2023

Safety-critical optimal control in autonomous traffic systems

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

SAFETY-CRITICAL OPTIMAL CONTROL IN AUTONOMOUS TRAFFIC SYSTEMS

by

KAIYUAN XU

B.S., Tsinghua University, 2016
M.S., Tsinghua University, 2019

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2023

Approved by

First Reader

Christos G. Cassandras, Ph.D
Distinguished Professor of Engineering
Professor of Electrical and Computer Engineering
Professor and Division Head of Systems Engineering

Second Reader

Calin A. Belta, Ph.D
Professor of Mechanical Engineering
Professor of Systems Engineering
Professor of Electrical and Computer Engineering

Third Reader

Ioannis Ch. Paschalidis, Ph.D
Distinguished Professor of Engineering
Professor of Electrical and Computer Engineering
Professor of Biomedical Engineering
Professor of Systems Engineering
Professor of Computing and Data Science

Fourth Reader

Sean B. Andersson, Ph.D
Professor and Chair of Mechanical Engineering
Professor of Systems Engineering

Acknowledgments

I would like to express my thanks to my advisor Prof. Christos G. Cassandras for his support and guidance through my 4-year PhD life at Boston University. He showed by example how to become a dedicated researcher and a responsible mentor, and gave me freedom to choose my research interest and help me formulate the research problems to make them concrete. Without his guidance and help, I would not be able to finish my PhD smoothly.

My gratitude also goes to my committee members, Prof. Ioannis Paschalidis, Prof. Calin Belta and Prof. Sean Andersson for their time and efforts given to my PhD thesis. Their constructive comments and suggestions broaden my horizon and help refine my thesis.

I am also in debt of gratitude to my family for their long time support throughout my student life. The past four years are extremely hard for both my family and myself. The outbreak of COVID-19 and the expired Visa has kept us separated since I came to USA. I would like to thank my family, especially my parents Jian Xu and Qin Liu, for your patience and understanding, as long as your encouragement which came all the way along my PhD life. I also want to thank my girlfriend Qingyi Wang for her company and consideration which brightens my life during the this hard time.

Last but not least, I would like to thank Boston and all the friends I met here. It has been a good time to live in this lovely city with these lovely people.

Kaiyuan Xu
Charles River Campus, Boston University
Jun. 29, 2023

SAFETY-CRITICAL OPTIMAL CONTROL IN AUTONOMOUS TRAFFIC SYSTEMS

KAIYUAN XU

Boston University, College of Engineering, 2023

Major Professor: Christos G. Cassandras, Ph.D
Distinguished Professor of Engineering
Professor of Electrical and Computer Engineering
Professor and Division Head of Systems Engineering

ABSTRACT

Traffic congestion is a central problem in transportation systems, especially in urban areas. The rapid development of Connected and Automated Vehicles (CAVs) and new traffic infrastructure technologies provides a promising solution to solve this problem. This work focuses on the safety-critical optimal control of CAVs in autonomous traffic systems.

The dissertation starts with the roundabout problem of controlling CAVs travelling through a roundabout so as to jointly minimize their travel time, energy consumption, and centrifugal discomfort while providing speed-dependent safety guarantees. A systematic approach is developed to determine the safety constraints for each CAV dynamically. The joint optimal control and control barrier function (OCBF) controller is applied, where the unconstrained optimal control solution is derived which is subsequently optimally tracked by a real-time controller while guaranteeing the satisfaction of all safety constraints.

Secondly, the dissertation deals with the feasibility problem of OCBF. The feasibility problem arises when the control bounds conflict with the Control Barrier Function (CBF) constraints and is solved by adding a single feasibility constraint to the Quadratic Problem (QP) in the OCBF controller to derive the feasibility guaranteed OCBF. The feasibility guaranteed OCBF is applied in the merging control problem which provably guarantees the feasibility of all QPs derived from the OCBF controller.

Thirdly, the dissertation deals with the performance loss of OCBF due to the improperly selected reference trajectory which deviates largely from the complete optimal solution especially when the vehicle limitations are tight. A neural network is used to learn the control policy from data retrieved by offline calculation from the complete optimal solutions. Tracking the learnt reference trajectory with CBF outperforms OCBF in simulation experiments.

Finally, a hierarchical framework of modular control zones (CZ) is proposed to extend the safety-critical optimal control of CAV from a single CZ to a traffic network. The hierarchical modular CZ framework is developed consisting of a lower-level OCBF controller and a higher-level feedback flow controller to coordinate adjacent CZs which outperforms a direct extension of the OCBF framework to multiple CZs without any flow control in simulation.

Contents

List of Tables

List of Figures

List of Abbreviations

BF	Barrier Functions
CAV	Connected and Automated Vehicle
CBF	Control Barrier Function
CLF	Control Lyapunov Function
LSTM	Long-Short Term Memory
MPC	Model Predictive Control
NN	Neural Network
OCBF	Optimal Control and Control Barrier Functions
QP	Quadratic Programming

Chapter 1

Introduction

1.1 Background

Traffic congestion has always been a hard problem to solve, especially in urban areas, which causes a significant economic loss including billions of hours of time and billions of gallons of fuel wasted on roads. Coordinating and controlling vehicles in a traffic system is a challenging problem in terms of ensuring safety and passenger comfort while minimizing congestion and energy consumption. Recently, the development of Intelligent Transportation Systems (ITS) including the rapid evolution of Connected and Automated Vehicles (CAVs) and the emergence of new traffic infrastructure technologies, has provided new potentials to solve this problem through better information utilization and more precise vehicle trajectory design. This thesis focuses on the coordination and control of CAVs to deal with traffic congestion.

A CAV is a typical autonomous system that is managed and supervised by itself without reliance of instructions from other entities and serves as an autonomous agent within a large autonomous system. With the characteristic of connectivity, a CAV has the ability to fully utilize the information from its neighbors (including the infrastructure) to design a safe and optimal trajectory in terms of minimizing travel time, energy consumption as well as discomfort in a decentralized way. Optimal control methods are often used in dynamic systems to achieve the optimal objective subject to known system dynamics and constraints. Employing optimal control methods to CAVs is straightforward but challenging. As safety is always the first priority in driving, the safety constraints are always tight to avoid collision, which makes the optimal control problem hard to solve. How to control the CAVs in real time in an optimal way (in terms of minimizing travel time, energy consumption and discomfort) while guaranteeing safety is the main problem to solve in this dissertation.

1.2 State of the Art

1.2.1 Optimal Control in Traffic Control Problems

The performance of traffic networks critically depends on the control of conflict areas such as intersections, roundabouts and merging roadways which define the main bottlenecks in these networks (Rios-Torres and Malikopoulos, 2016). The development of CAVs provides a promising solution to improve the performance in these conflict areas. One of the very early piece of research work on CAVs dates back to 1966, where an optimal linear feedback system was designed to regulate the position and velocity of high speed moving vehicles (Levine and Athans, 1966). A more systematic research was conducted in (Varaiya, 1993) where an automated intelligent vehicle/highway system was proposed.

Both centralized and decentralized methods have been studied to deal with the control and coordination of CAVs at conflict areas. Centralized mechanisms are often used in forming platoons in merging problems (Shladover et al., 1991; Xu et al., 2019a) and determining passing sequences at intersections (Xu et al., 2020; Xu et al., 2019b). These approaches tend to work better when the safety constraints are independent of speed and they generally require significant computation resources, especially when traffic is heavy. They are also not easily amenable to disturbances.

Decentralized mechanisms restrict all computation to be done on board each vehicle with information sharing limited to a small number of neighbor vehicles (Milanés et al., 2010; Rios-Torres et al., 2015; Bichiou and Rakha, 2018; Hult et al., 2016). Optimal control problem formulations are used in some of these approaches. The objectives in such problem formulations typically target the minimization of acceleration (Rios-Torres et al.,

2015) or the maximization of passenger comfort (measured as the acceleration derivative or jerk) (Ntousakis et al., 2016; Rathgeber et al., 2015). The optimal control approaches usually assume that no constraints are active in order to get simple analytical solutions. When the constraints are included, the derivation of the optimal trajectory becomes too complex to be solved in real time, even under simple vehicle dynamics (Malikopoulos et al., 2018; Ntousakis et al., 2016; Xiao et al., 2019).

Among the conflict areas, roundabouts are important components of a traffic network because they usually perform better than typical intersections in terms of efficiency and safety (Flannery and Datta, 1997). However, they can become significant bottleneck points as the traffic rate increases due to an inappropriate priority system, resulting in significant delays when the circulating flow is heavy. Previous studies mainly focus on conventional (human-driven) vehicles and try to solve the problem through improved road design or traffic signal control (Martin-Gasulla et al., 2016; Yang et al., 2004; Xu et al., 2016). More recently, however, researchers have proposed methods for decentralized optimal control of CAVs in a roundabout based on formulating an optimal control problem with an analytical solution provided in (Zhao et al., 2018). The problem is decomposed so that first the minimum travel time problem is solved under the assumption that all vehicles use the same maximum speed within the roundabout. Then, fixing this time, the control input that minimizes the energy consumption is derived analytically. The general framework for decentralized optimal control of CAVs used in intersections is implemented for roundabouts in (Chalaki et al., 2020). The analysis is similar to (Zhao et al., 2018) except that there is no circulating speed assumption.

1.2.2 Model Predictive Control and Control Barrier Function

Model Predictive Control (MPC) is a modern control strategy commonly used in control systems known for its capacity to provide optimized responses while accounting for state and input constraints of the system(Garcia et al., 1989). According to the system dynamics, MPC can be classified as Linear MPC (LMPC) and Nonlinear MPC (NMPC). The basic idea of MPC is to create a discretized predictive model (or a simplified one to reduce calculation) to predict the state of the system over a receding horizon. The MPC method can guarantee terminal constraints at the cost of heavy calculation due to the receding horizon it maintains. The computation becomes more expensive when it comes to NMPC due to repetitively solving Nonlinear Programs (NP).

Barrier Functions (BFs) are Lyapunov-like functions (Tee et al., 2009) well known in optimization problems (Boyd et al., 2004). Starting from a feasible solution, the BF can guarantee the solution after each searching step is also feasible. Utilizing this set invariance property, BFs have been extended to control problems as Control Barrier Functions (CBFs). It is shown that for nonlinear systems affine in control with cost functions quadratic in controls, an optimal control problem with safety constraints can be mapped to a sequence of quadratic programs (QPs) with some conservatism (Galloway et al., 2015). The core of the CBF method is the forward invariance property (Ames et al., 2014b) which refers to the set invariance of the feasible set defined by a constraint. In other words, any control input that satisfies the CBF constraint also satisfies the original constraint. In addition, a Control Lyapunov Function (CLF) (Ames et al., 2012) can guarantee the system state to converge to a desired one. Thus, the initial problem can be reformulated into a sequence of CBF-CLF-QPs. The CBF method outperforms MPC techniques in that the forward invariance guarantees the satisfaction of constraints in each time interval and it does not need discretized dynamics. The QPs in CBF method are usually less computationally expensive compared to MPC.

Both methods have been applied in traffic control problems in order for real time application. MPC techniques are often employed to account for additional constraints and to compensate for disturbances by re-evaluating optimal actions (Cao et al., 2015; Mukai et al., 2017; Nor and Namerikawa, 2018). As an alternative, CBFs are used in traffic merging problems (Xiao and Belta, 2022; Xiao et al., 2019) which provide provable guarantees that safety constraints are always satisfied under very general nonlinear vehicle dynamics affine in control. The joint optimal control and control barrier functions (OCBF) approach is proposed and applied in traffic control problems including merging roadways (Xiao et al., 2021b) and intersections (Xu et al., 2022b), which uses the unconstrained optimal solution as the referenced trajectory and optimally tracks the referenced trajectory subject to CBF constraints that guarantee safety.

1.2.3 Feasibility Problem of Control Barrier Functions

One of the main challenges for the CBF method is the feasibility problem. As mentioned before, the CBF method can easily map a nonlinear constraint to a new CBF constraint that is linear in control for affine control systems to form a series of CBF-CLF-QPs. By solving the problem one time step forward, it provably guarantees the satisfaction of the initial constraints due to the forward invariance property. However, the series of CBF-CLF-QPs may be infeasible due to multiple reasons and thus results in the loss of safety guarantees.

The first possibility is the violation of the initial constraint, potentially due to noise or model inaccuracy. Adaptive Control Barrier Functions (AdaCBFs) are proposed to accommodate time-varying control bounds and noise in the system dynamics and applied to a cruise control problem (Xiao et al., 2022a). There are also auxiliary methods in different fields to help enable the satisfaction of initial constraints. In traffic control, a Feasibility Enforcement Zone (FEZ) is proposed to enforce the satisfaction of initial constraints (Zhang et al., 2017).

Another reason is the inadequately small time step. In real applications, a CBF controller is usually realized by discretizing time and solving a series of CBF-CLF-QPs at each time step. The proof of the forward invariant property is based on the continuous dynamic system. The CBF constraints may become infeasible due to violation of the initial constraint during the time interval, especially when the system states change fast. Multiple methods are proposed to deal with this problem. Discrete Control Barrier Functions are proposed and applied in bipadal robot navigation (Agrawal and Sreenath, 2017) which define the CBFs in a discrete time form. Another method is to use event-driven CBF as opposed to time-driven ones(Xiao et al., 2021a). Safety constraint related events are defined to trigger the reformulation and solution of the CBF-CLF-QPs thus to guarantee the satisfaction of safety constraint at each time step.

The third reason is the myopic nature of CBF methods, especially when the control limit is tight which greatly increases the chance of conflicting with the CBF/CLF constraints. As aforementioned, with the forward invariance property, the CBF method is able to guarantee critical safety constraints by solving the problem just one time step forward, thus is more computationally efficient compared with MPC but is also more myopic. Different approaches to improve feasibility for specific applications have been proposed. For the adaptive cruise control (ACC) problem, the infeasibility issue is dealt with by adding an addition complex safety constraint related to the minimum braking distance (Ames et al., 2014b). A more general framework is proposed in (Xiao et al., 2022b) where an additional feasibility constraint is added to guarantee feasibility. However, in this work, the constructed feasibility constraints are limited to be independent of the control, which is not applicable for systems with control dependent constraints.

1.2.4 Machine Learning in Autonomous Traffic Systems

Neural network has become a hot research topic and has shown its power and potential in areas like computer vision, speech recognition, natural language processing due to the increasing computing power (Abiodun et al., 2018). Different types of neural networks are used in autonomous traffic systems for different tasks. A Convolutional Neural Network (CNN) is commonly used in image related tasks such as real-time object detection in autonomous driving (Wu et al., 2017). A Recurrent Neural Network (RNN) is often used in time related tasks due to the feedback introduced in the recurrent structure. Long-short

Term Memory (LSTM), a typical RNN, is used to predict vehicle trajectories on highway (Altché and de La Fortelle, 2017).

Apart from these supervised learning methods, reinforcement learning (RL) is also widely used in control systems as well as the autonomous traffic systems. Reinforcement learning enables an agent to learn an optimal policy to maximize its reward from scratch, with little or no prior knowledge of the model. Reinforcement learning is widely applied in autonomous driving research due to its model-free and online learning property (Sallab et al., 2017; Kiran et al., 2022). However, reinforcement learning has its drawbacks. First, reinforcement learning makes little utilization of model information and expert experience, thus wasting a lot of computing power and time. Imitation learning can be considered as an alternative which learns a control policy through imitating (Hussein et al., 2017). Also reinforcement learning learns from making mistakes by its nature, which makes it more effective in simulation-based scenarios while hard to implement in the real world. In safety-critical scenarios like autonomous driving, it is always risky to directly implement reinforcement learning without safety guarantees. Recent research aims at solving this problem by combining reinforcement learning with safety-critical control metrics like CBFs (Dawson et al., 2022). However, as the infeasibility problem of CBF is not considered, safety concern still remains.

1.3 Contributions and Thesis Outline

1.3.1 Summary of Contributions

The contributions of the dissertation are listed as follows and explained in detail in the following subsections. Firstly, the decentralized optimal control problem of CAVs in a roundabout is formulated and solved systematically using the OCBF controller. Secondly, the feasibility guaranteed OCBF is derived to provably resolve the feasibility problem arising from the roundabout problem. Thirdly, a neural network based method is proposed to learn a reference trajectory from the complete optimal solution to the traffic merging problem to resolve the performance loss problem. Last but not least, a hierarchical framework of modular control zones is proposed to extend the safety critical optimal control of CAVs from the previously studied single CZ scenario (e.g., traffic merging problem, roundabout problem) to multiple CZs and ultimately a general traffic network.

1.3.2 Decentralized Optimal Control of CAVs in a roundabout

The problem of controlling Connected and Automated Vehicles (CAVs) traveling through a roundabout is considered so as to jointly minimize their travel time, energy consumption, and centrifugal discomfort while providing speed-dependent and lateral roll-over safety guarantees, as well as satisfying velocity and acceleration constraints. We first develop a systematic approach to determine the safety constraints for each CAV dynamically, as it moves through different merging points in the roundabout. The unconstrained optimal control solution is derived and is subsequently optimally tracked by a real-time controller while guaranteeing that all constraints are always satisfied. Simulation experiments are performed to compare the proposed controller to a baseline of human-driven vehicles, showing its effectiveness under symmetric and asymmetric roundabout configurations, balanced and imbalanced traffic rates, and different sequencing rules for CAVs.

The main contributions are as follows. First, unlike (Zhao et al., 2018; Chalaki et al., 2020), the travel time and energy consumption are *jointly* minimized while also considering speed-dependent safety constraints at a set of MPs rather than merging zones, which makes solutions less conservative by improving roadway utilization. Second, not only the roundabouts consisting of only straight road segments are considered in (Xu et al., 2021) but also the circular roundabout configurations are analyzed where the comfort constraints are included and guaranteed to be satisfied. Third, the computational complexity of solving such an optimal control problem analytically is overcame by adopting the joint Optimal Control and Barrier Function (OCBF) approach in (Xiao et al., 2021b).

1.3.3 Feasibility guaranteed OCBF

Applying the joint optimal control and control barrier function (OCBF) method, a controller that optimally tracks the unconstrained optimal control solution while guaranteeing the satisfaction of all constraints is efficiently obtained by transforming the optimal tracking problem into a sequence of quadratic programs (QPs). However, these QPs can become infeasible, especially under tight control bounds, thus failing to guarantee safety constraints.

This dissertation resolves the QP feasibility problem when an OCBF controller is used in decentralized merging control, thus ensuring that feasible trajectories are always possible. The merging control problem is formulated as in (Xiao and Cassandras, 2021b) to jointly minimize the travel time and energy consumption subject to speed-dependent safety constraints as well as vehicle limitations. The OCBF approach is adopted where the feasibility of each QP problem is guaranteed by adding a single feasibility constraint to it following the strategy developed in (Xiao et al., 2022b) for general optimal control problems. While the feasibility constraints constructed in (Xiao et al., 2022b) are limited to be independent of the control, the structure of the safety constraints is exploited in the merging problem to derive control-dependent feasibility constraints and prove that all QPs needed to fully solve the merging problem are feasible. Extensive simulations of the merging control problem illustrate the effectiveness of this feasibility guaranteed controller.

1.3.4 Learning the Optimal Control Problem (OCP) Solution

Under the OCBF framework, the controller optimally tracks the unconstrained optimal control solution while guaranteeing the satisfaction of safety constraints using CBFs. However, when the scenario becomes complicated, there can be a large deviation between the unconstrained optimal trajectory and the complete optimal trajectory. Tracking the analytically unconstrained optimal trajectory easily results in saturated control or states when the vehicle limitations are tight, which may result in a poor performance.

To improve the potentially poor performance due to improperly selected referenced trajectory, a neural network (NN) is designed to learn the control policy from the data retrieved from the previously solved complete optimal solutions. A Long-short Term Memory (LSTM) based neural network is trained to learn the OCP solution. The OCBF approach is then applied to optimally track the trajectory learnt from neural network to guarantee safety (this method is named as NN+CBF). Simulations are performed comparing the trajectories and performance of the ego CAV using OCP, OCBF, NN+CBF respectively.

1.3.5 Scaling up the Optimal Safe Control of CAVs to a Traffic Network

The contribution of this section is to take a step towards a systematic extension of traffic control involving CAVs from one to multiple CZs and ultimately a complex traffic network. A hierarchical control framework is proposed consisting of modular interconnected CZs with two levels: (i) a decentralized lower level applied to each CZ, and (ii) an upper traffic flow control level. The interaction between the two levels relies on an interface that connects neighboring CZs and adjusts the values of velocity parameters controlled by the upper level to influence the performance of CAVs at the lower level, where CAVs are still controlled in a decentralized way. These velocity parameters effectively act as regulators of the traffic flow between adjacent CZs and can be dependent on global or only local information regarding the traffic network. The decentralized controllers used at the lower level are based on our prior work that combines Optimal Control with Control Barrier Functions (OCBF) (Xiao et al., 2021b). In the OCBF framework, a decentralized optimal control problem is first formulated with an objective which jointly minimizes (i) the travel time of each CAV over a given road segment from a point entering a Control Zone (CZ) to the eventual exit point, and (ii) a measure of its energy consumption. CBF-based constraints are included to *guarantee* that all safety constraints are satisfied at all times, taking advantage of the forward invariance property of CBFs (Xiao et al., 2023).

1.3.6 Outline of the Dissertation

This dissertation is organized as follows. In Chapter 2, the control barrier functions and its application in safety-critical control is reviewed as the prior knowledge of the dissertation. Chapter 3 describes the whole framework of solving the optimal control problem of CAVs in a roundabout using the joint optimal control and control barrier functions approach. Chapter 4 and 5 introduce the work on feasibility guaranteed OCBF and learning the complete optimal trajectory separately, in order to resolve the feasibility and performance problem arising from roundabout problem. Chapter 6 introduces a hierarchical framework of modular control zones which extends the study of safety-critical optimal control from a single control zone to the traffic network. Finally, Chapter 7 concludes the work and points out some future directions.

Chapter 2

Control Barrier Functions and Safety-Critical Optimal Control

In this chapter, the concept of control barrier functions and its application in safety-critical optimal control are briefly reviewed. This chapter does not serve as the contribution of the dissertation but is needed to set up a theoretical foundation for the rest of the thesis.

2.1 Control Barrier Functions

The term of control barrier functions motivates from *barrier certificates* that were introduced to formally prove safety of nonlinear systems (Prajna and Jadbabaie, 2004) and the *control Lyapunov functions* that aim to stabilizing a nonlinear system. Given a nonlinear system in the form:

$$\dot{x} = f(x), \tag{2.1}$$

where $x \in D \subset \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$, safety is defined in the context of enforcing the invariance of a safe set.

2.1.1 Set Invariance and Barrier Functions

Definition 2.1. *(Forward invariant set). A set $\mathcal{C}$ is forward invariant for system* (2.1) *if $\forall x(t_0) \in \mathcal{C}$ and $\forall t \geq t_0$, $x(t) \in \mathcal{C}$.*

A safe set $\mathcal{C}$ is normally defined as the superlevel set of a continuously differentiable function $h : D \subset \mathbb{R}^n \to \mathbb{R}$.

$$\mathcal{C} = \{x \in D \subset \mathbb{R}^n : h(x \geq 0)\} \tag{2.2}$$

This forward invaiance property helps define safety with respect to a set $\mathcal{C}$ in the sense that the solution or the future state of a system will never get out of safe set $\mathcal{C}$.

Definition 2.2. *(Safety) The system* (2.1) *is* <u>*safe*</u> *with respect to the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant.*

A necessary and sufficient condition for set invariance is given by Nagumo in the 1940s which focuses on boundary derivative properties of a safe set $\mathcal{C}$.

Theorem 2.1. *(Nagumo's Theorem, (Nagumo, 1942)) Given a dynamic system* (2.1)*, assuming that the safe set is defined as* (2.2)*, and that $\frac{\partial h}{\partial x}(x) \neq 0$ for all x such that $h(x) = 0$, then*

$$\mathcal{C} \text{ is invariant} \Leftrightarrow \dot{h}(x) \geq 0 \; \forall x \in \partial\mathcal{C}. \tag{2.3}$$

The Nagumo's Theorem is intuitive in the sense that a safe set always pushes the state of the system x to the interior of set $\mathcal{C}$ when x is at the boundary of $\mathcal{C}$. However, such condition is hard to justify and utilize in practice as it is a condition on the boundary of the safe set instead of for the whole set. In order to extend the condition for set invariance to the whole set, the concept of *barrier functions* is borrowed from optimization theory.

Definition 2.3. *(Class $\mathcal{K}$ function). (Khalil, 2015) A Lipschitz continuous function $\alpha : [0,a) \to [0,\infty), a > 0$ is a class $\mathcal{K}$ function if it is strictly monotonically increasing and $\alpha(0) = 0$.*

Definition 2.4. *(Extended class $\mathcal{K}$ function). (Ames et al., 2019) A Lipschitz continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is an extended class $\mathcal{K}_\infty$ function if it is strictly monotonically increasing and $\alpha(0) = 0$. (An extended class $\mathcal{K}_\infty$ function is defined on the entire real line.)*

Definition 2.5. *(Barrier function). (Ames et al., 2014b) The function $b : \mathbb{R}^n \to \mathbb{R}$ is a barrier function (BF) for systems* (2.1) *if there exist an (extended) class $\mathcal{K}$ function α such that*

$$\dot{b}(x) + \alpha(b(x)) \geq 0, \forall x \in \mathcal{C}. \tag{2.4}$$

Theorem 2.2. *(Glotfelter et al., 2017) Given a dynamic system* (2.1) *and a set $\mathcal{C}$ as in* (2.2)*, then*

$$\mathcal{C} \text{ is invariant} \Leftrightarrow \exists \text{ a BF } b : \mathcal{C} \to \mathbb{R}. \tag{2.5}$$

Barrier functions provide formal certificates for safety of nonlinear systems (2.1). However, system (2.1) only describes the dynamics of the systems without control.

2.1.2 Control Lyapunov Functions

To explore the relationship between formal safety and control, in the sequel, we consider the nonlinear affine control system in the form:

$$\dot{x} = f(x) + g(x)u, \tag{2.6}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^q$ are locally Lipschitz, $x \in D \subset \mathbb{R}^n$ (D denotes the set of admissible states) and $u \in U \subset \mathbb{R}^q$ (U denotes the set of admissible control).

Before establishing the control barrier function, we consider the stabilization problem of (2.6) where the control objective is to asymptotically stabilizing the nonlinear control system (2.6) to a point $x^* = 0$. The systems is stabilized to $V(x^*) = 0$, i.e. $x^* = 0$ if

$$\exists u = k(x) \text{ s.t. } \dot{V}(x, k(x)) \leq -\gamma(V(x)), \tag{2.7}$$

where

$$\dot{V}(x, k(x)) = L_f V(x) + L_g V(x) k(x),$$

L_f and L_g denote the Lie derivatives along f and g respectively, γ is a class $\mathcal{K}$ function, $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a positive definite function and furthermore, a *Lyapunov function*.

Instead of explicitly constructing a control law that satisfied (2.7), we can describe the set of all stabilizing controllers by defining the *control Lyapunov functions*.

Definition 2.6. *(Control Lyapunov function) A continously differentiable function $V : \mathbb{R}^n \to$*

$\mathbb{R}_{\geq 0}$ *is a* <u>*control Lyapunov function (CLF)*</u> *if it is positive definite and satisfies:*

$$\inf_{u \in U} [L_f V(x) + L_g V(x) u] \leq -\gamma(V(x)), \tag{2.8}$$

where γ is a class $\mathcal{K}$ function.

The set of all stabilizing controllers for every point $x \in D$ is described as:

$$K_{\text{clf}}(x) := \{u \in U \ : \ L_f V(x) + L_g V(x) u \leq -\gamma(V(x))\}. \tag{2.9}$$

Combining (2.7) and (2.8) yields the following theorem for stabilization with CLFs:

Theorem 2.3. *For the affine control system* (2.6)*, given a control Lyapunov function V, any Lipschitz continuous feedback controller $u(x) \in K_{\text{clf}}(x)$ asymptotically stabilizes the system to $x^* = 0$.*

2.1.3 Control Barrier Functions

Unlike the control Lyapunov function which aims at stabilizing the system to a given state x^*, safety is formally guaranteed in the context of enforcing the invariance of a safe set. Motivated by the concept of barrier functions, the control objective is to find a feedback controller $u = k(x)$ for the affine control system (2.6) such that the safe set $\mathcal{C}$ is invariant, this goal is achieved if

$$\exists u = k(x) \text{ s.t. } \dot{b}(x, k(x)) + \alpha(b(x)) \geq 0, \tag{2.10}$$

where

$$\dot{b}(x, k(x)) = L_f b(x) + L_g b(x) k(x),$$

L_f and L_g denote the Lie derivatives along f and g respectively, α is an (extended) class $\mathcal{K}_\infty$ function, b is a barrier function defined in (2.4). (2.10) has a similar form as (2.7), which intuitively yields the establishment of control barrier function.

Definition 2.7. *(Control barrier function). (Ames et al., 2012) Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $b : D \to \mathbb{R}$, then b is a control barrier* *function (CBF) if there exists an extended class $\mathcal{K}_\infty$ function α such that for the control system* (2.6)*:*

$$\sup_{u \in U} \left[L_f b(x) + L_g b(x) u \right] \geq -\alpha(b(x)) \tag{2.11}$$

for all $x \in D$.

Thus, the set of all controllers that render $\mathcal{C}$ safe for every point $x \in D$ is described as:

$$K_{\text{cbf}}(x) := \{u \in U \ : \ L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0\}. \tag{2.12}$$

Combining (2.10) and (2.11) yields the following theorem enforcing safe forward invariance, i.e. safety:

Theorem 2.4. *(Ames et al., 2014a) For the affine control system* (2.6)*, given a control barrier function b, any Lipschitz continuous feedback controller $u(x) \in K_{\text{cbf}}(x)$ renders the set $\mathcal{C}$ forward invariant.*

2.2 Safety-Critical Control using Control Barrier Functions

In the context of safety-critical control, given a feedback controller $u = k(x)$ which may be unsafe, an optimization problem utilizing the control barrier functions is formulated to amend the feedback controller to guarantee safety(Freeman and Kokotovic, 2008; Ames et al., 2014a):

$$\begin{aligned} u(x) = \underset{u \in U}{\arg\min}\ & \frac{1}{2}\|u - k(x)\|^2 \\ \text{s.t. } & L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0 \end{aligned} \tag{2.13}$$

where $b(x)$ is the control barrier function (in many cases, the safety constraint) and we assume $U \in \mathbb{R}$. By mapping the initial safety constraint $b(x) \geq 0$ to the CBF constraint $L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0$, we constrain the controllers in the feasible set to render the safe set $\mathcal{C}$ forward invariant and thus guarantee safety. Note that (2.13) is Quadratic Program (QP) as it has a quadratic objective and a linear CBF constraint, thus can be solved efficiently. When there is no actuation constraint ($U \in \mathbb{R}$), this CBF-QP is always feasible, thus provides a safety-critical controller which has a minimum deviation from $k(x)$.

The performance of the controller acquired by solving the CBF-QP (2.13) greatly relies on the performance of the feedback controller $u = k(x)$ itself. To achieve both optimality and safety guarantees, the **joint Optimal Control and Control Barrier Function (OCBF)** approach is proposed (Xiao et al., 2021b) which has the following steps:

(i) Formulate an optimal control problem:

$$\begin{aligned}&\min_{u} \int_{t_0}^{t_f} J(x(t), u(t))dt \\ &\text{s.t. } \dot{x} = f(x) + g(x)u, \quad b(x) \geq 0\end{aligned} \tag{2.14}$$

where J is the objective, $\dot{x} = f(x) + g(x)u$ is the system dynamics and $b(x)$ is the safe constraint. The complete optimal control problem is usually hard to solve, especially when the safety constraint is complex.

(ii) Solve the optimal control problem with no active safety constraints or only some tractable constraints to achieve the (unsafe) optimal controller $u^*(t)$. Note that the objective of (2.14) is an integration along time, the solution to (2.14) is often a control trajectory $u(t)$ instead of a feedback controller $u(x)$, especially when there are boundary conditions.

(iii) Construct a controller that optimally tracks a reference trajectory $u_{\text{ref}}(t)$ subject to CBF constraints:

$$\begin{aligned}&u(t) = \arg\min_{u(t)\in U} \int_{t_0}^{t_f} \frac{1}{2} \|u(t) - u_{\text{ref}}(t)\|^2 dt \\ &\text{s.t. } L_f b(x) + L_g b(x)u + \alpha(b(x)) \geq 0\end{aligned} \tag{2.15}$$

Additionally, a control Lyapunov constraint can be added to (2.15) to stabilize the state to a reference state trajectory. The OCBF controller optimally tracks the reference trajectory by solving the optimization problem:

$$\begin{aligned} u(t) = \underset{u(t),e_i(t)}{\arg\min} \int\limits_{t_0}^{t_f} \left(\frac{1}{2} \| u(t) - u_{\text{ref}}(t) \|^2 + \beta e_i^2(t) \right) dt \\ \text{s.t. } L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0 \\ L_f V(x) + L_g V(x) u + \gamma(V(x)) \leq e_i(t) \end{aligned} \tag{2.16}$$

where $\varepsilon > 0$, and $e_i(t)$ is a relaxation variable which makes the constraint soft.

(iv) Solve problem (2.16) by discretizing $[t_i^0, t_i^f]$ into intervals of equal length Δt and solving (2.16) over each time interval $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t]$. The decision variables $u_i(t)$ and $e_i(t)$ are assumed to be constant over each such interval. They can be easily obtained by solving the following CLF-CBF-QP, since all CBF constraints are linear in the decision variables $u_i(t)$ and $e_i(t)$ (fixed over each interval $[t_i^k, t_i^k + \Delta t]$):

$$\begin{aligned} \min_{u_i(t),e_i(t)} \ & \beta e_i(t)^2 + \frac{1}{2}(u_i(t) - u_{\text{ref}}(t))^2 \\ \text{s.t. } & L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0 \\ & L_f V(x) + L_g V(x) u + \gamma(V(x)) \leq e_i(t) \\ & t = t_i^0 + k\Delta t \end{aligned} \tag{2.17}$$

Chapter 3

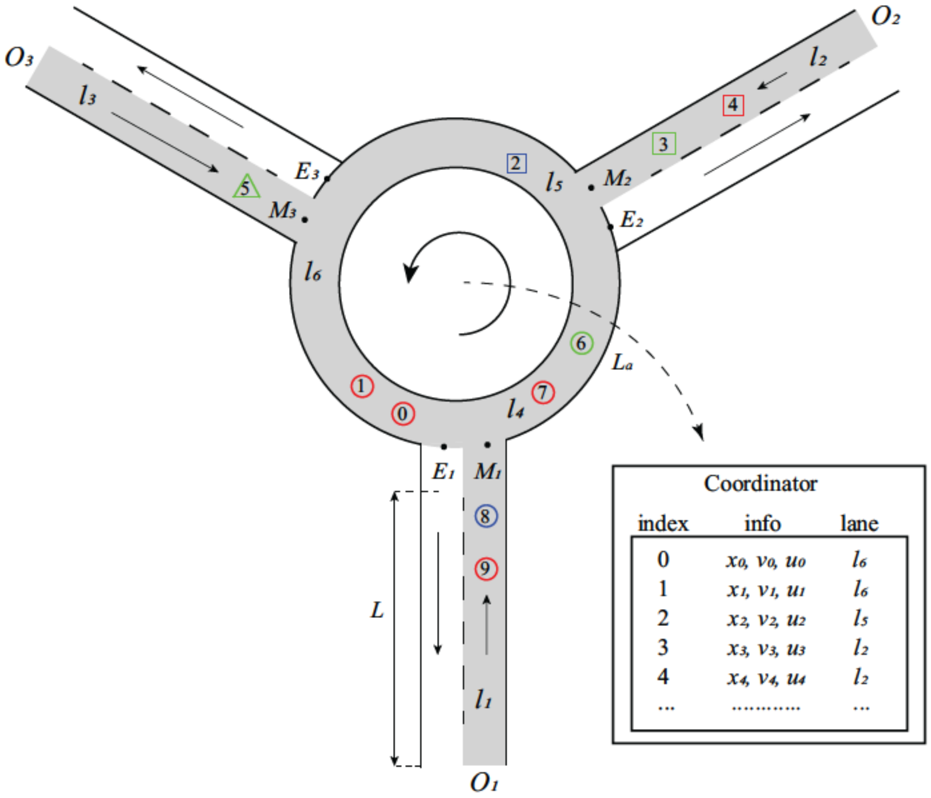
Roundabout Problem

In this chapter, the roundabout problem of controlling CAVs travelling through a roundabout is considered. The chapter starts with the problem formulation and proposes a general decentralized framework to control the CAVs. The solution to the unconstrained optimal problem is derived in details which is then combined with a joint optimal control and control barrier function (OCBF) controller to guarantee safety in real time. The effectiveness of the controller is validated through simulation experiments that compare the proposed controller to a baseline of human-driven vehicles.

