UNIVERSITY OF CALIFORNIA

Los Angeles

A Parallel Development and Testing Framework for Cooperative Driving Automation

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Civil Engineering

by

Xu Han

2024

ABSTRACT OF THE DISSERTATION

A Parallel Development and Testing Framework for Cooperative Driving Automation

by

Xu Han

Doctor of Philosophy in Civil Engineering

University of California, Los Angeles, 2024

Professor Jiaqi Ma, Chair

Automated driving technology has made significant strides, aiming to revolutionize transportation through enhanced safety and efficiency. However, significant gaps still exist in Cooperative Driving Automation (CDA) development and testing methods. Due to the scarcity of large-scale real-world deployments, CDA research often combines simulated and physical environments to achieve the realism that simulations alone cannot provide. Experiments require complex traffic scenarios and emphasize cooperation among multiple agents, which are challenging to manage with conventional methods. As a result, typical CDA research stops after initial benchmarking improvements without fully addressing system completeness, leaving a gap between prototypes and practical implementation.

To bridge this gap, this dissertation introduces a parallel development and testing framework inspired by Transportation 5.0 and the scenario engineering concepts [42,96], designed to address these unique challenges, accelerating prototyping and validation to ensure prototypes are robust, reliable, and ready for deployment.

Building upon the proposed framework, three key research projects were conducted to achieve real-world deployment of CDA systems. First, a cooperative platooning algorithm was developed and tested, enabling multi-lane platooning functions like cruising, lane changing, and adjusting platoon members across two lanes. Validated in simulation, it was later deployed on up to five Level 3-capable vehicles. Second, distributed tests were performed

using the VOICES platform with four participants across the country, each equipped with various CDA tools such as traffic simulators and Level 3-capable vehicles. This allowed heterogeneous, real-time interaction to jointly enhance system performance. Lastly, an ADS regulation-aware path planning algorithm was developed and deployed on Level 3-capable vehicles. It utilizes a machine-readable regulation database to extract real-world California vehicle codes and employs a vision-language model to interpret the environment from camera inputs, integrating regulations into the planning process to select optimal future actions.

In summary, this dissertation introduces a comprehensive, parallel development and testing framework for CDA, bridging the gap between prototyping and real-world deployment. By incrementally introducing risk factors and gradually validating systems through simulation and real-world testing, the framework ensures that CDA systems are robust, reliable, and ready for implementation. The successful application of this framework across key projects demonstrates its effectiveness in advancing the field of automated transportation.

The dissertation of Xu Han is approved.

Ali Mosleh

Ian Roberts

Yuan Tian

Xin Xia

Jiaqi Ma, Committee Chair

University of California, Los Angeles

2024

To Rugui, the pillar of our family, and our companion, Woody, for filling our lives with warmth and joy.

Contents

List of Figures

Acknowledgements

Growing up, I was never a top student, and I never thought about pushing myself or challenging myself for any significant advancement. I certainly never saw myself as a PhD candidate. But I was always curious, always wanting to go one step further to understand the reasons behind things in life—sports, cooking, automobiles, all of them.

My PhD journey began with this same curiosity about the field. I bluntly started this path, not knowing how much of a blessed opportunity I had grasped or how much it would change me—to become more persistent, attentive, and confident, and, most importantly, to challenge myself.

To this day, I hold my deepest respect and gratitude for my advisor. He had some faith in me that I had never had for myself. Without his recognition and, of course, his guidance and support, I would never have imagined myself completing my dissertation. He has also shaped me, teaching me the importance of staying curious and attentive to detail in research—qualities I believe will benefit me for a lifetime. We tend to think that events change the course of our lives, but I have come to realize that it is the person making these things possible who is the true gift in life.

I am deeply grateful to my dissertation committee, who have all contributed valuable and insightful suggestions for my work. It is also an honor to be part of my talented team, whose encouragement, collaboration, and expertise have helped me in so many ways and brought out the best in me. Thank you all for being part of this journey.

Lastly, my family's unwavering support and contribution are essential for my PhD journey. Throughout my life, I have never doubted for a moment that my parents would have my back. They always provided me with the best they could and served as such a great model for my own life. Because of them, I never had to worry for one second about my basic needs and could focus solely on my interests and goals. It is such an incredible privilege. I am also deeply blessed to have my wife, Rugui, and our companion, Woody. They are my dearest friends and the spine of my daily life, bringing enrichment and meaning to me from every morning to every dawn.

Curriculum Vita

Education

2020 - Present	PhD in Civil Engineering, University of California, Los Angeles
2017 - 2019	MS in Electrical Engineering, Washington University in St. Louis
2012 - 2017	BS in Electrical Engineering, University of Cincinnati

Publication

[1] Han, Xu, Runsheng Xu, Xin Xia, Anoop Sathyan, Yi Guo, Pavle Bujanović, Ed Leslie, Mohammad Goli, and Jiaqi Ma. "Strategic and tactical decision-making for cooperative vehicle platooning with organized behavior on multi-lane highways." *Transportation Research Part C: Emerging Technologies* 145 (2022): 103952.

[2] Han, Xu, Zonglin Meng, Xin Xia, Xishun Liao, Yueshuai He, Zhaoliang Zheng, Yutong Wang et al. "Foundation intelligence for smart infrastructure services in transportation 5.0." IEEE Transactions on Intelligent Vehicles (2024).

[3] Xu, Runsheng, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. "Opencda: an open cooperative driving automation framework integrated with co-simulation." In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 1155-1162. IEEE, 2021.

[4] Xia, Xin, Zonglin Meng, Xu Han, Hanzhao Li, Takahiro Tsukiji, Runsheng Xu, Zhaoliang Zheng, and Jiaqi Ma. "An automated driving systems data acquisition and analytics platform." Transportation research part C: emerging technologies 151 (2023): 104120.

Chapter 1

Introduction

In recent years, Automated Driving Systems (ADS) have achieved significant advancements, offering numerous benefits such as enhanced road safety, improved traffic efficiency, reduced fuel consumption, and various environmental and economic advantages. Additionally, Cooperative Driving Automation (CDA), as defined by SAE J3216 [67], utilizes machine-to-machine and Vehicle-to-Everything (V2X) communications to facilitate collaboration among vehicles, pedestrians, and infrastructure. This collaboration enhances the perception and decision-making capabilities of all ADS involved by enabling real-time data sharing and coordinated responses, leading to more accurate environmental awareness and optimized driving strategies. These developments have garnered substantial interest and investment due to their immense potential to fundamentally transform transportation [10, 13, 85, 89, 102, 103].

Despite significant advancements in ADS and CDA technologies, large-scale deployment remains challenging, mainly due to the lack of a robust framework to manage the complex development and testing required for CDA systems. Ma et al. [55] addresses validation and verification but focus solely on Decision-Making and Planning. Studies on scenario-based experiments and scene generation for automated vehicles [88, 101] concentrate on automated scenario generation, with limited emphasis on CDA development. Similarly, real-world CDA testing has been explored; Fremont et al. [28] propose a novel approach to scenario-based safety testing for autonomous vehicles in industrial settings, and Szalay [90] introduces an X-in-the-Loop framework leveraging advances in vehicle automation and testing requirements. However, these works do not address algorithm deployment or operational management processes. Most importantly, these efforts focus on specific tasks without offering a structured, systematic approach to CDA technologies. This lack of a structured framework hinders the integration of diverse components, the consistency of performance, and the validation of systems in dynamic, real-world conditions.

The lack of standardized methodologies for combining physical and virtual environments, handling complex traffic scenarios, and coordinating multiple CDA agents impedes the scalability and reliability necessary for widespread implementation. Consequently, without a unified framework to streamline development and testing, CDA technologies struggle to transition from promising prototypes to robust, real-world applications. To tackle these challenges, we present a comprehensive development and testing framework for Cooperative Driving Automation (CDA). The proposed development and testing framework can address the challenges in CDA by providing structured integration of diverse components, standardized methodologies for complex environments and traffic scenarios, and coordinated management of multiple agents. This ensures scalability and reliability, enabling CDA technologies to transition smoothly from promising prototypes to robust, real-world applications. This dissertation details this framework and showcases three real-world research projects—focusing on CDA development and testing in highway scenarios, intersection scenarios, and heterogeneous agent interactions—all conducted using this framework.

1.1 A Parallel Development and Testing Framework

The motivation for developing this parallel development and testing framework arises from shortcomings in current CDA methodologies. Limited large-scale real-world deployments of CDA compel researchers to rely on a combination of simulated and physical environments to achieve necessary realism, as simulations alone cannot address the complexities of modern research demands. Additionally, CDA experiments often involve intricate traffic scenarios, specific system objectives, and performance metrics that are challenging to manage with conventional testing methods. The need for cooperation among multiple CDA agents further complicates these experiments, making them difficult to execute using traditional approaches. As a result, many research efforts cease after reaching performance benchmarks without advancing to comprehensive validation or real-world deployment, creating a critical gap between prototypes and practical implementation.

