering control of string stable cooperative vehicle platoon”. In: *IFAC Proceedings Volumes* 46.21 (2013), pp. 750–755.

[71] Kyoungho Ahn et al. “Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels”. In: *Journal of transportation engineering* 128.2 (2002), pp. 182–190.

[72] Hakan Johansson et al. “Impact of EcoDriving on emissions”. In: *International Scientific Symposium on Transport and Air Pollution, Avignon, France.* Citeseer. 2003.

[73] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual.* 2018. URL: `http://www.gurobi.com`.

[74] Kamil Çağatay Bayindir, Mehmet Ali Gözüküçük, and Ahmet Teke. "A comprehensive overview of hybrid electric vehicle: Powertrain configurations, powertrain control techniques and electronic control units". In: *Energy conversion and Management* 52.2 (2011), pp. 1305–1313.

[75] Lorenzo Serrao, Simona Onori, and Giorgio Rizzoni. "A comparative analysis of energy management strategies for hybrid electric vehicles". In: *Journal of Dynamic Systems, Measurement, and Control* 133.3 (2011).

[76] Gino Paganelli et al. "General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles". In: *JSAE review* 22.4 (2001), pp. 511–518.

[77] Cristian Musardo et al. "A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management". In: *European Journal of Control* 11.4-5 (2005), pp. 509–524.

[78] Xuewei Qi et al. "Data-driven reinforcement learning–based real-time energy management system for plug-in hybrid electric vehicles". In: *Transportation Research Record* 2572.1 (2016), pp. 1–8.

[79] Yongkeun Choi et al. "Data-driven Energy Management Strategy for Plug-in Hybrid Electric Vehicles with Real-World Trip Information". In: *International Federation of Automatic Control*. 2020.

[80] Behrang Asadi and Ardalan Vahidi. "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time". In: *IEEE transactions on control systems technology* 19.3 (2010), pp. 707–714.

[81] Behrang Asadi and Ardalan Vahidi. "Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time". In: *IEEE transactions on control systems technology* 19.3 (2011), pp. 707–714.

[82] Matthew Barth et al. "Dynamic ECO-driving for arterial corridors". In: *2011 IEEE Forum on Integrated and Sustainable Transportation Systems, FISTS 2011* (2011), pp. 182–188.

[83] Kaijiang Yu, Junqi Yang, and Daisuke Yamaguchi. "Model predictive control for hybrid vehicle ecological driving using traffic signal and road slope information". In: *Control Theory and Technology* 13.1 (2015), pp. 17–28.

[84] Chao Sun et al. "Optimal eco-driving control of connected and autonomous vehicles through signalized intersections". In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 3759–3773.

[85] Frank Lattemann et al. *The predictive cruise control–a system to reduce fuel consumption of heavy duty trucks*. Tech. rep. SAE Technical paper, 2004.

[86] Valerio Turri, Bart Besselink, and Karl H Johansson. "Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning". In: *IEEE Transactions on Control Systems Technology* 25.1 (2017), pp. 12–28.

[87] Ugo Rosolia and Francesco Borrelli. “Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework.” In: *IEEE Transactions on Automatic Control* (2017).

[88] Ugo Rosolia, Ashwin Carvalho, and Francesco Borrelli. “Autonomous racing using learning model predictive control”. In: *American Control Conference (ACC), 2017*. IEEE. 2017, pp. 5115–5120.

[89] Maximilian Brunner et al. “Repetitive learning model predictive control: An autonomous racing example”. In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE. 2017, pp. 2545–2550.

[90] Lars Johannesson Nikolce Murgovski and Jonas Sjöberg. “Convex modeling of energy buffers in power control applications”. In: *IFAC Proceedings Volumes* 45 (30 2012), pp. 92–99.

[91] L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems*. Springer, 2013.

[92] Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli. “Robust learning model predictive control for iterative tasks: Learning from experience”. In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE. 2017, pp. 1157–1162.

[93] Md Abdus Samad Kamal et al. “On board eco-driving system for varying road-traffic environments using model predictive control”. In: *Control Applications (CCA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1636–1641.

[94] Jessica Van Brummelen et al. “Autonomous vehicle perception: The technology of today and tomorrow”. In: *Transportation research part C: emerging technologies* 89 (2018), pp. 384–406.

[95] Saif Al-Sultan et al. “A comprehensive survey on vehicular ad hoc network”. In: *Journal of network and computer applications* 37 (2014), pp. 380–392.

[96] Xiang Cheng et al. “5G-enabled cooperative intelligent vehicular (5GenCIV) framework: When Benz meets Marconi”. In: *IEEE Intelligent Systems* 32.3 (2017), pp. 53–59.

[97] Matthias Roeckl et al. “Sensing the environment for future driver assistance combining autonomous and cooperative appliances”. In: *Proceedings*. 2008, pp. 45–56.

[98] Seong-Woo Kim et al. “Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 5059–5066.

[99] Andreas Rauch et al. “Car2x-based perception in a high-level fusion architecture for cooperative perception systems”. In: *2012 IEEE Intelligent Vehicles Symposium*. IEEE. 2012, pp. 270–275.

[100] Andreas Rauch et al. “Inter-vehicle object association for cooperative perception systems”. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE. 2013, pp. 893–898.

[101] Moreno Ambrosin et al. “Object-level Perception Sharing Among Connected Vehicles”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 1566–1573.

[102] Florian Seeliger et al. “Advisory warnings based on cooperative perception”. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 246–252.

[103] Philippe Xu et al. “System architecture of a driverless electric car in the grand cooperative driving challenge”. In: *IEEE Intelligent Transportation Systems Magazine* 10.1 (2018), pp. 47–59.

[104] Dietrich Franken and Andreas Hupper. “Improved fast covariance intersection for distributed data fusion”. In: *2005 7th International Conference on Information Fusion*. Vol. 1. IEEE. 2005, 7–pp.

[105] Fabian de Ponte Müller, Estefania Munoz Diaz, and Ibrahim Rashdan. “Cooperative positioning and radar sensor fusion for relative localization of vehicles”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 1060–1065.

[106] U Bock and JU Varchmin. “Enhancement of the occupancy of railroads using virtually coupled train formations”. In: *World Congress on Railway Research (WCRR)*. 1999.

[107] U Bock and G Bikker. “Design and development of a future freight train concept–virtually coupled train formations”. In: *IFAC Proceedings Volumes* 33.9 (2000), pp. 395–400.

[108] “The Rail Joint Undertaking”. In: (2019). `http://www.shift2rail.org`.

[109] “Shift2Rail Strategic Master Plan, Version 1.0.” In: (Mar. 2015). `http://ec.europa.eu/transport/modes/rail/doc/2015-03-31-decisionn4-2015-adoption-s2r-masterplan.pdf`.

[110] Javier Goikoetxea. “Roadmap towards the wireless virtual coupling of trains”. In: *International Workshop on Communication Technologies for Vehicles*. Springer. 2016, pp. 3–9.

[111] Gonzalez R. *Simulacion del sistema CBTC en la Linea 1 de Metro de Ho Chi Minh*. `https://e-archivo.uc3m.es/handle/10016/12156`. 2011.