3.1 Problem Formulation

We consider the roundabout model shown in Fig. 3·1 with 3 entries and 3 exits. Extending our analysis to more than 3 entry and exit points is straightforward. We consider the roundabout model shown in Fig. 3·1 with 3 entries and 3 exits. Extending our analysis to more than 3 entry and exit points is straightforward. We consider the case where all traffic consists of CAVs which randomly enter the roundabout from three different origin points O_1, O_2 and O_3 and have randomly assigned exit points E_1, E_2 and E_3. The gray road segments which include the circular part of the roundabout (within which we assume all CAVs move in a counterclockwise direction), together with the three entry roads, form the Control Zone (CZ) within which CAVs can share information and thus be automatically controlled. The entry road segments are connected with the circular part at the three Merging Points (MPs) where CAVs from different road segments may potentially collide with each other. These MPs are labeled M_1, M_2 and M_3.

We assume that each road segment has a single lane (extensions to multiple lanes and MPs are possible following natural extensions we have already applied for merging (Xiao et al., 2020) and for signal-free intersections (Xu et al., 2022b)). The three entry road segments which are labeled l_1, l_2 and l_3 have the same length L, while the road segments in the circular part which are labeled l_4, l_5 and l_6 have the same length L_a (extensions to different lengths are straightforward). In Fig. 3·1, a circle, square and triangle represent CAVs entering from O_1, O_2 and O_3 respectively. The color red, green and blue represents exiting from E_1, E_2 and E_3 respectively. The sequence of MPs that a CAV trajectory must include can be determined by its entry and exit points.



Coordinator

index	info	lane
0	x_0, v_0, u_0	l_6
1	x_1, v_1, u_1	l_6
2	x_2, v_2, u_2	l_5
3	x_3, v_3, u_3	l_2
4	x_4, v_4, u_4	l_2
...	....	...

Figure 3·1: CAVs randomly enter the roundabout from three different origin points O_1, O_2 and O_3 and have randomly assigned exit points E_1, E_2 and E_3. A circle, square and triangle represent CAVs entering from O_1, O_2 and O_3 respectively. The color red, green and blue represents exiting from E_1, E_2 and E_3 respectively.

A coordinator, i.e., a Road Side Unit (RSU) associated with the roundabout, records the information associated with each CAV and maintains a queue table (see Fig. 3·1) with CAVs orfered as they enter the CZ. The CAVs communicate with the coordinator but are not controlled by it; rather, control inputs are evaluated on-board each CAV in a *decentralized* way. Each CAV is assigned a unique index upon arrival at the CZ which is used to determine its passing order (i.e., the order in which CAVs go through MPs). The most common scheme for maintaining such a passing order is the First-In-First-Out (FIFO) policy based on each CAV's arrival time at the CZ. The FIFO policy is one of the simplest schemes, yet works well in many occasions as also shown in (Xu et al., 2022a). For simplicity, in what follows we use the FIFO policy to illustrate the construction of the coordinator queue table, but we point out that any passing order policy may be used, such as the Dynamic Resequencing (DR) method in (Zhang and Cassandras, 2018). We also introduce an alternative policy in Section 3.2.

Let $S(t)$ be the set of CAV indices in the coordinator queue table at time t whose cardinality is $N(t)$. When a new CAV arrives, it is allocated the index $N(t)+1$. Each time a CAV i leaves the CZ, it is dropped and all CAV indices larger than i decrease by one. When CAV $i \in S(t)$ is traveling in the roundabout, there are several important *events* whose times are used in our analysis: (i) CAV i enters the CZ at tme t_i^0, (ii) CAV i arrives at MP M_k at time t_i^k, $k \in \{1,2,3\}$, (iii) CAV i leaves the CZ at time t_i^f. Based on this setting, we can formulate an optimal control problem as described next.

Vehicle Dynamics Denote the distance from the origin $O_j, j \in \{1,2,3\}$ to the current location of CAV i along its trajectory as $x_i^j(t)$. However, since the CAV's unique identity i contains the information about the CAV's origin O_j, we can use $x_i(t)$ instead of $x_i^j(t)$ (without any loss of information) to describe the vehicle dynamics as

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ u_i(t) \end{bmatrix} \tag{3.1}$$

where v_i is the velocity of CAV i along its trajectory and u_i is the acceleration (control input). In this chapter, we focus on planning optimal trajectories that guarantee safety while ignoring lateral vehicle motion. The lateral vehicle offset can also be dealt with by using detailed models (e.g., the 7-state model in (Xiao et al., 2021c)) and auxiliary methods like MPC following the same framework as in the Appendix B.

Objective 1 Minimizing the travel time

$$J_{i,1} = t_i^f - t_i^0 \tag{3.2}$$

where t_i^0 and t_i^f are the times CAV i enters and exits the CZ respectively.

Objective 2 Minimizing energy consumption:

$$J_{i,2} = \int_{t_i^0}^{t_i^f} C_i(u_i(t))dt \tag{3.3}$$

where $C_i(\cdot)$ is a strictly increasing function of its argument. Since the energy consumption rate is a monotonic function of the acceleration, we adopt this general form to achieve energy efficiency.

Objective 3 Maximizing centrifugal comfort:

$$J_{i,3} = \int_{t_i^0}^{t_i^f} \kappa(x_i(t))v_i^2(t)dt \tag{3.4}$$

where $\kappa(x_i)$ is the curvature of the road at position x_i. As the aim is to minimize the centrifugal force of the vehicle, the curvature $\kappa(x_i)$ has the form of $\frac{1}{r(x_i)}$, where $r(x_i)$ is the radius of the road at position x_i. Specifically, if the road is a straight road segment, $\kappa(x_i) = 0$ which infers that $r(x_i) = \infty$.

Constraint 1 (Rear-end safety constraint) Let i_p denote the index of the CAV which immediately precedes CAV i on the same road segment as i, if one exists. The distance between i_p and i, defined as $z_{i,i_p}(t) \equiv x_{i_p}(t) - x_i(t)$, should satisfy a speed-dependent constraint:

$$z_{i,i_p}(t) \geq \varphi v_i(t) + \delta, \;\; \forall t \in [t_i^0, t_i^f], \;\; \forall i \in S(t) \tag{3.5}$$

where φ denotes the reaction time (as a rule, $\varphi = 1.8s$ is suggested, see (Vogel, 2003)), and δ denotes the minimum safe distance between CAVs (in general, we may use δ_i to make this distance CAV-dependent but will use a fixed δ for simplicity). Note that the preceding CAV index i_p may change after road segment changing events and is updated by the method described in section 3.2.2.

Constraint 2 (Safe merging constraint) Let t_i^k, $k \in \{1,2,3\}$ be the arrival time of CAV i at MP M_k. Let i_m denote the index of the CAV that CAV i may collide with when arriving at its next MP M_k. The distance between i_m and i, defined as $z_{i,i_m}(t) \equiv x_{i_m}(t) - x_i(t)$, is constrained by:

$$z_{i,i_m}(t_i^k) \geq \varphi v_i(t_i^k) + \delta, \;\; \forall i \in S(t), \;\; k \in \{1,2,3\} \tag{3.6}$$

where i_m can be determined and updated by the method described in section 3.2.2.

We note that the rear-end safety constraint and the safe merging constraint take the time-to-collision into consideration, which is the most prevalent indicator used to identify collisions at roundabouts as reported, for example, in (Pinnow et al., 2021).

Constraint 3 (Lateral safety constraint) The moment generated by the centrifugal force needs to be smaller than the one generated by gravity in order to avoid rollover:

$$\kappa(x_i(t))v_i^2(t)h \leq w_h g \tag{3.7}$$

where h is the height of the center of gravity of the CAV with respect to the ground, w_h is the half width of the CAV (for simplicity, both assumed the same for all CAVs) and g is the gravity constant. The lateral safety constraint is obtained though the Zero Moment Point (ZMP) (Sardain and Bessonnet, 2004) method that balances the gravity and centrifugal force of the CAV.

Constraint 4 (Vehicle limitations) The CAVs are also subject to velocity and acceleration constraints due to physical limitations or traffic rules:

$$\begin{aligned} v_{i,\min} \leq v_i(t) \leq v_{i,\max}, \forall t \in [t_i^0, t_i^f], \forall i \in S(t) \\ u_{i,\min} \leq u_i(t) \leq u_{i,\max}, \forall t \in [t_i^0, t_i^f], \forall i \in S(t) \end{aligned} \tag{3.8}$$

where $v_{i,\max} > 0$ and $v_{i,\min} \geq 0$ denote the maximum and minimum speed for CAV i, while $u_{i,\max} > 0$ and $u_{i,\min} < 0$ denote the maximum and minimum acceleration for CAV i. We further assume common speed limits dictated by the traffic rules at the roundabout, i.e. $v_{i,\min} = v_{\min}$, $v_{i,\max} = v_{\max}$.

Boundary conditions The initial position and velocity of the CAV as well as the terminal position are given as

$$x_i(t_i^0) = 0, v_i(t_i^0) = v_i^0, x_i(t_i^f) = L_i \tag{3.9}$$

where v_i^0 is the velocity of the CAV when entering the control zone CZ and L_i is the distance it needs to travel from its entry point to its assigned exit (L_i is determined once CAV i enters CZ.)

Similar to the previous work (Xiao and Cassandras, 2021b), we construct a convex combination of the three objectives above:

$$J_i = \alpha_1 J_{i,1} + \alpha_2 \frac{J_{i,2}}{\frac{1}{2}\max\{u_{\max}^2, u_{\min}^2\}} + \alpha_3 \frac{J_{i,3}}{\kappa_{\max} v_{\max}^2} \tag{3.10}$$

where $\alpha_1, \alpha_2, \alpha_3 \in [0,1]$, $\alpha_1 + \alpha_2 + \alpha_3 = 1$, $J_{i,2}$ and $J_{i,3}$ are both properly normalized. Note that these weights are determined by each CAV according to its owner's relative preferences among time, energy, and comfort; they are set upon a CAV's arrival and remain unchanged during the CAV's trip. In particular, by defining

$$\beta_1 \equiv \frac{\alpha_1}{2\alpha_2} \max\{u_{\max}^2, u_{\min}^2\}, \tag{3.11}$$

$$\beta_2 \equiv \frac{\alpha_3 \max\{u_{\max}^2, u_{\min}^2\}}{2\alpha_2 \kappa_{\max} v_{\max}^2}, \tag{3.12}$$

we can rewrite this minimization problem as

$$J_i(u_i) = \int_{t_i^0}^{t_i^f} \left(\beta_1 + \frac{1}{2}u_i^2(t) + \beta_2 \kappa(x_i(t)) v_i^2(t)\right) dt \tag{3.13}$$

where β_1 and β_2 are the weight factors derived from α_1 and α_2. Finally, in what follows we simply choose in $J_{i,2}$ the quadratic function $C_i(u_i) = \frac{1}{2}u_i^2(t)$. Then, we can formulate the optimal control problem as follows:

Problem 1: For each CAV i following the dynamics (3.1), find the optimal control input $u_i(t)$ and terminal time t_i^f that minimizes (3.13) subject to constraints (3.1), (3.5), (3.6), (3.7), (3.7), (3.8), boundary conditions (3.9) and given t_i^0.

3.2 Decentralized Control Framework

In order to solve **Problem 1** for each CAV i, we need to first determine the corresponding i_p and i_m (when they exist) required in the safety constraints (3.5) and (3.6). Compared to the single-lane merging or intersection control problems where the constraints are determined and fixed immediately when CAV i enters the CZ, the main difficulty in a roundabout is that these constraints generally change after every event (defined earlier). In particular, for each CAV i at time t only the merging constraint related to the *next* MP ahead is considered. In other words, we need to determine at most one i_p to enforce (3.5) and one i_m to enforce (3.6) at any time instant.

There are two different ways to deal with this problem: (i) Treat the system as a single CZ with three MPs with advance knowledge of each CAV's sequence of MPs, or (ii) Decompose the roundabout into three separate merging problems corresponding to the three MPs, each with a separate CZ. The first approach heavily relies on the CAV sequencing policy used. If FIFO is applied, it is likely to perform poorly in a large roundabout, because a new CAV may experience a large delay in order to preserve the global FIFO passing sequence. In contrast, the second approach allows us to make use of the solution to the optimal merging problem (Xiao and Cassandras, 2021b) for each MP separately; it may, however, cause congestion if a roundabout is too small to provide adequate space for effective control at each separate CZ associated with each MP.

In what follows, we first address the task of determining the indices i_p and i_m for every CAV i in an event-driven manner which can be used in either of the two approaches above and for any desired sequencing policy. An extended queue table, an example of which is shown in Table 3.1 corresponding to Fig. 3·1, is used to record the essential state information and identify all conflicting CAVs. We specify the state-updating mechanism for this queue table so as to determine for each CAV i the corresponding i_p and i_m. Then, we focus on the second approach introduced above, and Section IV develops a general algorithm for solving **Problem 1** based on the OCBF method (Xiao et al., 2021b).

3.2.1 The Extended Coordinator Queue Table

Starting with the coordinator queue table shown in Fig. 3·1, we extend it to include 6 additional columns for each CAV i. The precise definitions are given below:

Table 3.1: The extended coordinator queue table $S(t)$

$S(t)$								
idx	state	curr.	ori.	1st MP	2nd MP	3rd MP	i_p	i_m
0	$\boldsymbol{x}_0$	l_6	l_1	M_1, M	M_2, M	M_3, M		
1	$\boldsymbol{x}_1$	l_6	l_1	M_1, M	M_2, M	M_3, M	0	
2	$\boldsymbol{x}_2$	l_5	l_2	M_2, M				
3	$\boldsymbol{x}_3$	l_2	l_2	M_2	M_3	M_1		2
4	$\boldsymbol{x}_4$	l_2	l_2	M_2	M_3		3	
5	$\boldsymbol{x}_5$	l_3	l_3	M_3	M_1			1
6	$\boldsymbol{x}_6$	l_4	l_1	M_1, M				
7	$\boldsymbol{x}_7$	l_4	l_1	M_1, M	M_2	M_3	6	4
8	$\boldsymbol{x}_8$	l_1	l_1	M_1	M_2			7
9	$\boldsymbol{x}_9$	l_1	l_1	M_1	M_2	M_3	8	

- *idx*: Unique CAV index, which allows us to determine the order in which the CAV will leave the roundabout according to some policy (e.g., FIFO is assumed in Table I since rows are ordered by the index value).

- *state*: CAV state $\boldsymbol{x}_i = (x_i, v_i)$ where x_i denotes the distance from the entry point to the location of CAV i along its current road segment.

- *curr.*: Current CAV road segment, which allows us to determine the rear-end safety constraints.

- *ori.*: Original CAV road segment, which allows us to determine its relative position when in road segment *curr*.

- *1st-3rd MP*: These columns record all the MPs on the CAV trajectory. If a CAV has already passed an MP, this MP is marked with an "M". Otherwise, it is unmarked. This marker is used to systematically determine the safety constraints in Sec. 3.2.2. As a CAV may not need to go through all three MPs in the roundabout, some of these columns may be blank.

- i_p: Index of the CAV that immediately precedes CAV i in the same road segment (if such a CAV exists).

- i_m: Index of the CAV that may conflict with CAV i at the next MP. CAV i_m is the last CAV that passes the MP ahead of CAV i. Note that if i_m and i are in the same road segment, then i_m $(= i_p)$ is the immediately preceding CAV. In this case, the safe merging constraint is redundant and need not be included.

Event-driven Update Process for $S(t)$: The extended coordinator queue table $S(t)$ is updated whenever an event (as defined earlier) occurs. In particular:

- A new CAV enters the CZ: The CAV is indexed and added to the bottom of the queue table.

- CAV i exits the CZ: All information of CAV i is removed. All rows with index larger than i decrease their index values by 1.
- CAV i passes an MP: Mark the MP with "M" and update the current road segment value *curr* of CAV i with the one it is entering.

Following each event, the values of i_p and i_m are also updated as detailed next.

3.2.2 Determination of Safety Constraints

Recall that for each CAV i in the CZ, we need to consider two different safety constraints (3.5) and (3.6). First, by looking at each row $j < i$ and the corresponding current road segment value *curr*, CAV i can determine its immediately preceding CAV i_p if one exists. This fully specifies the rear-end safety constraint (3.5).

Next, we determine i_m, the CAV (if it exists) which possibly conflicts with CAV i at the next MP it will pass so as to specify the safe merging constraint (3.6). To do so, we find in the extended queue table the last CAV $j < i$ that will pass or has passed the same MP as CAV i. In addition, if such a CAV is in the same road segment as CAV i, it coincides with the preceding CAV i_p. As an example, in Table 3.1 (a snapshot of Fig. 3·1), CAV 8 has no immediate preceding CAV in l_1, but it needs to yield to CAV 7 at M_1, its next MP: although CAV 7 has already passed M_1, when CAV 8 arrives at M_1 there needs to be adequate space between CAV 7 and 8 for CAV 8 to enter l_4). On the other hand, CAV 9 only needs to satisfy its rear-end safety constraint with CAV 8.

It is now clear that we can use the information in $S(t)$ in a systematic way to determine both i_p in (3.5) and i_m in (3.6). Thus, there are two functions $i_p(e)$ and $i_m(e)$ which need to be updated after event e if this event affects CAV i. First, the index i_p can be easily determined by looking at rows $j < i$, starting at row i and moving up in the list, until the first one is found with the same value *curr* as CAV i. For example, CAV 9 searches for its i_p from CAV 8 to the top and sets $i_p = 8$ as CAV 8 has the *curr* value l_1.

Next, the index i_m is determined. To do this, CAV i compares its MP information to that of each CAV in rows $j < i$, starting at row i and moving up in the list. The process terminates the first time that any one of the following two conditions is satisfied:

- The MP information of CAV j *matches* CAV i. We define j to "match" i if and only if the last marked MP or the first unmarked MP of CAV j is the same as the first unmarked MP of CAV i. Thus, $i_m = j$.
- All prior rows $j < i$ have been looked up and none of them matches the MP information of CAV i.

Combining the two updating processes for i_p and i_m together, there are four different cases as follows:

1. Both i_p and i_m exist. In this case, there are two possibilities: (i) $i_p \neq i_m$. CAV i has to satisfy the safe merging constraint (3.6) with $i_m < i$ and also satisfy the rear-end safety constraint (3.5) with $i_p < i$. For example, for $i = 7$, we have $i_p = 6$ and $i_m = 4$ (M_2 is the first unmarked MP for CAV 7 and that matches the first unmarked MP for CAV 4). (ii) $i_p = i_m$. CAV i only has to follow i_p and satisfy the rear-end safety constraint (3.5) with respect to i_p. Thus, there is no safe merging constraint for CAV i to satisfy. For example, $i = 4$ and $i_p = i_m = 3$.

2. Only i_p exists. In this case, there is no safe merging constraint for CAV i to satisfy. CAV i only needs to follow the preceding CAV i_p and satisfy the rear-end safety constraint (3.5) with respect to i_p. For example, $i = 1$ and $i_p = 0$.

3. Only i_m exists. In this case, CAV i has to satisfy the safe merging constraint (3.6) with the CAV i_m in $S(t)$. There is no preceding CAV i_p, thus there is no rear-end safety constraint. For example, $i = 5$ and $i_m = 1$ (M_3 is the first unmarked MP for CAV 5 and that matches the last marked MP for CAV 1 with no other match for $j = 4,3,2$).

4. Neither i_p nor i_m exists. In this case, CAV i does not have to consider any safety constraints. For example, $i = 2$.

3.2.3 Sequencing Policies Using Local Coordinator Queue Tables

The extended queue table $S(t)$ is based on a sequencing policy which applies to the whole roundabout. Thus, if a FIFO policy is adopted, we have seen how to use a systematic process for updating $i_p(e)$ and $i_m(e)$ after each event e in $S(t)$. However, FIFO may not be a good sequencing policy if applied to the whole roundabout. More generally, we wish to allow possible resequencing after a CAV passes a MP, based on the system state information at that time. This can be accomplished by introducing a *local* coordinator queue table $S_k(t)$ associated with each M_k, $k = 1,2,3$. This allows us to treat the problem of coordinating all CAVs with M_k as their next (or just passed) MP as a separate optimal merging control problem along the lines of (Xiao and Cassandras, 2021b). We define CZ$_k$ as the CZ corresponding to M_k that consists of the three road segments directly connected to M_k. A local coordinator queue table can be viewed as a subset of the extended coordinator queue table except that the CAVs appear in a different order in the two tables. As an example, Table 3.2 (a snapshot of Fig. 3·1) is the local coordinator queue table corresponding to M_1 (in this case, for simplicity, we still use the FIFO policy, but it is now applied only to CAVs involved with M_1).

Table 3.2: The local-coordinator queue table $S_1(t)$

$S_1(t)$								
idx	info	curr.	ori.	1st MP	2nd MP	3rd MP	i_p	i_m
6	$\boldsymbol{x}_6$	l_4	l_1	M_1, M				
7	$\boldsymbol{x}_7$	l_4	l_1	M_1, M	M_2	M_3	6	
8	$\boldsymbol{x}_8$	l_1	l_1	M_1	M_2			7
0	$\boldsymbol{x}_0$	l_6	l_1	M_1, M	M_2, M	M_3, M		
1	$\boldsymbol{x}_1$	l_6	l_1	M_1, M	M_2, M	M_3, M	0	
9	$\boldsymbol{x}_9$	l_1	l_1	M_1	M_2	M_3	8	

Event-driven Update Process for $S_k(t)$: The local-coordinator queue table $S_k(t)$ is updated as follows after each event that has caused an update of the extended coordinator queue table $S(t)$:

- For each CAV i in $S_k(t)$, update its information depending on the event observed: (i)

A new CAV j enters CZ$_k$ (either from an entry point to the roundabout CZ or a MP passing event): Add a new row to $S_k(t)$ and resequence the local-coordinator queue table according to the sequencing policy used. (ii) CAV j exits CZ$_k$: Remove all the information of CAV j from $S_k(t)$.

- Determine i_p and i_m in each local-coordinator queue table with the same method as described in section 3.2.2.
- Update CAV j's i_p and i_m in the extended coordinator queue table with the corresponding information in $S_k(t)$, while M_k is the next MP of CAV j.

Note that CAV j may appear in multiple local-coordinator queue tables with different i_p and i_m values. However, only the one in $S_k(t)$ where M_k is the next MP CAV j will pass is used to update the extended coordinator queue table $S(t)$. The information of CAV j in other local-coordinator queue tables is necessary for determining the safety constraints as CAV j may become CAV i_p or i_m of other CAVs.

Resequencing policy: The local-coordinator queue table allows resequencing when a CAV passes a MP. A resequencing policy evaluates a given criterion for each CAV and sorts the CAVs in the queue table when a new event happens. For example, FIFO takes the arrival time in the CZ as the criterion, while the Dynamic Resequencing (DR) policy(Zhang and Cassandras, 2018) uses the overall objective value in (3.13) as the criterion.

We propose here a straightforward yet effective resequencing policy for the roundabout as follows. Let $\tilde{x}_i^k \equiv x_i - d_j^k$ be the position of CAV i relative to M_k, where d_j^k denotes the fixed distance from the entry point (origin) O_j to merging point M_k along the trajectory of CAV i. Then, consider

$$y_i(t) = -\tilde{x}_i^k(t) \tag{3.14}$$

This resequencing criterion reflects the distance between the CAV and the next MP. The CAV which has the smallest $y_i(t)$ value is allocated first, thus this is referred to as the

Shortest Distance First (SDF) policy. This simple resequencing policy is tested in Section 3.5. Other policies can also be easily implemented with the help of the local-coordinator queue tables.

3.3 Unconstrained Optimal Control Solution

We now return to the solution of **Problem 1**, i.e., the minimization of (3.13) subject to constraints (3.1), (3.5), (3.6), (3.8), (3.7), the initial condition $x_i(t_i^0) = 0$, and given t_i^0, v_i^0 and $x_i(t_i^f)$. The problem formulation is complete since we have used the local coordinator queue tables to determine i_p and i_m (needed for the safety constraints) associated with the closest MP to CAV i.

As pointed out in (Xiao and Cassandras, 2021b), when one or more constraints become active, the solution to **Problem 1** becomes computationally intensive. The problem here is exacerbated by the fact that the values of i_p and i_m dynamically change due to different events in the roundabout system. To ensure that a solution can be obtained in real time while also *guaranteeing that all safety constraints are always satisfied*, we adopt the OCBF approach (Xiao et al., 2021b) briefly introduced in the introduction and further discussed in Section 3.4: we first determine the solution of the *unconstrained* optimal control problem and then solve a problem of optimally tracking this solution while guaranteeing the satisfaction of all constraints through the use of Control Barrier Functions (CBFs).

Thus, our first task is to obtain a solution to the unconstrained roundabout problem through Hamiltonian analysis. Denoting by $\mathcal{X}_i(t) := (x_i(t), v_i(t))$ and $\lambda_i(t) := (\lambda_i^x(t), \lambda_i^v(t))$

the state vector and costate vector respectively, the Hamiltonian of the system in (3.13) is

$$\begin{aligned}\mathcal{H}_i(X_i,\lambda_i,u_i) =&\beta_1+\frac{1}{2}u_i^2+\beta_2\kappa(x_i)v_i^2+\lambda_i^x v_i+\lambda_i^v u_i\\ &+\mu_i^a(u_i-u_{\max})+\mu_i^b(u_{\min}-u_i)\\ &+\mu_i^c(v_i-v_{\max})+\mu_i^d(v_{\min}-v_i)\\ &+\mu_i^e(x_i+\varphi v_i+\delta-x_{i_p})\\ &+\mu_i^f(\kappa(x_i)v_i^2 h-w_h g)\end{aligned} \tag{3.15}$$

where $\mu_i^a,\mu_i^b,\mu_i^c,\mu_i^d,\mu_i^e,\mu_i^f$ are Lagrange multipliers associated with the constraints (3.5), (3.6), (3.8) and (3.7). Since the terminal state constraint $\psi_{i,1} := x_i(t_i^f)-L_i=0$ is not an explicit function of time, the transversality condition is

$$H_i(X_i(t),\lambda_i(t),u_i(t))|_{t=t_i^f}=0 \tag{3.16}$$

The necessary conditions for optimality are

$$\dot{\lambda}_i^x=-\frac{\partial\mathcal{H}_i}{\partial x_i}=-\mu_i^e-\beta_2\frac{\partial\kappa(x_i)}{\partial x_i}v_i^2-\mu_i^f\frac{\partial\kappa(x_i)}{\partial x_i}v_i^2 \tag{3.17}$$

$$\begin{aligned}\dot{\lambda}_i^v=-\frac{\partial\mathcal{H}_i}{\partial v_i}=&-2\beta_2\kappa(x_i)v_i-\lambda_i^x-(\mu_i^c-\mu_i^d)\\ &-\varphi\mu_i^e-2\kappa(x_i)\mu_i^f v_i\end{aligned} \tag{3.18}$$

$$0=\frac{\partial\mathcal{H}_i}{\partial u_i}=u_i+\lambda_i^v+\mu_i^a-\mu_i^b \tag{3.19}$$

Under the unconstrained assumption, none of the constraints (3.5), (3.6), (3.8) and (3.7) is active, therefore, $\mu_i^a=\mu_i^b=\mu_i^c=\mu_i^d=\mu_i^e=\mu_i^f=0$. Then, the most complex part in solving the equations above is due to $\frac{\partial\kappa(x_i)}{\partial x_i}$ appearing in (3.17). In (Xiao and Cassandras, 2021a), a single curved road segment was modeled by assuming $\kappa(x_i)$ to be a constant taken to be the average curvature $\bar{\kappa}$ or the maximum curvature $\kappa_{\max}$. However, in a roundabout road configuration, we cannot take $\kappa(x_i)$ to be a constant over an entire CAV trajectory, but rather define it as a piecewise constant function which is 0 when $x_i \in [0,L)$ and jumps to $1/r$ at $x_i = L$. This discontinuity in $\kappa(x_i)$ causes complications in (3.17).

There are several ways to deal with this discontinuity problem in $\kappa(x_i)$. For example, one can approximate it through a smooth (e.g., sigmoid) function. However, even the simplest sigmoid function results in a set of complex nonlinear equations too hard to solve in real time. In what follows, we resolve this issue by transforming **Problem 1** into a bi-level optimal control problem as described next.

At the lower level, we separate the problem for each CAV i into two parts: one for the straight road segment and one for the circular part. Both problems are parameterized by the terminal speed of CAV i, v_i^m, at the end of the straight road segment, which is also the initial speed for the circular part. Then, the upper level problem consists of determining an optimal value for the parameter v_i^m.

To formulate the two lower-level problems, let t_i^m be the time a CAV enters the circular part of the roundabout from the straight road segment. The boundary conditions for the speed of CAV i when entering the straight road segment and the circular part are v_i^0 and v_i^m respectively. Then, we formulate the two *lower-level problems* corresponding to the straight line and the circular part (both parameterized by v_i^m) as follows:

$$\begin{aligned}
&\min_{u_i(t)} J_i^{\mathrm{S}}(u_i(t); v_i^m) = \int_{t_i^0}^{t_i^m} f_i^{\mathrm{S}}(u_i(t))dt \\
&s.t. \;\; (3.1),(3.5),(3.6),(3.8) \\
&\qquad v_i(t_i^0) = v_i^0, v_i(t_i^m) = v_i^m \\
&\qquad x_i(t_i^m) = 0, x_i(t_i^f) = L_{i,1}
\end{aligned} \tag{3.20}$$

$$\begin{aligned}
&\min_{u_i(t)} J_i^{\mathrm{C}}(u_i(t);v_i^m) = \int_{t_i^m}^{t_i^f} f_i^{\mathrm{C}}(u_i(t),\boldsymbol{v}_i(t))dt \\
&s.t.\;\; (3.1),(3.5),(3.6),(3.8),(3.7) \\
&\qquad v_i(t_i^m) = v_i^m \\
&\qquad x_i(t_i^m) = L_{i,1}, x_i(t_i^f) = L_i
\end{aligned} \tag{3.21}$$

where, for notational simplicity, we have defined

$$f_i^{\mathrm{S}}(u_i(t)) = \beta_1 + \frac{1}{2}u_i^2(t)$$

$$f_i^{\mathrm{C}}(u_i(t),\boldsymbol{v}_i(t)) = \beta_1 + \frac{1}{2}u_i^2(t) + \beta_2\kappa v_i^2(t)$$

Note that f_i^{S} and f_i^{C} are both special cases of the integrand in (3.13). In f_i^{S} the curvature is $\kappa(x_i) = 0$, while in f_i^{C} the curvature is a constant $\kappa(x_i) = \hat{\kappa}$. Given the speed parameter v_i^m, solving the lower level problems yields two optimal costs $J_i^{*\mathrm{S}}(v_i^m)$ and $J_i^{*\mathrm{C}}(v_i^m)$, both functions of v_i^m. We then formulate the following *upper level problem* which aims at finding the optimal terminal velocity v_i^m:

$$\begin{aligned}
&\min_{v_i^m} J_i(v_i^m) = J_i^{*\mathrm{S}}(v_i^m) + J_i^{*\mathrm{C}}(v_i^m) \\
&s.t.\;\; v_{\min} \leq v_i^m \leq v_{\max}
\end{aligned} \tag{3.22}$$

3.3.1 Lower level problem 1 – Circular road segment

The circular part problem is a special case of **Problem 1** where the curvature $\kappa(x_i)$ is constant, i.e. $\kappa = \hat{\kappa}$. Under the unconstrained assumption, $\mu_i^a = \mu_i^b = \mu_i^c = \mu_i^d = \mu_i^e = \mu_i^f = 0$

and the necessary conditions (3.17), (3.18), (3.19) yield:

$$\dot{\lambda}_i^x = -\frac{\partial \mathcal{H}_i}{\partial x_i} = 0 \tag{3.23}$$

$$\dot{\lambda}_i^v = -\frac{\partial \mathcal{H}_i}{\partial v_i} = -2\beta_2\hat{\kappa}v_i - \lambda_i^x \tag{3.24}$$

$$0 = \frac{\partial \mathcal{H}_i}{\partial u_i} = u_i + \lambda_i^v \tag{3.25}$$

Since (3.23) implies that λ_i^x is a constant, set $\lambda_i^x = a_i$. Then, combining (3.24) and (3.25) yields

$$\ddot{v}_i - 2\beta_2\hat{\kappa}v_i - a_i = 0 \tag{3.26}$$

Solving this equation gives an explicit solution for the speed:

$$v_i^*(t) = b_i e^{\sqrt{2\beta_2\hat{\kappa}}t} + c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t} - \frac{a_i}{2\beta_2\hat{\kappa}} \tag{3.27}$$

where b_i, c_i are integration constants. Applying (3.1), the optimal solution for the unconstrained problem is obtained as follows:

$$u_i^*(t) = \sqrt{2\beta_2\kappa}(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t} - c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t}) \tag{3.28}$$

$$x_i^*(t) = \frac{1}{\sqrt{2\beta_2\kappa}}(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t} - c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t}) - \frac{a_i}{2\beta_2\kappa}t + d_i \tag{3.29}$$

where d_i is also an integration constant.

Given the boundary conditions $v_i(t_i^m) = v_i^m$, $x_i(t_i^m) = L_{i,1}$, $\lambda_i^v(t_i^f) = 0$, $x_i(t_i^f) = L_i$, as well as the transversality condition (3.16), we can obtain a_i, b_i, c_i, d_i and t_i^f by solving the set of nonlinear algebraic equations:

$$\begin{aligned}
&b_i e^{\sqrt{2\beta_2\hat{\kappa}}t_i^m} + c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^m} - \frac{a_i}{2\beta_2\hat{\kappa}} = v_i^m \\
&\frac{1}{\sqrt{2\beta_2\hat{\kappa}}}(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t_i^m} - c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^m}) - \frac{a_i}{2\beta_2\kappa}t_i^m + d_i = L_{i,1} \\
&\frac{1}{\sqrt{2\beta_2\hat{\kappa}}}(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t_i^f} - c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^f}) - \frac{a_i}{2\beta_2\kappa}t_i^f + d_i = L_i \\
&\sqrt{2\beta_2\hat{\kappa}}(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t_i^f} - c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^f}) = 0 \\
&\beta_1 + \beta_2\hat{\kappa}(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t_i^f} + c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^f} - \frac{a_i}{2\beta_2\hat{\kappa}})^2 \\
&\qquad + a_i(b_i e^{\sqrt{2\beta_2\hat{\kappa}}t_i^f} + c_i e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^f} - \frac{a_i}{2\beta_2\hat{\kappa}}) = 0
\end{aligned} \tag{3.30}$$

Thus, by solving (3.30) for each $i \in S(t)$, we can obtain all the integration constants a_i, b_i, c_i, d_i and the terminal time t_i^f for CAV i as a function of v_i^m. Since (3.30) is usually hard to solve due to the presence of the exponential terms, we will present in Section 3.3.3 a computationally efficient approach to calculate these five constants, as well as the optimal cost.