Concurrently, inspired by the Internet of Things (IoT) and advancements in AI and large models, researchers have recognized the need to transform how revolutionary technologies are implemented in transportation. Wang et al. [96] introduced Transportation 5.0, an integrated, autonomous, and decentralized framework that uses Distributed Autonomous Organizations (DAOs) to enhance the safety, security, and sustainability of intelligent transportation systems. This approach facilitates decentralized control and collaborative decision-making across agents, promoting system resilience and trust. Similarly, Li et al. [42] highlight the importance of scenario engineering in AI, emphasizing a shift from traditional feature engineering to scenario-based frameworks that rigorously test AI behaviors in realistic, complex situations. Together, these concepts promote a visionary approach to transportation technology, blending autonomous control with rigorous validation to build safe, trustworthy, and adaptable systems.

Inspired by these visionary approaches, this dissertation introduces a parallel development and testing framework specifically designed to streamline the prototyping and validation processes for CDA. Reflecting principles from Transportation 5.0 and scenario-based engineering, this framework combines mixed-reality testing environments with coordinated parallel operations, creating an adaptable setup that evaluates CDA systems under diverse, realistic conditions. Through structured scenario engineering and staged parallel validation processes, the framework ensures that CDA applications are robust, reliable, and ready for real-world deployment in complex transportation environments. Chapter 2 presents the proposed framework in detail, bridging existing gaps by integrating physical and virtual environments to enable experiments that blend real-world and simulated elements, enhancing both realism and validity. The framework supports scenario engineering and parallel validation at both the traffic level and the vehicle level, providing relevant performance indices that comprehensively evaluate system goals with experiments that are designed in a safe and strategic manner. It also manages parallel operations capable of handling multiple CDA agents, facilitating seamless cooperation among automated systems. By incorporating these key capabilities, the framework systematically supports the development and testing of CDA systems, making them robust, reliable, and prepared for real-world deployment. Overall, the parallel framework provides a thorough and structured approach to CDA development and validation, enabling efficient prototyping and deployment. Its adaptable design allows researchers to apply selected components or the entire framework, aligning with specific project requirements while ensuring a strong foundation for dependable CDA systems.

1.2 CDA Development and Testings in Highway Environment: Cooperative Multi-lane Platooning

The motivation for developing this project arises from the crucial role that highway scenarios play in the large-scale deployment of Cooperative Driving Automation (CDA) systems. Highways are integral to real-world transportation networks, where efficient and safe vehicle interactions are essential for achieving goals such as improved traffic flow, reduced congestion, and enhanced safety. While Cooperative Adaptive Cruise Control (CACC) has demonstrated effectiveness in managing longitudinal vehicle movements, there remains a significant gap in developing true cooperative methods that extend beyond single-lane or longitudinal control. Most existing solutions focus narrowly on single-lane or longitudinal dynamics, lacking the comprehensive coordination and advanced decision-making required for multi-lane highway environments. Addressing this gap is vital for the successful and widespread adoption of automated driving technologies on highways.

Chapters 3 and 4 of the dissertation focus on multi-lane platooning algorithms, where Chapter 3 emphasizes algorithm prototype development with simulation testing, and Chapter 4 elaborates on real-world deployment and field tests, both following the proposed parallel framework. The project aims to develop and validate cooperative strategies that operate effectively across multiple lanes, enhancing coordination and decision-making among vehicles. By utilizing the structured approach provided by the framework, the project ensures that the multi-lane platooning algorithms are extensively tested under realistic and complex highway conditions, thereby enhancing their reliability and scalability for real-world application. In particular, during the development phase (i.e., Chapter 3), the simulated testing environment is utilized to demonstrate the completeness of the multi-lane planning capability with various scenarios and testing data sources. The operations cover single-lane gap regulation, adjacent lane platoon joining and departing, and the intelligence controller's decision-making for a complex on-ramp merging scenario. For the real-world deployment and testing phase (i.e., Chapter 4), a combination of virtual and physical environments is integrated to fully validate the system's performance under real-world noises, delays, and disturbances. The corresponding scenarios, involving multiple L3-capable ADS vehicles, covers both same-lane regulation and adjacent lane joining, validating the system's performance on real-vehicle running in a controlled testing track, with operations covers comprehensively from communication protocols, wireless signal receive and process, algorithm reasoning and vehicle control.

1.3 CDA Development and Testings in Intersection Environment: Regulation-aware Path Planning

The motivation for this project arises from the critical importance of intersection scenarios in the large-scale deployment of CDA systems. Intersections are inherently complex environments where multiple vehicles and vulnerable road users interact, making adherence to traffic regulations particularly challenging. One of the most demanding tasks in these settings is ensuring compliance with multiple traffic laws simultaneously while navigating a dynamic traffic stream that includes not only vehicles but also pedestrians, cyclists, and other vulnerable road users (VRUs). Additionally, intersection scenarios often involve overlapping regulations, further complicating decision-making processes. Conventional methods typically rely on high-level interpretations or fixed universal principles to manage these complexities, which are insufficient for the nuanced and real-time demands of modern traffic environments. In contrast, CDA systems are designed to interpret the environment accurately, process relevant traffic regulations, and integrate this information into the planning process to make informed and compliant driving decisions.

Chapter 5 of the dissertation presents the proposed parallel development and testing framework applied to the development of a regulation-aware path planning system for CDA. This framework integrates both physical and virtual testing environments, enabling comprehensive scenario engineering and the effective management of multiple CDA agents interacting within complex intersection scenarios. The proposed algorithm dynamically interprets and applies relevant traffic laws based on real-time environmental data. Unlike traditional methods, this approach allows the CDA system to adaptively relate to applicable regulations during the planning and execution of driving maneuvers, ensuring legal and safe driving practices. The framework supports rigorous validation through both simulated and real-world tests, ensuring that the intersection management system is robust, reliable, and capable of handling the multifaceted demands of real-world intersections. The details of the development and testing processes, along with the implementation of the regulation-aware decision-making algorithm, are thoroughly explored in Chapter 5, demonstrating how the framework effectively bridges the gap between innovative prototypes and practical, deployable solutions for automated driving technologies.

1.4 CDA Distributed Testings with Heterogeneous Agents: VOICES Distributed Testings

The motivation for developing and conducting the distributed testing framework arises from the significant challenges posed by heterogeneous agents and the unobservability of other agents' internal processes within Cooperative Driving Automation (CDA) systems. Although there is cooperation among CDA agents through the exchange of standardized messages such as Signal Phase and Timing (SPaT) and Basic Safety Messages (BSM), the internal decision-making processes of each agent remain opaque, effectively making them black boxes. This lack of transparency complicates the accurate prediction and integration of behaviors across different CDA systems, especially when they originate from diverse manufacturers with varying algorithms and protocols. Traditional testing methods, which rely on isolated simulations or physical tests, are inadequate for assessing how these heterogeneous agents interact and cooperate in real-time environments. Furthermore, the complexity of managing multiple CDA agents, each with its own proprietary logic, exacerbates integration issues and hinders the seamless deployment of large-scale CDA solutions. To overcome these obstacles, a comprehensive distributed testing framework is essential. Such a framework must facilitate real-time interaction and collaboration among diverse CDA agents, enabling a more accurate and reliable assessment of their cooperative behaviors and ensuring that CDA systems can operate cohesively within a complex transportation ecosystem.

Chapter 6 of the dissertation presents the proposed distributed testing framework designed to enhance the development and validation of CDA systems by focusing on the integration of heterogeneous agents and addressing the unobservability of their internal processes. This framework establishes a secure, mixed-reality testing environment where multiple organizations can simultaneously utilize their existing simulated and physical test assets to interact with each other's CDA systems in real-time. By enabling comprehensive scenario engineering and the effective management of diverse CDA agents, the framework allows for the evaluation of cooperative behaviors and decision-making processes in a controlled yet realistic setting. The project introduces advanced coordination algorithms that facilitate seamless interaction among heterogeneous agents, ensuring that each system can interpret and respond to the actions of others despite the lack of transparency in their internal processes. Additionally, the framework supports rigorous validation through both simulated and real-world tests, ensuring that the CDA systems are robust, reliable, and capable of operating cohesively within a complex transportation ecosystem. Chapter 6 details the implementation of this distributed testing architecture, demonstrating how it effectively bridges the gap between diverse CDA prototypes and their practical, deployable solutions, thereby advancing the integration and scalability of automated driving technologies.

This dissertation presents a development and testing framework for CDA systems, addressing challenges in transitioning from prototypes to real-world deployment. The proposed framework integrates physical and virtual environments, supports complex scenario engineering and validation processes, and manages multi-agent interactions to ensure robust, reliable CDA systems. Note that the proposed framework is a comprehensive structure that covers from development to real-world validation and deployment, but not all models are applied when it comes to real-world application. In particular, the proposed framework is applied across three key projects: cooperative multi-lane platooning in highway settings, regulation-aware path planning in intersections, and distributed testing of heterogeneous agents. Each project demonstrates how the framework enables effective development and validation processes using the modules that are most creative to the project scope and research goals. On a high level, they all involve a comprehensive testing environment, sophisticated scenario engineering, extensive validation, and seamless coordination, bridging the gap between experimental outcomes and real-world application, ultimately enhancing the scalability and reliability of CDA technologies.