3.3.2 Lower level problem 2 – Straight road segment

In this case, the Hamiltonian in (3.15) becomes

$$\begin{aligned}
\mathcal{H}_i(\mathcal{X}_i, \lambda_i, u_i) =& \beta_1 + \frac{1}{2}u_i^2 + \lambda_i^x v_i + \lambda_i^v u_i \\
& + \mu_i^a(u_i - u_{\max}) + \mu_i^b(u_{\min} - u_i) \\
& + \mu_i^c(v_i - v_{\max}) + \mu_i^d(v_{\min} - v_i) \\
& + \mu_i^e(x_i + \varphi v_i + \delta - x_{i_p})
\end{aligned} \tag{3.31}$$

with the transverality condition

$$\mathcal{H}_i(\mathcal{X}_i, \lambda_i, u_i)\big|_{t=t_i^m} = 0 \tag{3.32}$$

The necessary conditions for optimality are

$$\dot{\lambda}_i^x = -\frac{\partial \mathcal{H}_i}{\partial x_i} = -\mu_i^e \tag{3.33}$$

$$\dot{\lambda}_i^v = -\frac{\partial \mathcal{H}_i}{\partial v_i} = -\lambda_i^x - (\mu_i^c - \mu_i^d) - \varphi \mu_i^e \tag{3.34}$$

$$0 = \frac{\partial \mathcal{H}_i}{\partial u_i} = u_i + \lambda_i^v + \mu_i^a - \mu_i^b \tag{3.35}$$

Again, under the unconstrained assumption, $\mu_i^a = \mu_i^b = \mu_i^c = \mu_i^d = \mu_i^e = 0$. Therefore, solving the equations above, we can explicitly obtain the optimal solution to the unconstrained problem as:

$$\begin{aligned} u_i^*(t) &= a_i t + b_i \\ v_i^*(t) &= \frac{1}{2} a_i t^2 + b_i t + c_i \\ x_i^*(t) &= \frac{1}{6} a_i t^3 + \frac{1}{2} b_i t^2 + c_i t + d_i \end{aligned} \tag{3.36}$$

The boundary conditions for this problem are $v_i(t_i^0) = v_i^0$, $v_i(t_i^m) = v_i^m$, $x_i(t_i^0) = 0$ and $x_i(t_i^m) = L_{i,1}$. Combining these boundary conditions with the tranversality condition (3.32) yields the following set of equations from which all the integration constants a_i, b_i, c_i, d_i and the terminal time t_i^m can be obtained, again as a function of v_i^m:

$$\begin{aligned} &\frac{1}{2} a_i \cdot (t_i^0)^2 + b_i \cdot t_i^0 + c_i = v_i^0, \\ &\frac{1}{2} a_i \cdot (t_i^m)^2 + b_i \cdot t_i^0 + c_i = v_i^m, \\ &\frac{1}{6} a_i \cdot (t_i^0)^3 + \frac{1}{2} b_i \cdot (t_i^0)^2 + c_i t_i^0 + d_i = 0, \\ &\frac{1}{6} a_i \cdot (t_i^m)^3 + \frac{1}{2} b_i \cdot (t_i^m)^2 + c_i t_i^m + d_i = L, \\ &\beta - \frac{1}{2} b_i^2 + a_i c_i = 0. \end{aligned} \tag{3.37}$$

This set equations is not difficult to solve; in practice, the solution can be obtained within $\ll$ 1s using MATLAB.

3.3.3 The upper level problem

Once the solution $u_i^*(t)$ in (3.36) is obtained in conjunction with (3.37), we have the optimal cost $J_i^{*\mathrm{S}}(v_i^m)$ available in (3.22). Similarly, once $v_i^*(t)$ and $u_i^*(t)$ in (3.27) and (3.28) are obtained in conjunction with (3.30), then $J_i^{*\mathrm{C}}(v_i^m)$ is available and (3.22) becomes a standard nonlinear programming problem whose solution gives an optimal v_i^m. The difficulty, however, is that explicit analytical expressions for $J_i^{*\mathrm{S}}(v_i^m)$ and $J_i^{*\mathrm{C}}(v_i^m)$ are generally unavailable. The numerical evaluation of these two terms, given a particular v_i^m value, is computationally expensive and this is especially true for $J_i^{*\mathrm{C}}(v_i^m)$ where exponential terms are involved. A straightforward and intuitive way to solve the upper level problem in real time is to use a regression approach to obtain explicit expressions for the two optimal cost functions. This provides a computationally efficient solution at the expense of some accuracy in determining the optimal v_i^m.

What is important to note is that the aforementioned regression approach can be carried out *off line*. To explain this, let us consider the first lower-level problem for the circular part. For any CAV i with a given origin and destination, the physical parameter values of $L_{i,1}$, L_i and $\hat{\kappa}$ in (3.21) are fixed. Therefore, given the parameters β_1, β_2, for any fixed v_i^m, a shift in the arrival time t_i^m will only result in a shift in the optimal control but does not influence the optimal travel time or the optimal cost. In other words, for $i, j \in S(t)$, if $v_i^m = v_j^m$ and $\Delta t = t_j^m - t_i^m$, we have $u_i^*(t) = u_j^*(t+\Delta t)$ as well as $t_i^f - t_i^m = t_j^f - t_j^m$. It follows that

$$\begin{aligned} J_i^{*\mathrm{C}}(v_i^m) &= \int_{t_i^m}^{t_i^f} f_i^{\mathrm{C}}(u_i^*(t), \boldsymbol{x}_i^*(t))dt \\ &= \int_{t_i^m+\Delta t}^{t_i^f+\Delta t} f_j^{\mathrm{C}}(u_j^*(t), \boldsymbol{x}_j^*(t))dt = J_j^{*,\mathrm{C}}(v_j^m) \end{aligned} \tag{3.38}$$

where $f_i^{\mathrm{C}}(u(t),x(t)) = f_j^{\mathrm{C}}(u(t),x(t))$ for given β_1 and β_2. This implies that the optimal cost function $J_i^{*,\mathrm{C}}(v_i^m)$ is independent of the initial time t_i^m. Therefore, given β_1, β_2 for a roundabout with physical parameters $\hat{\kappa}, L_{i,1}$ and L_i, we can get the optimal cost $J_i^{*\mathrm{C}}(v_i^m)$ for any fixed $v_i^m \in [v_i^{\min}, v_i^{\max}]$ by solving (3.30) off line with $t_i^m = 0$. Following this approach, we can calculate a number of optimal cost and initial speed pairs $(J_i^{*\mathrm{C}}(v_i^m), v_i^m)$ off line and use any standard regression method to fit the optimal cost function:

$$\mathcal{R}_J^C(v_i^m) = J_i^{*\mathrm{C}}(v_i^m) + \varepsilon(v_i^m) \tag{3.39}$$

where $\mathcal{R}_J^C : \mathbb{R} \to \mathbb{R}$ denotes the regression model for the optimal cost, and ε denotes the regression error.

Using the same technique, another regression model $\mathcal{R}_J^S(v_i^m)$ can be calculated off line as an approximation to the optimal cost function $J_i^{*\mathrm{S}}(v_i^m)$ for CAV i in the straight road segment. Thus, the objective function of the upper level problem can be explicitly expressed by the regression model:

$$J_i(v_i^m) = \mathcal{R}_J^S(v_i^m) + \mathcal{R}_J^C(v_i^m) \tag{3.40}$$

Then, the upper-level problem becomes a nonlinear programming problem with an explicit objective function. This is usually easy to solve and the optimal v_i^m is readily obtained in real time.

Once the optimal velocity v_i^m is determined, the explicit solution of the two lower-level optimal control problems can be obtained by solving (3.30) and (3.37) respectively. Regarding (3.30), as mentioned in Section 3.3.1, a solution can be computationally expensive to obtain due to the exponential terms involved. To accelerate this solution process, we can eliminate the variable t_i^f in (3.30), hence obtaining a much simpler system of only four equations to solve; this is accomplished by employing a similar regression technique for t_i^f in terms of v_i^m as explained next.

As already pointed out, for a specific roundabout where $L_{i,1}$ and L_i are fixed, the total travel time $t_i^f - t_i^m$ is independent of the arrival time t_i^m. Thus, given β_1, β_2 and the physical parameters $\hat{\kappa}, L_{i,1}$ and L_i, we can calculate a number of travel time and initial speed pairs $(t_i^f - t_i^m, v_i^m)$ off line for any fixed v_i^m. A regression model $\mathcal{R}_t^C(v_i^m)$ is then used to fit the travel time as follows:

$$\mathcal{R}_t^C(v_i^m) = (t_i^f - t_i^m) + \varepsilon(v_i^m) \tag{3.41}$$

where $\mathcal{R}_t : \mathbb{R} \to \mathbb{R}$ denotes the regression model for the optimal travel time and $\varepsilon(v_i^m)$ represents the regression error. Thus, for any observed initial velocity $v_i^m \in [v_{\min}, v_{\max}]$, we can use this regression model to obtain the solution of t_i^f:

$$t_i^f = t_i^m + \mathcal{R}_t^C(v_i^m) \tag{3.42}$$

Given β_1, β_2 and v_i^m and the regression model $\mathcal{R}_t^C(v_i^m)$, the optimal terminal time can be immediately obtained using (3.42). Therefore, the problem (3.21) is reduced into an optimal control problem with fixed terminal time. The transversality condition (the last equation in (3.30)) is no longer needed. Then, the integration constants can be obtained easily and quickly through a simple matrix multiplication:

$$\begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} -\frac{1}{2\beta_2\hat{\kappa}} & e^{\sqrt{2\beta_2\hat{\kappa}}t_i^m} & e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^m} & 0 \\ -\frac{1}{2\beta_2\hat{\kappa}} & \frac{e^{\sqrt{2\beta_2\hat{\kappa}}t_i^m}}{2\beta_2\hat{\kappa}} & -\frac{e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^m}}{2\beta_2\hat{\kappa}} & 1 \\ -\frac{1}{2\beta_2\hat{\kappa}} & \frac{e^{\sqrt{2\beta_2\hat{\kappa}}t_i^f}}{2\beta_2\hat{\kappa}} & -\frac{e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^f}}{2\beta_2\hat{\kappa}} & 1 \\ 0 & e^{\sqrt{2\beta_2\hat{\kappa}}t_i^f} & e^{-\sqrt{2\beta_2\hat{\kappa}}t_i^f} & 0 \end{bmatrix}^{-1} \begin{bmatrix} v_i^m \\ L_{i,1} \\ L_i \\ 0 \end{bmatrix} \tag{3.43}$$

Although (3.37) for the second lower-level problem is easy to solve, the solution can still be accelerated by a similar regression approach: a number of travel time pairs $(t_i^m - t_i^0, v_i^m)$ can be calculated off line to generate a regression model:

$$\mathcal{R}_t^S(v_i^m) = (t_i^m - t_i^0) + \varepsilon(v_i^m) \tag{3.44}$$

which efficiently calculates the optimal terminal time. This reduces (3.37) to only four equations which are easily and quickly solved by a matrix multiplication:

$$\begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(t_i^0)^2 & t_i^0 & 1 & 0 \\ \frac{1}{2}(t_i^m)^2 & t_i^m & 1 & 0 \\ \frac{1}{6}(t_i^0)^3 & \frac{1}{2}(t_i^0)^2 & t_i^0 & 1 \\ \frac{1}{6}(t_i^m)^3 & \frac{1}{2}(t_i^m)^2 & t_i^m & 1 \end{bmatrix}^{-1} \begin{bmatrix} v_i^0 \\ v_i^m \\ 0 \\ L \end{bmatrix} \tag{3.45}$$

3.3.4 Bi-level optimal control problem transformation

Using the regression technique introduced in Section 3.3.3, the bi-level problem for an explicit solution of the unconstrained optimal control problem can be solved efficiently in real time at the expense of some accuracy due to regression errors. We can further simplify this process by transforming the bi-level problem into the following optimal control problem:

$$\begin{aligned} \min_{u_i(t)} J_i(u_i(t)) = & \int_{t_i^0}^{t_i^m} f_i^{\mathrm{S}}(u_i(t), \boldsymbol{x}_i(t))dt + R_J^C(v_i(t_i^m)) \\ s.t.\ \ & (3.1), (3.5), (3.6), (3.8) \\ & v_i(t_i^0) = v_i^0 \\ & x_i(t_i^0) = 0, x_i(t_i^m) = L_{i,1} \end{aligned} \tag{3.46}$$

This problem has a similar form to (3.20) except that it has a terminal state cost term $R_J^C(v_i(t_i^m))$ reflecting the performance of the terminal state under an undetermined terminal velocity. The advantage of this alternative problem formulation is that only one regression model, $R_J^C(v_i(t_i^m))$, needs to be pre-calculated, thus reducing the overall regression error. The Hamiltonian analysis to this problem is similar to that in Section 3.3.2 which, under the unconstrained assumption, generates the same form of optimal control as (3.36).

As the term $\mathcal{R}_J^C(v_i(t_i^m))$ is included, the transversality condition and the boundary condition of (3.46) differ from (3.22). The transversality condition is

$$\mathcal{H}_i(\mathcal{X}_i, \lambda_i, u_i)\big|_{t=t_i^m} = \frac{\partial R_J^C(v_i(t))}{\partial t}\bigg|_{t=t_i^m} \tag{3.47}$$

with costate boundary condition

$$\lambda_i^v(t_i^m) = \left.\frac{\partial R_J^C(v_i(t))}{\partial v_i}\right|_{v_i = v_i(t_i^m)}. \tag{3.48}$$

Combining the boundary conditions $v_i(t_i^0) = v_i^0, x_i(t_i^0) = 0, x_i(t_i^m) = L_{i,1}$ and (3.48) with the transversality condition (3.47) yields the following equations from which we can solve for all the integration constants a_i, b_i, c_i, d_i and the terminal time t_i^m:

$$\begin{aligned}
&\frac{1}{2}a_i \cdot (t_i^0)^2 + b_i \cdot t_i^0 + c_i = v_i^0, \\
&-a_i t_i^m - b_i = \left.\frac{\partial R_J^C(v_i(t))}{\partial v_i}\right|_{v_i = \frac{1}{2}a_i \cdot (t_i^m)^2 + b_i t_i^m + c_i}, \\
&\frac{1}{6}a_i \cdot (t_i^0)^3 + \frac{1}{2}b_i \cdot (t_i^0)^2 + c_i t_i^0 + d_i = 0, \\
&\frac{1}{6}a_i \cdot (t_i^m)^3 + \frac{1}{2}b_i \cdot (t_i^m)^2 + c_i t_i^m + d_i = L, \\
&\beta - \frac{1}{2}b_i^2 + a_i c_i = \left.\frac{\partial R_J^C(v_i(t))}{\partial t}\right|_{t = t_i^m}.
\end{aligned} \tag{3.49}$$

This set of equations is usually easy to solve if the regression model is not too complex (e.g., using polynomial regression). Hence, the whole problem can be solved efficiently with only the approximation error of $R_J^C(v_i^m)$ included. We emphasize that this is only an alternative to the full bi-level problem solution which can reduce the computational cost for determining the unconstrained optimal solution needed for the OCBF approach described in Section 3.4.

3.3.5 Example

The purpose of this example is to illustrate the difference between the bi-level optimal control approach (or the simpler transformed version in Section 3.3.4) to the alternative of separating the roundabout into two segments where the optimal control for a CAV in each segment is calculated and applied every time the CAV enters a new segment.

Consider a CAV entering a roundabout which consists of a 100m straight road segment and a circular road segment with radius 25m. Set $\alpha_1 = 0.2, \alpha_2 = 0.2, \kappa = 1/25, L = 50\pi, u_{\max} = 5m/s^2, v_{\max} = 20m/s$. The cost of the bi-level approach in which we deal with the roundabout as a whole is 421.2, lower than 435.9, the cost of the method which separates the roundabout into two segments. The corresponding velocity trajectories are shown in Fig. 5. We remind the reader that these are obtained under the assumption that no constraint is active. Note that in the latter approach the speed violates the constraint $v_{\max} = 20m/s$ and there is a significant jerk as the CAV enters the circular part.

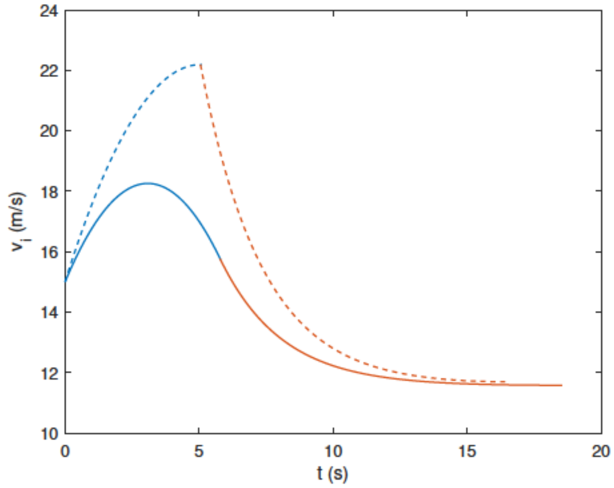


Figure 3·2: Velocity trajectory comparison: Blue curve: CAV speed in straight road segment. Red curve: CAV speed in circular part. Solid curve: solution of bi-level problem. Dotted curve: velocity trajectory obtained by treating the roundabout as two separate road segments.

3.4 Joint optimal control and control barrier function controller (OCBF)

In Sec. 3.3, the optimal solution to **Problem 1** with all constraints inactive was obtained. This solution forms the basis of the OCBF approach (Xiao et al., 2021b): (i) the solution for the *unconstrained* optimal control problem is used as a reference control, (ii) the resulting control reference trajectory is optimally tracked subject to the bounds (3.8), as well as a set of CBF constraints enforcing (3.5), (3.6) and (3.7) (iii) This optimal tracking problem is efficiently solved by discretizing time and solving a simple Quadratic Problem (QP) at each discrete time step. The significance of CBFs in this approach is twofold: first, their forward invariance property (Xiao et al., 2021b) guarantees that all constraints they enforce are satisfied at all times if they are initially satisfied; second, CBFs impose *linear* constraints on the control which is what enables the efficient solution of the tracking problem through the sequence of QPs in (iii) above.

Once we obtain the unconstrained optimal control solution for the straight (3.36) and circular road segment (3.28) respectively, we define it as a control reference trajectory $u_{ref}(t)$. More generally, we can define any function $h(u_i^*(t), x_i^*(t), x_i(t))$ as a control reference $u_{ref}(t) = h(u_i^*(t), x_i^*(t), x_i(t))$, where $x_i(t)$ provides feedback from the actual observed CAV trajectory to add robustness to the solution. We normally choose the simplest and most straightforward choice $u_{ref}(t) = u_i^*(t)$ where $u_i^*(t)$ is the unconstrained optimal control solution obtained from (3.36) and (3.28). We will, however, revisit the case with feedback in Section 3.4.1.

Next, we design a controller that optimally tracks $u_{ref}(t)$ while satisfying all constraints. First, let $\boldsymbol{x}_i(t) \equiv (x_i(t), v_i(t))$. Based on the vehicle dynamics (3.1), define $f(\boldsymbol{x}_i(t)) = [v_i(t), 0]^T$ and $g(\boldsymbol{x}_i(t)) = [0, 1]^T$. Each of the constraints (3.5), (3.6) and (3.8) is expressed in the form $b_k(\boldsymbol{x}_i(t)) \geq 0, k \in \{1, ..., n\}$ where n is the number of constraints. The CBF method maps a constraint $b_k(\boldsymbol{x}_i(t)) \geq 0$ onto a new constraint which directly involves the control $u_i(t)$ in *linear* fashion and takes the general form

$$L_f b_k(\boldsymbol{x}_i(t)) + L_g b_k(\boldsymbol{x}_i(t)) u_i(t) + \gamma(b_k(\boldsymbol{x}_i(t))) \geq 0, \tag{3.50}$$

where L_f, L_g denote the Lie derivatives of $b_k(\boldsymbol{x}_i(t))$ along f and g respectively and $\gamma(\cdot)$ denotes any class-$\mathcal{K}$ function (Xiao et al., 2021b). The forward invariance property of CBFs guarantees that a control input that keeps (3.50) satisfied will also keep $b_k(\boldsymbol{x}_i(t)) \geq 0$. In other words, the constraints (3.5), (3.6), (3.8) and (3.7) are never violated (this comes at the expense of potential conservativeness in the control since the CBF constraint is a sufficient condition for ensuring its associated original problem constraint.)

Considering all constraints in **Problem 1**, the rear-end safety constraint (3.5), the vehicle limitations (3.8) and the lateral safety constraint (3.7) are all straightforward to transform into a CBF form by directly applying (3.50). As an example, consider (3.5) by setting $b_1(\boldsymbol{x}_i(t)) = z_{i,i_p}(t) - \varphi v_i(t) - \delta = x_{i_p}(t) - x_i(t) - \varphi v_i(t) - \delta$. As $b_1(\boldsymbol{x}_i(t))$ is differentiable, we can calculate the Lie derivatives $L_f b_1(\boldsymbol{x}_i(t)) = v_{i_p} - v_i$ and $L_g b_1(\boldsymbol{x}_i(t)) = -\varphi$. Choosing a linear class-$\mathcal{K}$ function $\gamma(x) = k_1 x$, the CBF constraint (3.50) can be obtained as

$$b_{\text{cbf}_1}(\boldsymbol{x}_i, u_i) = v_{i_p} - v_i - \varphi u_i + k_1 b_1(\boldsymbol{x}_i) \geq 0 \tag{3.51}$$

The safe merging constraint (3.6) differs from the rest in that it only applies to a single specific time instants $t_i^{m_k}$. This poses a technical complication due to the fact that a CBF must always be in a continuously differentiable form. We can convert (3.6) to such a form using the technique in (Xiao et al., 2021b) to obtain

$$z_{i,i_m}(t) - \Phi(x_i(t))v_i(t) - \delta \geq 0, \ \ t \in [t_i^{k,0}, t_i^k] \tag{3.52}$$

where $t_i^{k,0}$ denotes the time CAV i enters the road segment connected to M_k and $\Phi(\cdot)$ is any strictly increasing function as long as it satisfies the boundary constraints $z_{i,i_m}(t_i^{k,0}) - \phi v_i(t_i^{k,0}) - \delta \geq 0$ and $z_{i,i_m}(t_i^k) - \phi v_i(t_i^k) - \delta \geq 0$ (the latter is precisely (3.6)). Note that we need to satisfy (3.52) when a CAV changes road segments in the roundabout and the value of i_m changes. Since $z_{i,i_m}(t_i^{k,0}) \geq -L_{i_m} + L_i$, where L_i is the length of the road segment CAV i is in, to guarantee the feasibility of (3.52), we set $\Phi(x_i(t_i^{k,0}))v_i(t_i^{k,0}) + \delta = -L_{i_m} + L_i$. Then, from (3.6), we get $\Phi(x_i(t_i^k)) = \varphi$. Simply choosing a linear $\Phi(\cdot)$ as follows:

$$\Phi(x_i(t)) = \left(\varphi + \frac{L_{i_m} - L_i + \delta}{v_i(t_i^{k,0})}\right)\frac{x_i(t)}{L_i} - \frac{L_{i_m} - L_i + \delta}{v_i(t_i^{k,0})} \tag{3.53}$$

it is easy to check that it satisfies the boundary requirements. Note that when implementing the OCBF controller, $x_i(t)$ needs to be transformed into a relative position $\tilde{x}_i^k + L_i$, which reflects the distance between CAV i and the origin of the current road segment. Thus, z_{i,i_p} and z_{i,i_m} are calculated after this transformation, where $z_{i,i_p} = \tilde{x}_{i_p}^k - \tilde{x}_i^k$, $z_{i,i_m} = \tilde{x}_{i_m}^k - \tilde{x}_i^k$.

The last step is to provide the OCBF controller with the capability to optimally track the reference speed trajectory. This is accomplished by using a Control Lyapunov Function (CLF) $V(\boldsymbol{x}_i(t))$ which is similar to a CBF. Letting $V(\boldsymbol{x}_i(t)) = (v_i(t) - v_{ref}(t))^2$, the CLF constraint takes the form

$$L_f V(\boldsymbol{x}_i(t)) + L_g V(\boldsymbol{x}_i(t)) u_i(t) + \varepsilon V(x_i(t)) \leq e_i(t), \tag{3.54}$$

where $\varepsilon > 0$, and $e_i(t)$ is a relaxation variable which makes this a soft constraint.

We can now formulate the problem that the OCBF controler must solve, i.e., to optimally track the reference trajectory by solving the optimization problem:

$$\min_{u_i(t), e_i(t)} \int_{t_i^0}^{t_i^f} \left(\beta e_i^2(t) + \frac{1}{2}(u_i(t) - u_{ref}(t))^2 \right) dt \tag{3.55}$$

subject to the vehicle dynamics (3.1), the CBF constraints (3.50) derived from (3.5), (3.6), (3.8), (3.7) and the CLF constraint (3.54). As already mentioned, we select $u_{ref}(t) = u_i^*(t)$ and, similarly, $v_{ref}(t) = v_i^*(t)$ in the CLF $V(\boldsymbol{x}_i(t)) = (v_i(t) - v_{ref}(t))^2$, but extend these in Section 3.4.1.

With all constraints converted to CBF constraints in (3.55), we can solve this problem by discretizing $[t_i^0, t_i^f]$ into intervals $[t_i^0, t_i^0 + \Delta], \ldots, [t_i^0 + k\Delta, t_i^0 + (k+1)\Delta], \ldots$ of equal length Δ and solving (3.55) over each time interval. The decision variables $u_k = u_i(t_k)$ and $e_k = e_i(t_k)$ are assumed to be constant on each such time interval and can be easily obtained by solving a Quadratic Program (QP) problem:

$$\min_{u_k, e_k} \beta e_k^2 + \frac{\Delta}{2}(u_k - u_{ref}(t_i^0 + k\Delta))^2 \tag{3.56}$$

subject to the CBF constraints (3.50) and the CLF constraint (3.54), all evaluated at t_k, where all CBF and CLF constraints are linear in the decision variables u_k and e_k. By repeating this process until CAV i exits the CZ, the solution to (3.55) is obtained.

The computational cost in using OCBF is that of solving a Quadratic Program (QP) as shown in (3.56) at each time step. The cost of QP solutions is minimal and, in practice, it is less than 0.01 sec. By comparison, a complete solution of the optimal control problem (3.13), may require one or two orders of magnitude more (about 0.3 to 30 seconds). Similarly, an MPC-based approach is also an order of magnitude slower (about 0.5 second). All the computation times are measured in MATLAB.

3.4.1 Reference Trajectory with Feedback

As already mentioned, in (3.55) we can select the simplest and most straightforward $u_{ref}(t) = u_i^*(t)$ along with $v_{ref}(t) = v_i^*(t)$ in the CLF $V(\boldsymbol{x}_i(t)) = (v_i(t) - v_{ref}(t))^2$. These simple reference trajectories work well in problems where the deviations $(u_i(t) - u_{ref}(t))$ are not exceedingly large, as observed, for instance, in optimal merging (Xiao et al., 2021b) and intersection control (Zhang and Cassandras, 2019) problems. However, when the constraints become complex, especially when traffic in the roundabout becomes heavy, tracking these simple reference trajectories in an open-loop way often results in such large deviations from the unconstrained optimal solution (as illustrated in Section 3.5). Thus, a reference trajectory which includes feedback is introduced to solve this issue. In particular, we set

$$v_{\text{ref}}(t) = \frac{x^*(t)}{x(t)} v^*(t), \quad u_{\text{ref}}(t) = \frac{x^*(t)}{x(t)} u^*(t), \tag{3.57}$$

where $x(t)$ is the actual observed CAV position.

We also introduce another type of position-feedback to resolve the problem related to large deviations in t between the reference and actual trajectories, as illustrated through a specific numerical example in Fig. 3·8, further discussed in Section 3.5.2. We calculate a reference time t_{ref} by solving the following equation at any t:

$$x^*(t_{\text{ref}}) = x(t) \tag{3.58}$$

where $x^*(\cdot)$ is the optimal unconstrained position of a given CAV in (3.29). Then, we choose the unconstrained optimal trajectory at t_{ref} as the reference $v_{\text{ref}}(t) = v^*(t_{\text{ref}})$ and $u_{\text{ref}}(t) = u^*(t_{\text{ref}})$ or

$$v_{\text{ref}}(t) = v^*((x^*)^{-1}(x(t))), \quad u_{\text{ref}}(t) = u^*((x^*)^{-1}(x(t))) \tag{3.59}$$

We will show how this approach can improve performance in Section 3.5.2.

3.5 Simulation Results

In this section, we use Vissim, a multi-model traffic flow simulation platform, as a baseline to evaluate traffic performance in roundabouts with human-driven vehicles and compare it to the performance obtained using our OCBF controller (for all CAVs). We use the model shown in Fig. 3·1 constructed in Vissim and use the same vehicle arrival patterns in the human-driven vehicle baseline and under the OCBF controller for consistent comparison purposes.

3.5.1 Virtual roundabout example

We first conduct a case study based on a virtual roundabout as shown in Fig. 3·1. The parameter settings are as follows: $L_a = 100m$, $L = 100m$, $\delta = 0m$, $\varphi = 1.8s$, $v_{\max} = 20m/s$, $v_{\min} = 0$, $u_{\max} = 5m/s^2$, $u_{\min} = -5m/s^2$. This example considers a *symmetric* configuration in the sense that $L_a = L$. Different weights $\alpha_1, \alpha_2, \alpha_3$ directly influence the unconstrained optimal trajectory of a CAV. To explore this impact and determine a proper parameter setting for the weights, we start with the analysis of a single CAV that enters O_1 with initial speed $v_0 = 15m/s$ and exits from E_2 and plot the unconstrained optimal velocity trajectory

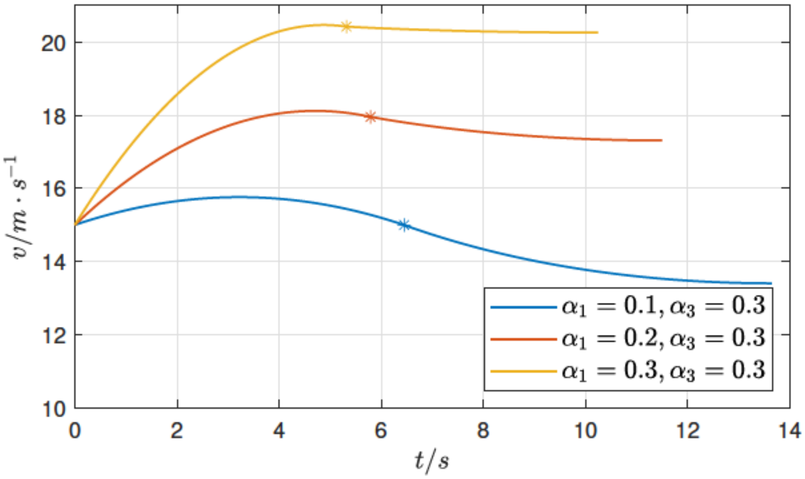


Figure 3·3: Velocity trajectory under different travel time weights α_1. The asterisk shows the point when the CAV enters the circular part of the roundabout from the straight road segment.

under different weight settings in Figs. 3·3 and 3·4. In Fig. 3·3 the weight for travel time α_1 differs from the weight for comfort fixed at $\alpha_3 = 0.3$, while in Fig. 3·4 α_1 is fixed at 0.2 with changing α_3. Figure 3·3 shows that when comfort is taken into consideration with weight $\alpha_3 = 0.3$, the optimal control as the weight on time α_1 changes is for the CAV to go through the roundabout with a similar velocity trajectory but with higher speed. Figure 3·4 shows the influence of the comfort weight α_3. When α_3 decreases with α_1 fixed, the CAV adopts a higher speed in the circular part.

In what follows we will focus on the weight parameter settings: $\alpha_1 = 0.2, \alpha_2 = 0.5, \alpha_3 = 0.3$ to better illustrate the benefits of OCBF when the unconstrained velocity trajectory is within the speed limit. We will further discuss the benefits of a reference trajectory with feedback when this is not the case in Section 3.5.2. We compare the OCBF controller with the human-driven vehicle Vissim baseline by simulating the same *symmetric* vehicle arrival pattern. The traffic in the three incoming roads is generated through Poisson processes and is *symmetric* in the sense that they all have the same traffic rates. Two scenarios corresponding to a normal traffic rate of 400 CAVs/h and a high traffic rate of 600 CAVs/h are

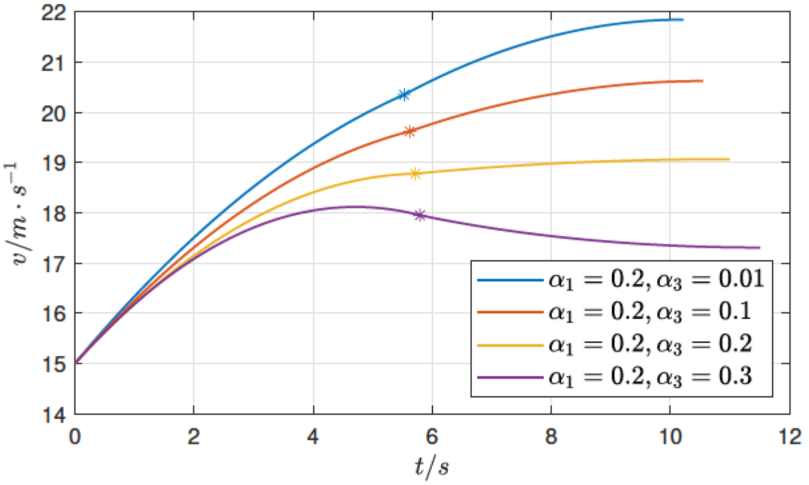


Figure 3·4: Velocity trajectory under different comfort weights α_3. The asterisk shows the point when the CAV enters the circular part of the roundabout from the straight road segment.

simulated. The former traffic rate allows vehicles to smoothly pass the roundabout with occasional wait or stop in Vissim, while the latter can sometimes form queues at the entry points before the merging points. The two traffic rates are selected based on observations of Vissim simulations so as to present scenarios with different road saturation (roughly 66.7% and 100% respectively). A total number of approximately 200 CAVs are simulated in both cases. In this example, the OCBF controller is applied with the SDF rule (see Section 3.2.3) and the reference trajectory is generated *without* feedback. The results comparing the performance of OCBF to that of human-driven vehicles in Vissim are shown in Table 3.3.

When the traffic rate is 400 CAVs/h, there is a 6% loss in the average travel time of CAVs using OCBF compared to that in Vissim due to the emphasis on energy and comfort which are improved by 49% and 21% respectively. The OCBF controller also shows a 12.5% improvement in the total objective over the human-driven performance in Vissim. On the other hand, when the traffic rate becomes 600 CAVs/h (which results in congestion), there is an improvement of 23% in the travel time (despite the low weight of $\alpha_1 = 0.2$ on

Table 3.3: Performance comparison for a symmetric roundabout under symmetric traffic input

Traffic rate	400 CAVs/h		600 CAVs/h	
Methods	OCBF	Vissim	OCBF	Vissim
Ave. time (s)	21.00	19.77	28.49	36.87
Ave. energy	17.67	34.78	32.73	57.35
Ave. comfort	58.36	74.10	47.98	60.96
Ave. obj. (energy)	175.20	200.32	218.37	296.56
Ave. fuel (mL)	19.64	13.39	19.92	14.59
Ave. obj. (fuel)	177.17	178.92	205.56	253.81

travel time), 43% in the energy consumption, 20% in the comfort cost, and 26.5% in the total objective using OCBF relative to Vissim. This improvement in all the metrics is to be expected as the CAVs using the OCBF method never stop and wait for CAVs in other road segments to go through, which is the case in Vissim.

In Table 3.3, we also include another performance metric that captures fuel consumption through a detailed model compared to the simple "energy" metric $\frac{1}{2}u^2$. The fuel consumption is measured using the model introduced in (Kamal et al., 2011) using the set of parameters given in (Kamal et al., 2013b):

$$f_V = f_{\text{cruise}} + f_{\text{accel}} \tag{3.60}$$

where $f_{\text{cruise}} = b_0 + b_1 v + b_2 v^2 + b_3 v^3$ represents the fuel consumption of cruising, $f_{\text{accel}} = u(c_0 + c_1 v + c_2 v^2)$ represents the fuel consumption of acceleration. The OCBF controller (which is not designed to explicitly minimize f_V) consumes about 40% more fuel than the car-following model used in Vissim; however, since we do not know the details of the Vissim fuel consumption model, the importance of this comparison should be discounted. Nonetheless, a higher fuel consumption under OCBF is justified due to the following two reasons: (i) a CAV consumes more fuel when cruising at a higher speed - which is selected by CAVs to improve travel times, (ii) the quadratic criterion $\frac{1}{2}u^2$ discourages large deceleration, whereas deceleration consumes no fuel according to (Kamal et al., 2011). On the other hand, leaving deceleration unpenalized slows traffic down and promotes backlog. Nevertheless, despite this difference, the total objective with fuel consumption included is still improved using OCBF as shown in Table 3.3.

Finally, we note that if fuel consumption as measured through f_V above becomes an optimization objective, this can be accomplished using a numerical optimization approach which makes use of CBFs to still guarantee all constraints; this was shown in (Xiao et al., 2021b).

3.5.2 Real Roundabout

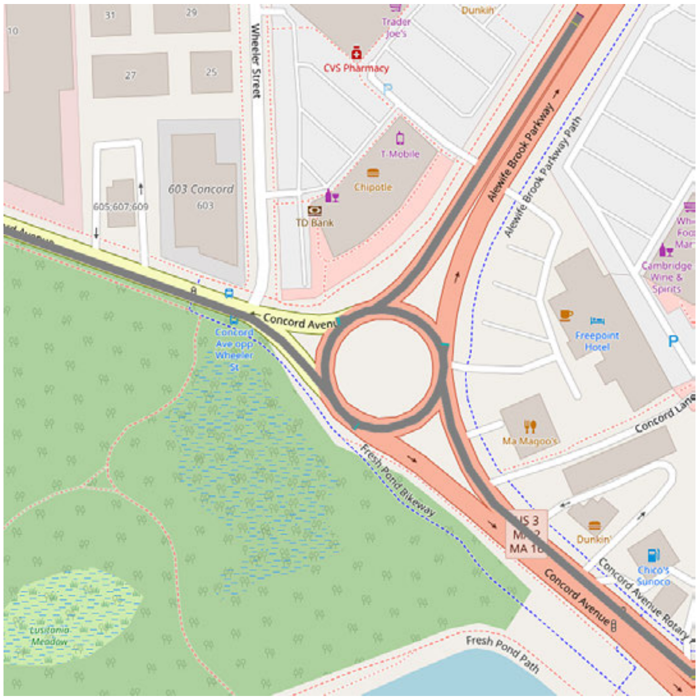


Figure 3·5: An asymmetric roundabout near Fresh Pond in Boston, MA

We consider next a real roundabout as shown in Fig. 3·5 located near Fresh Pond in Boston, MA, with the geometric parameters $L_1 = 186m, L_2 = 165m, L_3 = 196m, L_{a,1} = L_{a,2} = 53m, L_{a,3} = 63m$. This roundabout is asymmetric with a small circle and three long entries. The remaining settings are the same as in Section 3.5.1, i.e., $\delta = 0m$, $\varphi = 1.8s$,

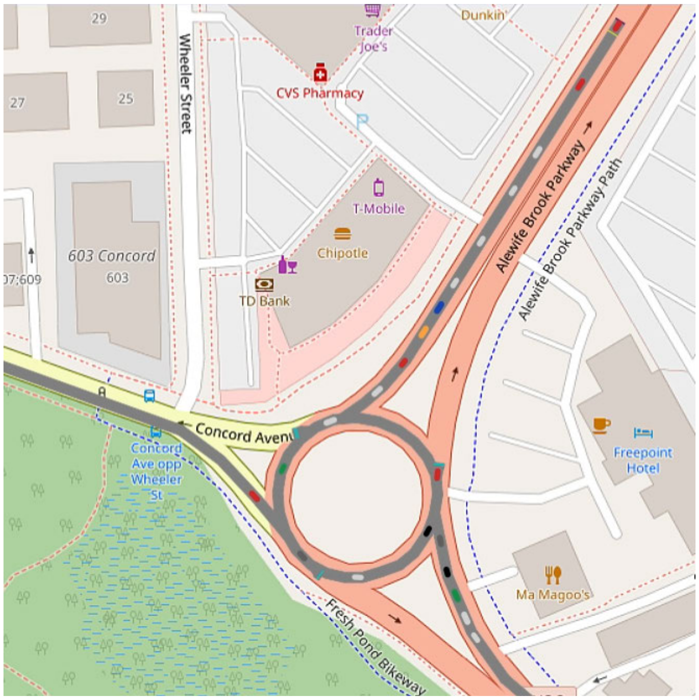


Figure 3·6: Vissim simulation of human-driven behavior

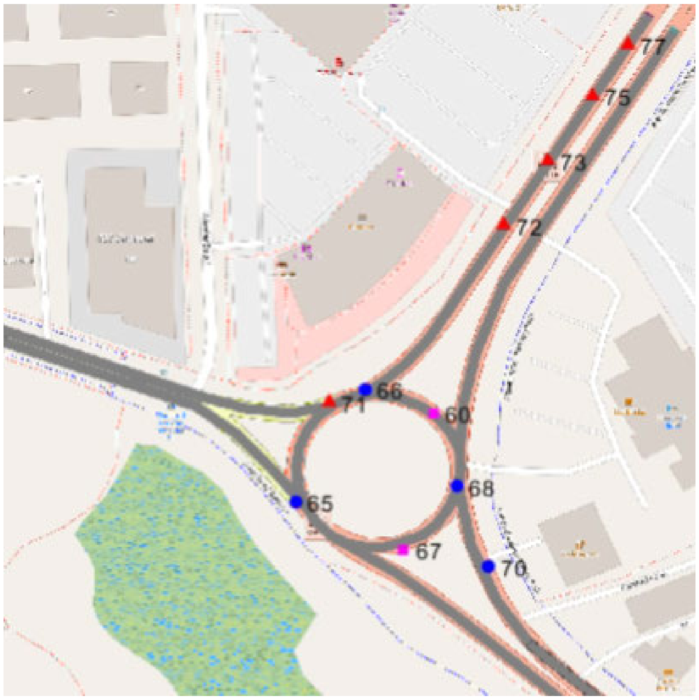


Figure 3·7: Matlab simulation of OCBF controller

$v_{\max} = 20m/s$, $v_{\min} = 0$, $u_{\max} = 5m/s^2$, $u_{\min} = -5m/s^2$. We start with the same weight parameter settings: $\alpha_1 = 0.2, \alpha_2 = 0.5, \alpha_3 = 0.3$ and compare the performance of OCBF (with the SDF rule and no feedback) to the human-driven vehicle performance in Vissim under two different traffic rates 600 CAVs/h and 800 CAVs/h. The simulation results are shown in Table 3.4. Sample snapshots of the Vissim and MATLAB simluation respectively are shown in Fig. 3·6 and Fig. 3·7 at the same time instant.

Table 3.4: Performance comparison for an asymmetric real roundabout under symmetric traffic input

Traffic rate	600 CAVs/h		800 CAVs/h	
Methods	OCBF	Vissim	OCBF	Vissim
Ave. time (s)	18.97	27.44	22.28	39.13
Ave. energy	49.50	62.60	63.82	74.34
Ave. comfort	41.65	35.83	37.83	30.22
Ave. obj. (energy)	165.44	217.94	194.35	285.30
Ave. fuel	21.57	11.66	19.96	12.49
Ave. obj. (fuel)	137.52	167.00	150.48	223.45

As illustrated in Fig. 3·6, a backlog always forms in the Vissim simulation which requires vehicles to stop and queue before entering the circular road segment. The OCBF method, however, allows the CAVs to move faster and smoothly with *guaranteed* safety constraints without forming any backlogs (see Fig. 3·7). Although δ is set to 0, the CAVs using OCBF still keep an appropriate safety distance because of the φv_i term in (3.5); interestingly, these safety distances are actually larger than those observed between human-driven vehicles that use the car-following model in Vissim.

As shown in Table 3.4, CAVs using the OCBF controller improve performance on average by 31% in time, 21% in energy consumption and 24% in the total objective when the traffic rate is 600 CAVs/h, which is consistent with the results of the example in Section 3.5.1. An additional 8% improvement in the total objective is achieved under the high traffic rate of 800 CAVs/h. Note that although the comfort cost is included in the total objective with weight 0.3, the OCBF controller still incurs a 16% higher comfort cost. This can be explained by the backlog formed by human-driven vehicles in Vissim shown in Fig. 3·6. As the queuing vehicles stop before entering and re-accelerate from zero speed in the circle, they result in a lower *average* velocity than those using the OCBF controller (thus, the actual comfort experience wihin the circular road segment is misrepresented by the average comfort data). The fuel consumption and the associated total objective using the model in (Kamal et al., 2011), instead of the simple metric $\frac{1}{2}u^2$, are also included in Table 3.4 showing similar results to those of Section 3.5.1.

In the following subsections, the settings of the OCBF controller are changed and simulated to explore the influence of weights, sequencing rules, feedback as well as asymmetric incoming traffic.

Time vs. comfort

In this case study, the travel time weight is fixed to $\alpha_1 = 0.2$ while the weight of comfort α_3 is changed from 0.3 to 0.01, representing almost no emphasis on comfort. The OCBF controller uses the SDF rule (see Section 3.2.3) and tracks the reference trajectory with feedback. Simulation results under the traffic rate of 800 CAVs/h is recorded in Table 3.5.

Table 3.5: Performance comparison for an asymmetric real roundabout under symmetric traffic input

Methods	OCBF		Vissim	
Weight of comfort α_3	0.3	0.01	0.3	0.01
Ave. time (s)	22.65	21.84	39.13	
Ave. energy	64.61	61.95	74.34	
Ave. comfort	37.12	39.27	30.22	
Ave. obj. (energy)	196.63	131.47	285.30	198.49

As shown in Table 3.5, the OCBF controller achieves better performance both when $\alpha_3 = 0.3$ and $\alpha_3 = 0.01$, with improvements of over 30% in the total objective. The average travel time as well as energy consumption decreases while the comfort cost increases when $\alpha_3 = 0.01$. This is reasonable as more emphasis is placed on energy consumption instead of comfort. As the comfort cost is a quadratic function of speed, a larger comfort cost indicates a higher velocity which results in shorter travel time.

Effect of Sequencing Rules

In this set of simulations, the effect of different sequencing rules is explored using both a symmetric roundabout (the virtual one) and an asymmetric roundabout (the real one). Simulation results on the performance of OCBF with FIFO and OCBF with the SDF sequencing policy are shown in Table 3.6.

Table 3.6: Performance comparison for different sequencing rules under symmetric traffic input

Roundabout geometry	Asymmetric		Symmetric	
Sequencing rule	FIFO	SDF	FIFO	SDF
Ave. time (s)	57.51	18.97	34.31	28.49
Ave. energy	141.37	49.50	39.61	32.73
Ave. comfort	14.27	41.65	43.66	47.98
Ave. obj. (energy)	436.14	165.44	250.44	218.37

When OCBF+FIFO is applied in a symmetric roundabout, it performs worse than OCBF+SDF in average travel time (20%), energy consumption (20%) as well as the total objective (14%). Comparing Table 3.6 with Table 3.3, it can be seen that the CAVs still benefit from the OCBF controller regardless of the sequencing policy selected. However, when OCBF+FIFO is applied to an asymmetric roundabout, the traffic becomes congested and the results become unstable even after simulating only 50 CAVs, indicating that FIFO works poorly in an asymmetric roundabout. For example, when a CAV enters segment l_4, it has to wait for another CAV that has entered l_2 just before it to drive more than 100 meters for safe merging. This is unreasonable and may also result in some extreme cases where the OCBF problem becomes infeasible. On the other hand, OCBF+SDF still achieves better and reliable performance in an asymmetric setting.

Reference Trajectory with Feedback

Here, the influence of feedback is studied. As mentioned in Section 3.4.1, the reference trajectory without feedback may become unreliable due to the exponential terms that appear in (3.27) and (3.28) which result in large values when the time in the roundabout t is large. This is illustrated with the simulation results shown in Table 3.7 for two different symmetric traffic rates 600 CAVs/h and 800 CAVs/h with the weight parameters set to $\alpha_1 = 0.2, \alpha_3 = 0.01$. When the traffic rate is 600 CAVs/h, using OCBF with feedback has a minimal effect on performance, contributing only a 4% improvement in the total objective. However, when the traffic rate increases to 800 CAVs/h, the roundabout becomes congested when CAVs are CAVs are controlled by OCBF without feedback, resulting in the explosive average travel time and energy consumption reflected in Table 3.7. This instability can be explained by the extended unconstrained optimal speed trajectory of a typical CAV plotted in Fig. 3·8.

Table 3.7: Performance comparison for an asymmetric real roundabout under symmetric traffic input

Traffic rate	600 CAVs/h		800 CAVs/h	
Feedback	No	Yes	No	Yes
Ave. time (s)	19.22	18.92	111.70	21.84
Ave. energy	52.97	49.64	99.82	61.95
Ave. comfort	42.87	43.61	15.26	39.27
Ave. obj. (energy)	114.26	109.99	453.46	131.47

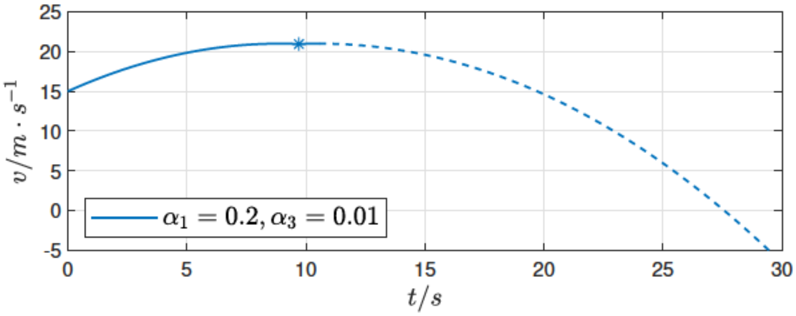


Figure 3·8: Extended unconstrained optimal trajectory

In Fig. 3·8, the unconstrained optimal speed trajectory is shown as the solid blue line, where the terminal time t_f is marked with an asterisk. Note that t_f here corresponds to the unconstrained problem and is determined by (3.30), so in the actual constrained trajectory implemented through the OCBF controller the CAV is still in the CZ for times $t > t_f$. This "extended" speed trajectory is shown with the dashed line. When the traffic is heavy and results in considerable delay, the reference trajectory required by the OCBF controller is the one shown by the dashed line in Fig. 3·8. The reference speed will even drop to 0 after 27s in this example. Thus, the OCBF controller will track the inappropriate reference speed, which induces the CAV to stop and consequently block traffic. However, when a referenced trajectory with feedback (3.59) is used, the controller will map the current state of the CAV to the point on the solid curve in Fig. 3·8 and thus solve the issue related to selecting the proper time t on the reference trajectory.

Imbalanced Traffic

The purpose of this case study is to investigate the effect of traffic volume. A total number of approximately 200 CAVs is simulated under imbalanced incoming traffic (900 CAVs/h from O_1, 450 CAVs/h from O_2 and O_3). The simulation results of the performance under OCBF+SDF compared to that of the human-driven vehicles in Vissim under imbalanced incoming traffic are shown in Table 3.8.

Table 3.8: Performance comparison for an asymmetric real roundabout under asymmetric traffic input

Method	CAV Origin	Time	Energy	Comfort	Ave. Obj
OCBF	All	18.28	45.63	42.24	158.45
	O_1	19.38	44.01	42.14	162.27
	O_2	16.13	41.68	39.35	142.27
	O_3	18.62	53.22	45.70	169.48
Vissim	All	28.87	69.25	36.09	231.88
	O_1	33.88	76.03	34.67	263.03
	O_2	26.90	71.27	33.24	222.61
	O_3	21.43	53.89	42.06	182.36

Comparing Table 3.8 with Table 3.4, it is seen that imbalanced traffic causes longer travel times ($\sim$2s) and higher energy consumption ($\sim$7%), although the total traffic rates are the same. The imbalanced traffic results in an imbalanced performance of CAVs from different origins. The CAVs originating from O_1 with heavy traffic perform worse than those from O_2 and O_3 where traffic is lighter. However, when OCBF+SDF is applied to the system, the imbalanced traffic brings no performance loss and becomes more balanced compared to human-driven vehicle traffic. This result is interesting because the OCBF approach does not explicitly take into account the fact that traffic is *imbalanced*. An explanation of this phenomenon is that the SDF policy gives CAVs from O_1 a higher priority as they are more likely to be the closest ones to the MP, while OCBF allows the CAVs to go through the roundabout quickly without stopping; therefore, the CAVs from a heavy traffic flow are less likely to get congested.

Chapter 4

Feasibility-Guaranteed OCBF

As aforementioned in the introduction, the feasibility problem of CBFs arises when the control limits are tight, which often happens in the roundabout. To resolve this issue, we start from the traffic merging problem where the traffic from two different roads join together. The merging roadway is a basic component of a roundabout, which has a similar problem structure but less complexity. Through the analysis of the merging problem, we can provide a feasibility guaranteed OCBF approach that is extendable to the roundabout problem.

4.1 Feasibility-Guaranteed OCBF

We first consider the general OCBF controller to provide a sufficient condition for it to be feasible. Consider the nonlinear affine control system in the form:

$$\dot{x} = f(x) + g(x)u, \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^q$ are locally Lipschitz, $x \in D \subset \mathbb{R}^n$ (D denotes the set of admissible states) and $u \in U \subset \mathbb{R}^q$ (U denotes the set of admissible control). As introduced in Chapter 2, the CBF method maps $b(\boldsymbol{x}(t)) \geq 0$ to a new constraint which directly involves the control $u(t)$ and takes the (linear in the control) form

$$b_{\mathrm{CBF}}(\boldsymbol{x}(t), u(t)) = L_f b(\boldsymbol{x}(t)) + L_g b(\boldsymbol{x}(t)) u_i(t) + \gamma(b(\boldsymbol{x}(t))) \geq 0, \tag{4.2}$$

where L_f, L_g denote the Lie derivatives of $b(\boldsymbol{x}(t))$ along f and g respectively and $\gamma(\cdot)$ denotes some class-$\mathcal{K}$ function (Xiao and Belta, 2019).

To optimally track the reference speed trajectory, a CLF function $V(\boldsymbol{x}(t))$ is used. A CLF function is not a necessity in OCBF but often helps tracking the velocity trajectory. The CLF constraint takes the form

$$L_f V(\boldsymbol{x}(t)) + L_g V(\boldsymbol{x}(t)) u(t) + \varepsilon V(\boldsymbol{x}(t)) \leq e(t), \tag{4.3}$$

where $\varepsilon > 0$, and $e_i(t)$ is a relaxation variable which makes the constraint soft.

We solve an optimal tracking problem by discretizing $[t^0, t^m]$ into intervals of equal length Δt and solving a Quadratic Program (QP) problem (4.4)

$$\begin{aligned} \min_{u(t), e(t)} \quad & \beta e(t)^2 + \frac{1}{2}(u(t) - u_{\text{ref}}(t))^2 \\ \text{s.t.} \quad & L_f b(\boldsymbol{x}(t)) + L_g b(\boldsymbol{x}(t)) u(t) + \gamma(b(\boldsymbol{x}(t))) \geq 0, \\ & L_f V(\boldsymbol{x}(t)) + L_g V(\boldsymbol{x}(t)) u(t) + \varepsilon V(\boldsymbol{x}(t)) \leq e(t), \\ & u_{\min} \leq u(t) \leq u_{\max}, \\ & t = t^0 + k\Delta t \end{aligned} \tag{4.4}$$

where the control limit

$$u_{\min} \leq u(t) \leq u_{\max} \tag{4.5}$$

is also included. This approach is simple and computationally efficient. However, it is also myopic since each QP is solved over a single time step, which may lead to infeasible QPs at future time steps, especially when the control limits (4.5) is tight.

To avoid the infeasibility caused by the myopic QP solving approach in the CBF method, an additional "feasibility constraint" $b_F(\boldsymbol{x}(t), u(t)) \geq 0$ is introduced in (Xiao et al., 2022b).

Definition 4.1. *(Feasibility Constraint) A feasibility constraint is defined as a constraint that makes the QP corresponding to the next time interval feasible, thus, in the case of* (4.4)*, a feasibility constraint has the following properties:*

(i) it guarantees that the CBF constraint (4.2) *and the control bound* (4.5) *do not conflict,*

(ii) the feasibility constraint itself conflicts with neither the CBF constraint (4.2) *nor the control bound* (4.5).

Definition 4.2. *(Conflict free) Any two state and/or control constraints (e.g., any CBF constraint)* conflict-free *if their intersection is non-empty in terms of the control.*

In (Xiao et al., 2022b), a sufficient condition for feasibility is provided based on the assumption that the feasibility constraint $b_F(\boldsymbol{x}(t), u(t))$ is independent of the control $u(t)$. This assumption restricts the generality of the sufficient condition as many systems (e.g. traffic merging problem) do not have the beautiful nature of independence. In what follows, we show that it is possible to find a feasibility constraint $b_F(\boldsymbol{x}(t), u(t)) \geq 0$ without this assumption and explicitly derive this constraint which can provably guarantee feasibility. This provably guaranteed feasibility also relies on the property of the system, however, we will show that this property assumption is looser than the assumption in (Xiao et al., 2022b). We will also show in the case study that the traffic merging control problem satisfies such property, thus we are able to explicitly derive a feasibility constraint $b_F(\boldsymbol{x}(t), u(t)) \geq 0$ which can provably guarantee feasibility.

4.1.1 Sufficient condition to guarantee feasibility

We now consider the QP problem (4.4). Due to the existence of the slack variable $e_i(t)$ in the CLF constraint, the CLF constraint (4.3) never conflicts with other constraint. Thus, the QP problem (4.4) only becomes infeasible when the CBF constraint (4.2) conflicts with (4.5). We consider the case that $u(t)$ is one dimensional. Without loss of generalization, we assume $L_g(b(x(t)) \leq 0$ and multiplies $-L_g(\boldsymbol{b}(x(t))$ to both side of (4.5) which yields:

$$-L_g(b(\boldsymbol{x}(t))u_{\min} \leq -L_g(b(\boldsymbol{x}(t))u(t) \leq -L_g(b(\boldsymbol{x}(t))u_{\max} \tag{4.6}$$

Note that (4.2) can be rewritten as

$$-L_g b(\boldsymbol{x}(t))u(t) \leq L_f b(\boldsymbol{x}(t)) + \gamma(b(\boldsymbol{x}(t))) \tag{4.7}$$

where $-L_g b(\boldsymbol{x}(t))u(t) \leq -L_g b(\boldsymbol{x}(t))u_{\max}$ never conflicts with (4.7) as they have the same inequality direction. Thus, we can guarantee that (4.2) and (4.6) are conflict-free by adding a feasibility constraint:

$$b_F(\boldsymbol{x}) = L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})u_{\min} + \gamma(b(\boldsymbol{x})) \geq 0 \tag{4.8}$$

where the argument t of the functions above is omitted for simplicity. Note that there is no control u in (4.8), thus (4.8) is equivalent to $\boldsymbol{x} \in \mathcal{C}_F$, where

$$\mathcal{C}_F = \{\boldsymbol{x} : b_F(\boldsymbol{x}) \geq 0\} \tag{4.9}$$

We can now consider this feasibility constraint as a new CBF and apply (4.2) to transform it into a CBF constraint to enforce the forward invariance of set $\mathcal{C}_F$. Choosing the linear function $\gamma(x) = k_1 x$ and $\gamma_F(x) = k_F x$ as the class $\mathcal{K}$ functions, the corresponding CBF constraint is:

$$\begin{aligned} &L_f^2 b(\boldsymbol{x}) + L_f L_g b(\boldsymbol{x})u_{\min} + k_1 L_f b(\boldsymbol{x}) + \big(L_g L_f b(\boldsymbol{x}) + L_g^2 b(\boldsymbol{x})u_{\min} \\ &+ k_1 L_g b(\boldsymbol{x})\big)\, u + k_F\left(L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})u_{\min} + k_1(b(\boldsymbol{x}))\right) \geq 0 \end{aligned} \tag{4.10}$$

Next, we determine a feasible constraint to be added to every QP so that it guarantees the QP of the next time interval is feasible. This is done by choosing k_F so that $k_F = k_1$ and (4.10) becomes

$$\begin{aligned} &L_f^2 b(\boldsymbol{x}) + L_f L_g b(\boldsymbol{x})u_{\min} + (L_g L_f b(\boldsymbol{x}) + L_g^2 b(\boldsymbol{x})u_{\min})u \\ &+ k_1(L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})u_{\min}) + k_1(L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})u + k_1(b(\boldsymbol{x})) \geq 0 \end{aligned} \tag{4.11}$$

Define a "candidate function" $\eta(\boldsymbol{x}_i, u_i)$ (Xiao et al., 2022b) as

$$\begin{aligned}\eta(\boldsymbol{x}, u) = {} & L_f^2 b(\boldsymbol{x}) + L_f L_g b(\boldsymbol{x}) u_{\min} + \left(L_g L_f b(\boldsymbol{x}) + L_g^2 b(\boldsymbol{x}) u_{\min}\right) u \\ & + k_1 \left(L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x}) u_{\min}\right)\end{aligned} \tag{4.12}$$

Then, replacing the first four terms of the feasibility CBF constraint (4.11) with $\eta(\boldsymbol{x}, u)$ and noting that the remaining terms are given by $b_{\text{cbf}}(\boldsymbol{x}, u)$ defined in (4.2) to substitute them with $b_{\text{cbf}}(\boldsymbol{x}, u)$, (4.11) becomes

$$\eta_1(\boldsymbol{x}, u) + k_1 b_{\text{cbf}}(\boldsymbol{x}, u) \geq 0 \tag{4.13}$$

Since $b_{\text{cbf}}(\boldsymbol{x}, u) \geq 0$ is required in (4.2), it follows that (4.13) will be satisfied if

$$\eta(\boldsymbol{x}, u) \geq 0 \tag{4.14}$$

Setting

$$b_\eta(\boldsymbol{x}) = L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x}) u_{\min} \tag{4.15}$$

in (4.12), we can view $b_\eta(\boldsymbol{x})$ as a CBF and apply (4.2) to observe that the corresponding CBF constraint coincides with (4.14). Adding the CBF constraint (4.14) to the QP (4.4), we will show next that (4.14) is a constraint that guarantees the feasibility of the QP corresponding to the next time interval. Before establishing this result, we make the following assumption.

Assumption 4.1. *The affine system* (4.1) *and the CBF function $b(\boldsymbol{x})$ has the property such that:*

$$\forall \boldsymbol{x} \in \mathcal{C}, \quad \sup_{u \in U} h(\boldsymbol{x}, u) \geq 0, \tag{4.16}$$

where

$$h(\boldsymbol{x}, u) = L_f^2 b(\boldsymbol{x}) + L_f L_g b(\boldsymbol{x}) u_{\min} + \left(L_g L_f b(\boldsymbol{x}) + L_g^2 b(\boldsymbol{x}) u_{\min}\right) u \tag{4.17}$$

$$\mathcal{C} = \{\boldsymbol{x} : b(\boldsymbol{x}) \geq 0\}$$

This assumption is required to guarantee that (4.14) and (4.6) are conflict-free. It is a system dependent assumption which provides a guidance in designing the CBF and can be proved to be true for systems like the traffic merging problem.

Assumption 4.2. *Δt is adequately small such that the forward invariance property of CBFs remains in force.*

This assumption is made to utilize the forward invariance property of CBFs to guarantee safety. It can be met by decreasing the time interval or by using the recently proposed event-driven technique (Xiao et al., 2021a) which uses a tunable inter-QP interval (instead of a fixed time-driven one) which is guaranteed to preserve constraint satisfaction.

Theorem 4.1. *Under Assumption 4.1 and 4.2, if $b_\eta(\boldsymbol{x}(t)) \geq 0$ and the QP* (4.4) *subject to* (4.2)*,* (4.6) *and* (4.14) *is feasible at time t, then the QP corresponding to time $t+\Delta t$ is also feasible.*

Proof. By *Assumption 4.1*, there always exists a control input $u(t) \in [u_{\min}, u_{\max}]$ such that $h(\boldsymbol{x}, u) \geq 0$. As $b_\eta(\boldsymbol{x}(t)) \geq 0$, applying (4.12), we can always find a feasible control $u(t)$ such that $\eta(\boldsymbol{x}(t), u(t)) \geq 0$. As $\eta(\boldsymbol{x}(t), u(t)) \geq 0$ is the CBF constraint corresponding to $b_\eta(\boldsymbol{x}(t)) \geq 0$, using the forward invariance property of CBFs under *Assumption 4.2*, we have $b_\eta(\boldsymbol{x}(t+\Delta t)) \geq 0$. Thus, there always exists a control $u(t) \in [u_{\min}, u_{\max}]$ such that $\eta(\boldsymbol{x}(t+\Delta t), u(t+\Delta t)) \geq 0$. Hence, (4.14) and (4.6) are conflict-free at $t+\Delta t$.

Since $-L_g b(\boldsymbol{x}(t)) \geq 0$, (4.2) constrains the control $u(t+\Delta t)$ with an upper bound. Similarly, $u(t+\Delta t)$ is also constrained by an upper bound through (4.14). Thus, (4.2) and (4.14) are conflict-free at $t+\Delta t$.

Since the QP (4.4) subject to (4.2), (4.6) and (4.14) is feasible at time t, it follows that $b_{\text{cbf}}(\boldsymbol{x}(t), u(t)) \geq 0$, $b_F(\boldsymbol{x}(t)) \geq 0$. As $\eta(\boldsymbol{x}(t), u_i(t)) \geq 0$ always has a solution $u(t) \in [u_{\min}, u_{\max}]$, there exists a control under which (4.13) is satisfied. Since (4.13) is the CBF constraint of (4.8), using the forward invariance of CBFs under *Assumption 4.2*, we have $b_F(\boldsymbol{x}(t+\Delta t)) \geq 0$, which implies that (4.2) and (4.6) are conflict-free at time $t+\Delta t$.

Thus, all constraints of the QP (4.4) are conflict-free at $t+\Delta t$ and the QP corresponding to time $t+\Delta$ is feasible. □

Assumption 4.3. *The following initial conditions are satisfied:*

$$b(\boldsymbol{x}(t^0)) \geq 0, b_F(\boldsymbol{x}(t^0)) \geq 0, b_\eta(\boldsymbol{x}(t^0)) \geq 0$$

.

This assumption requires the system to be safe and feasible at the initial time point such that this property can maintain through the OCBF method. In the context of set invariance, this is a common assumption to require the initial state $\boldsymbol{x}$ to be in the feasible safe set.

Theorem 4.2. *Under Assumptions 4.1, 4.2 and 4.3, the QP* (4.4) *subject to* (4.2)*,* (4.6) *and* (4.14) *corresponding to any time interval* $[t^0+k\Delta t, t^0+(k+1)\Delta t] \subset [t^0, t^f]$ *is feasible.*

Proof. See Appendix A. □

4.1.2 Trade-off between feasibility and performance

In this section, we go back to the feasibility condition (4.8) and view it in the context of set invariance: a controller u that enforces the forward invariance of $\mathcal{C}_F$ guarantees the feasibility of (4.8). Furthermore, we only need to enforce the forward invariance of a nonempty subset of $\mathcal{C}_F$ to guarantee feasibility at the expense of conservation, such that he feasibility guaranteed OCBF controller becomes more conservative if the subset is smaller.

One of the goals of the OCBF controller (4.4) is to guarantee the invariance of a safe set such that:

$$b(\boldsymbol{x}(t)) \geq 0, \tag{4.18}$$

thus we can naturally seperate $b_F(\boldsymbol{x})$ into two parts: $b_\eta(\boldsymbol{x})$ and $\gamma(b(\boldsymbol{x}))$, where $b_\eta(\boldsymbol{x}) = L_f b(\boldsymbol{x}) + L_g(\boldsymbol{x})u_{\min}$ is defined in (4.12), and $\gamma(b(\boldsymbol{x})) \geq 0$. $b_\eta(\boldsymbol{x})$ provides the feasibility guarantee in *Theorem 4.1 and 4.2* where a CBF constraint (4.14) is found to enforce the forward invariance of $b_\eta(\boldsymbol{x}) \geq 0$, while $\gamma(b(\boldsymbol{x}))$ gives some relaxation to avoid some conservation. A trade-off can be made between feasibility and conservation by defining the following function:

$$b_\eta(\boldsymbol{x}, \alpha) = L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})u_{\min} + \alpha\gamma(b(\boldsymbol{x}))) \tag{4.19}$$

where $\alpha \in [0,1]$. Defining

$$\mathcal{C}_\eta(\alpha) = \{\boldsymbol{x} : b_\eta(\boldsymbol{x}, \alpha) \geq 0\}, \tag{4.20}$$

it is obvious that

$$\mathcal{C}_\eta(\alpha) \subset \mathcal{C}_F \tag{4.21}$$

The function $b_\eta(\boldsymbol{x}, \alpha)$ describes a series of feasibility constraints, where a trade-off between feasibility and performance is made by α. When $\alpha = 0$,

$$b_\eta(\boldsymbol{x}, 0) = L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x}) u_{\text{min}} \tag{4.22}$$

which yields the tightest feasibility set $\mathcal{C}_\eta(\alpha)$ to enforce invariance and thus the most conservative feasibility condition (which is used in *Theorem 4.1 and 4.2*). When $\alpha = 1$,

$$b_\eta(\boldsymbol{x}, 1) = L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x}) u_{\text{min}} + \gamma(b(\boldsymbol{x})), \tag{4.23}$$

then $b_\eta(x, 1) = b_F(\boldsymbol{x})$ yields the least conservative feasibility constraint with respect to $b_F(\boldsymbol{x})$ and $\mathcal{C}_\eta(\alpha)$ becomes C_F.

Consider the case when $\alpha = 0$. Applying (4.2) to $b_\eta(\boldsymbol{x}, 0)$ yields the following CBF constraint which enforces the forward invariance of $\mathcal{C}_\eta(0)$:

$$L_f^2 b(\boldsymbol{x}) + L_f L_g b(\boldsymbol{x}) u_{\text{min}} + \left(L_g L_f b(\boldsymbol{x}) + L_g^2(b(\boldsymbol{x})) u_{\text{min}}\right) u + \gamma_2(b_\eta(\boldsymbol{x}, 0)) \geq 0 \tag{4.24}$$

where γ_2 is again a class $\mathcal{K}$ function. This is exactly the candidate function defined in (4.12). Furthermore, we can prove *Theorem 4.2* in the context of set invariance without choosing the class $\mathcal{K}$ function to be linear.

Proof. Consider the set $\mathcal{C} \cap \mathcal{C}_\eta(0)$, for each state $\boldsymbol{x}(0) \in \mathcal{C} \cap \mathcal{C}_\eta(0)$, $\boldsymbol{x}(0) \in \mathcal{C}$ and $\boldsymbol{x}(0) \in \mathcal{C}_\eta(0)$ by definition. Under Assumption 4.1, there always exists a controller u such that (4.24) is satisfied, thus renders $\mathcal{C}_\eta(0)$ forward invariant, i.e. $\boldsymbol{x}(t) \in \mathcal{C}_\eta(0)$ for all $t \geq t_0$.

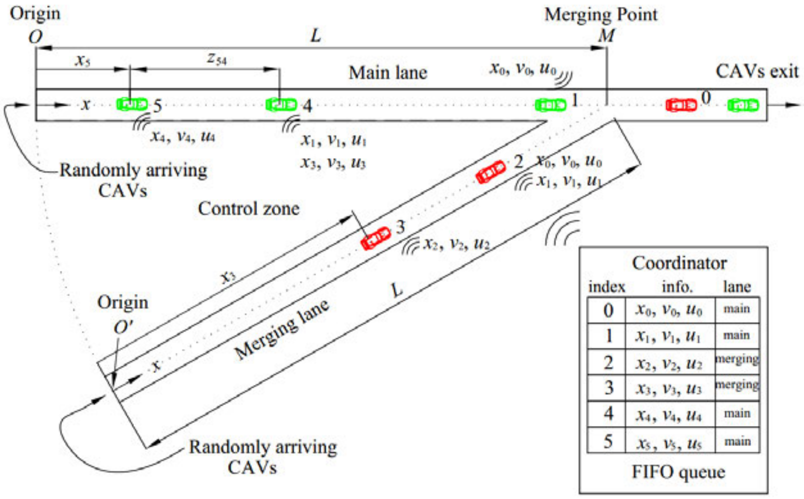
As $\mathcal{C}_\eta(0) \subset \mathcal{C}_F$, according to the definition of a feasibility constraint, the CBF constraint (4.2) is always feasible for $\boldsymbol{x} \in \mathcal{C}_\eta(0)$, thus renders set $\mathcal{C}$ forward invariant, i.e. $\forall \boldsymbol{x}(0) \in \mathcal{C}_\eta(0) \cap \mathcal{C}$, $\boldsymbol{x}(t) \in \mathcal{C}$ for all $t \geq t_0$.