Chapter 2

A Parallel Development and Testing Framework for Cooperative Driving Automation

Cooperative Driving Automation (CDA) has significant potential to improve road safety, traffic efficiency, and environmental sustainability by fostering collaboration between vehicles, infrastructure, and road users through machine-to-machine and Vehicle-to-Everything (V2X) communication. However, scaling CDA deployment presents challenges due to the complex testing and validation requirements. Current methodologies often lack the integration of physical and virtual environments, the ability to manage diverse traffic scenarios, and the coordination of multiple automated agents. To address these gaps, this chapter introduces a parallel development and testing framework aimed at bridging the divide between prototype testing and real-world deployment.

The proposed parallel framework includes several guiding modules—physical and virtual environments, scenario engineering, parallel operations, and a newly added parallel validation module. Each module provides essential support for testing, but not all are necessary for every project. The parallel validation module provides a flexible testing framework that introduces progressively realistic disturbances, beginning with simulation tests, advancing through software-in-the-loop and hardware-in-the-loop testing, and extending to field and distributed testing environments. This comprehensive approach ensures that each phase effectively evaluates system performance, compatibility, and adaptability under increasingly realistic conditions. For each specific project or chapter, only the relevant modules will be selected according to project requirements, with explanations provided for these choices. By incorporating flexible yet cohesive elements, the framework supports robust CDA validation, enhancing reliability, scalability, and interoperability and facilitating the transition from prototype development to large-scale deployment.

2.1 Introduction

Intelligent Transportation Systems (ITS) are transforming the transportation landscape by utilizing advanced technologies to overcome traditional infrastructure limitations. These systems have shown significant potential to enhance transportation system goals, such as safety and traffic efficiency [5, 93]. According to SAE J3216 [67], Cooperative Driving Automation (CDA) involves vehicle automation that relies on Machine-to-Machine communication to enable coordinated interactions among multiple entities—such as vehicles, vulnerable road users (VRUs), and infrastructure components—equipped with communication capabilities. CDA represents a significant advancement in the field of automated transportation, offering the potential to revolutionize road safety, traffic flow, and environmental sustainability [5, 30, 107]. By facilitating collaboration between vehicles, infrastructure, and VRUs through advanced communication methods like Vehicle-to-Everything (V2X), CDA systems enable coordinated decision-making and perception across multiple entities. This coordination helps address complex challenges such as navigating intersections, merging lanes, and managing mixed traffic environments, which are difficult for single automated vehicles to handle independently [33, 102, 103]. As such, CDA has attracted substantial interest and investment from both industry and academia due to its promise to transform the transportation landscape.

However, despite these advancements, large-scale deployment of CDA technologies remains limited. Given the high costs of development and deployment, as well as the increased risks and legal constraints associated with a larger number of devices, a common approach for validation is to use a simulated virtual testing environment [47, 62, 74]. Simulated environments allow for the creation of controlled scenarios without property damage, legal constraints, or safety risks. However, most simulation platforms suffer from unrealistic environmental conditions and traffic behaviors. Moreover, they often feature simplified communication systems that rely on artificial noises that follow certain distributions [48, 80], which is particularly problematic when evaluating the real-world performance of CDA systems that rely on sophisticated V2X communication methodologies. Therefore, it is essential to incorporate some level of realistic environment to better capture the complexities and interactions that occur in real-world conditions while maintaining a mixture of virtual environments to limit the risk.

Integrating a virtual testing environment with a physical testing environment (i.e., digital twin) ensures that algorithms, which may perform well in controlled, simulated settings, are also robust enough to withstand the disturbances, noise, and delays introduced by real-world conditions. Such integration is essential for validating that CDA systems can maintain their performance and reliability when tested in a more robust testing scenario and environment [4,8,79]. The challenge lies in managing and integrating both virtual and physical testing environments into a coherent system. It is crucial to establish a systematic approach to handle these complexities and maintain consistency between the two environments. With this approach, the simulation portion provides valuable insights and knowledge about the CDA system's performance across various scenarios, serving as both prior experience for system improvements and a performance baseline for validation. Meanwhile, the physical system delivers realistic performance data, which can be used to refine and enhance the quality of the virtual testing environment, creating a more accurate and comprehensive testing process.

On the other hand, another primary obstacle in large-scale deployment is the complexity involved in developing and validating CDA systems with multiple agents. Unlike single-vehicle automated systems, which focus primarily on the behavior of a single entity [43, 103], CDA requires precise coordination and interaction among multiple agents, each contributing unique capabilities to the overall system. This involves not just vehicles but also roadside units, infrastructure elements, and even other VRUs. Scenario engineering plays a crucial role in this process, as it involves designing detailed, realistic testing environments that can capture the full spectrum of interactions that CDA systems might encounter in the real world. These scenarios are usually engineered to include diverse variables such as varying weather conditions, different types of road users, and unexpected events like emergency stops or lane changes.

With increasingly complex hybrid testing environments and challenging scenarios, CDA testing is becoming more involved and demanding. Combining virtual and physical environments, CDA systems must perform reliably across both simulated and real-world conditions. In scenarios specifically designed to emphasize cooperation, additional challenges emerge in managing timing, multi-agent interactions, and adapting to unexpected dynamics. To address these complexities, it is essential to design a validation protocol that includes a range of tests, each introducing different levels of real-world complexity—such as randomness, noise, and disturbances. These tests do not follow a strict sequence but instead represent a collection of options, each suited to various development stages with distinct focuses. By carefully selecting appropriate tests based on project emphasis and risk tolerance, the protocol allows developers to test CDA systems incrementally, minimizing unnecessary risk factors while focusing on core research objectives. This approach supports flexible and adaptive validation, enabling CDA systems to be evaluated under increasingly realistic conditions without requiring all tests to be completed at once.

Lastly, operations among the CDA systems introduce additional complexity to the development and validation processes by requiring the simultaneous management of processes that guide agent interactions and system adaptation. These operations involve coordinating communication, testing performance across diverse conditions, and continuously refining decision-making, ensuring that the system can respond effectively to dynamic, real-world scenarios. Well-managed operations among agents allow seamless collaboration and allow the system to learn from each interaction and experiment, making it an essential attribute for developing a robust CDA system that can adapt to the unpredictable nature of real-world deployment. In addition, effective operational management requires careful oversight of data flow and process alignment to maintain consistency and ensure that outcomes are both reliable and scalable.

2.2 Related Work

The development and validation of CDA systems have been extensively studied, and various frameworks have been proposed to address the challenges of testing or integration.

Many of these frameworks rely heavily on simulation environments due to their ability to replicate diverse scenarios without the risks associated with real-world testing. Feng [26] proposed a new framework for safety assessment of highly automated driving systems that integrates an augmented reality (AR) testing platform and a testing scenario library generation (TSLG) method. The AR testing platform generates simulated background traffic in test tracks, which interact with subject ADS under test to create a realistic traffic environment. Nalic [61] proposed a novel stress testing method (STM) based on traffic flow simulation software (TFSS) PTV Vissim and the vehicle simulation software IPG CarMaker. With this method, traffic participants are manipulated in the vicinity of the vehicle under test in order to provoke SCS derived from statistical accident data on motorways in Austria. ADS assessment in terms of scenarios is also discussed, where naturalistic driving data are used to generate test cases for Monte-Carlo simulations of ADS [18]. Because real-life data is used, the assessment allows conclusions to be drawn on how the ADS would perform in real traffic.

In terms of development, the simulated virtual environment has been extensively used as the foundation for cooperative driving automation frameworks, such as full-stack CDA development framework to test different CDA algorithms at both levels of traffic and individual autonomy [102], fully customizable and simulation platform that support the research of generalizable reinforcement learning algorithms for machine autonomy [41].

However, nearly all of these methods failed to utilize any form of real-life system or environment that captures realistic communication or operation arrangement. A mixture of virtual and physical environments, also known as the digital-twin environment, has gained popularity, providing a high-fidelity virtual representation of physical systems to test CDA algorithms in controlled conditions. Some efforts focused on utilizing a digital environment that replicated the world to design cooperative driving systems, allowing connected vehicles to cooperate with each other to cross intersections without any full stops using a slot reservation algorithm [97] or establish an Advanced Driver Assistance Systems (ADAS) that recommend vehicle behavior based on a digital replicated roadway segment with simulated live traffic stream [98]. Young et al., [105] establishes an infrastructure-based cooperative perception fusion engine based on a complete 3D digital representation of the current traffic state with measurable accuracy to support a wide range of downstream applications such as intelligent signal control, safety, and energy applications, and CDA applications.

These platforms enable researchers to simulate complex traffic scenarios, integrate virtual sensors, and analyze the behavior of multiple automated agents, providing a safe and controlled environment for testing before real-world deployment. While they offer greater realism compared to purely simulation-based methods, most existing work primarily uses these environments without establishing a comprehensive framework for testing. This approach can address aspects such as sensor noise, environmental variability, and the intricate dynamics of V2X communication but often lacks the structured methodology needed to integrate these elements into a cohesive testing and development process.