Combing the two forward invariance properties, we can always solve the QP (4.4) to find a controller u such that $x(t) \in \mathcal{C} \cap \mathcal{C}_\eta(0)$ which indicates that the set $\mathcal{C} \cap \mathcal{C}_\eta(0)$ is rendered forward invariant. □

Remark. The system-dependent *Assumption 4.1* is important to guarantee feasibility. In the case study of the traffic control problem which is introduced in the following sections, the satisfaction of this assumption can be proved. Generally speaking, $b_\eta(0)$ provides the most conservative condition and to some extend the best we can do. If *Assumption 4.1* is not satisfied, we should redesign the CBF to enhance feasibility. If we have the feasibility guarantee when *Assumption 4.1* is satisfied but poor performance due to the conservation introduced, α can be adjusted to make a trade-off between feasibility and performance.

4.2 Case Study: Merging Problem Formulation

We consider the traffic merging problem in the sequel to illustrate the application of the feasibility guaranteed OCBF approach. The merging problem arises when traffic must be joined from two different roads, usually associated with a main lane and a merging lane as shown in Fig.4·1. We consider the case where all traffic consists of CAVs randomly arriving at the two roads joined at the Merging Point (MP) M where a collision may occur. A coordinator is associated with the MP whose function is to maintain a First-In-First-Out (FIFO) queue of CAVs based on their arrival time at the CZ and enable real-time Vehicle-to Infrastructure (V2I) communication with the CAVs that are in the CZ, as well as the last one leaving the CZ. The segment from the origin O or O' to the MP M has a length L for both roads, where L is selected to be as large as possible subject to the coordinator's communication range and the road network's configuration and it defines the CZ. Since we consider single-lane roads in this merging problem, CAVs may not overtake each other in the CZ (extensions to multi-lane merging are given in (Xiao et al., 2020)). The FIFO assumption imposed so that CAVs cross the MP in their order of arrival is made for simplicity (and often to ensure fairness), but can be relaxed through dynamic resequencing schemes as in (Xiao and Cassandras, 2020) where optimal solutions are similarly derived but for different selected CAV sequences.



Coordinator

index	info.	lane
0	x_0, v_0, u_0	main
1	x_1, v_1, u_1	main
2	x_2, v_2, u_2	merging
3	x_3, v_3, u_3	merging
4	x_4, v_4, u_4	main
5	x_5, v_5, u_5	main

FIFO queue

Figure 4·1: The merging problem. CAVs randomly arrive at the origins of the main and merging roads. Collisons may occur at the MP. A coordinator is associated with the MP to maintain the FIFO queue and share information among CAVs as needed.

Let $S(t)$ be the set of FIFO-ordered indices of all CAVs located in the CZ at time t along with the CAV (whose index is 0 as shown in Fig.6·2) that has just left the CZ. Let $N(t)$ be the cardinality of $S(t)$. Thus, if a CAV arrives at time t it is assigned the index $N(t)$. All CAV indices in $S(t)$ decrease by one when a CAV passes over the MP and the vehicle whose index is -1 is dropped.

The vehicle dynamics for each CAV $i \in S(t)$ along the lane to which it belongs take the form

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ u_i(t) \end{bmatrix} \tag{4.25}$$

where $x_i(t)$ denotes the distance to the origin O (O') along the main (merging) lane if the vehicle i is located in the main (merging) lane, $v_i(t)$ denotes the velocity, and $u_i(t)$ denotes the control input (acceleration). We consider two objectives for each CAV subject to three constraints, as detailed next.

Objective 1 (Minimizing travel time): Let t_i^0 and t_i^m denote the time that CAV $i \in S(t)$ arrives at the origin O or O' and the MP M, respectively. We wish to minimize the travel time $t_i^m - t_i^0$ for CAV i.

Objective 2 (Minimizing energy consumption): We also wish to minimize energy consumption for each CAV $i \in S(t)$ expressed as

$$J_i(u_i(t)) = \int_{t_i^0}^{t_i^m} C(u_i(t))dt, \tag{4.26}$$

where $C(\cdot)$ is a strictly increasing function of its argument.

Constraint 1 (Safety constraints between i and i_p): Let i_p denote the index of the CAV which physically immediately precedes i in the CZ (if one is present). We require that the distance $z_{i,i_p}(t) := x_{i_p}(t) - x_i(t)$ be constrained so that

$$z_{i,i_p}(t) \geq \varphi v_i(t) + \delta, \;\; \forall t \in [t_i^0, t_i^m], \tag{4.27}$$

where $v_i(t)$ is the speed of CAV $i \in S(t)$ and φ denotes the reaction time (as a rule, $\varphi = 1.8$s is used, e.g., (Vogel, 2003)). If we define z_{i,i_p} to be the distance from the center of CAV i to the center of CAV i_p, then δ is a constant determined by the length of these two CAVs (generally dependent on i and i_p but taken to be a constant for simplicity).

Constraint 2 (Safe merging between i and $i-1$): When $i-1 = i_p$, this constraint is redundant since (6.3) is enforced, but when $i-1 \neq i_p$ there should be enough safe space at the MP M for a merging CAV to cut in, i.e.,

$$z_{i,i-1}(t_i^m) \geq \varphi v_i(t_i^m) + \delta. \tag{4.28}$$

Note that the safe merging is a terminal constraint, which is only active when the ego vehicle arrives at the merging point.

Constraint 3 (Vehicle limitations): Finally, there are constraints on the speed and acceleration for each $i \in S(t)$, i.e.,

$$v_{\min} \leq v_i(t) \leq v_{\max}, \forall t \in [t_i^0, t_i^m], \tag{4.29}$$

$$u_{i,\min} \leq u_i(t) \leq u_{i,\max}, \forall t \in [t_i^0, t_i^m], \tag{4.30}$$

where $v_{\max} > 0$ and $v_{\min} \geq 0$ denote the maximum and minimum speed allowed in the CZ, while $u_{i,\min} < 0$ and $u_{i,\max} > 0$ denote the minimum and maximum control input, respectively.

Optimization Problem Formulation. Our goal is to determine a control law (as well as optimal merging time t_i^m) to achieve objectives 1-2 subject to constraints 1-3 for each $i \in S(t)$ governed by the dynamics (4.25). The common way to minimize energy consumption is by minimizing the control input effort $u_i^2(t)$, noting that the OCBF method allows for more elaborate fuel consumption models, e.g., as in (Kamal et al., 2013a). By normalizing travel time and $u_i^2(t)$, and using $\alpha \in [0,1)$, we construct a convex combination as follows:

$$J_i(u_i(t), t_i^m) = \int_{t_i^0}^{t_i^m} \left(\alpha + \frac{(1-\alpha)\frac{1}{2}u_i^2(t)}{\frac{1}{2}\max\{u_{i,\max}^2, u_{i,\min}^2\}} \right) dt. \tag{4.31}$$

If $\alpha = 1$, then we solve (4.31) as a minimum time problem. Otherwise, by defining

$$\beta := \frac{\alpha \max\{u_{i,\max}^2, u_{i,\min}^2\}}{2(1-\alpha)}$$

and multiplying (4.31) by $\frac{\beta}{\alpha}$, we have:

$$J_i(u_i(t), t_i^m) := \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} \frac{1}{2} u_i^2(t) dt, \tag{4.32}$$

where $\beta \geq 0$ is a weight factor that can be adjusted to penalize travel time relative to the energy cost, subject to (4.25), (4.27)-(4.30) and the terminal constraint $x_i(t_i^m) = L$, given $t_i^0, x_i(t_i^0), v_i(t_i^0)$.

4.3 Feasibility-Guaranteed OCBF in Merging Problem

In the merging control problem, the form of the safety constraint depends on whether CAV i and CAV $i-1$ are in the same road. If so, $i-1=i_p$ and the rear-end safety constraint needs to be considered. Otherwise, the safe merging constraint (4.28) must be included.

4.3.1 Rear-end Safety Constraint

When CAV i and $i-1=i_p$ are in the same road, only the rear-end safety constraint needs to be considered:

$$b_1(\boldsymbol{x}(t)) = z_{i,i_p}(t) - \varphi v_i(t) + \delta \geq 0 \tag{4.33}$$

Note that in (4.33) only CAV i_p's position $x_{i_p}(t)$ is needed in addition to CAV i's state $\boldsymbol{x}_i(t)$, which can be easily implemented in a decentralized way. As $b_1(\boldsymbol{x}(t))$ is differentiable, we can calculate the Lie derivatives $L_f(b_1(\boldsymbol{x}(t))) = v_{i_p} - v_i(t)$, $L_g(b_1(\boldsymbol{x}(t))) = -\varphi$. Applying (4.2) and choosing a linear function $\gamma(x) = k_1 x$ as the class-$\mathcal{K}$ function, the rear-end safety constraint (4.33) can be directly transformed into the CBF constraint:

$$b_{\text{cbf}_1}(\boldsymbol{x}, u_i) = v_{i_p} - v_i - \varphi u_i + k_1 b_1(\boldsymbol{x}) \geq 0 \tag{4.34}$$

As $-L_g(b_1(\boldsymbol{x}(t))) = \varphi > 0$, multiplying both sides of the control bound (4.30) by $-L_g(b_1(\boldsymbol{x}(t)))$ will not change the direction of the inequalities. Hence, we have

$$\varphi u_{i,\min} \leq \varphi u_i(t) \leq \varphi u_{i,\max} \tag{4.35}$$

Note that (4.34) can be rewritten as

$$\varphi u_i(t) \leq v_{i_p}(t) - v_i(t) + k_1 b_1(\boldsymbol{x}(t)) \tag{4.36}$$

where $\varphi u_i(t) \leq u_{i,\max}$ never conflicts with (4.36) as they have the same inequality direction.

Thus, we can guarantee that (4.34) and (4.35) are conflict-free by adding

$$b_F(\boldsymbol{x}(t)) = v_{i_p}(t) - v_i(t) + k_1 b_1(\boldsymbol{x}(t)) - \varphi u_{i,\min} \geq 0 \tag{4.37}$$

Following the derivation in Sec 4.1.1, we can now consider this feasibility constraint as a new CBF and apply (4.2) to transform it into a CBF constraint. Choosing a linear function $\gamma(x) = k_1 x$ as the class $\mathcal{K}$ function, the corresponding CBF constraint becomes

$$\begin{aligned} &u_{i_p} - u_i + k_1(v_{i_p} - v_i - \varphi u_{i,\min}) \\ &\qquad + k_1(v_{i_p} - v_i - \varphi u_i + k_1(z_{i,i_p} - \varphi v_i + \delta)) \geq 0 \end{aligned} \tag{4.38}$$

Define a candidate function $\eta(\boldsymbol{x}, u_i)$ as

$$\eta_1(\boldsymbol{x}, u_i) = u_{i_p} - u_i + k_1(v_{i_p} - v_i - \varphi u_{i,\min}) \tag{4.39}$$

Then, replacing the first three terms of the feasibility CBF constraint (4.38) with $\eta_1(\boldsymbol{x}, u_i)$ and noting that the remaining terms are given by $b_{\text{cbf}_1}(\boldsymbol{x}, u_i)$ defined in (4.34), to substitute the second row with $b_{\text{cbf}_1}(\boldsymbol{x}, u_i)$, (4.38) becomes

$$\eta_1(\boldsymbol{x}, u_i) + k_1 b_{\text{cbf}_1}(\boldsymbol{x}, u_i) \geq 0 \tag{4.40}$$

Since $b_{\text{cbf}_1}(\boldsymbol{x}, u_i) \geq 0$ is required in (3.51), it follows that (4.38) will be satisfied if

$$\eta_1(\boldsymbol{x}, u_i) \geq 0 \tag{4.41}$$

Following the derivation in Sec 4.1.1, we have

$$b_{\eta_1}(\boldsymbol{x}) = v_{i_p} - v_i - \varphi u_{i,\min} \tag{4.42}$$

$$h_1(\boldsymbol{x}, u) = u_{i,p} - u_i \tag{4.43}$$

We will show next that (4.41) is a constraint that guarantees the feasibility of the QP corresponding to the next time interval. Before establishing this result, we make the following assumption.

Assumption 4.4. *All CAVs have the same minimum acceleration, i.e.* $u_{i,\min} = u_{\min}$.

This is a weak assumption guaranteeing that (4.27) and (4.30) are conflict-free, thus *Assumption 4.1* is true. It can be easily enforced since all CAVs are operating within the same CZ, i.e., they can reach agreement on a common $u_{\min} = \min_i\{u_{i,\min}\}$.

Lemma 4.1. *Under Assumption 4.4, Assumption 4.1 is satisfied, i.e.*

$$\forall \boldsymbol{x} \in \mathcal{C}, \quad \sup_{u \in U} h_1(\boldsymbol{x}, u) \geq 0,$$

where $\mathcal{C} = \{\boldsymbol{x} : b_1(\boldsymbol{x}) \geq 0\}$ *and* $h_1(\boldsymbol{x}, u)$ *is defined in* (4.43).

Proof. By *Assumption 4.4*, $u_{i_p} \in [u_{\min}, u_{\max}]$. Taking $u = u_{i_p}$, $h_1(\boldsymbol{x}, u) = 0$, thus there always exists a control input $u(t) \in [u_{\min}, u_{\max}]$ such that $h_1(\boldsymbol{x}, u) \geq 0$. □

Theorem 4.3. *If* $b_{\eta_1}(\boldsymbol{x}(t)) \geq 0$ *and the QP* (4.4) *subject to* (4.34), (4.30) *and* (4.41) *is feasible at time t, then the QP corresponding to time* $t + \Delta t$ *is also feasible.*

Proof. See Appendix A. □

Assumption 4.5. *The following initial conditions are satisfied:*

$$b_1(\boldsymbol{x}(t_i^0)) \geq 0, b_F(\boldsymbol{x}(t_i^0)) \geq 0, b_{\eta_1}(\boldsymbol{x}(t_i^0)) \geq 0$$

The constraint $b_1(\boldsymbol{x}(t_i^0)) \geq 0$ requires CAV i to meet the rear-end safety with the immediately preceding CAV (if one exists) when entering the CZ. In addition, $b_F(\boldsymbol{x}(t_i^0)) \geq 0$ requires that the CBF constraint is initially conflict-free with the control bounds and $b_{\eta_1}(\boldsymbol{x}(t_i^0)) \geq 0$ indicates that CAV i should not be too faster than the preceding CAV. These constraints are reasonable and can be met using a Feasibility Enforcement Zone (FEZ) (Zhang et al., 2017) that precedes the CZ.

Theorem 4.4. *Under Assumptions 4.2, 4.4 and 4.5, the QP* (4.4) *subject to* (4.34), (4.30) *and* (4.41) *corresponding to any time interval* $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t] \subset [t_i^0, t_i^m]$ *is feasible.*

Proof. See Appendix A. □

4.3.2 Safe Merging Constraint

When CAVs i and $i-1$ are in different roads, they should also satisfy the merging safety constraint

$$z_{i,i-1}(t_i^m) - \varphi v_i(t_i^m) - \delta \geq 0 \tag{4.44}$$

This differs from the rear-end safety constraint in that it only applies to specific time instants t_i^m. This poses a technical complication as a CBF must always be in a continuously differentiable form. We can convert (4.44) to such a form using a technique similar to the one in (Xiao et al., 2021b) to define

$$b_2(\boldsymbol{x}(t)) = z_{i,i-1}(t) - \Phi(\boldsymbol{x}(t))v_i(t) - \delta \geq 0 \tag{4.45}$$

where $\Phi(\boldsymbol{x}(t)) = \frac{\varphi}{x_i(t_i^m)} x_i(t)$. Note that $\Phi(\boldsymbol{x}(t_i^m)) = \varphi$ consistent with (4.44). Setting $\varphi_2 = \frac{\varphi}{x_i(t_i^m)}$ and omitting the argument t, we get $L_f(b_2(\boldsymbol{x})) = v_{i-1} - v_i - \varphi_2 v_i^2$ and $L_g(b_2(\boldsymbol{x})) = -\varphi_2 x_i$. Thus, using (4.2) and choosing $\gamma(x) = k_2 x$, (4.44) is mapped onto the CBF constraint

$$b_{\text{cbf}_2}(\boldsymbol{x}, u_i) = v_{i-1} - v_i - \varphi_2 v_i^2 - \varphi_2 x_i u_i + k_2 b_2(\boldsymbol{x}) \geq 0 \tag{4.46}$$

Proceeding as in Sec. 4.3.1, we define

$$h_2(\boldsymbol{x}, u) = u_{i-1} - u_i - 2\varphi_2 v_i u_i - \varphi_2 v_i u_{\min} \tag{4.47}$$

$$b_{\eta_2}(\boldsymbol{x}) = v_{i-1} - v_i - \varphi_2 v_i^2 - \varphi_2 x_i u_{\min} \tag{4.48}$$

$$\eta_2(\boldsymbol{x}, u_i) = h(\boldsymbol{x}, u) + k_2(b_{\eta_2}(\boldsymbol{x})) \tag{4.49}$$

We can then obtain similar theorems as before by deriving a condition corresponding to (4.41):

$$\eta_2(\boldsymbol{x}, u_i) \geq 0 \tag{4.50}$$

Lemma 4.2. *Under Assumption 4.4, if $v_i \geq 0, u_{\min} \leq 0$,, then Assumption 4.1 is satisfied,* *i.e.*

$$\forall \boldsymbol{x} \in \mathcal{C}, \quad \sup_{u \in U} h_2(\boldsymbol{x}, u) \geq 0,$$

where $\mathcal{C} = \{\boldsymbol{x} : b_2(\boldsymbol{x}) \geq 0\}$ *and* $h_2(\boldsymbol{x}, u)$ *is defined in* (4.47).

Proof. See Appendix A. □

Theorem 4.5. *If* $b_{\eta_2}(\boldsymbol{x}(t)) \geq 0$*,* $v_i \geq 0, u_{\min} \leq 0$ *and the QP* (4.4) *subject to* (4.46), (4.30) *and* (4.50) *is feasible at time t, then the QP corresponding to time* $t + \Delta t$ *is also feasible.*

Proof. See Appendix A. □

Assumption 4.6. *The following initial conditions are satisfied:*

$$b_2(\boldsymbol{x}(t_i^0)) \geq 0, b_F(\boldsymbol{x}(t_i^0)) \geq 0, b_{\eta_2}(\boldsymbol{x}(t_i^0)) \geq 0$$

Theorem 4.6. *Under Assumptions 4.2, 4.4, 4.6, if* $v_i \geq 0, u_{\min} \leq 0$*, the QP* (4.4) *subject to* (4.46), (4.30) *and* (4.50) *corresponding to any time interval* $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t] \subset [t_i^0, t_i^m]$ *is feasible.*

Proof. See Appendix A. □

Note that when both the rear-end safety constraint and the safe merging constraint are included, the feasibility of the QP can still be guaranteed by adding one feasibility constraint corresponding to each CBF constraint, i.e. (4.41) and (4.50).

Theorem 4.7. *If* $b_{\eta_1}(\boldsymbol{x}(t)) \geq 0$*,* $b_{\eta_2}(\boldsymbol{x}(t)) \geq 0$*,* $v_i \geq 0, u_{\min} \leq 0$*, the QP* (4.4) *subject to* (4.34), (4.46), (4.30), (4.41) *and* (4.50) *is feasible at time t, then the QP corresponding to time* $t + \Delta t$ *is also feasible.*

Proof. See Appendix A. □

Theorem 4.8. *Under* Assumption 4.5 *and* Assumption 4.6*, if* $v_i \geq 0, u_{\min} \leq 0$*, the QP* (4.4) *subject to* (4.34), (4.46), (4.30), (4.41) *and* (4.50) *corresponding to any time interval* $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t] \subset [t_i^0, t_i^m]$ *is feasible.*

Proof. See Appendix A. □

4.4 Simulation Results

In this section, simulations are conducted to validate and assess the effect of the feasibility constraints added onto the OCBF framework. All simulations are performed in MATLAB using quadprog as the solver for the QPs.

We first build the model shown in Fig. 4·1 with simulation parameters $L = 400, u_{\min} = -2m/s^s, u_{\max} = 3m/s^2$ and adopt the OCBF controller without feasibility guarantee for each CAV. In some situations, a QP for optimally tracking the unconstrained optimal control trajectory of a CAV i becomes infeasible. We record the indices of such CAVs and consider two different cases corresponding to the rear-end safety constraint and the safe merging constraint separately. We re-run the simulations of the two cases with feasibility constraints added to the ego CAV, keeping all other conditions same. The simulation results are detailed in what follows.

4.4.1 Rear-end Safety Constraint

A particular CAV, labeled “CAV 25”, is chosen as the first case study. As CAV 25 and CAV 24 are in the same road, the possibly active constraint of interest is the rear-end safety constraint. We adopt the OCBF controller and run the simulation twice to derive the two trajectories of CAV 25, one with the feasibility guarantee and the other without. Note that in the merging problem, the control $u_i(t)$ is 1-dimensional, thus the feasible set of the QP is an interval. To illustrate the performance of the feasibility constraint, the evolution of the feasible set of the QPs over time are plotted in Fig. 4·2.

In Fig. 4·2, the solid blue curve shows the control history $u(t)$ generated by the OCBF controller. The shaded blue area shows the feasible set of the QP. For each time t, the shaded blue area marks the maximum and minimum acceleration allowed by the QP. Note that when $t = 85s$, the QP becomes infeasible and the control is set to be $-2m/s^2$ to continue the program execution. The solid red line is the control history generated by the OCBF

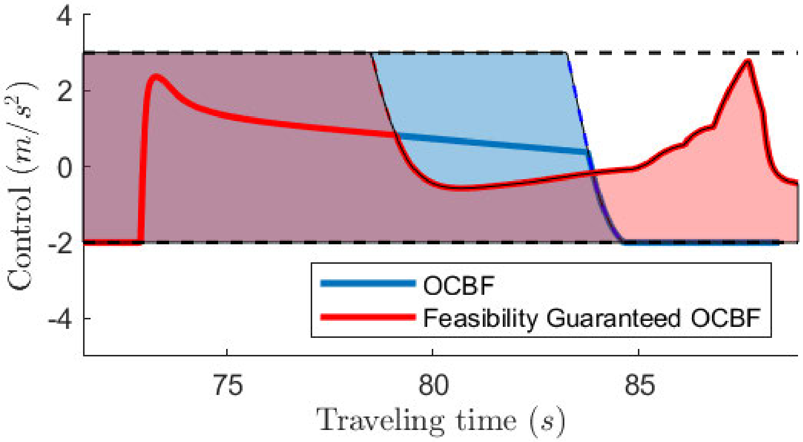


Figure 4·2: Control History Comparison

controller with the feasibility constraint added to the QP. The shaded red area shows the feasible set corresponding to the revised QP. The dashed black lines shows the control bounds of the CAV.

The figure shows that the OCBF and feasibility guaranteed OCBF have the same feasible set before 79s, which generates the same control history. However, after 79s, the feasible set of the feasibility guaranteed OCBF shrinks due to the feasibility constraint while the myopic OCBF approach keeps the same feasible set. This leads to a large difference after 84s. The feasible set of the myopic OCBF approach rapidly shrinks and even becomes empty, indicating that the QP is infeasible. The feasibility guaranteed OCBF, however, remains feasible with the help of the advance action introduced by the feasibility constraint.

The infeasible QP makes the safety constraints unguaranteed as we no longer benefit from the forward invariance property of the CBF. The rear-end safety constraints of the two cases are plotted in Fig. 4·3, where the blue curve corresponds to the OCBF controller and the red curve corresponds to the feasibility guaranteed OCBF controller. From the figure, we can see that the rear-end safety constraint is violated after 85s using the OCBF controllers. This corresponds to the infeasible QP shown in Fig. 4·2 after 85s. With the

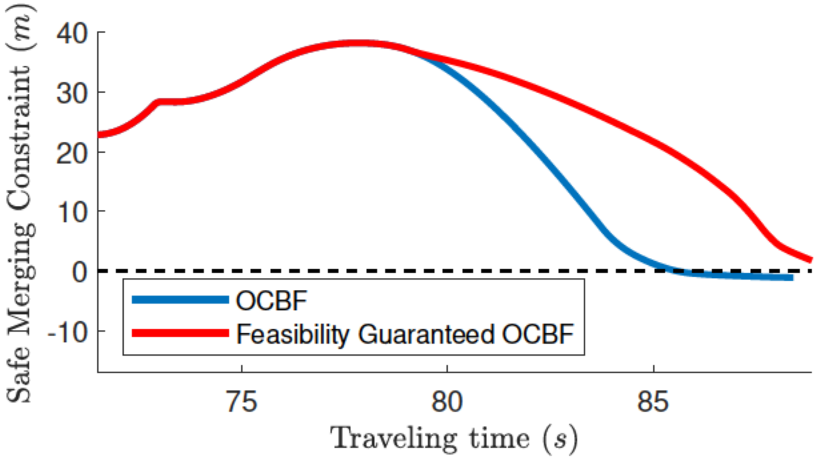


Figure 4·3: Rear-end Safety Constraint Satisfaction Comparison

help of the feasibility constraint, the red curve remains positive, indicating that the rear-end safety constraint is always satisfied.

4.4.2 Safe Merging Constraint

The second case study focuses on CAV 17. As CAV 17 and CAV 16 are in different roads, the safe merging constraint needs to be considered. Following the same steps, we plot the control histories, as well as the safe merging constraints, of the two different approaches in Fig. 4·4 and Fig. 4·5.

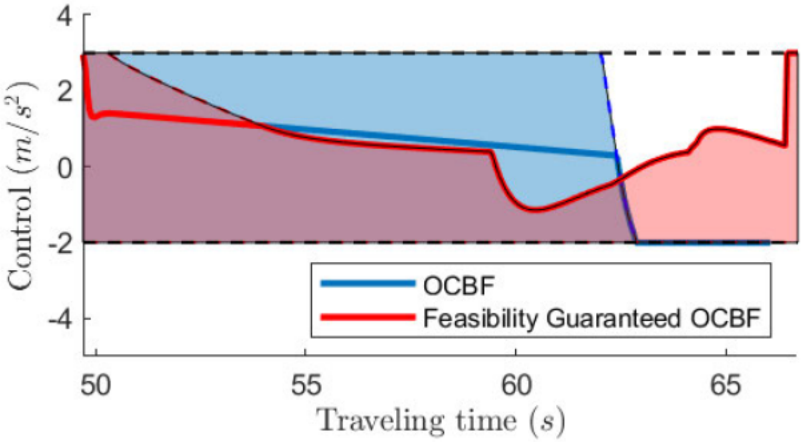


Figure 4·4: Control History Comparison

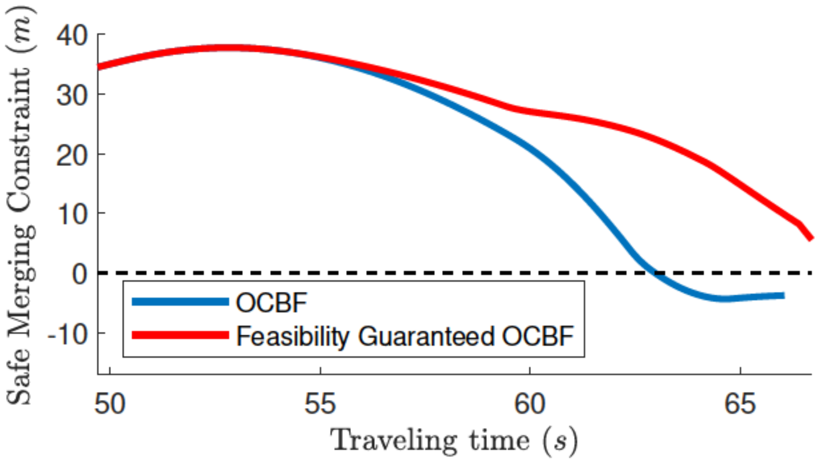


Figure 4·5: Safe Merging Constraint Satisfaction Comparison

The simulation results are similar to the first case. The feasibility guaranteed OCBF is less myopic than the standard OCBF method and controls the CAV to decelerate earlier for safety, before it is impossible to do so. Unlike the rear-end safety constraint which is relatively easier to be satisfied (as shown in Fig. 4·3), the safe merging constraint is more likely to be violated. As is shown in Fig. 4·5, when the QP becomes infeasible after 63s using OCBF, the safe merging constraint is violated and finally results in a potential car crash at 66s when merging. The feasibility guaranteed OCBF, however, avoids this problem with the feasibility constraint included.

4.5 Summary

In this chapter, we have presented a feasibility guaranteed decentralized control framework combining optimal control and CBFs leading to OCBF controllers for merging CAVs jointly minimizing both the travel time and the energy consumption subject to speed-dependent safety constraints, as well as vehicle limitations. We resolve the QP feasibility problem which arises when the OCBF controller is used in decentralized merging control by adding a single control-dependent feasibility constraint corresponding to each CBF constraint. Proofs are provided and the results are illustrated through simulations of a merging problem to demonstrate the effectiveness of this approach. The explicit control-dependent feasibility constraints we have derived rely on the special velocity-dependent safety constraint structure of the merging control problem.

Chapter 5

Learning the Optimal Control Problem Solution

In this chapter, we introduce the machine learning method to find a better reference trajectory for the OCBF controller to track which has a huge influence on the performance of the controller. The unconstrained optimal solution used in the previous roundabout problem often deviates largely from the optimal trajectory, especially when the vehicle limitation is tight. To find a better reference trajectory that is achievable in real time, we also start from the merging control problem as what we did in deriving the feasibility guaranteed OCBF (See Chapter 4. Although the complete optimal solution can not be calculated in real time, they are achievable in off-line calculation, thus gives us the possibility to learn a better referenced trajectory using neural network based on data generated from the complete Optimal Control Problem (OCP) solution.

5.1 Problem Formulation

We consider the traffic merging problem in the sequel to learn an optimal solution from. The merging problem is formulated in detail in Section 4.2 and revisited briefly in the follow sections.

5.1.1 Merging Problem Revisit

The merging problem arises when traffic must be joined from two different roads, usually associated with a main lane and a merging lane as shown in Fig.4·1, where we consider the case where all traffic consists of CAVs randomly arriving at the two roads joined at the

Merging Point (MP) M where a collision may occur.

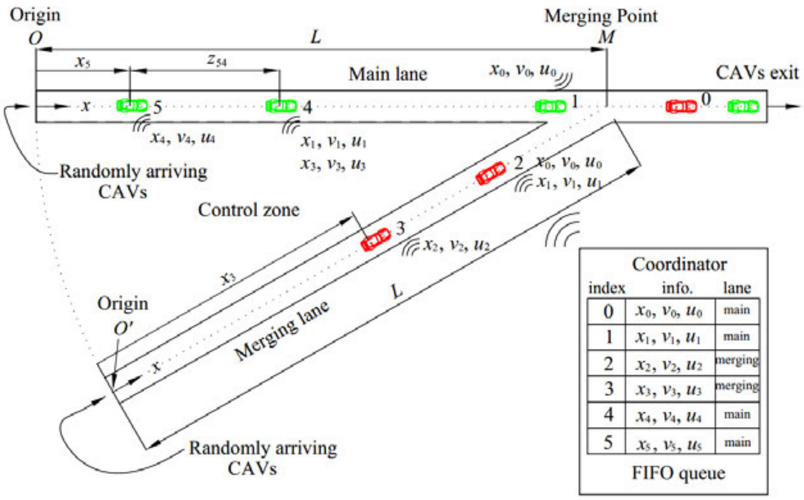


Figure 5·1: The merging problem. CAVs randomly arrive at the origins of the main and merging roads. Collisons may occur at the MP. A coordinator is associated with the MP to maintain the FIFO queue and share information among CAVs as needed.

Let $S(t)$ be the set of FIFO-ordered indices of all CAVs located in the CZ at time t along with the CAV (whose index is 0 as shown in Fig.5·1) that has just left the CZ. The vehicle dynamics for each CAV $i \in S(t)$ along the lane to which it belongs take the form

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ u_i(t) \end{bmatrix} \tag{5.1}$$

where $x_i(t)$ denotes the distance to the origin O (O') along the main (merging) lane if the vehicle i is located in the main (merging) lane, $v_i(t)$ denotes the velocity, and $u_i(t)$ denotes the control input (acceleration). We consider the convex combination of two objectives, minimizing travel time and minimizing energy consumption, for each CAV subject to rear-end safety constraint, safe merging constraint and vehicle limitations and thus formulate the optimal control problem as follows:

$$\min_{u_i, t_i^m} \quad J_i(u_i(t), t_i^m) := \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} \frac{1}{2} u_i^2(t) dt \tag{5.2}$$

$$\text{s.t.} \quad z_{i,i_p}(t) \geq \varphi v_i(t) + \delta, \quad \forall t \in [t_i^0, t_i^m], \tag{5.3}$$

$$z_{i,i_m}(t_i^m) \geq \varphi v_i(t_i^m) + \delta, \tag{5.4}$$

$$v_{\min} \leq v_i(t) \leq v_{\max}, \quad \forall t \in [t_i^0, t_i^m], \tag{5.5}$$

$$u_{i,\min} \leq u_i(t) \leq u_{i,\max}, \quad \forall t \in [t_i^0, t_i^m], \tag{5.6}$$

$$x_i(t_i^m) = L, \quad \text{given } t_i^0, x_i(t_i^0), v_i(t_i^0) \tag{5.7}$$

Detailed explanations on the objective (5.2), the safety constraints (5.3), (5.4) and the vehicle limitations (5.5), (5.6) can be found in Section 4.2.

5.1.2 Solution to the Merging Problem

The solution to the merging problem can be obtained through Hanmiltonian analysis. The Hanmiltonian analysis starts from the unconstrained problem where all constraints are assumed to be inactive. Following the same analysis as in Section 3.3.2 where we calculate the unconstrained optimal solution to the lower level problem of straight road segments in the roundabout problem, the unconstrained optimal solution to the merging problem can be explicitly obtained as:

$$\begin{aligned} u_i^*(t) &= a_i t + b_i \\ v_i^*(t) &= \frac{1}{2} a_i t^2 + b_i t + c_i \\ x_i^*(t) &= \frac{1}{6} a_i t^3 + \frac{1}{2} b_i t^2 + c_i t + d_i \end{aligned} \tag{5.8}$$

where the integration constants a_i, b_i, c_i, d_i and the terminal time t_i^m can be obtained by solving the set of equations

$$\begin{aligned}
&\frac{1}{2}a_i \cdot (t_i^0)^2 + b_i \cdot t_i^0 + c_i = v_i^0, \\
&a_i t_i^0 + b_i = 0, \\
&\frac{1}{6}a_i \cdot (t_i^0)^3 + \frac{1}{2}b_i \cdot (t_i^0)^2 + c_i t_i^0 + d_i = 0, \\
&\frac{1}{6}a_i \cdot (t_i^m)^3 + \frac{1}{2}b_i \cdot (t_i^m)^2 + c_i t_i^m + d_i = L, \\
&\beta - \frac{1}{2}b_i^2 + a_i c_i = 0,
\end{aligned} \tag{5.9}$$

which is not difficult to solve numerically. By solving the equations and replacing the integration constants into (5.8), the unconstrained optimal solution to the merging problem can be achieved in real time.

When one of more constraints becomes active, the solution to the optimal control problem becomes computationally intensive, thus restricts its application in real time. However, the complete solution to the merging problem is still achievable in reasonable time using the method in (Xiao and Cassandras, 2021b), thus provides the ability to generate the complete optimal trajectories off-line and learn a reference trajectory using the neural network. Denote the optimal trajectory to the merging problem as $f_{OCP}(\boldsymbol{x}, t)$, the goal of learning a reference trajectory is to find a function $f_{NN}(\boldsymbol{x}, \alpha)$ in the form of a neural network that has the minimum derivation from the optimal trajectory:

$$\min_{\alpha} \int_{t_i^0}^{t_i^m} \frac{1}{2} \| f_{OCP}(\boldsymbol{x}, t) - f_{NN}(\boldsymbol{x}, \alpha) \|^2 dt \tag{5.10}$$

where α denotes the parameters of the neural network with a specific structure and mean square error is used as the performance criterion.

5.2 Learning Framework

The whole process of learning a reference trajectory from the complete optimal trajectory of the merging problem follows the standard framework in machine learning consisting of data generation and data set construction, network design and training and is explained in details in the sequel.

5.2.1 Generation of training data

As aforementioned, the complete optimal solution to the merging control problem can be completely solved through intensive calculation, which restricts its application in real time. Due to the long computation time to get a complete optimal trajectory, it is inefficient and unreasonable to generate such a trajectory corresponding to an initial state of an ego CAV during the training process like what is commonly done in reinforcement learning and imitation learning. However, as the computation can be done in reasonable time off-line, it is possible to generate the data of optimal trajectories corresponding to different initial states and learn a reference trajectory from the data using supervised learning.

The raw data set is defined to include the state of the ego CAV i and the CAV i_c that may conflict with the ego vehicle, as well as the optimal control u_i^* acquired from the complete optimal solution. Each data point in the data set records the discrete time series of a 7-dimensional state $\boldsymbol{x}_{i,\text{raw}}(t)$ consisting of:

(i) State of the ego CAV $x_i(t), v_i(t)$.

(ii) State of the potentially conflicting vehicle $x_{i_c}(t), v_{i_c}(t)$, where i_c refers to the index of the CAV that may conflict with CAV i. Under the FIFO assumption of the merging problem, for each CAV i, only one safety constraint (either rear-end safety or safety merging) needs to be considered. If the rear-end safety constraint is considered, $i_c = i_p$. Otherwise, if the safe merging constraint is considered, $i_c = i_m$.

(iii) Lane information l_i and l_{i_c}. The lane information is necessary to indicate whether i and i_c are in the same lane. For the merging problem, both l_i and l_{i_c} are boolean variables which records the index of the lane CAV i and i_c belongs to respectively. Furthermore, $l_i = l_{i_c}$ indicates that the two CAVs are in the same lane.

(iv) The optimal control $u_i^*(t)$ obtained from the complete optimal solution.

We first build the model shown in Fig. 4·1 in simulation with parameters $\alpha = 0.25, L = 400m, u_{\min} = -2m/s^s, u_{\max} = 3m/s^2$ and calculate the complete optimal solution for each CAV i. In each simulation, CAVs arrive at the merging roadway following the Poisson process with arrival rates carefully selected such that the OCP may encounter some active constraints but is still solvable. The state $\boldsymbol{x}_{i,\text{raw}}(t)$ is recorded in a discrete manner for all $t \in [t_i^0, t_i^m]$ where $t = t_i^0 + k\Delta$, $k \in \mathbb{Z}$ and thus forms a data point in the raw data set. Following the process, the raw data set is constructed which includes the trajectories of around 8000 CAVs in the simulation.

The raw data set is separated into a train set and a test set following the standard 7:3 ratio such that the train set contains 70% of the trajectories in the data set and the test set contains the complementary 30%. Additionally, pre-processing may be needed to adjust the structure of the data based on the neural network structure.

5.2.2 Network Structure

As aforementioned, our aim is to learn a reference trajectory from the complete optimal solution. Unlike the classification and prediction task which only relies on the current state, the trajectory learning task requires information of past states, thus time must be introduced to get a better performance. The standard way to take time into consideration in neural network is to include a Recurrent Neural Network (RNN) such as the Long-short Term Memory (LSTM). A LSTM network is a RNN aimed to deal with the vanishing gradient problem which is composed of a cell, an input gate, an output gate and a forget gate(Gers et al., 1999). The cell remembers values over arbitrarily long terms and the three gates regulates the flow of information into and out of the cell to provide achievable short-term memory.

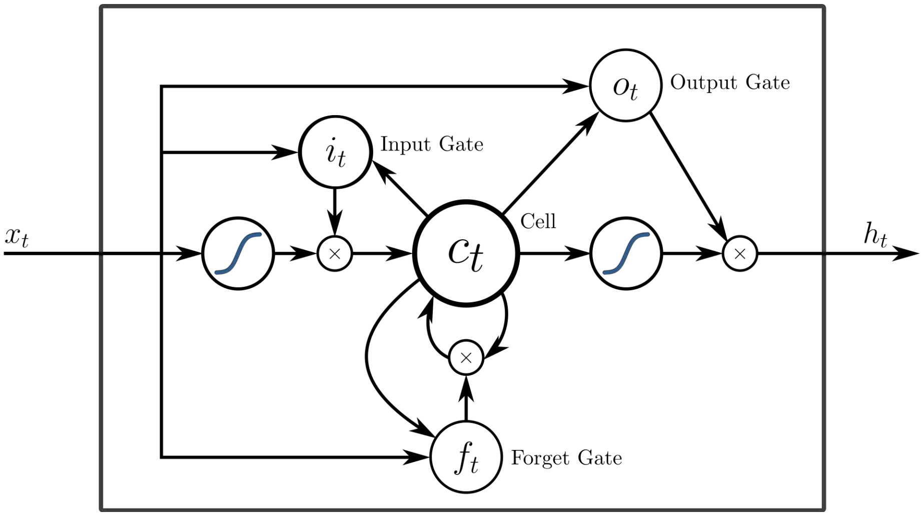


Figure 5·2: LSTM model(Gers et al., 1999)

Based on a LSTM, we design the basic structure of the neural network as shown in 5·3.

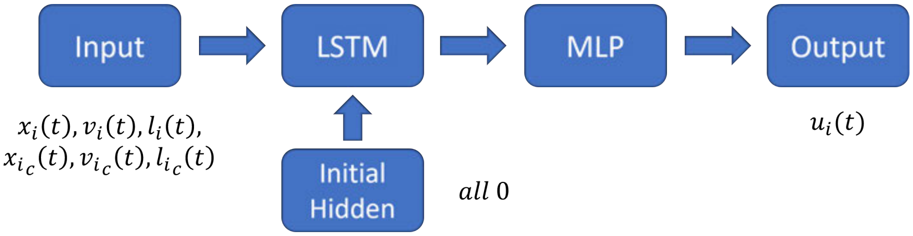

Input
LSTM
MLP
Output
$x_i(t), v_i(t), l_i(t),$
$x_{i_c}(t), v_{i_c}(t), l_{i_c}(t)$
Initial Hidden
$all\ 0$
$u_i(t)$

Figure 5·3: Basic Structure of the Neural Network

The input is a 6-dimensional vector $\boldsymbol{x}(t) = (x_i(t), v_i(t), l_i(t), x_{i_c}(t), v_{i_c}(t), l_{i_c}(t))$, which includes the states and lane information of both the ego CAV and the potentially conflicting CAV. The LSTM has both the hidden layer size and the output size set to be 256 and is connected to a Multilayer Perceptron (MLP) with hidden layer size also 256. The output size of the MLP is 1, which refers to the control learnt from the data and is compared to the label $u_i^*(t)$. The mean square loss is selected as the criterion to learn the parameters in

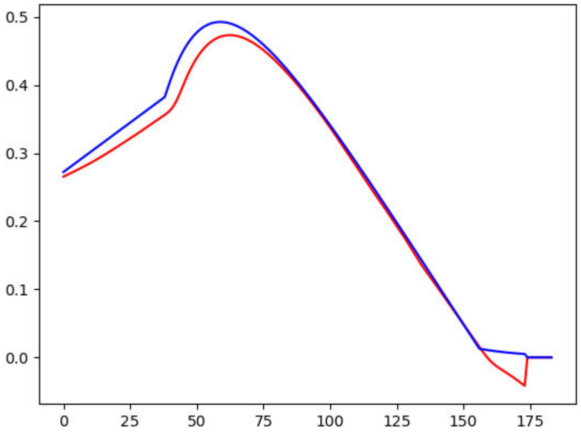


Figure 5·4: Comparison of the learnt trajectory and the optimal trajectory

the neural network model and the Adam optimizer is chosen to learn the parameters during training.

The neural network works in the way as follows. Given a series of data $\boldsymbol{x}(t)$ corresponding to a trajectory from t_i^0 to t_i^m, the input $\boldsymbol{x}(t_i^0)$ is first fed into the LSTM, along with all hidden nodes initialized as 0. The LSTM generates an output which is passed to the MLP to get the control $u(t_i^0)$ and updates the hidden layer. The same process is followed to deal with the data points along the trajectory, however, the hidden layer gets updated in the previous time step and is no longer all 0. In this way, time is introduced to the network by including the history information with the hidden layer in LSTM. Finally, the output $u_i(t)$ is compared with the label $u_i^*(t)$ and the MSE loss is calculated.

This model works well in both training and testing with an MSE loss of around 10^{-6}, which indicates the average deviation from the predicted control and the label is around $10^{-3}m/s^2$. As $u_i(t)$ distributes in $[-2m/s^2, 3m/s^2]$ and heavily accumulates within the range of $[0.2m/s^2, 0.5m/s^2]$, the loss is adequately small. The comparison of the learnt trajectory and the optimal trajectory shown in Fig. 5·4 also shows that the network has learnt the trend of the complete optimal solution.

However, when this model is directly implemented in the simulation such that the control $u = f_{NN}(\boldsymbol{x})$ is exerted on CAV i at each time step, a huge performance loss occurs due to the dependency between $x_i(t), v_i(t)$ and $u_i(t)$. In the offline training session, all data are previously generated. $x_i(t), v_i(t)$ are always accurate and independent with the output $u_i(t)$. However, when the model is tested in an online environment in the simulation, $x_i(t+\Delta t)$ and $v_i(t+\Delta t)$ are influenced by $u_i(t)$. The errors in $u_i(t)$ accumulate through this feedback process and thus results in a large deviation from the label trajectory.

Adjustments are made to the neural network model to resolve the problem of performance difference between offline and online testing due to the input-output dependency. The structure of the network is shown in Fig. 5·5 which decouples $x_i(t), v_i(t)$ with $u_i(t)$ in the training. In this model, $x_i(t)$ and $v_i(t)$ no longer serves as the input, thus the input is decoupled from the output and remains accurate in both offline training and online testing. Only $x_i(t_0)$ and $v_i(t_0)$ are used to initialize the hidden layer of the LSTM, the rest state information of CAV i are used as the label. The adjusted network has an input size of 4 $(x_{i_c}(t), v_{i_c}(t), l_i(t), l_{i_c}(t))$ and an output size of 3 $(x_i(t), v_i(t), u_i(t))$. Besides learning a reference trajectory from the optimal solution, the network now has an underlying goal of learning the dynamics of the CAV at the same time. In the adjusted model, the mean square error is also chosen as the loss function with properly selected weights. As the adjusted model has fewer inputs and more outputs and is more complex than the previous one, it is harder to be trained. However, a small MSE loss can still be acquired at around 10^{-4} both in off-line training and online testing.

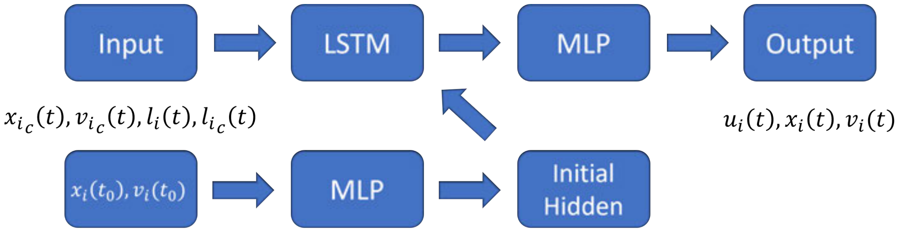


Figure 5·5: Neural Network Model 2

5.3 OCBF with Neural Network

As introduced in Sec. 3.4, the OCBF controller transforms the optimal control problem into an optimal trajectory tracking problem with CBFs providing safety guarantees and CLFs providing convergence to desired states. The problem is decretized along time with time step Δt to form a series of QPs (5.11) which can be solved efficiently in real time.

$$\begin{aligned}\min_{u(t),e(t)} \quad & \beta e(t)^2 + \frac{1}{2}(u(t) - u_{\text{ref}}(t))^2 \\ \text{s.t.} \quad & L_f b(\boldsymbol{x}(t)) + L_g b(\boldsymbol{x}(t))u(t) + \gamma(b(\boldsymbol{x}(t))) \geq 0, \\ & L_f V(\boldsymbol{x}(t)) + L_g V(\boldsymbol{x}(t))u(t) + \varepsilon V(\boldsymbol{x}(t)) \leq e(t), \\ & u_{\min} \leq u(t) \leq u_{\max}, \\ & t = t_0 + k\Delta t\end{aligned} \tag{5.11}$$

where $V(\boldsymbol{x}(t)) = (v(t) - v_{\text{ref}}(t))^2$ is the CLF.

Combing the OCBF with the neural network model proposed in Sec. 5.2 is straightforward. Denote the neural network as a function $\boldsymbol{y}(t) = f_{NN}(\boldsymbol{x}(t), h(t))$ where $\boldsymbol{x}(t) = (x_{i_c}(t), v_{i_c}(t), l_i(t), l_{i_c}(t))$, $\boldsymbol{y}(t) = \left(x_i^{NN}(t), v_i^{NN}(t), u_i^{NN}(t)\right)$ and $h(t)$ is the hidden layer in the LSTM. Starting with $t = t_0$, $h(t_0)$ is initialized with the initial state of the ego vehicle $x_i(t_0)$ and $v_i(t)$. The learnt trajectory $\boldsymbol{y}(t_0)$ is given by taking $\boldsymbol{x}(t_0)$ and $h(t_0)$ into $f_{NN}(\boldsymbol{x}, h)$ which also updates the hidden state into $h(t_1)$. Taking $\boldsymbol{y}(t_0)$ as the reference state such that $u_{\text{ref}}(t_0) = u_i^{NN}(t_0)$ and $v_{\text{ref}}(t_0) = v_i^{NN}(t_0)$ and solving (5.11) gives the control $u_i(t_0)$. Executing the control to the vehicle updates the states of both the ego vehicle and the conflicting vehicle.

For $t = t_k = t_0 + k\Delta t$ and $k = 1 \ldots k_{\max}$, the hidden state $h(t_k)$ is no longer initialized. By taking $\boldsymbol{x}(t_k)$ and $h(t_k)$ into $f_{NN}(\boldsymbol{x}, h)$, both the learnt state $\boldsymbol{y}(t_k)$ and the updated hidden state $h(t_{k+1})$ are acquired. Tracking $\boldsymbol{y}(t_k)$ with (5.11) generates the control $x_i(t_k)$ which is then exerted on the ego vehicle.

Following this process, the control that optimally tracks the learnt trajectory from the neural network with safety guarantees can be acquired in real time.

5.4 Simulation Results

In this section, a simulation is build on the model shown in Fig. 5·1 with the parameters $\alpha = 0.25, L = 400, u_{\min} = -2m/s^2, u_{\max} = 3m/s^2$ same as those when generating the data set. CAVs arrive at the CZ following a Poisson process with different initial velocities and follow their own complete optimal trajectories which are previously calculated offline. Among the optimal trajectories, a trajectory with partially active constraints generated by CAV i is selected as the baseline. The simulation is then rerun following the same process except that CAV i is controlled by an OCBF controller. Both the unconstrained optimal trajectory and the trajectory generated by a neural network are used as the reference trajectory in OCBF and measured in the simulation. The comparison of the standard OCBF (with unconstrained optimal trajectory), NN-based OCBF (with learnt reference trajectory) and the complete optimal solution is plotted in Fig. 5·6.

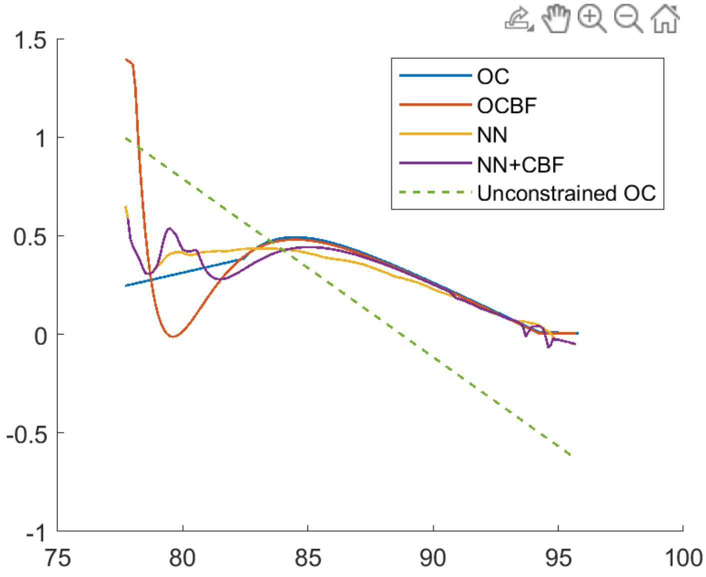


Figure 5·6: Control Trajectory: NN+CBF vs OCBF

The complete optimal control solution is plotted in blue in Fig. 5·6. The CAV enters the merging roadway at around 78s and keeps a linearly increasing acceleration until it reaches the safety constraint at 83s. Given the same initial condition, the unconstrained optimal control trajectory is calculated and plotted as the dashed green line. The unconstrained optimal solution is always linear in time *t* as it neglect all the safety constrains and thus has a large deviation from the OCP solution. Using it as the reference trajectory in OCBF ans tracking it with CBF and CLF generates the red curve, in which we can see a huge drop in the control at the beginning. This will result in a waste in energy consumption and a jerk that may bring discomfort. In comparison, the neural network model manage to generate a trajectory that is closer to the OCP solution as shown in the yellow curve. It has a smaller acceleration at the beginning which reduces the energy cost and manage to follow the tendency of the OCP trajectory to reduce the acceleration when the constraint becomes active. Using the control trajectory generated by the neural network and tracking it using CBFs, the NN-based OCBF controller generates a curve closer to the OCP trajectory as shown in purple (Note: The NN-based OCBF controller is also noted as NN+CBF in the figures and tables). It has small acceleration in the beginning and gets closer to the OCP curve when the constraint becomes active.

Table 5.1: Performance Comparison of OCP, OCBF and NN+CBF

Method	OCP	OCBF	NN+CBF
time (s)	18.20	18.20	18.20
energy	0.967	1.387	0.982
obj.	49.50	49.92	49.52

The performance comparison of OCP, OCBF and NN+CBF are listed in Table 5.1. As the time resolution of the simulation is 0.1s (which means the CAVs in the simulation are updated and recorded every 0.1 simulation seconds), the travel time of all three method are not distinguishable. However, there is a huge difference in the energy consumption and thus brings a difference in the total objective. Comparing with the OCP baseline, OCBF

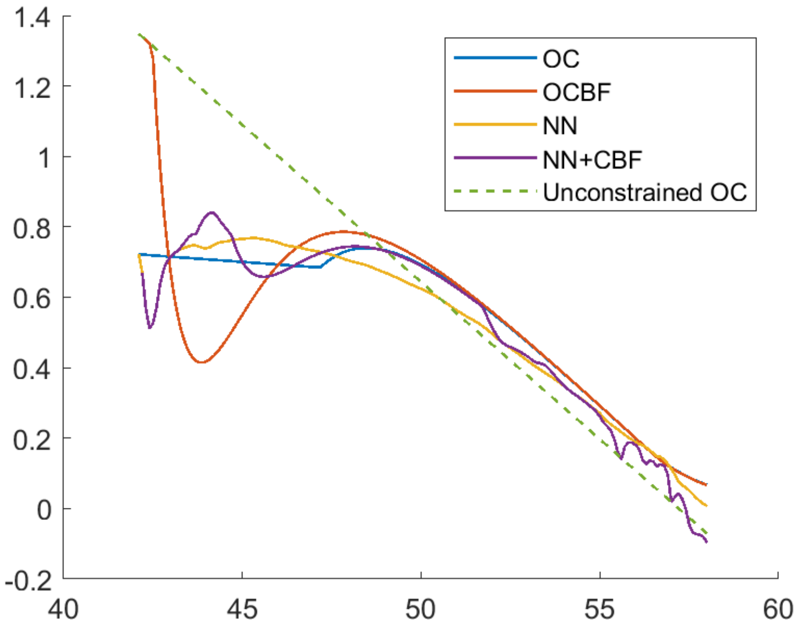


Figure 5·7: Control Trajectory2: NN+CBF vs OCBF

costs about 40% more energy while NN+CBF costs only 1.5% more, which results in a performance loss of 1% and 0.04% respectively. The results shows that for this particular CAV, the neural network manages to generate a referenced trajectory that outperforms the standard OCBF and improves the performance of the OCBF controller.

Multiple trajectories are selected as baselines to illustrate the improvement to the reference trajectory brought by the neural network. Some selected cases are shown in Fig. 5·7 and Fig. 5·8 respectively. The figures show that the NN-based trajectory is much closer to the complete OCP solution than the unconstrained optimal solution although there remains some deviation between the NN curve and the complete OCP curve. Due to the limited samples of constrained optimal solution in the data set, the neural network cannot full reconstruct the complete OCP solution, but manage to learn a trajectory between the unconstrained optimal solution and the OCP solution. As shown in Fig. 5·8, the NN curve in yellow gives the CAV the instruction to accelerate when entering the CZ, which

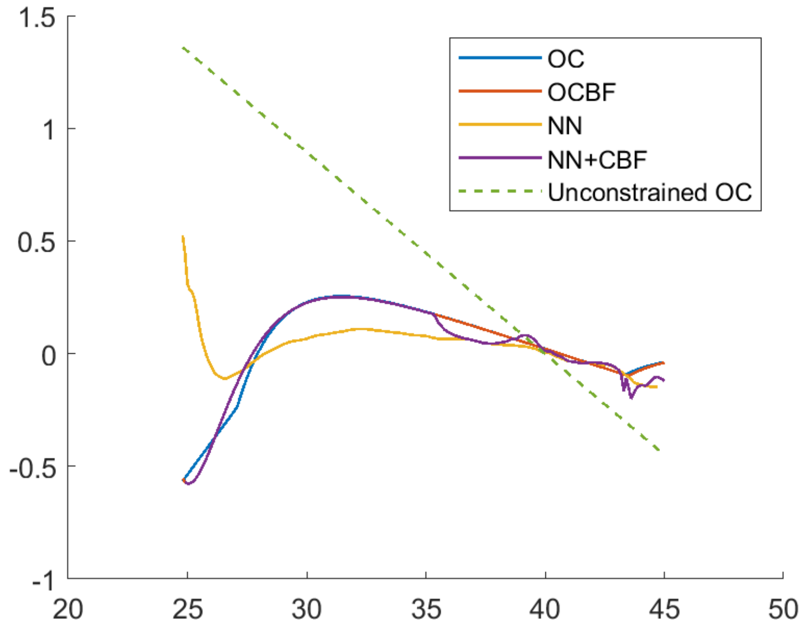


Figure 5·8: Control Trajectory3: NN+CBF vs OCBF

is opposite to the deceleration command given by the OCP solution. However, instead of the aggressive $1.5m/s^2$ control given by the unconstrained optimal trajectory (shown in dashed green), the NN-based trajectory suggests a control $0.5m/s^2$ which happens to be the average of the OCP trajectory and the unconstrained one. The simulation results show that by using a neural network, a better reference trajectory can be generated and utilized to improve the performance of the OCBF controller comparing to using the unconstrained optimal trajectory. Further improvements can be made to enhance the performance of the NN-based trajectory, including adjustment to the NN model, better tuning and learning on a larger data set.

Chapter 6

A Hierarchical Framework of Modular Control Zones

The transition from a single CZ to *multiple* interconnected CZs is particularly challenging for several reasons including the following. First, when studying a CZ in isolation through an optimization problem, it is assumed that the initial conditions for each CAV satisfy the constraints of the problem. When one CZ is followed by another, there is no obvious way to ensure that the state of a CAV exiting the first can always satisfy the initial feasibility conditions of the next CZ's optimization problem. Second, when two adjacent CZs are physically separated by a short distance, the optimal control of CAVs in either CZ may in fact cause congestion in the other, including traffic blocking effects. Thus, both performance degradation and lack of safety guarantees result from a direct application of the techniques developed to date for a single CZ without utilizing the global information of the traffic flow in a system or at least some local information from neighboring CZs.

6.1 Problem Formulation

We model the control of CAVs in a general traffic network under the modular control zone architecture shown in Fig. 6·1. A *Control Zone* (CZ), defined as an area within which CAVs can communicate with each other and with a coordinator, is usually associated with a specific conflict area, including (but not limited to) merging roadways, intersections, and roundabouts. A coordinator (Road Side Unit (RSU)) is associated with each CZ to maintain a certain passing sequence of CAVs based on their arrival time at the CZ and their states; it

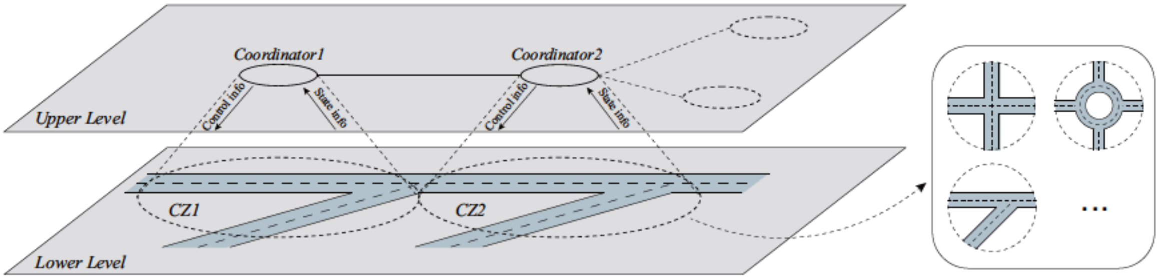


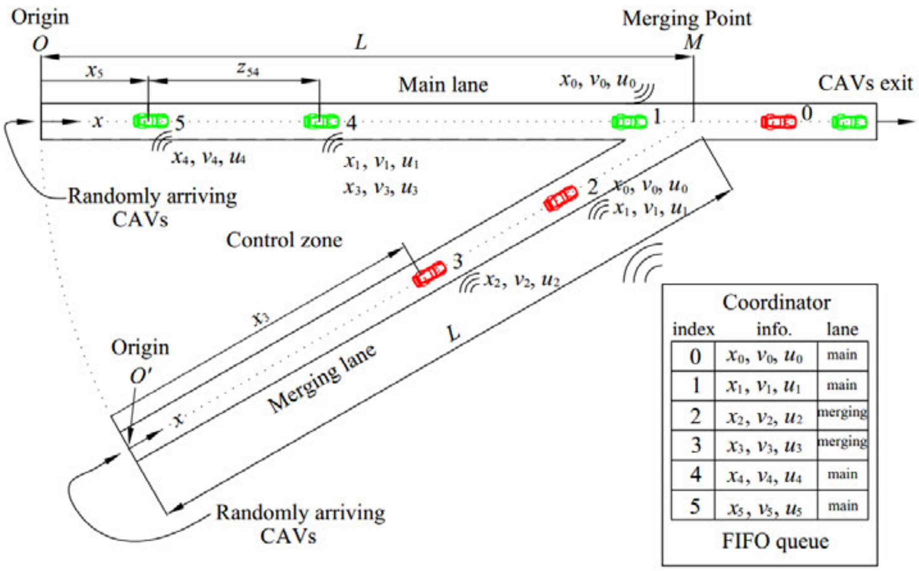
Figure 6·1: Modular Control Zone Architecture

also enables real-time Vehicle-to-Infrastructure (V2I) communication with the CAVs that are in the CZ. In prior work involving a single CZ (Xiao et al., 2021b), the coordinator does not have any control functionality over individual CAVs and is only responsible for maintaining and sharing state information with the CAVs. In the proposed modular CZ framework, a coordinator has the additional function of controlling traffic flows between CZs by sending information to CAVs based on the real-time CZ information acquired through Infrastructure-to-Infrastructure (I2I) communication. To maintain the *decentralized* framework within each CZ, the signals sent by the coordinator to a CAV do not directly affect its control; they are instead parameters that influence the dynamics of each CAV (e.g., its target terminal velocity upon exiting the CZ). These signals also act as an interface between CZs to control their traffic flows.

In this chapter, we limit ourselves to the merging control problem in consecutive CZs as shown in Fig. 6·1 so as to illustrate the framework of modular CZs. The problem is decomposed into (i) a lower level with some modifications to the established decentralized merging control problem (Xiao et al., 2021b), and (ii) a new upper level with the flow control problem. These two levels are detailed next.

6.1.1 Merging Control: Lower Level Problem

We consider the same merging problem as introduced in Sec. 3.1 and review it in details with adjustment to a new terminal constraint. The problem arises when traffic must be joined from two different roads, usually associated with a main road and a merging road as shown in Fig. 6·2. We consider the case where all traffic consists of CAVs randomly arriving at the two roads joined at the *Merging Point* (MP) M where a collision may occur. A coordinator is associated with the MP to maintain a First-In-First-Out (FIFO) queue of CAVs based on their arrival time at the CZ. The segment from the origin O or O' to M has a length L for both roads, where L is selected to be as large as possible subject to the coordinator's communication range and the road network's configuration. Since we consider single-lane roads in this merging problem, CAVs may not overtake each other in the CZ (extensions to multi-lane merging are given in (Xiao et al., 2020)). The FIFO assumption imposed so that CAVs cross the MP in their order of arrival is made for simplicity (and often to ensure fairness), but can be relaxed through dynamic resequencing schemes as in (Xiao and Cassandras, 2020) where optimal solutions are similarly derived but for different selected CAV sequences.



Coordinator

index	info.	lane
0	x_0, v_0, u_0	main
1	x_1, v_1, u_1	main
2	x_2, v_2, u_2	merging
3	x_3, v_3, u_3	merging
4	x_4, v_4, u_4	main
5	x_5, v_5, u_5	main

FIFO queue

Figure 6·2: The merging problem: CAVs randomly arrive at the origins of the main and merging roads. Collisons may occur at the Merging Point M. A coordinator is associated with M to maintain a queue and share information among CAVs as needed.

Let $S(t)$ be the set of FIFO-ordered indices of all CAVs located in the CZ at time t

along with the CAV (whose index is 0 as shown in Fig.6·2) that has just left the CZ. Let $N(t)$ be the cardinality of $S(t)$. Thus, if a CAV arrives at time t, it is assigned the index $N(t)$. All CAV indices in $S(t)$ decrease by one when a CAV passes over the MP and the vehicle whose index is -1 is dropped.

The vehicle dynamics for each CAV $i \in S(t)$ along the lane to which it belongs take the form

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ u_i(t) \end{bmatrix} \tag{6.1}$$

where $x_i(t)$ denotes the distance to the origin O (O') along the main (merging) lane if the vehicle i is located in the main (merging) lane, $v_i(t)$ denotes the velocity, and $u_i(t)$ denotes the control input (acceleration). We consider two objectives for each CAV subject to four constraints, as detailed next.

Objective 1 (Minimizing travel time): Let t_i^0 and t_i^m denote the time that CAV $i \in S(t)$ arrives at the origin O or O' and the MP M, respectively. We wish to minimize the travel time $t_i^m - t_i^0$ for CAV i.

Objective 2 (Minimizing energy consumption): We also wish to minimize energy consumption for each CAV $i \in S(t)$ expressed as

$$J_i(u_i(t)) = \int_{t_i^0}^{t_i^m} C(u_i(t))dt, \tag{6.2}$$

where $C(\cdot)$ is a strictly increasing function of its argument.

Constraint 1 (Rear end safety constraints): Let i_p denote the index of the CAV which physically immediately precedes i in the CZ (if one is present). We require that the distance $z_{i,i_p}(t) := x_{i_p}(t) - x_i(t)$ be constrained so that

$$z_{i,i_p}(t) \geq \varphi v_i(t) + \delta, \ \ \forall t \in [t_i^0, t_i^m], \tag{6.3}$$

where $v_i(t)$ is the speed of CAV $i \in S(t)$ and φ denotes the reaction time (as a rule, $\varphi = 1.8$s is used, e.g., (Vogel, 2003)). If we define z_{i,i_p} to be the distance from the center of CAV i to the center of CAV i_p, then δ is a constant determined by the length of these two CAVs (generally dependent on i and i_p but taken to be a constant for simplicity).

Constraint 2 (Safe merging (terminal constraint) between i and $i-1$): When $i-1 = i_p$, this constraint is redundant since (6.3) is enforced, but when $i-1 \neq i_p$ there should be enough safe space at the MP M for a merging CAV to cut in, i.e.,

$$z_{i,i-1}(t_i^m) \geq \varphi v_i(t_i^m) + \delta. \tag{6.4}$$

Constraint 3 (Vehicle limitations): There are constraints on the speed and acceleration for each $i \in S(t)$, i.e.,

$$v_{\min} \leq v_i(t) \leq v_{\max}, \forall t \in [t_i^0, t_i^m], \tag{6.5}$$

$$u_{i,\min} \leq u_i(t) \leq u_{i,\max}, \forall t \in [t_i^0, t_i^m], \tag{6.6}$$

where $v_{\max} > 0$ and $v_{\min} \geq 0$ denote the maximum and minimum speed allowed in the CZ, while $u_{i,\min} < 0$ and $u_{i,\max} > 0$ denote the minimum and maximum control input, respectively.

Constraint 4 (Terminal velocity): In a single CZ problem (Xiao et al., 2021b), the *optimal* terminal velocity is determined by solving the associated optimal control problem. In contrast, here we impose a constraint v^m determined by the upper (flow control) level:

$$v_i(t_i^m) = v^m \tag{6.7}$$

Problem 1: Our goal is to determine a control law (as well as optimal merging time t_i^m) to achieve objectives 1-2 subject to constraints 1-4 for each $i \in S(t)$ governed by the dynamics (6.1). The common way to minimize energy consumption is by minimizing the control input effort $u_i^2(t)$, noting that the OCBF method allows for more elaborate fuel consumption models, e.g., as in (Kamal et al., 2013a). By normalizing travel time and

$u_i^2(t)$, and using $\alpha \in [0,1)$, we construct a convex combination as follows:

$$J_i(u_i(t), t_i^m; v^m) = \int_{t_i^0}^{t_i^m} \left(\alpha + \frac{(1-\alpha)\frac{1}{2}u_i^2(t)}{\frac{1}{2}\max\{u_{i,\max}^2, u_{i,\min}^2\}} \right) dt. \tag{6.8}$$

If $\alpha = 1$, then we solve (6.8) as a minimum time problem. Otherwise, by defining $\beta := \frac{\alpha \max\{u_{i,\max}^2, u_{i,\min}^2\}}{2(1-\alpha)}$ and multiplying (6.8) by $\frac{\beta}{\alpha}$, we have:

$$(u_i(t), t_i^m; v^m) := \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} \frac{1}{2}u_i^2(t)dt, \tag{6.9}$$

where $\beta \geq 0$ is a weight factor that can be adjusted to penalize travel time relative to the energy cost, subject to (6.1), (6.3)-(6.7) and the terminal constraint $x_i(t_i^m) = L$, given $t_i^0, x_i(t_i^0), v_i(t_i^0)$.

6.1.2 Merging Control: Upper (Flow Control) Level

The function of the coordinator of CZ $j = 1, 2, \ldots$ now includes dynamically setting $v^m = v_j^m$ in the terminal velocity constraint (6.7). The traffic in CZ j is influenced by v_j^m in two ways: (i) CAV i's objective $J_i(u_i(t), t_i^m; v_j^m)$ becomes a function of v_j^m, and (ii) The initial velocity of CAVs in CZ j is exactly the terminal velocity $v_{j_u}^m$ in CZ j's upstream CZ, j_u. Thus, by setting such terminal velocity constraints, the coordinators can control the traffic flow in CZ j based on observations including, but not limited to, the number of CAVs in CZ j, N_j; the number of CAVs in CZ j's neighboring CZs , $\{N_{j_n} : j_n \in \mathcal{N}(j)\}$; the length of the road segment L; and the terminal velocity of the neighboring CZs $v_{j_n}^m$.