To address these challenges, we propose a parallel development and testing framework designed specifically for CDA. This framework integrates virtual and physical testing environments, detailed scenario engineering, and coordinated testing operations to provide a comprehensive approach for validating CDA systems. By leveraging a parallel setup, the framework ensures that CDA algorithms are thoroughly tested in simulated environments before being exposed to the complexities of real-world conditions, allowing for a smoother transition from prototype to deployment. This structured approach not only enhances the scalability and reliability of CDA technologies but also bridges the critical gap between innovative research and practical application, paving the way for safer and more efficient transportation systems.

2.3 Framework Contributions and Key Insights

This section explores the key contributions and insights derived from the implementation of the proposed parallel development and testing framework across multiple projects. As mentioned in Chapter 1, this dissertation introduces three projects (in four chapters) that demonstrate the practical application of the proposed parallel framework. This section will present the contributions, experiences, and insights gained from the process of applying the framework to these projects, highlighting its practical value and relevance. It emphasizes the benefits of adhering to the framework, such as improved system robustness, enhanced interoperability, and reliable performance validation under diverse conditions. These insights validate the framework as a comprehensive tool for advancing CDA technologies and provide valuable guidance for future research and practical applications, ensuring a scalable and effective pathway to real-world deployment.

2.3.1 Integration of Mixed-Reality Environment for CDA Development and Testing

The integration of physical and virtual environments is a critical contribution of the framework, addressing the limitations of using either environment alone for CDA testing. Virtual simulations provide scalability and controlled testing conditions but often lack real-world complexity, such as communication delays and environmental noise. Physical testing ensures realism but is resource-intensive and risky for early-stage development. By combining these environments, the framework allows algorithms to be iteratively refined in simulations and validated under real-world conditions, creating a balanced, efficient, and comprehensive testing process.

This integration leverages the strengths of both environments, enabling seamless synchronization. Virtual environments replicate real-world conditions as digital twins, supporting safe pre-testing and scenario refinement. Physical tests validate system performance under unpredictable, real-world conditions, and their data feeds back into simulations for further optimization. This dual-environment approach accelerates development, reduces risks, and ensures the systems are robust and deployment-ready.

The **Multi-Lane Platooning Algorithm** project involves both virtual and physical systems. The framework significantly contributed to the testing of a multi-lane platooning algorithm detailed in Chapters 3 and 4. In the virtual testing phase, CARLA was used to simulate scenarios such as on-ramp merging, adjacent-lane coordination, and gap regulation. The framework's integration of scenario engineering allowed the team to incrementally test the algorithm under controlled yet progressively complex conditions. Knowledge gained from these simulation tests informed improvements and adjustments to decision-making parameters and communication protocols, such as target platoon gap and V2V communication range. During physical testing at the SunTrax facility, the same framework enabled the transition by replicating the virtual scenarios in real-world environments. Level 3-capable vehicles validated the algorithm's robustness under real-world disturbances, including communication delays and road surface variability. This iterative process ensured that virtual testing seamlessly prepared the system for real-world deployment.

The **Regulation-Aware Path Planning** project also involves both virtual and physical systems. The framework facilitated the development and validation of a regulation-aware path planning algorithm in Chapter 5. Virtual testing environments were crucial for refining the vision-language model (VLM) by simulating complex intersection scenarios with dynamic traffic and overlapping regulations. The VLM relies on text prompts to identify key traffic elements and make context-aware decisions. Using simulation as a safe and convenient platform, these text prompts were fine-tuned for various scenarios, enabling the system to accurately interpret each situation and focus on the most critical traffic elements. The simulation phase ensured that the prompts were closely aligned with real-world conditions, creating a solid foundation for deployment.

Following the framework, the fine-tuned text prompts were directly applied in real-world testing, significantly reducing the time and effort required during field tests. At a real intersection, the algorithm leveraged these optimized prompts to identify and interpret key traffic elements, such as signals, road signs, and dynamic obstacles, enhancing its performance in noisy and dynamic environments. This dual-environment approach ensured the algorithm's efficiency and readiness for practical applications by smoothly transitioning from virtual to real-world testing.

2.3.2 Scenario Engineering for Comprehensive System Evaluation

The framework's structured scenario engineering process ensures CDA systems are comprehensively evaluated under diverse, realistic conditions. Unlike traditional testing approaches that rely on ad-hoc or overly simplified scenarios, the framework employs a systematic approach to create, manage, and validate scenarios aligned with specific research goals. This approach allows researchers to progressively introduce complexities—such as dynamic traffic conditions, environmental factors, and communication delays—ensuring systems are tested incrementally for both functionality and robustness.

Scenarios are designed with specific objectives in mind, such as optimizing gap regulation, ensuring compliance with traffic laws, or improving cooperative decision-making. Each scenario is parameterized to cover a wide range of conditions, starting with simple setups in simulations and advancing to real-world environments. Tailored performance metrics—such as response time, compliance rate, and coordination efficiency—are used to evaluate systems objectively across different testing phases. This structured progression ensures systems are resilient and adaptable to real-world complexities, providing confidence that they can scale effectively from prototypes to deployment.

The framework's scenario engineering process was crucial in testing the **Multi-Lane Platooning Algorithm**, as detailed in Chapters 3 and 4. Structured scenarios were designed to evaluate the algorithm's capabilities, including gap regulation in single-lane platoons, adjacent-lane merging, and on-ramp coordination. These scenarios were progressively refined, introducing challenges such as varying traffic densities and unpredictable maneuvers by human-driven vehicles to test the algorithm under increasingly realistic and complex conditions. This iterative approach allowed the algorithm's decision-making logic to be systematically improved at each stage.

For the **Regulation-Aware Path Planning** algorithm in Chapter 5, the framework facilitated the creation of scenarios specifically designed to include co-existing traffic regulations. These scenarios incorporated diverse traffic laws and dynamic elements, such as pedestrians and cyclists, to evaluate the system's ability to interpret and comply with multiple regulations under controlled conditions. The algorithm demonstrated the capability to handle complex legal environments while focusing on critical regulatory requirements. In physical testing, the scenarios were adapted to reflect real-world conditions, retaining key similarities to ensure the system could perform effectively. Real-world tests included live traffic signals, variable road conditions, and environmental disturbances, validating the algorithm's robustness and its ability to make legally compliant decisions in dynamic, real-time settings with overlapping regulations.

2.3.3 Parallel Validation with Multi-Stage Risk Management

The framework's multi-stage parallel validation process ensures the robustness and scalability of CDA systems by progressively increasing testing complexity. It begins with simulation-based testing, allowing researchers to explore algorithms under controlled and repeatable conditions. From there, it transitions to software-in-the-loop (SIL) and hardware-in-the-loop (HIL) phases, where real-time controls and physical hardware are introduced. These stages prepare systems for real-world conditions while maintaining a safe and cost-effective testing environment.

This process reduces risk by allowing issues to be identified and resolved early in development. The simulation stage focuses on initial functionality, while SIL and HIL tests introduce real-time elements like delays and sensor noise. Field testing validates system performance in real-world conditions, and distributed testing assesses multi-agent interactions under diverse scenarios. By gradually increasing complexity, the framework ensures that systems are tested comprehensively without unnecessary risks during early phases. The modularity of the framework allows researchers to tailor the validation process to project-specific needs, focusing on critical aspects such as communication protocols, decision-making algorithms, or sensor integration. This ensures efficient use of resources and targeted testing, resulting in systems that are reliable and ready for deployment.

The **VOICES distributed testing framework** introduced in Chapter 6 highlights the framework's capability to facilitate incremental validation with increasing complexity and risk. Beginning with simulation tests, the framework validated the distributed system's foundational integrity by ensuring reliable cloud-based interactions among multiple entities in a controlled environment. This established a baseline for performance, providing a foundation for subsequent phases. SIL testing introduced standardized SAE messages, such as Basic Safety Messages (BSM) and Signal Phase and Timing (SPaT), alongside additional encoders and decoders, enabling real-time interactions and increasing the system's exposure to operational noise. Finally, distributed testing incorporated heterogeneous interactions with mixed-reality assets, simulating realistic and diverse conditions while validating the system's scalability and adaptability. The framework ensured a structured progression at each step, maintaining manageable risk levels and reinforcing system robustness.

The **Platooning field test**, detailed in Chapters 3 and 4, illustrates the framework's role in structuring and enhancing the progression from simulation to real-world application while managing incremental complexity and risk. In the SIL phase, the framework integrated simulation with ROS to validate multi-lane platooning behaviors, such as gap regulation and multi-lane control, in a simulated three-vehicle environment with added communication costs. The framework's scenario engineering ensured that test cases reflected realistic traffic conditions while maintaining safety, allowing for the refinement of control algorithms and communication protocols in a controlled environment. For the HIL phase, the framework facilitated the transition to two physical vehicles, incorporating real-world dynamics such as communication delays and sensor noise. By leveraging insights and validated parameters from the SIL phase, the framework reduced development time and minimized potential failures. In the final field test, the framework supported scaling to a three-vehicle, multi-lane platoon, requiring synchronized coordination across lanes and adaptation to dynamic traffic conditions. Its structured validation approach ensured that the system was tested incrementally, with each phase building on the success of the previous, improving the efficiency and reliability of the overall development process.