Problem 2 Our goal is to design a terminal velocity constraint controller for CZ j that minimizes the average of CAV objectives:

$$J_j(v_j^m(t)) = \frac{1}{N_j(t)} \sum_{i}^{N_j(t)} J_i^*(v_j^m(t)) \tag{6.10}$$

where $N_j(t)$ denotes the total number of CAVs in CZ j at time t and $J_i^*(v_j^m(t))$ is the optimal cost in (6.9). The problem is re-solved at every t when a CAV arrival/departure event occurs that causes a change in $N_j(t)$ and the set of feasible values of $v_j^m(t)$ depends on observations from CZ j and its neighboring CZs $j_n \in \mathcal{N}(j)$. This is a challenging problem to solve on line. In the sequel, we limit ourselves to a simpler feedback-based policy for $v_j^m(t)$, as described in Section 6.2.2.

6.2 Hierarchical Control Problem Solution

In this section, we provide solutions to **Problems 1-2** formulated above as shown in Fig. 6·3. **Problem 1** is solved adopting the OCBF approach (Xiao et al., 2021b). **Problem 2** is solved by using several event-driven methods including a simple fixed v_m and a feedback control method to set the terminal velocity dynamically.

6.2.1 Problem 1: Lower Level Merging Control

The merging control problem can be analytically solved, however, as pointed out in (Xiao et al., 2021b), it becomes computationally intensive when one of more constraints become active. To solve the merging problem in real time while still *guaranteeing* that all safety constraints are satisfied, we adopt the OCBF approach (Xiao et al., 2021b) through the following steps: (i) an optimal control solution for the *unconstrained* optimal control problem (6.9) is first obtained as a reference control, (ii) the resulting reference trajectory is optimally tracked subject to the control bounds (6.6) as well as a set of CBF constraints enforcing (6.3),(6.4). Using the forward invariance property of CBFs (Xiao et al., 2023), these constraints are guaranteed to be satisfied at all times if they are initially satisfied. The importance of CBFs is that they impose *linear* constraints on the control which, if satisfied, guarantee the satisfaction of the associated original constraints that involve the state and/or control. This whole process leads to a sequence of QPs solved over discrete time steps,

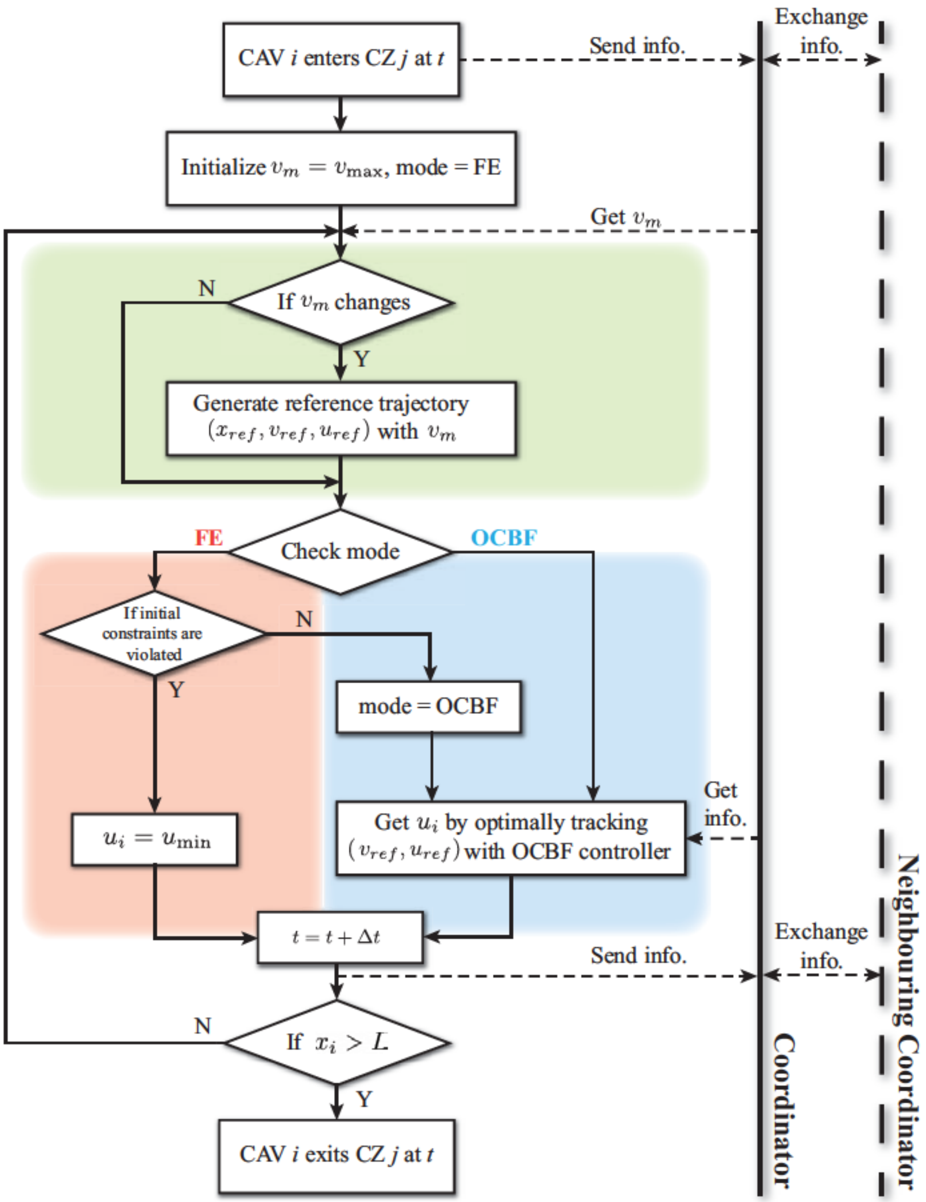

CAV i enters CZ j at t
Send info.
Exchange info.
Initialize $v_m = v_{\max}$, mode = FE
Get v_m
N
If v_m changes
Y
Generate reference trajectory $(x_{ref}, v_{ref}, u_{ref})$ with v_m
FE
Check mode
OCBF
If initial constraints are violated
N
Y
mode = OCBF
$u_i = u_{\min}$
Get info.
Get u_i by optimally tracking (v_{ref}, u_{ref}) with OCBF controller
$t = t + \Delta t$
Send info.
Exchange info.
N
If $x_i > L$
Y
CAV i exits CZ j at t
Coordinator
Neighbouring Coordinator

Figure 6·3: Hierarchical Control Framework over Modular Control Zones. The green, red and blue shaded areas represent the three main components: reference trajectory generation, feasibility enforcement mode, and optimal tracking controller with CBFs.

since the objective function is quadratic and the CBF constraints are linear in the control.

Unconstrained optimal control solution. The unconstrained solution refers to the solution of (6.9) with all safety constraints (6.3) and (6.4) inactive. However, the terminal velocity constraint (6.7) is retained as a boundary condition. Using standard Hamiltonian analysis (Bryson and Ho, 1969), the optimal control, velocity, and position trajectories of CAV *i* have the form:

$$u_i(t) = a_i t + b_i \tag{6.11}$$

$$v_i(t) = 1/2 \cdot a_i t^2 + b_i t + c_i \tag{6.12}$$

$$x_i(t) = 1/6 \cdot a_i t^3 + 1/2 \cdot b_i t^2 + c_i t + d_i \tag{6.13}$$

where the parameters a_i, b_i, c_i, d_i and t_i^m are obtained by solving the following five equations:

$$1/2 \cdot a_i (t_i^0)^2 + b_i t_i^0 + c_i = v_i^0 \tag{6.14}$$

$$1/2 \cdot a_i (t_i^m)^2 + b_i t_i^m + c_i = v^m \tag{6.15}$$

$$1/6 \cdot a_i (t_i^0)^3 + 1/2 \cdot b_i (t_i^0)^2 + c_i t_i^0 + d_i = 0 \tag{6.16}$$

$$1/6 \cdot a_i (t_i^m)^3 + 1/2 \cdot b_i (t_i^m)^2 + c_i t_i^m + d_i = L \tag{6.17}$$

$$\beta - 1/2 \cdot b_i + a_i c_i = 0 \tag{6.18}$$

where (6.15) corresponds to the terminal speed constraint (6.7).

Optimal reference tracking controller with CBFs: Once we obtain the unconstrained optimal control solution (6.11)-(6.13), we set a reference control $u_{ref}(t) = h(u_i^*(t), x_i^*(t), x_i(t))$ for some appropriately designed $h(u_i(\cdot)$ that provides feedback $x_i(t)$ from the actual observed CAV trajectory; in the simplest case, we just set $u_{ref}(t) = u_i^*(t)$, the solution of the unconstrained problem in (6.11). We then design a controller that minimizes $\int_{t_i^0}^{t_i^m} \frac{1}{2}(u_i(t) - u_{\text{ref}}(t))^2 dt$ subject to all constraints (6.3) through (6.6). This is accomplished by introducing CBFs for the safety constraints to take advantage of their simple form (linear in the control) while still guaranteeing the satisfaction of all constraints through the forward invariance property of CBFs. We review this approach next (see also (Xiao et al., 2021b)).

First, let $\boldsymbol{x}_i(t) \equiv (x_i(t), v_i(t))$. Based on the vehicle dynamics (6.1), define $f(\boldsymbol{x}_i(t)) = [v_i(t), 0]^T$ and $g(\boldsymbol{x}_i(t)) = [0,1]^T$. All state constraints, such as (6.3) and (6.4), can be expressed in the form $b_k(\boldsymbol{x}_i(t)) \geq 0, k \in \{1,...,B\}$ where B is the number of constraints. The CBF method maps each constraint $b_k(\boldsymbol{x}_i(t)) \geq 0$ to a new constraint which directly involves the control $u_i(t)$ and takes the (linear in the control) form

$$L_f b_k(\boldsymbol{x}_i(t)) + L_g b_k(\boldsymbol{x}_i(t)) u_i(t) + \gamma(b_k(\boldsymbol{x}_i(t))) \geq 0, \tag{6.19}$$

where L_f, L_g denote the Lie derivatives of $b_k(\boldsymbol{x}_i(t))$ along f and g respectively and $\gamma(\cdot)$ denotes some class-$\mathcal{K}$ function (Xiao et al., 2023), usually taken to be linear for simplicity. The forward invariance property of CBFs guarantees that a control input that satisfies (6.19) will also enforce $b_k(\boldsymbol{x}_i(t)) \geq 0$ at all times. In other words, the safety constraints (6.3), (6.4) are never violated.

To optimally track the reference speed trajectory, a Control Lyapunov Function (CLF) $V(\boldsymbol{x}_i(t))$ is used. Letting $V(\boldsymbol{x}_i(t)) = (v_i(t) - v_{\text{ref}}(t))^2$, the CLF constraint takes the form

$$L_f V(\boldsymbol{x}_i(t)) + L_g V(\boldsymbol{x}_i(t)) u_i(t) + \varepsilon V(x_i(t)) \leq e_i(t), \tag{6.20}$$

where $\varepsilon > 0$, and $e_i(t)$ is a relaxation variable which makes the constraint soft. Then, the OCBF controller optimally tracks the reference trajectory by solving the optimization problem:

$$\min_{u_i(t), e_i(t)} \int_{t_i^0}^{t_i^m} \left(\beta e_i^2(t) + \frac{1}{2}(u_i(t) - u_{\text{ref}(t)})^2 \right) \tag{6.21}$$

subject to the B CBF constraints (6.19), the CLF constraints (6.20) and the control bounds (6.6).

We can now solve problem (6.21) by discretizing $[t_i^0, t_i^m]$ into intervals of equal length Δt and solving (6.21) over each time interval $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t]$. The decision variables $u_i(t)$ and $e_i(t)$ are assumed to be constant over each such interval. They can be easily obtained by solving the following Quadratic Program (QP) problem, since all CBF constraints are linear in the decision variables $u_i(t)$ and $e_i(t)$ (fixed over each interval $[t_i^k, t_i^k + \Delta t]$):

$$\begin{aligned} \min_{u_i(t), e_i(t)} \ & \beta e_i(t)^2 + \frac{1}{2}(u_i(t) - u_{\text{ref}}(t))^2 \\ s.t. \ & (6.19), (6.20), (6.6), \quad t = t_i^0 + k\Delta t \end{aligned} \tag{6.22}$$

By repeating this process until CAV i exits the CZ, the solution to (6.21) is obtained as long as (6.22) is *feasible for each time interval* (this issue is further addressed below). Unlike the CBF constraints (6.19) that guarantee safety, the CLF constraints (6.20) provide convergence to the referenced velocity trajectory, designed so that $v_{\text{ref}}(t_i^m) = v^m$. If convergence to v^m is not fully attained by the time CAV i exits the CZ, this only causes some loss in performance but no violation to any safety constraint.

Control and speed reference trajectories. The reference trajectories $u_{\text{ref}}(t)$ and $v_{\text{ref}}(t)$ are designed in a position-feedback manner such that $v_{\text{ref}}(t)$ ends with v^m, i.e. when $x_i(t) = L$, $v_{\text{ref}}(t_i^m) = v^m$. We introduce a reference time t_{ref} by solving $x^*(t_{\text{ref}}) = x(t)$ for any t, where $x^*(\cdot)$ is the optimal unconstrained position of a given CAV in (6.13). For computational efficiency, we may use an approximate solution $t_{\text{ref}} = t - (x^*(t) - x(t))/v^*(t)$instead. Then, we choose the unconstrained optimal trajectory at t_{ref}: $v_{\text{ref}}(t) = v^*(t_{\text{ref}})$, $u_{\text{ref}}(t) = u^*(t_{\text{ref}})$.

Feasibility Enforcement Mode. When solving (6.9) for a single CZ, it is assumed that all safety constraints are initially satisfied. When OCBF is applied, the additional initial CBF constraints in (6.19) are assumed to be satisfied as well. However, when multiple CZs are considered, this assumption is no longer reasonable, as the initial state of a CAV entering a CZ depends directly on the control applied at its upstream CZ. Thus, a control mechanism is needed to enforce the initial feasibility which can no longer be assumed.

Inspired by the Feasibility Enforcement Zone (FEZ) concept (Zhang et al., 2017), where initial constraints are enforced in a road segment prior to entering the CZ, we introduce a *Feasibility Enforcement (FE) mode* as shown in Fig. 6·3. When CAV i enters a CZ at time t, it is set to the FE mode with a constant control input of maximum deceleration which enforces the satisfaction of initial constraints as fast as possible up to a maximum distance limit (normally $L/4$) within which feasibility must be attained. If none of the constraints is violated, the CAV's mode is switched to *OCBF mode* with the OCBF controller optimally tracking the reference trajectory, and never changed back. Of course, the FE mode mechanism does not guarantee the desired initial feasibility, especially when the road length L is short and the initial velocity v_i^0 is high. Thus, the coordinator is required to constrain the terminal velocity of CAVs appropriately so as to further enforce feasibility and smooth the traffic flow. In addition, the average time of a CAV in FE mode reflects the robustness of the system: the shorter the average FE mode is in a CZ, the safer and more robust this CZ is.

OCBF Control Feasibility. As already pointed out, the feasibility of the OCBF controller after each time step in (6.22) is not guaranteed. This is because the time discretization keeps the control constant over a time step and may cause a conflict with the control limits which cannot be predicted by the myopic nature of each QP in the sequence of QPs that are solved. However, in merging control, we can *enforce this feasibility* by adding a single feasibility guarantee constraint corresponding to each CBF constraint using the method in (Xu et al., 2022c). Specifically, the rear-end CBF constraint (6.19) applied to (6.3) is $v_{i_p} - v_i - \varphi u_i + k_1 b_1(\boldsymbol{x}_i) \geq 0$ and it was shown in (Xu et al., 2022c) that it can be enforced by any u_i and $k_1 > 0$ that satisfy

$$u_{i_p} - u_i + k_1(v_{i_p} - v_i - \varphi u_{i,\min}) \geq 0 \tag{6.23}$$

Similarly, the safe-merging CBF constraint (6.19) applied to (6.4) is $v_{i-1} - v_i - \varphi_2 v_i^2 -$

$\varphi_2 x_i u_i + k_2 b_2(\boldsymbol{x}_i) \geq 0$ and it can be shown to be enforced by any u_i and $k_2 > 0$ that satisfy

$$u_{i-1} - u_i - 2\varphi_2 v_i u_i - \varphi_2 v_i u_{\min} + k_2(v_{i-1} - v_i - \varphi_2 v_i^2 - \varphi_2 x_i u_{\min}) \geq 0$$

6.2.2 Problem 2: Upper Level Flow Control

Flow control is performed by dynamically setting the terminal velocity v_j^m for each CZ j in an event-driven way: when a CAV enters or exits CZ j or its neighbors, coordinator j communicates with all its neighboring coordinators $j_n \in \mathcal{N}(j)$ and adjusts v_j^m according to accumulated statistical information at the CZ and its neighbors. Coordinator j then broadcasts v_j^m to all CAVs within CZ j. Once a new v_j^m is received, CAV i generates a new reference trajectory through (6.11), (6.12), (6.13) and tracks it with the OCBF controller. This event-driven method avoids frequent rescheduling of CAVs through the lower level control.

The performance of the system is affected by v_j^m in two ways. First, v_j^m directly influences the CAV objectives at the lower level. Specifically, when the weight α on travel times is relatively large, a large v_j^m often results in better performance. On the other hand, an aggressive exit velocity v_j^m may result in a higher probability of violating the safety constraints upon entering the downstream CZs, which increases the average time spent in FE mode and consumes more energy as a consequence.

In this chapter, rather than explicitly trying to solve **Problem 2** through (6.10), we design a feedback controller to regulate the terminal velocity v_j^m according to the number of CAVs in CZ j and its neighbors which is monitored by each coordinator. The feedback controller has the form:

$$v_j^m(t) = v_b - kN_{j_n}(t) \tag{6.24}$$

where v_b is a baseline velocity, $k \geq 0$ is a feedback gain parameter, and $N_{j_n}(t)$ is the number of CAVs in the downstream neighboring CZ j_n. When $k = 0$, this becomes a fixed speed controller. A simple, yet effective, selection for v_b is the average optimal terminal velocity

$v_b = \frac{1}{N_S}\sum_{i\in S}\arg\min_{v^m} J_i^*(v^m)$, where where $J_i^*(v^m)$ is the optimal cost in (6.9), S is a set of CAVs sampled with different entry velocity, and N_S is the cardinality of S.

The controller can be modified by replacing $N_{j_n}(t)$ by $N_{j_n}(l,t)$, defined as the number of CAVs in $[0,l]$, the first segment of the CZ with length l, since the performance of the downstream CZ is critically affected by the average time in FE mode, therefore, the states of CAVs in this segment is more important than the rest. In this case, l is a parameter to be selected and an alternative selection for v_b is the average velocity of CAVs in the critical zone $[0,l]$, denoted by $\bar{v}(l)$.

Efficient trajectory generation when v^m changes. Clearly, every time v^m changes in (6.15), the reference trajectories of all CAVs in the CZ affected by this change need to be re-generated. The event-driven nature of the flow controller above limit the frequency of such changes. Nonetheless, under heavy traffic, this frequency may still be high and motivates an efficient trajectory regenerating method that address this problem.

We first observe that the unconstrained optimal solution to **Problem 1** (6.11)-(6.13) is invariant to a time shift, i.e. $u_i(t) = u_i'(t-t')$ where the parameters of $u_i'(t)$ are obtained from (6.14)-(6.18) with $t_i^0 - t'$ replacing t_i^0 and $t_i^m - t'$ replacing t_i^m. Thus, by shifting time $t' = t_i^0$ forward and replacing t_m by $t_i^m - t_i^0$, (6.14)-(6.18) are equivalent to:

$$c_i = v_i^0 \tag{6.25}$$

$$1/2 \cdot a_i t_m^2 + b_i t_m + c_i = v^m \tag{6.26}$$

$$d_i = 0 \tag{6.27}$$

$$1/6 \cdot a_i t_m^3 + 1/2 \cdot b_i t_m^2 + c_i t_m + d_i = L \tag{6.28}$$

$$\beta - 1/2 \cdot b_i + a_i c_i = 0 \tag{6.29}$$

These can be combined into a single cubic equation in t_m (for notational simplicity, v_i^0 and

v^m are replaced with v_0 and v_m respectively in the sequel):

$$\beta t_m^3 + (2v_0 + v_m)t_m^2 + (-3L + 6v_0 v_m + 6v_0^2)t_m - 12Lv_0 = 0 \tag{6.30}$$

Denoting the left side of (6.30) by $f(t_m, v_m)$, we get

$$\begin{aligned}
\frac{\partial f(t_m, v_m)}{\partial v_m} &= t_m^2 + 6v_0 t_m \\
\frac{\partial f(t_m, v_m)}{\partial t_m} &= 3\beta t_m^2 + (4v_0 + 2v_m)t_m - 3L + 6v_0 v_m + 6v_m^2
\end{aligned}$$

Considering a small perturbation in v_m, we have $f(t_m, v_m + \Delta v) = f(t_m, v_m) + \frac{\partial f(t_m, v_m)}{\partial v_m}\Delta v$, which results in a perturbation in t_m such that $f(t_m + \Delta t, v_m + \Delta v) = f(t_m, v_m) = 0$. Then, using a first-order Taylor expansion,

$$f(t_m + \Delta t, v_m + \Delta v) = f(t_m, v_m + \Delta v) + \frac{\partial f(t_m, v_m)}{\partial t_m}\Delta t \tag{6.31}$$

which yields

$$\begin{aligned}
\Delta t &= -\left(\frac{\partial f(t_m, v_m)}{\partial v_m}\right) \Big/ \left(\frac{\partial f(t_m, v_m + \Delta v)}{\partial t_m}\right)\Delta v \\
&= -\frac{t_m^2 + 6v_0 t_m}{3\beta t_m^2 + (4v_0 + 2v'_m)t_m - 3L + 6v_0 v'_m + 6v_0^2}\Delta v
\end{aligned} \tag{6.32}$$

where $v'_m = v_m + \Delta v$ is the perturbed v_m. This provides an analytical expression of the perturbed terminal time $t'_m = t_m + \Delta t$ when v_m is perturbed by Δv.

Now the problem turns into finding the unconstrained optimal solution to **Problem 1** with fixed terminal time t'_m. The solution shares the form (6.11)-(6.13) with parameters obtained by solving (6.14)-(6.17). By similarly shifting time $t' = t_i^0$ forward and replacing t_m by $t_i^m - t_i^0$, (6.14)-(6.17) are equivalent to (6.25)-(6.28) which has a unique analytical solution if $t_m \neq 0$ (which obviously holds):

$$\begin{aligned} a_i &= 6/t_m^2(v^m - v_i^0) - 12/t_m^3(L - v_i^0 t_m) \\ b_i &= -2/t_m(v^m - v_i^0) + 6/t_m^2(L - v_i^0 t_m) \\ c_i &= v_i^0, \\ d_i &= 0 \end{aligned} \tag{6.33}$$

Thus, every time v^m is changed by Δv, we can efficiently generate the unconstrained optimal trajectories by setting $t'_m = t_m + \Delta t$ using (6.32) and then using t'_m into (6.33) to update all the parameters without re-solving (6.14)-(6.18).

6.3 Simulation Results

In this section, we use the multi-model traffic flow simulation platform Vissim to construct the two-CZ configuration in Fig. 6·1. We replicate this model in a Python-based simulation with the same traffic input and apply the lower level controller only (without flow control using terminal velocity constraints) as a *baseline* to evaluate the traffic performance in two consecutive merging roadways and compare it to the performance obtained using our proposed modular control zone framework. In the simulations, only the terminal velocity v_1^m of CZ1 is set for flow control, while v_2^m is set to be the average exiting velocity of CAVs passing CZ2 with the lower level controller only for fair comparison.

Flow control with fixed v^m vs. No flow control. We start with the flow controller (6.24) with fixed v^m. The parameter settings are as follows: $L_1 = L_2 = 200m, \delta = 0m, \varphi = 1.8s$, $v_{\max} = 30m/s$, $v_{\min} = 0$, $u_{\max} = -u_{\min} = 4m/s^2$, $\alpha = 0.0625$, $v_2^m = 18.5m/s$. We use a fixed traffic rate 400 CAVs/h in each road segment and simulate around 120 CAVs. Simulations are performed using a fixed v_1^m value, which varies over $12 - 20m/s$ with step size $0.5m/s$. The objective of each CZ as well as the total objective corresponding to different v_1^m are shown in Fig. 6·4 where dashed lines show the baseline. The figure shows that a higher

v_1^m results in a better performance in CZ1, but worse in CZ2, which well aligns with our expectation as explained in Sec. 6.2.2. The speed v_1^m constrains the aggressive exit velocity which has a high probability of violating the safety constraints when entering CZ2; thus, it improves the performance in CZ2 when v_1^m is low. The total objective can be approximated by a quadratic function with the minimum value occurring when v_1^m is around $15m/s$.

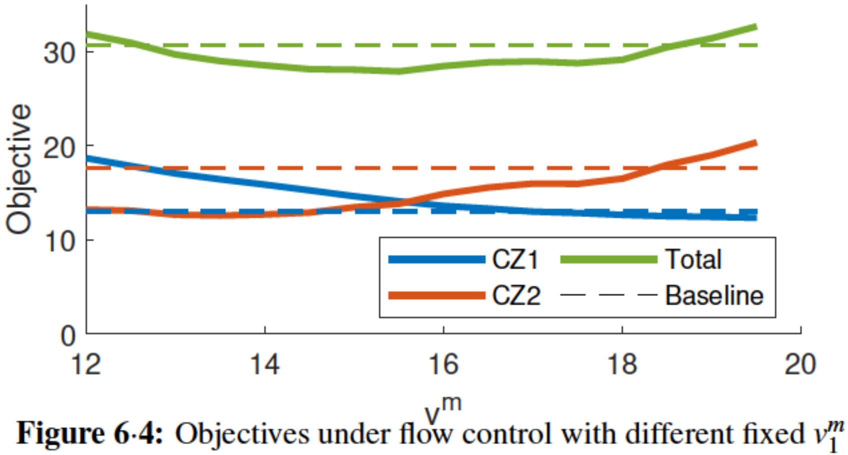


Figure 6·4: Objectives under flow control with different fixed v_1^m

The effectiveness of the flow controller with fixed v_1^m can also be illustrated in terms of the average time CAVs spend in FE mode (FEm) entering CZ2. We show the key feasibility metrics in Table 6.1 including the total number of CAVs with FE mode active, the average active FE mode time, and the total number of CAVs with infeasible QPs. Both FEm count and average FEm time drop when flow control is exerted with fixed v_1^m by 25.8% and 60.5% respectively. Since we have already included the feasibility guarantee constraints (6.23) and (6.24) in the OCBF controller, the only reason for the existence of infeasible QPs using no flow control is that the *initial* feasibility fails to be enforced within the FE mode distance limitation. Both the initial feasibility and the QP feasibility of CAV are more easily enforced with fixed v_1^m flow control, thus making the system safer and more robust.

Feedback control on v^m. We apply a simple feedback controller as in (6.24) with the same parameter settings as before under a fixed traffic rate of 400 CAVs/h and parameters $v_b = 18m/s$, $k = 1/2$ and $l = L/4$. We compare the performance of the feedback controller

Table 6.1: Feasibility Comparison of Flow Control vs. no flow control

	FEm count	FEm Time (s)	Infeasible Count
No flow control	62	0.86	1
$v_1^m = 15m/s$	46	0.34	0
$v_1^m = 12m/s$	26	0.15	0

with the fixed value flow controller ($v_1^m = 15m/s$) and the baseline using only the lower level controller as shown in Table. 6.2.

Table 6.2: Performance Comparison

	Metric	Feedback Control	Fixed v^m	No flow control
CZ1	t	12.45	13.44	10.65
	e	4.51	5.66	5.81
	obj.	12.81	14.62	12.92
CZ2	t	10.31	12.11	10.23
	e	8.32	5.38	13.31
	obj.	15.19	13.46	20.13
Total	obj.	28.00	28.07	33.03

Table 6.2 shows that the flow controller with fixed v^m and feedback control on v^m both outperform the baseline in the sense of total objective, with approximately 15% improvement. CAVs using only the lower level controller always have a shorter travel time than the modular CZ method with flow control at the expense of an energy consumption rise in CZ2 causing a total performance loss. There is no significant performance increase using the feedback controller due to the constant traffic rate used in the simulation. As can be seen in Fig. 6·4, $v_1^m = 15m/s$ is already close to optimal, leaving little room for extra improvement. However, a feedback controller pays off when the traffic rate increases over time.

Varying traffic input. We construct a varying traffic input with rates

$$\lambda(t) = \begin{cases} 300, & t \in [\ \ 0,\ 400] \\ 200, & t \in [\,400,\ 800] \\ 400, & t \in [\,800, 1200] \\ 100, & t \in [1200, 1600] \\ 400, & t \in [1600, 2000] \end{cases} \quad \text{CAVs/h}$$

The same parameter settings are used except $\alpha = 0.25$ to put more emphasis on travel time. The flow controller with fixed value $v_1^m = 15m/s$ and the feedback controller $v_1^m = \bar{v} + 2 - 1/2n(L/4)$ are both applied and compared with the baseline of no flow control. The total objective of each CAV is shown in Fig. 6·5 with the exponential smoothing technique used for visualization. CAVs using flow control with fixed v^m (solid blue line) outperform those using only lower-level control (dashed orange line) when the traffic rate is high. However, the fixed v^m constraint becomes too conservative under low traffic, thus restricting CAV performance. However, by applying feedback flow control (solid green line), CAVs can achieve a similar near-optimal performance as using only the lower level control under low traffic rate and outperform it when the system becomes congested.

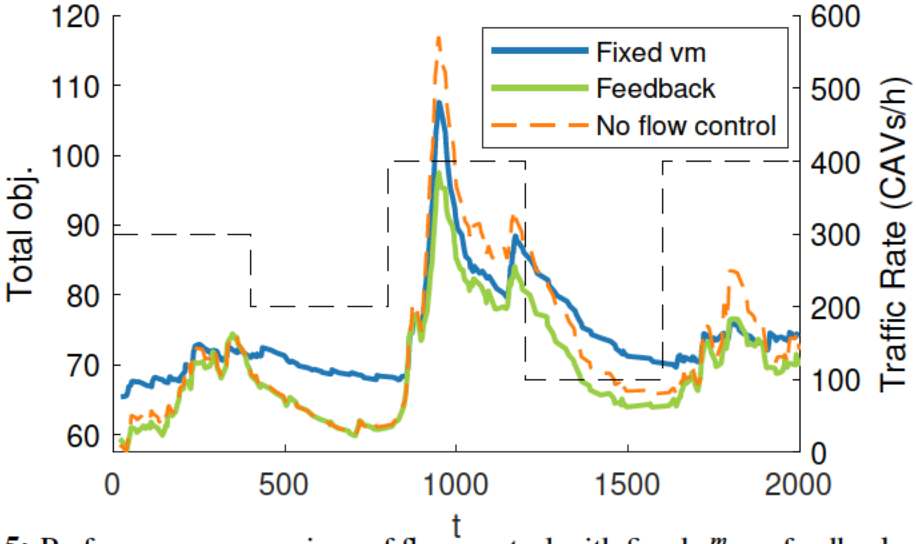


Figure 6·5: Performance comparison of flow control with fixed v^m vs. feedback control on v^m vs. no flow control under changing traffic rate. The changing traffic rate is shown in the dashed black line.

The numerical results in Table 6.3 also illustrate the benefit of feedback control on v^m: it constrains the aggressive exit velocity of CAVs in heavy traffic and adjusts v^m according to traffic conditions to avoid being too conservative, thus outperforming both no flow control and fixed v^m. The performance increase is not as significant as in Table 6.2 mainly due to $\alpha = 0.25$ giving more emphasis on travel time, thus favoring the aggressive no flow control

Table 6.3: Performance Comparison

	Metric	Feedback Control	Fixed v^m	No flow control
CZ1	t	10.69	12.48	10.66
	e	8.45	5.58	5.81
	obj.	36.96	38.85	34.23
CZ2	t	9.72	10.69	10.23
	e	9.88	9.41	13.31
	obj.	35.82	37.92	40.59
Total	obj.	72.78	76.77	74.72

method.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

In this dissertation, we develop and tackled four problems closely related to safety-critical optimal control in autonomous traffic systems:

(1) Optimal control of CAVs in a roundabout (Chapter 3),

(2) Feasibility guaranteed OCBF (Chapter 4),

(3) Learning the OCP solution (Chapter 5),

(4) A hierarchical framework of modular control zones (Chapter 6).

These works are related in the following way. Following our proposed framework in (1), the CAVs in a roundabout can be controlled in a decentralized way with significant improvements in the performance shown in the simulation. However, when the traffic conditions become more complex and the vehicle limitations become tighter, the feasibility problem arises and is often accompanied with the problem of performance loss. The feasibility guaranteed OCBF is derived to resolve the feasibility problem, where we adopt the OCBF approach and guarantee the feasibility of each QP problem by adding a single feasibility constraint to it. To improve the performance, a neural network is designed and trained to learn a referenced trajectory from the OCP solution. As the roundabout problem consists of several coupled merging control problems and the latter is easier to analyze, work (2) and (3) are both conducted in the merging control scenario. Work (1)-(3) successfully tackle the decentralized optimal control problem of controlling CAVs travelling through a roundabout scenario and resolve the potential feasibility and performance problem but are restricted to a single control zone. For the completeness and generality of the dissertation, a hierarchical framework is proposed and validated in (4) to scale up the optimal safe control of CAVs to a traffic network.

7.2 Future Directions

7.2.1 Application in a Generalized Complex Traffic Network

The decentralized control framework for CAVs proposed in the dissertation are built and tested in simple scenarios comparing to a complex traffic network. The effectiveness of the framework remains to be validated and adjustments may be needed when applying the framework to a more complicated scenario. Therefore, the application of the safety-critical optimal control methods to CAVs in simulators with a built-in town map (like CARLA) will be a future direction. A benchmark can also be a potential output.

7.2.2 Adaptive Control Barrier Function with Feasibility Guarantee

The feasibility of the OCBF controller is guaranteed by adding a feasibility constraint at the price of conservation. As pointed out in Chapter 4, a parameter α can make a trade-off between feasibility and conservation. However, in the dissertation, α is a fixed value predefined according to a user's preference instead of a decision variable. The idea of adaptive CBF can be borrowed to bring the parameter α into the optimization process to provide a less conservative feasibility guarantee.

7.2.3 Control Barrier Functions with Reinforcement Learning and Online Learning

In the dissertation, the OCBF controller tracks a reference trajectory that is fixed either by analytical solution or by pre-trained neural network. Therefore the controller lacks the capability to learn from the environment and adjust according to noises and environment changes. A combination of CBFs with reinforcement learning and online learning has the potential to resolve this problem and thus improve the performance.

7.2.4 Mixed Traffic Scenario

The dissertation limits itself to the scenario of all CAVs with no human interference. The problem becomes interesting and challenging when human driven vehicles (HDVs) are included. The decentralized OCBF framework still has the potential to solve the problem under mixed traffic scenario, however, the interaction between HDVs and CAVs remains to be studied.

Appendix A

Proofs

Proof of Theorem 4.2

Theorem 4.2 *Under Assumptions 4.1, 4.2 and 4.3, the QP* (4.4) *subject to* (4.2)*,* (4.6) *and* (4.14) *corresponding to any time interval* $[t^0+k\Delta t, t^0+(k+1)\Delta t] \subset [t^0, t^f]$ *is feasible.*

Proof. Since $b_F(\boldsymbol{x}(t^0)) \geq 0$, the CBF constraint (4.2) does not conflict with the control bounds (4.6). Using (4.12) under $b_\eta(\boldsymbol{x}(t^0)) \geq 0$ and *Assumption 4.1*, there always exists a control $u(t^0) \in [u_{\min}, u_{\max}]$ such that (4.14) is satisfied, which indicates that (4.14) and (4.6) are conflict-free. As (4.2) and (4.14) have the same inequality direction for $u(t^0)$, they are conflict-free as well. Thus, the QP (4.4) subject to (4.2), (4.6) and (4.14) is feasible at time t^0.