2.3.4 Scalability and Flexibility in Framework Design

The framework's modular design provides scalability and adaptability, enabling researchers to tailor its components to meet specific project requirements. Rather than adhering to a rigid structure, researchers can selectively apply or combine elements from scenario engineering and parallel validation based on project objectives. For instance, the Regulation-Aware Path Planning framework introduced in Chapter 5 utilized a combination of field testing and HIL. While the field test was not fully automated, requiring a human driver to operate the vehicle, it incorporated hardware components and real-world traffic conditions to enhance realism. This flexibility ensures efficient use of resources, eliminates redundant testing, and allows the framework to support a wide range of projects, from small-scale experiments to large-scale, real-world implementations.

Scalability is further supported by the framework's ability to expand the testing scope. It transitions smoothly from individual vehicle tests to complex multi-agent systems involving roadside infrastructure and connected vehicles. For example, small-scale tests can focus on single-function validation, while large-scale scenarios can test system-wide interactions and real-world dynamics. This adaptability ensures that CDA systems are extensively tested across various levels of complexity, making the framework suitable for academic research and industrial applications.

2.4 System Overview

The proposed parallel development and testing framework for CDA is designed to address the complexities of developing and validating systems that require coordination among multiple agents. As shown in Figure 2.1, it integrates four core components: parallel systems, scenario engineering, parallel validation, and parallel operations. Each component plays a crucial role in establishing a comprehensive testing environment that enables CDA systems to be effectively evaluated across varied conditions. Parallel systems allow simultaneous testing in both virtual and physical settings, balancing the safety of controlled experimentation with the realism needed for real-world validation. Scenario engineering provides detailed and realistic testing conditions, replicating the complexity and variability of real-world traffic situations. Parallel validation is a comprehensive testing suite structured with increasing levels of real-world risk and disturbance, ranging from simulation to software-in-the-loop, hardware-in-the-loop, field, and distributed tests, allowing for adaptable validation pathways depending on project requirements. Finally, parallel operations manage coordination, testing, and iterative adaptation, ensuring that the system continuously learns and improves. Together, these components create a cohesive framework that supports thorough testing and validation, guiding CDA systems from prototype stages to large-scale deployment with enhanced reliability, scalability, and real-world readiness.

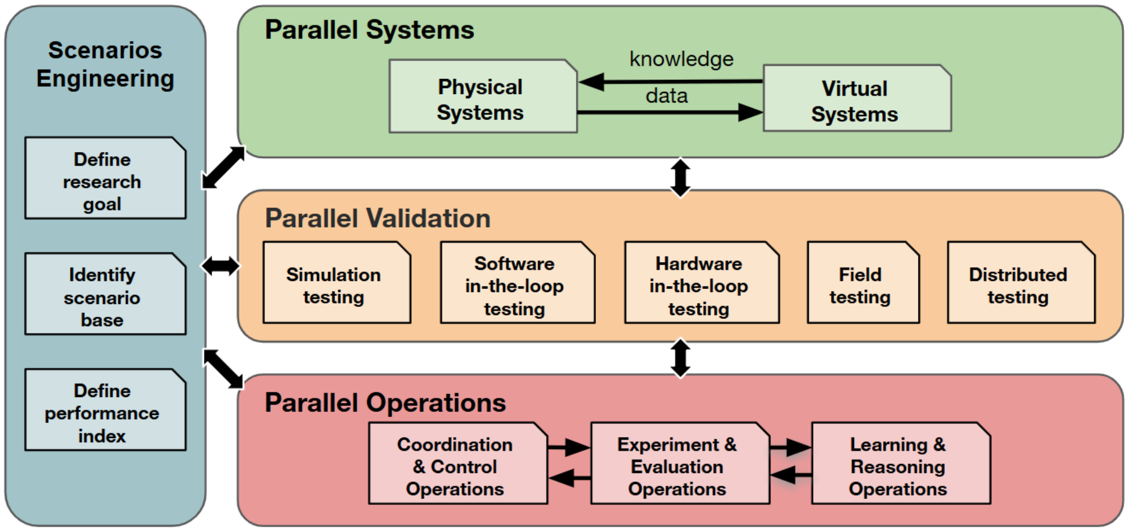


Figure 2.1: Overview of parallel development and testing framework.

2.4.1 Parallel Systems

The parallel systems component of the framework integrates both virtual and physical testing environments, creating a mixed-reality setup that allows for a more comprehensive evaluation of CDA systems, as shown in Figure.2.2. Virtual environments, such as digital-twin simulations using platforms like CARLA or SUMO, replicate real-world conditions with high fidelity, enabling researchers to test various scenarios in a controlled and safe digital space. These simulations can include diverse traffic conditions, complex road geometries, and different weather patterns, allowing CDA algorithms to be tested against a wide array of potential challenges without any risk to physical assets.

The physical testing environments include advanced vehicle platforms and roadside infrastructure that provide crucial real-world data about interactions and environmental factors. For example, a Level 3-capable C-ADS vehicle equipped with a ride-by-wire system,

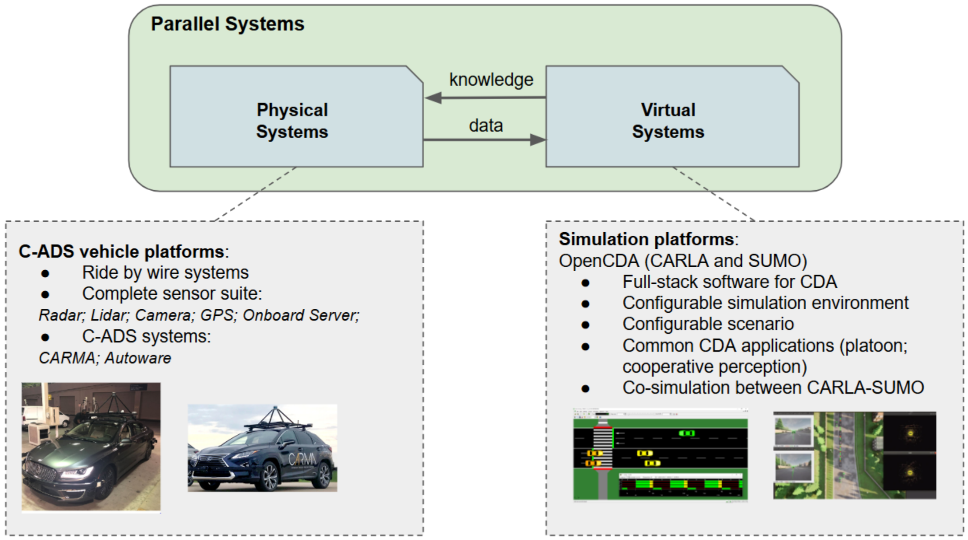


Figure 2.2: Overview of parallel systems module of the development and testing framework.

a complete sensor suite, and integrated C-ADS software such as CARMA or Autoware can be used to test CDA algorithms in real driving conditions. These vehicles can seamlessly integrate and test new algorithms, offering valuable insights into their performance outside of simulation. Additionally, physical systems might include roadside sensor suites combined with edge computing middleware as a central processing hub. An example of such a setup, as shown in Figure.2.3 is UCLA's smart intersection [53], where LiDAR, radar, cameras, and GPS sensors are installed on traffic signal poles, providing a comprehensive view of the intersection environment. This setup is complemented by two roadside middleware units that are connected to the sensors, allowing for real-time perception data sharing with nearby CDA units. Such physical testing setups enable the study of real-world challenges, such as coordinating sensor data from infrastructure with vehicle-based data, making them an essential complement to virtual testing in the overall framework.

The virtual systems component of the framework enables detailed, controlled testing in simulated environments, offering a safe and flexible way to evaluate CDA algorithms. A virtual environment can serve as a digital replica of a real-world setting, such as a digital twin of UCLA's smart intersection. This digital version replicates the road geometry, nearby

Figure 2.3: Overview of UCLA Smart Intersection setup.

buildings, and traffic signals based on the actual intersection layout and can be loaded into simulation platforms like CARLA. Within this virtual space, traffic streams can be spawned and managed, allowing for precise control over vehicle behaviors, pedestrian movements, and other dynamic elements, making it ideal for testing specific scenarios that would be difficult to reproduce consistently in the real world. Virtual environments can also utilize conventional simulation platforms, generally categorized into two types: traffic-level and vehicle-level simulators. Traffic-level simulators, such as SUMO [7], focus on modeling larger traffic flows and interactions within complex networks, making them suitable for assessing the impact of CDA systems on broader traffic efficiency and safety. Vehicle-level simulators, such as CARLA [21] or META Drive [41], provide a high-fidelity environment for testing the behavior and decision-making of individual automated vehicles, focusing on detailed sensor modeling and vehicle dynamics. Moreover, as an additional virtual environment alternative, running different simulation platforms in parallel through co-simulation allows each platform to handle the tasks it is best suited for. For example, SUMO might simulate overall traffic conditions, while CARLA focuses on the detailed interactions of a specific vehicle within that traffic. This co-simulation approach enhances the realism and comprehensiveness of virtual testing, allowing researchers to analyze both micro- and macro-level behaviors of CDA systems in a highly configurable environment.

By running these virtual and physical systems simultaneously, researchers can correlate the results, using insights from virtual tests to refine real-world trials and vice versa. For example, a virtual model might identify edge cases that are then validated through controlled physical tests, or real-world data from physical tests might be used to enhance the accuracy of simulations. This parallel approach ensures that the CDA system's performance in a simulated environment aligns closely with its behavior in actual driving conditions, creating a robust foundation for real-world deployment. Such a method allows the framework to balance the safety and scalability of virtual testing with the realism and unpredictability of physical trials, resulting in a well-rounded and reliable validation process.

2.4.2 Scenario Engineering

The concept of scenario engineering, as shown in Figure 2.4, is beyond just generating diverse driving conditions to challenge CDA systems; it is about creating a systematic framework that can guide the testing and validation process to ensure that the tested CDA applications are prepared for real-world deployment and validation. At its core, scenario engineering involves developing a structured approach to CDA testing that is closely aligned with well-defined research goals and establishes a scenario base suitable for all phases of testing. This approach is essential for developing and validating robust algorithms, ensuring that CDA experiments remain consistent with research objectives and are capable of handling diverse and challenging situations with reliability and safety.

The first step in scenario engineering is identifying and interpreting the specific research goals that align with the broader objectives of CDA development. These goals can vary widely, encompassing tasks such as cooperative perception, decision-making in dynamic traffic scenarios, regulation-aware driving, and vehicle cooperation during complex maneuvers like lane merging or platoon formation. At this stage, researchers must consider what capabilities

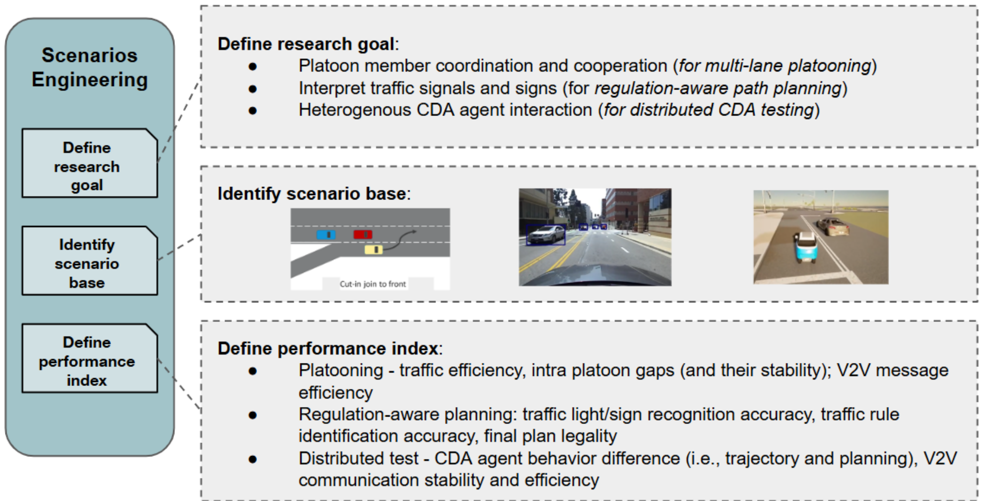


Figure 2.4: Detailed overview of scenario engineering.

and behaviors the CDA system needs to demonstrate, such as accurately sharing sensor data between vehicles, interpreting complex traffic regulations, or managing interactions with vulnerable road users (VRUs). For example, a goal might be to ensure that CDA vehicles can safely navigate intersections by interpreting traffic signals and pedestrian crossings, or to optimize vehicle cooperation in a platoon, where multiple vehicles need to adjust their speeds and positions smoothly.

Once the research goals are clearly defined, the next step is to generate a scenario base that covers a range of testing conditions, introducing varying levels of difficulty, noise, and interruptions. These scenarios are designed to progressively validate CDA algorithms, starting with basic functionality tests and advancing to more challenging real-world simulations. For example, initial scenarios might involve straightforward vehicle cooperation tasks, such as merging onto a highway or forming a basic platoon under ideal conditions. As testing progresses, more complexity is introduced, such as variable communication delays, unexpected obstacles such as road debris, or sudden changes in road conditions like icy patches or heavy rain. Similarly, for tasks like cooperative perception, early scenarios might involve clear visibility and perfect sensor conditions, while later scenarios introduce factors like fog or partial sensor failures to test the robustness of the system's ability to maintain situational awareness. These layered scenarios ensure that CDA systems are tested incrementally, with each level building upon the previous one, thus creating a robust testing environment that closely mimics the unpredictable nature of real-world traffic.

Lastly, defining performance indices is a critical aspect of scenario engineering, particularly when the scenario base includes a wide variety of scenarios, each designed with specific research intentions. Performance indices serve as measurable metrics that align with the objectives of individual scenarios, ensuring that the evaluation process remains focused and meaningful. Given the diverse research goals in CDA development—ranging from cooperative perception and decision-making in dynamic traffic conditions to regulation-aware driving and complex vehicle coordination—each scenario requires tailored indices that reflect its unique intent. For example, a scenario testing cooperative perception might prioritize indices like sensor accuracy and data-sharing efficiency, while a platooning scenario could emphasize metrics such as inter-vehicle gap consistency and synchronization of speed adjustments. By defining and applying appropriate performance indices, researchers can ensure that CDA systems are not only tested comprehensively but also evaluated against the specific capabilities and behaviors that the scenarios are designed to challenge. This targeted approach enables a systematic validation process, making it possible to assess the readiness of CDA systems for real-world deployment while addressing the varied demands of different testing phases.

Scenario engineering serves as a critical component in the framework, rooted in a methodical approach that starts with identifying research goals and evolves through the creation of tailored scenarios that meet those objectives. By systematically introducing challenges and increasing complexity, scenario engineering allows researchers to comprehensively evaluate the readiness of CDA systems for real-world conditions. This structured method ensures that CDA technologies can be validated in a way that not only tests their core functions but also prepares them for the diverse and often unpredictable situations they will encounter upon deployment. The result is a more resilient and adaptive CDA system, capable of transitioning from experimental phases to effective, large-scale application in real-world transportation systems.

2.4.3 Parallel Validation

The concept of parallel validation extends beyond merely verifying CDA system functionality; it provides a structured and layered testing approach to prepare CDA applications for real-world deployment, introducing progressively increasing complexity, risk, and disturbance. Parallel validation is integral to the framework, forming a comprehensive suite of testing methods that systematically introduces varied environments and scenarios to rigorously assess system readiness. This approach is essential for refining CDA algorithms, ensuring that each testing phase aligns with the system's developmental goals and supports reliable, safe interactions under diverse and challenging conditions.

The parallel validation suite begins with **simulation testing**, where foundational algorithms are tested in a controlled virtual environment, allowing for early experimentation without physical risks. This simulation phase establishes a baseline for system performance, helping identify areas for improvement while keeping testing constraints manageable. By focusing on core capabilities within a predictable setup, simulation testing enables researchers to explore initial CDA responses, interactions, and decision-making logic in a safe yet informative environment.

Following simulation, the validation process advances to **SIL** and **HIL** testing, which incrementally increase testing fidelity by introducing real-time control and physical hardware components. In SIL, the CDA software interacts with a live virtual environment, bringing real-time responsiveness and enabling the system to handle dynamic data inputs. SIL serves as an essential bridge, integrating software performance with more realistic virtual scenarios. HIL builds on this by incorporating actual hardware components, such as sensors and actuators, exposing the system to physical disturbances like sensor noise and delays. SIL and HIL phases also add layers of heterogeneity, as independent agents, functioning as black boxes to one another, must interact under realistic conditions. This black-box interaction closely mirrors real-world CDA environments, where each system is isolated in its internal workings and capabilities. Testing how these autonomous agents interact is crucial for ensuring reliable and coordinated CDA performance in complex, multi-agent settings.

As development progresses, **field tests** allow the CDA system to operate in live environments, interacting directly with real road conditions, infrastructure, and road users. These field tests assess how CDA systems handle real-world complexity, such as unexpected events or environmental constraints, providing critical feedback for final adjustments. **Distributed testing** serves as the final stage, validating multi-agent CDA systems' ability to coordinate across varied geographic locations and platforms, and assessing how heterogeneous agents interact within larger networks. In distributed testing, each agent continues to act as a black box, independently interpreting the environment and responding based on its unique decision-making framework. This heterogeneity, mirroring real-world CDA conditions, is critical for ensuring that CDA systems can seamlessly operate alongside other autonomous systems, creating a cohesive and scalable deployment environment.