According to *Theorem* 4.1, the QP corresponding to time $t^0+\Delta t$ is feasible. *Assumption 4.2* preserves the forward invariance property of CBFs which guarantees that $b(\boldsymbol{x}(t^0+\Delta t)) \geq 0$ and $b_\eta(\boldsymbol{x}(t^0+\Delta t)) \geq 0$ under *Assumption 4.3*.

Following the same procedure, it is easy to prove that if the QP is feasible at $t^0+k\Delta t$, $b(\boldsymbol{x}(t^0+k\Delta t)) \geq 0$ and $b_\eta(\boldsymbol{x}(t^0+k\Delta t)) \geq 0$, then the QP corresponding to $t^0+(k+1)\Delta t$ is feasible, $b(\boldsymbol{x}(t^0+(k+1)\Delta t)) \geq 0$ and $b_\eta(\boldsymbol{x}(t^0+(k+1)\Delta t)) \geq 0$ and the proof of the theorem is completed by induction. □

Proof of Theorem 4.3

Theorem 4.3 *If* $b_{\eta_1}(\boldsymbol{x}(t)) \geq 0$ *and the QP* (4.4) *subject to* (4.34)*,* (4.30) *and* (4.41) *is feasible at time t, then the QP corresponding to time* $t+\Delta t$ *is also feasible.*

Proof. By *Assumption 4.4*, $\min_i\{u_i\} = u_{\min} \leq u_{i_p}(t)$, thus there always exists a control input $u_i(t) \in [u_{\min}, u_{i,\max}]$ such that $u_{i_p}(t) - u_i(t) \geq 0$. As $b_{\eta_1}(\boldsymbol{x}(t)) \geq 0$, applying (4.39), we can always find a feasible control $u_i(t)$ such that $\eta_1(\boldsymbol{x}_i(t), u_i(t)) \geq 0$. As $\eta_1(\boldsymbol{x}_i(t), u_i(t)) \geq 0$

is the CBF constraint corresponding to $b_{\eta_1}(\boldsymbol{x}_i(t)) \geq 0$, using the forward invariance property of CBFs under *Assumption 4.2*, we have $b_{\eta_1}(\boldsymbol{x}(t+\Delta t)) \geq 0$. Thus, there always exists a control $u_i(t) \in [u_{\min}, u_{i,\max}]$ such that $\eta_1(\boldsymbol{x}_i(t+\Delta t), u_i(t+\Delta t)) \geq 0$. Hence, (4.41) and (4.30) are conflict-free at $t+\Delta t$.

Since $\varphi \geq 0$, (4.34) constrains the control $u_i(t+\Delta t)$ with an upper bound. Similarly, $u_i(t+\Delta t)$ is also constrained by an upper bound through (4.41). Thus, (4.34) and (4.41) are conflict-free at $t+\Delta t$.

Since the QP (4.4) subject to (4.34), (4.30) and (4.41) is feasible at time t, it follows that $b_{\text{cbf}_1}(\boldsymbol{x}_i(t), u_i(t)) \geq 0$, $b_F(\boldsymbol{x}(t)) \geq 0$. As $\eta_1(\boldsymbol{x}_i(t), u_i(t)) \geq 0$ always has a solution $u_i(t) \in [u_{\min}, u_{i,\max}]$, there exists a control under which (4.40) is satisfied. Since (4.40) is the CBF constraint of (4.37), using the forward invariance of CBFs under *Assumption 4.2*, we have $b_F(\boldsymbol{x}(t+\Delta t)) \geq 0$, which implies that (4.34) and (4.30) are conflict-free at time $t+\Delta t$.

Thus, all constraints of the QP (4.4) are conflict-free at $t+\Delta t$ and the QP corresponding to time $t+\Delta$ is feasible. □

Proof of Theorem 4.4

Theorem 4.4 *Under Assumptions 4.2, 4.4 and 4.5, the QP* (4.4) *subject to* (4.34)*,* (4.30) *and* (4.41) *corresponding to any time interval* $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t] \subset [t_i^0, t_i^m]$ *is feasible.*

Proof. Since $b_F(\boldsymbol{x}_i(t_i^0)) \geq 0$, the CBF constraint (4.34) does not conflict with the control bounds (4.30). Using (4.39) under $b_{\eta_1}(\boldsymbol{x}_i(t_i^0)) \geq 0$ and *Assumption 4.4*, there always exists a control $u_i(t_i^0) \in [u_{\min}, u_{i,\max}]$ such that (4.41) is satisfied, which indicates that (4.41) and (4.30) are conflict-free. As (4.34) and (4.41) have the same inequality direction for $u_i(t_i^0)$, they are conflict-free as well. Thus, the QP (4.4) subject to (4.34), (4.30) and (4.41) is feasible at time t_i^0.

According to *Theorem* 4.3, the QP corresponding to time $t_i^0 + \Delta t$ is feasible. *Assumption 4.2* preserves the forward invariance property of CBFs which guarantees that $b_1(\boldsymbol{x}_i(t_i^0 + \Delta t)) \geq 0$ and $b_{\eta_1}(\boldsymbol{x}(t_i^0 + \Delta t)) \geq 0$ under *Assumption 4.5*.

Following the same procedure, it is easy to prove that if the QP is feasible at $t_i^0 + k\Delta t$, $b_1(\boldsymbol{x}_i(t_i^0 + k\Delta t)) \geq 0$ and $b_{\eta_1}(\boldsymbol{x}_i(t_i^0 + k\Delta t)) \geq 0$, then the QP corresponding to $t_i^0 + (k+1)\Delta t$ is feasible, $b_1(\boldsymbol{x}_i(t_i^0 + (k+1)\Delta t)) \geq 0$ and $b_{\eta_1}(\boldsymbol{x}_i(t_i^0 + (k+1)\Delta t)) \geq 0$ and the proof of the theorem is completed by induction. □

Proof of Lemma 4.2

Theorem 4.2 *Under Assumption 4.4, Assumption 4.1 is satisfied, i.e.*

$$\forall \boldsymbol{x} \in \mathcal{C}, \quad \sup_{u \in U} h_2(\boldsymbol{x}, u) \geq 0,$$

where $\mathcal{C} = \{\boldsymbol{x} : b_2(\boldsymbol{x}) \geq 0\}$ *and* $h_2(\boldsymbol{x}, u)$ *is defined in* (4.47).

Proof. Lemma 4.2 is equivalent to: if $v_i \geq 0, u_{\min} \leq 0$, there exists a control input $u_i(t) \in [u_{\min}, u_{i,\max}]$, such that

$$u_{i-1} - u_i - 2\varphi_2 v_i u_i - \varphi_2 v_i u_{\min} \geq 0 \tag{A.1}$$

Note that (A.1) can be rewritten as

$$u_i + \frac{1}{2}u_{\min} \leq \frac{1}{1+2\varphi_2 v_i}(u_{i-1} + \frac{1}{2}u_{\min}) \tag{A.2}$$

Under *Assumption 4.4*,

$$\frac{1}{1+2\varphi_2 v_i}(u_{i-1} + \frac{1}{2}u_{\min}) \geq \frac{1}{1+2\varphi_2 v_i}\frac{3}{2}u_{\min} \tag{A.3}$$

As $v_i \geq 0$, we have $1 + 2\varphi_2 v_i \geq 1$. Additionally, $u_{\min} \leq 0$, thus

$$u_{\min} \leq \frac{1}{1+2\varphi_2 v_i}u_{\min} \tag{A.4}$$

As $\min_{u_i}(u_i + \frac{1}{2}u_{\min}) = \frac{3}{2}u_{\min}$, we have

$$\min_{u_i}(u_i + \frac{1}{2}u_{\min}) \leq \frac{1}{1+2\varphi_2 v_i}\frac{3}{2}u_{\min} \tag{A.5}$$

Combining (A.5) with (A.3), there always exists a control input $u_i(t) \in [u_{\min}, u_{i,\max}]$ such that (A.2) is satisfied. Hence, *Lemma 4.2* is proved. □

Proof of Theorem 4.5

Theorem 4.5 *If* $b_{\eta_2}(\boldsymbol{x}(t)) \geq 0$, $v_i \geq 0, u_{\min} \leq 0$ *and the QP* (4.4) *subject to* (4.46), (4.30) *and* (4.50) *is feasible at time* t, *then the QP corresponding to time* $t + \Delta t$ *is also feasible.*

Proof. According to *Lemma* 4.2, there exists a control input $u_i(t) \in [u_{\min}, u_{i,\max}]$ such that (A.1) is satisfied, thus *Assumption* 4.1 is satisfied. According to *Theorem* 4.1, the QP (4.4) subject to (4.46), (4.30) and (4.50) is feasible at time t, then the QP corresponding to time $t+\Delta t$ is also feasible. □

Proof of Theorem 4.6

Theorem 4.6 *Under Assumptions 4.2, 4.4, 4.6, if* $v_i \geq 0, u_{\min} \leq 0$*, the QP* (4.4) *subject to* (4.46)*,* (4.30) *and* (4.50) *corresponding to any time interval* $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t] \subset [t_i^0, t_i^m]$ *is feasible.*

Proof. The proof is by induction following the same steps as in the proof of *Theorem* 4.4 and invoking *Theorem 4.5*. □

Proof of Theorem 4.7

Theorem 4.7 *If* $b_{\eta_1}(\boldsymbol{x}(t)) \geq 0$*,* $b_{\eta_2}(\boldsymbol{x}(t)) \geq 0$*,* $v_i \geq 0, u_{\min} \leq 0$*, the QP* (4.4) *subject to* (4.34)*,* (4.46)*,* (4.30)*,* (4.41) *and* (4.50) *is feasible at time t, then the QP corresponding to time* $t + \Delta t$ *is also feasible.*

Proof. Theorem 4.3 shows that (4.34), (4.30) and (4.41) are conflict-free at $t+\Delta t$. *Theorem 4.5* shows that (4.46), (4.30) and (4.50) are conflict-free at $t+\Delta t$. In order to guarantee the feasibility of the QP corresponding to time $t+\Delta t$, we need to further prove (4.34), (4.46), (4.41) and (4.50) are conflict-free. By analyzing the coefficient of $u_i(t)$, we find $\varphi \geq 0$ in (4.34), $1 \geq 0$ in (4.41), $\varphi_2 x_i \geq 0$ in (4.46) and $1 + 2\varphi_2 v_i \geq 0$ in (4.50). This indicates that all these constraints have the same inequality direction, thus (4.34), (4.46), (4.41) and (4.50) are conflict-free. Hence, the QP corresponding to time $t+\Delta t$ is feasible. □

Proof of Theorem 4.8

Theorem 4.8 *Under* Assumption 4.5 *and* Assumption 4.6*, if* $v_i \geq 0, u_{\min} \leq 0$*, the QP* (4.4) *subject to* (4.34)*,* (4.46)*,* (4.30)*,* (4.41) *and* (4.50) *corresponding to any time interval* $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t] \subset [t_i^0, t_i^m]$ *is feasible.*

Proof. *Theorem* 4.8 is proved by induction following the same steps of the proof of *Theorem* 4.4 using *Theorem 4.7*. □

Appendix B

Tracking the reference trajectory with MPC

In this appendix, an auxiliary method is introduced to deal with the lateral offset while tracking the trajectory given by the OCBF controller. The OCBF controller focuses on generating a reference trajectory with longitudinal safety guarantees in real time. The lateral error can be dealt using the Model Predictive Control (MPC) approach.

B.1 Vehicle Dynamics

The same vehicle dynamics is adopted as in (Xiao et al., 2021c). Ego dynamics are defined in a 2-dimension coordinate system with respect to a reference trajectory, where the x coordinate lies along the referenced trajectory and the y coordinate refers to the lateral offset.

$$\underbrace{\begin{bmatrix} \dot{s} \\ \dot{d} \\ \dot{\mu} \\ \dot{v} \\ \dot{a} \\ \dot{\delta} \\ \dot{\omega} \end{bmatrix}}_{\dot{\boldsymbol{x}}} = \underbrace{\begin{bmatrix} \frac{v\cos(\mu+\beta)}{1-d\kappa} \\ v\sin(\mu+\beta) \\ \frac{v}{l_r}\sin\beta - \kappa\frac{v\cos(\mu+\beta)}{1-d\kappa} \\ a \\ 0 \\ \omega \\ 0 \end{bmatrix}}_{f(\boldsymbol{x})} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{g(\boldsymbol{x})} \underbrace{\begin{bmatrix} u_{jerk} \\ u_{steer} \end{bmatrix}}_{\boldsymbol{u}} \tag{B.1}$$

where $s \in \mathbb{R}$ is the along-trajectory distance and $d \in \mathbb{R}$ is the lateral offset; μ is the vehicle local heading error; v, a denote the vehicle linear speed and acceleration; δ, ω denote the steering angle and steering rate; u_{jerk} and u_{steer} denote the two control inputs for jerk and steering acceleration; κ is the curvature of the reference trajectory at the projection point

$(s,0)$; $\beta = \arctan\left(\frac{l_r}{l_r+l_f}\tan\delta\right)$ where l_f (l_r)+ is the length of the vehicle from head (tail) to the CoG.

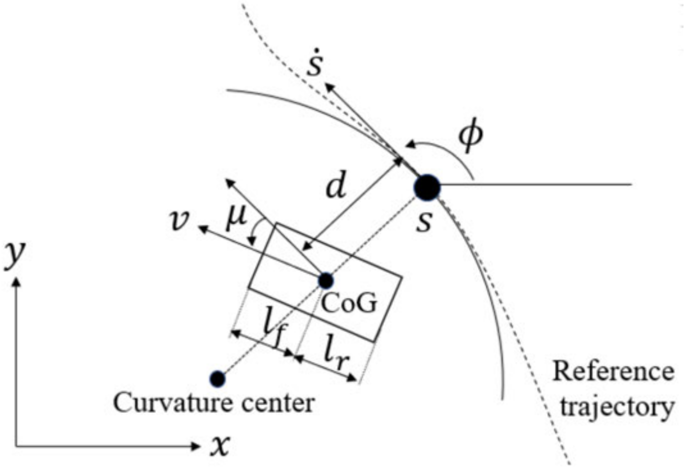


Figure B·1: Coordinates of ego w.r.t a reference trajectory

B.2 MPC controller design

To track the referenced trajectory under the complex vehicle dynamics as shown in Sec. B.1, we propose a receding horizon (MPC) approach that solves the following optimization problem with receding horizon H at each time $t \geq 0$:

$$(\boldsymbol{u}_{mpc}, \boldsymbol{x}_{mpc}) = \arg \min_{\boldsymbol{u}(0:H-1), \boldsymbol{x}(1:H)} \sum_{i=1}^{H} C(\boldsymbol{x}(i)) + \sum_{i=0}^{H-1} J(\boldsymbol{u}(i)) \tag{B.2}$$

where $C(\boldsymbol{x}(i))$ describes the tracking error with respect to the reference trajectory, $J(\boldsymbol{u}(i))$ is the penalty term for jerk and steering acceleration. For this specific problem, the two functions are defined as follows:

$$C(\boldsymbol{x}) = w_1(s - s_{ref})^2 + w_2(d - d_{ref}^2) + w_3(v - v_{ref})^2 \tag{B.3}$$

$$J(\boldsymbol{u}) = u_{jerk}^2 + u_{steering}^2 \tag{B.4}$$

For the prediction part, we are using the following predictive model derived from (B.1) using the Adomian Decomposition Method (ADM):

$$\begin{bmatrix} s(k+1) \\ d(k+1) \\ \mu(k+1) \\ v(k+1) \\ a(k+1) \\ \delta(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} s(k) + \frac{v(k)\cos(\mu(k)+\beta(k))}{1-d(k)\kappa(k)}\Delta T \\ d(k) + v(k)\sin(\mu(k)+\beta(k))\Delta T \\ \mu(k) + (\frac{v(k)}{l_r}\sin\beta(k) - \kappa(k)\frac{v(k)\cos(\mu(k)+\beta(k))}{1-d(k)\kappa(k))\Delta T} \\ v(k) + a(k)\Delta T + \frac{1}{2}u_{jerk}\Delta T^2 \\ a(k) + u_{jerk}\Delta T \\ \delta(k) + \omega\Delta T + \frac{1}{2}u_{steer}\Delta T^2 \\ \omega(k) + u_{steer}\Delta T \end{bmatrix} \tag{B.5}$$

The control bounds and state constraints are also included as follows:

$$\begin{aligned} v_{\min} \leq & v(t) \leq v_{\max} \\ a_{\min} \leq & a(t) \leq a_{\max} \\ \delta_{\min} \leq & \delta(t) \leq \delta_{\max} \\ \omega_{\min} \leq & \omega(t) \leq \omega_{\max} \\ u_{jerk,\min} \leq & u_{jerk}(t) \leq u_{jerk,\max} \\ u_{steer,\min} \leq & u_{steer}(t) \leq u_{steer,\max} \end{aligned} \tag{B.6}$$

B.3 Simulation Results

In this simulation, we build a scenario of roundabout which connects a 3/4 circle with radius $r = 27m$ to a 100m length straight road segment. We use the proposed MPC approach to track the reference trajectory given by our OCBF controller. The first simulation is conducted without adding noises. The tracking performance of the MPC controller is shown in Fig. B·2, where we can see the MPC trajectory precisely follows the center line of the road (also the reference trajectory).

The tracking errors of the three states (lateral offset d, velocity v and along-trajectory distance s) are shown in Fig. B·3, B·4 and B·5 respectively.

The tracking error of the lateral offset is almost zero except at around 8s when the

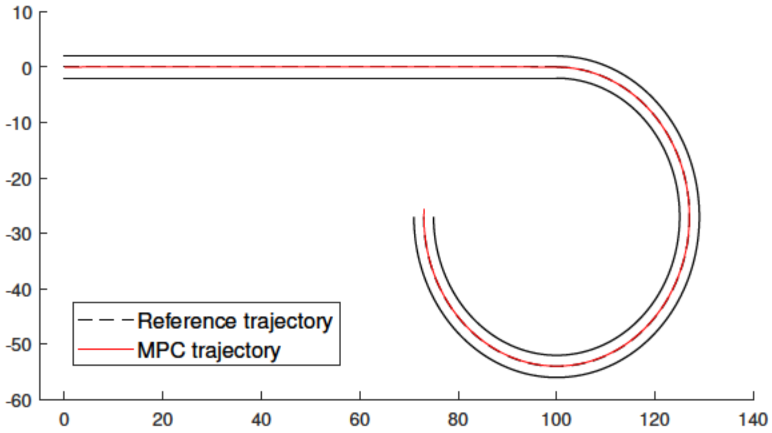


Figure B·2: Tracking performance of MPC

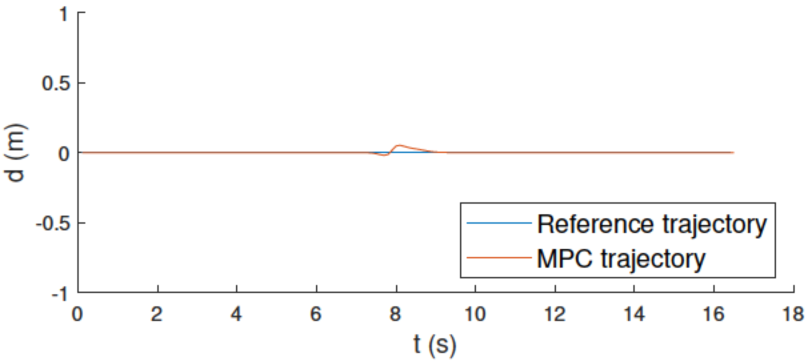


Figure B·3: Tracking error of d

vehicle enters the circular part of the road. The change of the road curvature results in a lateral offset of 8cm, which is neglectable comparing to the size of the roundabout. While avoiding the lateral offset, the controller can also track the referenced velocity as well as position accurately at the same time, as shown in Fig. B·4 and B·5. This indicates that by using an auxiliary method like MPC, we can deal with the lateral offset while tracking our trajectory given by the OCBF controller without much effort.

We further illustrate the power of the controller by adding noises to the vehicle dynamics. Uniform noises range from $[0,1]$(m/s) are added both to the along-trajectory velocity $\dot{s}$ and the lateral offset velocity $\dot{\mu}$ which represents model inaccuracy as well as some vehicle

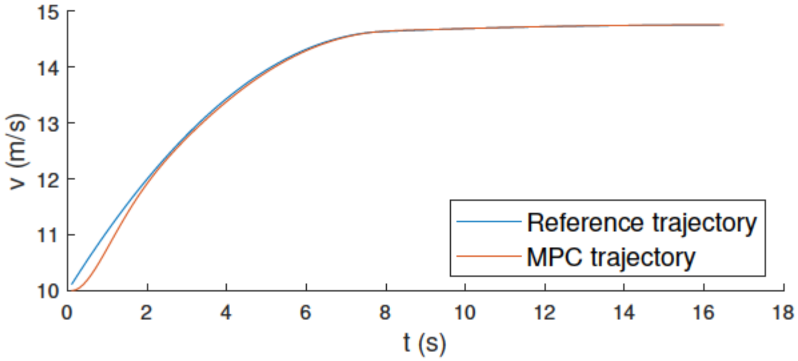


Figure B·4: Tracking error of v

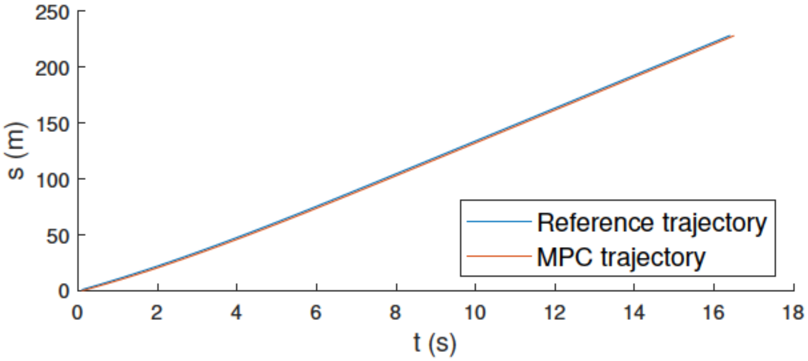


Figure B·5: Tracking error of s

malfunction like imbalanced type pressure which may result in biased error. The tracking performance of d, v and s with noises added are shown in Fig. B·6, B·7 and B·8.

Even when the noise is added, the vehicle can still track the referenced trajectory well. The lateral offset error increases to around 20cm due to the large biased noise in $\dot{d}$ but is still quite small. Furthermore, it always keeps its error level within this acceptable range and never diverge. The controller also performs well in tracking the velocity trajectory and the position trajectory when noise is included.

From the simulation results, we argue that by using an auxiliary method like MPC, we are able to deal with the lateral offset while tracking the trajectory given by our OCBF controller. The OCBF controller also has the power to deal with the lateral offset by directly deriving CBFs from the vehicle dynamics (B.1).

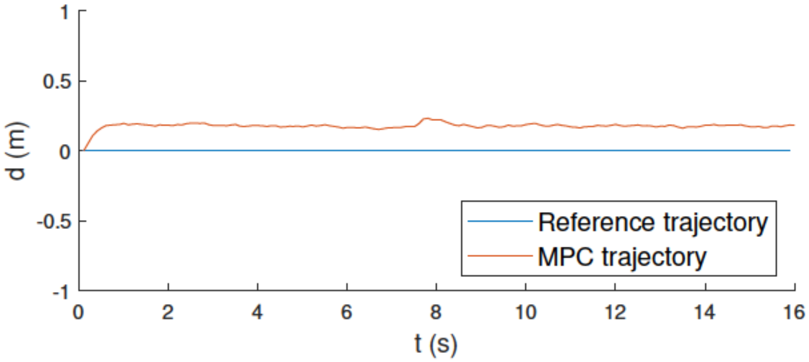


Figure B·6: Tracking error of *d* with noise

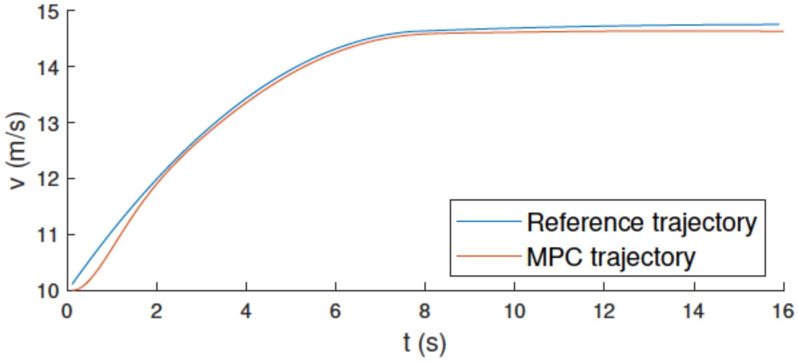


Figure B·7: Tracking error of *v* with noise

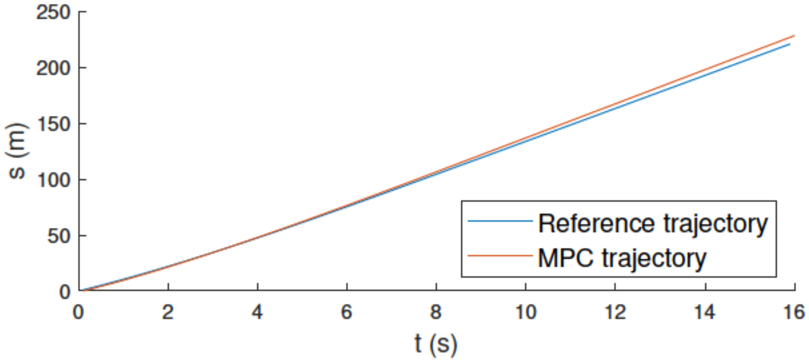


Figure B·8: Tracking error of *s* with noise

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938.

Agrawal, A. and Sreenath, K. (2017). Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Robotics: Science and Systems*, volume 13. Cambridge, MA, USA.

Altché, F. and de La Fortelle, A. (2017). An lstm network for highway trajectory prediction. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pages 353–359. IEEE.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE.

Ames, A. D., Galloway, K., and Grizzle, J. W. (2012). Control lyapunov functions and hybrid zero dynamics. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 6837–6842. IEEE.

Ames, A. D., Galloway, K., Sreenath, K., and Grizzle, J. W. (2014a). Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891.

Ames, A. D., Grizzle, J. W., and Tabuada, P. (2014b). Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278. IEEE.

Bichiou, Y. and Rakha, H. A. (2018). Developing an optimal intersection control system for automated connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1908–1916.

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bryson and Ho (1969). *Applied Optimal Control*. Ginn Blaisdell.

Cao, W., Mukai, M., Kawabe, T., Nishira, H., and Fujiki, N. (2015). Cooperative vehicle path generation during merging using model predictive control with real-time optimization. *Control Engineering Practice*, 34:98–105.

Chalaki, B., Beaver, L. E., and Malikopoulos, A. A. (2020). Experimental validation of a real-time optimal controller for coordination of cavs in a multi-lane roundabout. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 775–780. IEEE.

Dawson, C., Qin, Z., Gao, S., and Fan, C. (2022). Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, pages 1724–1735. PMLR.

Flannery, A. and Datta, T. (1997). Operational performance measures of american roundabouts. *Transportation research record*, 1572(1):68–75.

Freeman, R. and Kokotovic, P. V. (2008). *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media.

Galloway, K., Sreenath, K., Ames, A. D., and Grizzle, J. W. (2015). Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332.

Garcia, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348.

Gers, F., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2.

Glotfelter, P., Cortés, J., and Egerstedt, M. (2017). Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315.

Hult, R., Campos, G. R., Steinmetz, E., Hammarstrand, L., Falcone, P., and Wymeersch, H. (2016). Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation. *IEEE Signal Processing Magazine*, 33(6):74–84.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35.

Kamal, M., Mukai, M., Murata, J., and Kawabe, T. (2013a). Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Transactions on Control Systems Technology*, 21(3):831–841.

Kamal, M. A. S., Mukai, M., Murata, J., and Kawabe, T. (2011). Ecological vehicle control on roads with up-down slopes. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):783–794.

Kamal, M. A. S., Mukai, M., Murata, J., and Kawabe, T. (2013b). Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Transactions on Control Systems Technology*, 21(3):831–841.

Khalil, H. K. (2015). *Nonlinear control*, volume 406. Pearson New York.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., and Pérez, P. (2022). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926.

Levine, W. and Athans, M. (1966). On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, 11(3):355–361.

Malikopoulos, A. A., Cassandras, C. G., and Zhang, Y. J. (2018). A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93:244–256.

Martin-Gasulla, M., García, A., and Moreno, A. T. (2016). Benefits of metering signals at roundabouts with unbalanced flow: Patterns in spain. *Transportation Research Record*, 2585(1):20–28.

Milanés, V., Godoy, J., Villagrá, J., and Pérez, J. (2010). Automated on-ramp merging system for congested traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):500–508.

Mukai, M., Natori, H., and Fujita, M. (2017). Model predictive control with a mixed integer programming for merging path generation on motor way. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 2214–2219. IEEE.

Nagumo, M. (1942). Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 24:551–559.

Nor, M. H. B. M. and Namerikawa, T. (2018). Merging of connected and automated vehicles at roundabout using model predictive control. In *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 272–277. IEEE.

Ntousakis, I. A., Nikolos, I. K., and Papageorgiou, M. (2016). Optimal vehicle trajectory planning in the context of cooperative merging on highways. *Transportation research part C: emerging technologies*, 71:464–488.

Pinnow, J., Masoud, M., Elhenawy, M., and Glaser, S. (2021). A review of naturalistic driving study surrogates and surrogate indicator viability within the context of different road geometries. *Accident Analysis & Prevention*, 157:106185.

Prajna, S. and Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In Alur, R. and Pappas, G. J., editors, *Hybrid Systems: Computation and Control*, pages 477–492, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rathgeber, C., Winkler, F., Kang, X., and Müller, S. (2015). Optimal trajectories for highly automated driving. *International Journal of Mechanical and Mechatronics Engineering*, 9(6):969–975.

Rios-Torres, J., Malikopoulos, A., and Pisu, P. (2015). Online optimal control of connected vehicles for efficient traffic flow at merging roads. In *2015 IEEE 18th international conference on intelligent transportation systems*, pages 2432–2437. IEEE.

Rios-Torres, J. and Malikopoulos, A. A. (2016). A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077.

Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76.

Sardain, P. and Bessonnet, G. (2004). Forces acting on a biped robot. center of pressure-zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 34(5):630–637.

Shladover, S. E., Desoer, C. A., Hedrick, J. K., Tomizuka, M., Walrand, J., Zhang, W.-B., McMahon, D. H., Peng, H., Sheikholeslam, S., and McKeown, N. (1991). Automated vehicle control developments in the path program. *IEEE Transactions on vehicular technology*, 40(1):114–130.

Tee, K. P., Ge, S. S., and Tay, E. H. (2009). Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927.

Varaiya, P. (1993). Smart cars on smart roads: problems of control. *IEEE Transactions on automatic control*, 38(2):195–207.

Vogel, K. (2003). A comparison of headway and time to collision as safety indicators. *Accident analysis & prevention*, 35(3):427–433.

Wu, B., Iandola, F., Jin, P. H., and Keutzer, K. (2017). Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 129–137.

Xiao, W. and Belta, C. (2019). Control barrier functions for systems with high relative degree. In *Proc. of 58th IEEE Conference on Decision and Control*, pages 474–479, Nice, France.

Xiao, W. and Belta, C. (2022). High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662.

Xiao, W., Belta, C., and Cassandras, C. G. (2021a). Event-triggered safety-critical control for systems with unknown dynamics. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 540–545.

Xiao, W., Belta, C., and Cassandras, C. G. (2022a). Adaptive control barrier functions. *IEEE Transactions on Automatic Control*, 67(5):2267–2281.

Xiao, W., Belta, C. A., and Cassandras, C. G. (2022b). Sufficient conditions for feasibility of optimal control problems using control barrier functions. *Automatica*, 135:109960.

Xiao, W. and Cassandras, C. G. (2020). Decentralized optimal merging control for connected and automated vehicles with optimal dynamic resequencing. In *2020 American Control Conference (ACC)*, pages 4090–4095.

Xiao, W. and Cassandras, C. G. (2021a). Decentralized optimal merging control for connected and automated vehicles on curved roads. In *2021 IEEE Conference on Decision and Control (CDC)*, pages 2677–2682. IEEE.

Xiao, W. and Cassandras, C. G. (2021b). Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees. *Automatica*, 123:109333.

Xiao, W., Cassandras, C. G., and Belta, C. (2019). Decentralized merging control in traffic networks with noisy vehicle dynamics: A joint optimal control and barrier function approach. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3162–3167. IEEE.

Xiao, W., Cassandras, C. G., and Belta, C. (2020). Decentralized optimal control in multi-lane merging for connected and automated vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Xiao, W., Cassandras, C. G., and Belta, C. (2023). *Safe Autonomy with Control Barrier Functions: Theory and Applications*. Springer Nature.

Xiao, W., Cassandras, C. G., and Belta, C. A. (2021b). Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles. *Automatica*, 129:109592.

Xiao, W., Mehdipour, N., Collin, A., Bin-Nun, A. Y., Frazzoli, E., Tebbens, R. D., and Belta, C. (2021c). Rule-based evaluation and optimal control for autonomous driving. *ArXiv:2107.07460.*

Xu, H., Cassandras, C. G., Li, L., and Zhang, Y. (2022a). Comparison of cooperative driving strategies for cavs at signal-free intersections. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7614–7627.

Xu, H., Feng, S., Zhang, Y., and Li, L. (2019a). A grouping-based cooperative driving strategy for cavs merging problems. *IEEE Transactions on Vehicular Technology*, 68(6):6125–6136.

Xu, H., Xiao, W., Cassandras, C. G., Zhang, Y., and Li, L. (2022b). A general framework for decentralized safe optimal control of connected and automated vehicles in multi-lane signal-free intersections. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17382–17396.

Xu, H., Zhang, K., and Zhang, D. (2016). Multi-level traffic control at large four-leg roundabouts. *Journal of Advanced Transportation*, 50(6):988–1007.

Xu, H., Zhang, Y., Cassandras, C. G., Li, L., and Feng, S. (2020). A bi-level cooperative driving strategy allowing lane changes. *Transportation research part C: emerging technologies*, 120:102773.

Xu, H., Zhang, Y., Li, L., and Li, W. (2019b). Cooperative driving at unsignalized intersections using tree search. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4563–4571.

Xu, K., Cassandras, C. G., and Xiao, W. (2021). Decentralized time and energy-optimal control of connected and automated vehicles in a roundabout. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 681–686.

Xu, K., Xiao, W., and Cassandras, C. G. (2022c). Feasibility guaranteed traffic merging control using control barrier functions. In *2022 American Control Conference (ACC)*, pages 2309–2314. IEEE.

Yang, X., Li, X., and Xue, K. (2004). A new traffic-signal control for modern roundabouts: method and application. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):282–287.

Zhang, Y. and Cassandras, C. G. (2018). A decentralized optimal control framework for connected automated vehicles at urban intersections with dynamic resequencing. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 217–222. IEEE.

Zhang, Y. and Cassandras, C. G. (2019). Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees. *Automatica*, 109:108563.

Zhang, Y., Cassandras, C. G., and Malikopoulos, A. A. (2017). Optimal control of connected automated vehicles at urban traffic intersections: A feasibility enforcement analysis. In *2017 American Control Conference (ACC)*, pages 3548–3553.

Zhao, L., Malikopoulos, A., and Rios-Torres, J. (2018). Optimal control of connected and automated vehicles at roundabouts: An investigation in a mixed-traffic environment. *IFAC-PapersOnLine*, 51(9):73–78.

CURRICULUM VITAE

[redacted]