The progression from simulation to distributed testing is structured to adapt flexibly to project requirements, offering a pathway that incorporates increasingly complex testing environments. By gradually building layers of interaction and disturbance through a mix of virtual, hybrid, and physical tests, parallel validation enables each CDA system to demonstrate readiness for real-world scenarios. This approach ensures that CDA applications are not only robust and adaptable but also capable of managing the complex, heterogeneous agent interactions essential for real-world CDA functionality.

2.4.4 Parallel Operations

The parallel operations component of the framework is designed to ensure that various processes critical to the functioning of CDA systems work together seamlessly. It consists of three interconnected types of operations: Coordination and Control, Experiment and Evaluation, and Learning and Reasoning. Each of these operations plays a distinct role in enhancing the robustness, adaptability, and effectiveness of CDA systems, making them capable of handling the complex challenges encountered in real-world scenarios. An overview of this module is presented in Figure.2.5.

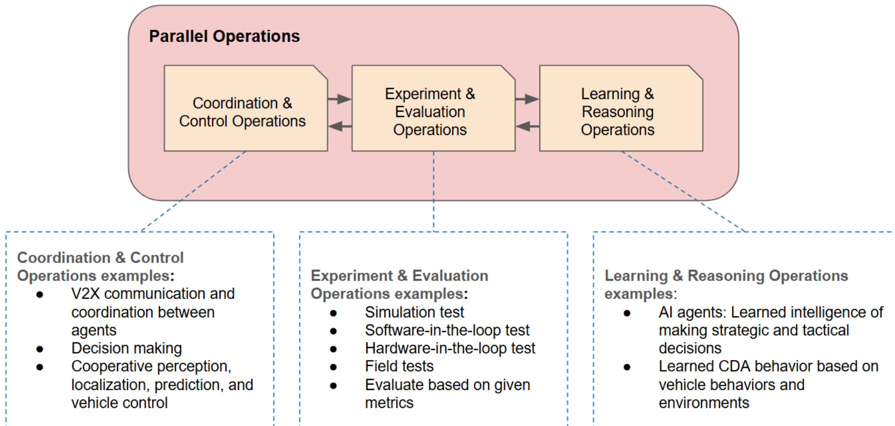


Figure 2.5: Overview of parallel operations module of the development and testing framework.

Coordination Control Operations involve managing the interactions and communications between different agents within the CDA system. This includes Vehicle-to-Everything (V2X) communication, where data is exchanged between vehicles, infrastructure elements like roadside units (RSUs), and even pedestrians equipped with communication devices. For example, a vehicle approaching an intersection can receive real-time updates about traffic signal phases, nearby pedestrian movements, or the intentions of other vehicles. Such information enables smoother and safer decision-making, like adjusting speed to allow a safe passage through an intersection. Coordination and Control also cover decision-making processes, where the system must process shared information to make cooperative decisions, such as determining the order of vehicle passage at a four-way stop or coordinating lane changes in a platoon. Additionally, this operation includes cooperative perception, where agents share sensor data (e.g., LiDAR or camera feeds) to create a richer, more accurate understanding of the environment. Localization and prediction help vehicles determine their precise position relative to other agents and predict their future movements, while vehicle control ensures that all entities adjust their behavior accordingly to maintain safe distances and coordinated maneuvers.

Experiment and Evaluation Operations focus on rigorously testing the performance of CDA systems across a variety of settings and conditions. This process is multi-faceted, encompassing several testing methodologies that address different stages of development. Simulation tests, for example, allow researchers to run thousands of scenarios virtually to understand how a CDA system might behave in diverse situations, such as heavy rain, foggy conditions, or sudden lane closures. Software-in-the-loop (SIL) tests extend this by integrating real software components with virtual models, providing a higher level of interaction with the simulation environment. These tests are particularly useful for evaluating how real-world code handles edge cases that may not frequently occur. Hardware-in-the-loop (HIL) tests take this a step further by incorporating physical hardware elements—like sensors or communication devices—into the testing loop, allowing a more accurate representation of the real-world interactions between software and physical components. Finally, field tests are conducted using actual CDA vehicles on test tracks or in controlled public environments. These tests verify the system's ability to handle real-world conditions, such as varying road textures, dynamic traffic conditions, and the influence of physical obstructions. Throughout all these testing stages, specific performance metrics, such as reaction time, collision avoidance, and fuel efficiency, are tracked to ensure that the system meets safety and efficiency standards.

Learning and Reasoning Operations are crucial for enabling the CDA system to adapt and improve over time based on the data gathered during the testing phases. This involves the use of AI agents that develop learned intelligence, allowing the system to make strategic and tactical decisions dynamically. For instance, AI-based reasoning can help a vehicle determine the best way to merge into high-speed traffic or adjust its speed during cooperative maneuvers like platooning. The learning process is supported by a continuous feedback loop from previous simulations and real-world data, allowing the AI agents to refine their decision-making models. This is particularly useful in situations that require quick adaptation, such as adjusting to a sudden change in traffic flow or reacting to unpredictable behavior from human drivers. The system's ability to learn from its experiences means that over time, it becomes better at predicting the actions of other road users and optimizing its responses. Moreover, the reasoning component allows the system to integrate context-aware decision-making, such as prioritizing safety over speed when navigating areas with high pedestrian density. This continuous learning helps ensure that the CDA system not only meets initial performance benchmarks but also remains effective as conditions evolve and new challenges emerge.

Together, these parallel operations create a cohesive framework for developing, testing, and refining CDA systems. By integrating coordination, evaluation, and learning processes, the framework ensures that the system can adapt to complex scenarios and consistently achieve high performance. For example, the ability to coordinate and control multiple agents allows for smoother traffic flow in congested urban environments, while the iterative testing process ensures that systems are robust enough to handle unexpected real-world conditions. Simultaneously, the learning and reasoning processes ensure that the system is not static but continues to improve as it encounters new situations. This comprehensive approach makes CDA systems more resilient and better suited for practical, real-world applications, ultimately facilitating the transition from controlled testing environments to large-scale deployment on public roads.

2.5 Summary

The parallel development and testing framework for CDA provides a comprehensive approach to overcoming the complexities of developing, validating, and deploying CDA systems. It integrates four core components—parallel systems, scenario engineering, parallel validation, and parallel operations—to create a structured, adaptable environment that allows for rigorous testing and refinement of CDA technologies.

- **Parallel Systems** module of the framework combines both virtual and physical testing environments, enabling researchers to balance the scalability and safety of simulations with the realism of real-world trials. Importantly, virtual environments provide valuable knowledge and insights through controlled experimentation and scenario analysis, while physical environments offer critical real-world data to validate and refine these findings while enhancing the virtual environment.

- **Scenario Engineering** module is central to the framework's ability to test CDA systems effectively. This process begins with identifying clear research goals and designing scenarios tailored to meet those objectives. Research objectives could range from vehicle-oriented tasks to RSU cooperation tasks, while scenarios typically introduce progressively challenging conditions to ensure that CDA algorithms are validated across a spectrum of challenges and that the tests are transitioned from prototype validations to real-world tests.

- **Parallel Validation** Parallel validation is a comprehensive testing framework that evaluates CDA systems across multiple environments, progressively increasing in complexity and real-world disturbance. By including simulation, SIL, HIL, field, and distributed tests, it enables adaptive, multi-stage validation that ensures systems are robust, scalable, and ready for real-world deployment.

- **Parallel Operations** module further enhances the framework by organizing processes into coordination, experimentation, and learning. In particular, Coordination and Control Operations focus on communication and decision-making among agents, Experiment and Evaluation Operations involve multi-stage testings, and Learning and Reasoning Operations use AI-driven analysis to help systems adapt and improve continuously, refining their strategies based on real-world data and past experiences.

Together, these elements create a cohesive environment that bridges the gap between theoretical development and practical deployment, making the framework an essential tool for advancing CDA technologies. Note that not all components within this framework are necessary for every phase of development and testing; rather, they serve as recommended guiding steps that form a complete, adaptable approach. The efficacy of this framework is demonstrated through three key CDA research projects detailed in the subsequent chapters, which focus on multi-lane cooperative platooning, regulation-aware path planning in intersections, and distributed testing with heterogeneous agents. For each project in the later chapters, specific modules from the parallel framework will be selectively applied based on project objectives, with rationale provided for each selection. Overall, these projects successfully utilized the framework for both prototype development and real-world deployment, proving its value in translating innovative concepts into scalable, deployable solutions for CDA.

Chapter 3

Strategic and Tactical Decision Making for Multi-lane Cooperative Platooning - Phase One: Prototype Development

Driving automation and vehicle-to-vehicle (V2V) communication enable cooperative automated driving systems (C-ADS) to meet essential transportation goals, such as sustainability, safety, and efficiency. Vehicle platooning, in particular, shows great promise for achieving these goals. This study proposes a multi-lane platooning algorithm structured through a hierarchical framework that leverages a mission-motion structure, enabling it to navigate complex multi-lane highway environments. The strategic level of the algorithm employs a deterministic finite state machine (FSM) protocol to guide platooning behavior, ensuring structured and organized vehicle operations. To address the limitations of heuristic approaches, a genetic fuzzy system is also incorporated, trained with the FSM as a baseline, to enhance algorithm performance in scenarios like on-ramp merging. On the tactical motion level, the algorithm generates trajectories for general ADS maneuvers, such as lane following and lane changing, while also supporting coordinated platooning through intent-sharing features that take into account the planned paths of other vehicles.

Aligned with the parallel development and testing framework, this study employs a parallel environment that integrates both vehicle-level and traffic-level simulation platforms to thoroughly evaluate the multi-lane platooning algorithm's performance. Scenario engineering is tailored to create diverse, platoon-specific highway conditions, testing the algorithm's behavior across various scenarios. Although only simulation testing from the parallel validation framework is involved at this phase, extensive real-world data are used to ensure that the algorithm can robustly handle real-world conditions. Parallel operations manage all experimental tasks, enabling real-time interactions among platoon members and coordinating decision-making processes across agents, providing a comprehensive assessment within complex, multi-agent settings. This approach effectively supports system validation, confirming that the proposed multi-lane platooning algorithm meets safety, efficiency, and system management goals for C-ADS-equipped vehicles within dynamic, realistic conditions.

3.1 Introduction

Intelligent Transportation Systems (ITS) are reshaping transportation and have demonstrated a promising potential to elevate transportation system management, operations, safety, and efficiency. As one of the essential sub-fields in ITS, Cooperative Driving Automation (CDA) is defined in SAE J3216 [67] and refers to vehicle automation that uses machine-to-machine communication to enable cooperation among two or more entities (e.g., vehicles, infrastructure components). Traffic efficiency, energy consumption, and driving safety can be significantly improved [86] with the potential benefit of significantly enhanced perception performance (i.e., perception accuracy and perception range) [102].

multiple vehicles that closely follow each other, has been extensively studied [3, 30, 60] to enhance the longitudinal behavior of advanced driver-assistance system via cooperation. Several highway traffic simulations conducted by the California Partners for Advanced Transportation Technology [30, 60] showed that autonomous adaptive cruise control (ACC) alone had little effect on lane capacity, even at high market penetration rates (MPRs). On-the-road experiments [60] showed that a stream of autonomous ACC vehicles is string unstable and more likely to restrain the lane capacity. Therefore, the combination of vehicle-to-vehicle (V2V) communication and ACC system allows an ADS-equipped vehicle to leverage the information from the preceding automated driving systems (ADS)-equipped vehicle to maintain an optimal following distance, leading to a string stable vehicle string. The primary improvement brought by CACC is reducing traffic congestion by improving highway capacity, increasing throughput, and attenuating traffic flow disturbances. In particular, the deployment of V2V communication has the potential to reduce the mean following time gap from about 1.4 s when driving manually to approximately 0.6 s when using CACC [60], significantly increasing highway lane capacity. Notably, because CACC systems allow much narrower following distance by establishing a V2V communication hierarchy, the lane capacity could be increased from 2,200 vehicles per hour (vph) to almost 4,000 vph at 100 percent market penetration [30, 60].

One major drawback of the existing CACC vehicle string is the lack of explicit consideration of lateral V2V cooperation. The CACC's capability of handling cut-in HDVs has been enhanced in a later study [60], but the improvement is limited to handling cut-in human-driven vehicles (HDVs) where the controller only considers the longitudinal direction. Various studies have recently focused on CACC applications in multi-lane settings (i.e., considered lateral control). Firoozi [27] utilized a pre-defined lookup table to optimize platooning behaviors among multiple lanes. The platoon can switch formation between single or multiple adjacent lanes to avoid congestion under the leader's command. This work established a leader–follower relationship but the lack of intent-sharing limit this algorithm as a passive reformation method with a fixed number of members. Pizarro [71] proposes a distributed scheme for lane changes inside a platoon that harnesses graph theoretical formulation for reference generation (during multi-lane formation) and distributed model predictive control (DMPC) for vehicle commands to achieve smooth multi-lane platoon maneuvers. The intentions of the peer members are shared and considered because each connected and automated vehicle (CAV) tracks relative state differences with regard to its neighbors. However, both multi-lane platooning methods assume a fixed platoon length, only managing existing vehicles within the platoon. Meanwhile, none of the existing work considers complex cooperation scenarios such as on-ramp mergers. In this regard, Uno's team [95] was the first to apply the "virtual vehicle" concept whereby a "ghost" leading CAV was mapped onto the highway before the merging maneuver happened, allowing a safer and smoother lane change control. Later, a distributed consensus-based highway on-ramp merging system [99] was developed, where two connected CACC systems are formed on the mainline and on-ramp, further enhancing the on-ramp merging bottleneck. In particular, the merging sequence is determined based on the estimated time-to-arrival calculated by road side unit, whereby each vehicle's intention (i.e., merging time and speed) is estimated. However, the work lacks active member management, which, when combined with intend-sharing, supports intention-based V2V cooperation such as joining and departing from a platoon. In addition, most current on-ramp merging studies either follow simple sequencing logic (i.e., first-in-first-out) or directly assume the existence of a near optimal sequence, leaving another research gap for an integrated multi-lane platooning algorithm that can assign near-optimal merging sequence with the objective to improve network capacity.

On the other hand, extensive studies [23, 37, 82] have been conducted on developing and deploying comprehensive state-of-the-art (SOTA) ADS platforms (i.e., end-to-end software stacks including perception, planning, and control modules) on capable vehicles to accomplish ADS functions and V2V communication. The SOTA ADS platforms have unique frameworks, in which the planning module includes a mission planning step configuring the semantic goal and a motion planning step that generates the corresponding trajectories. Such a framework allows the planning module to interact and cooperate with other cooperative automated driving systems (C-ADS) modules, such as the perception, control modules, and drive-by-wire modules, to achieve full ADS functions on a real-world C-ADS-equipped vehicle.

The necessity of developing and utilizing the SOTA ADS platform is prominent. For one, microscopic simulators, such as SUMO [49] and VISSIM [25], directly assign the speed and/or acceleration without explicitly considering vehicle dynamics or any form of ADS framework. Specifically, Mena-Oreja et al. [59] developed an open-source platooning simulator (i.e., PERMIT) based on SUMO to verify the effectiveness of platooning in terms of fuel reduction and congestion improvement. Though simulation data provide convincing results, the PERMIT platform considered traffic level control but did not consider vehicle dynamics or a proper ADS planning module. Secondly, most ADS studies that are based on vehicle-level simulators, such as CARLA [21], over-simplify the perception and planning module, thus directly loading simulation results and generating waypoints, limiting the software's scope to a prototype algorithm. Moreover, due to the computational limitations, though providing sufficient platoon-related results, studies that adopted CARLA [34, 91] do not consider traffic level performance and introduce no background traffic. Lastly, the scope of conventional CACC studies [30, 60, 72] is limited to longitudinal control without explicitly considering any lateral movements among the ADS vehicles. Despite the definition difference, though the "cut-in" condition (i.e., randomly merged humandriven vehicle) is considered, the scope of Milan´es [60] work still falls in longitudinal control but with additional consideration to damped the disturbance.

Accordingly, this study proposes a multi-lane platooning algorithm with a complete two-step (i.e., strategic mission planning and tactical motion planning) software framework. In particular, we define platooning as an organized behavior whereby each vehicle in a platoon has a responsibility towards the rest of the platoon to abide by certain agreed-upon rules. Regarding ADS structure, the algorithm utilizes perception results and composes a mission plan and a corresponding high-resolution motion plan that is executable for ADS control modules.

coordinated by the platoon leader, aim to move through traffic as safely and efficiently as possible. In contrast, with CACC, independent vehicles only receive front vehicles' real-time information, establishing a longitudinal, feed-forward ad hoc string (i.e., a vehicle string where each member considers solely its preceding vehicle's status) that allows reduced following gaps. Once a platoon has formed, or the members have subsequently joined, they must follow a certain set of rules and protocols that the leader commonly sets. A platoon leader may set the rules simply by passing what was received from infrastructure (e.g., the maximum amount of vehicles allowed in a platoon or speed limit), whereas other rules may be set by the leader of his own volition (e.g., protocols for allowing vehicles to join a platoon). Additionally, because platooning is a group activity, cooperation among each member is allowed in situations when the leader would coordinate the relevant members to accomplish a more involving task, such as lane-change, on-ramp merge, and a cut-in join. It is important to emphasize such distinction, as most conventional studies use CACC and platooning interchangeably, regardless of these fundamental differences. Table 3.1 lists the differences between platooning and CACC.

The other focus is on adhering to the existing SOTA ADS platform's framework that harnesses onboard computation power and sensor suites (i.e., camera, Lidar, GPS, and DSRC) to establish end-to-end ADS vehicle automation and communication. Existing SOTA ADS platforms, such as CARMA℠ [82], Autoware™ [37], and Apollo™ [23], share similar ideologies in which the ADS tasks are accomplished by three sequential modules – perception, planning, and control. As one of the planning applications, the multi-lane platooning algorithm relies on the perception module (i.e., localization and V2V communication) to acquire the relative speed and position of the surrounding ADS vehicles, by selecting proper maneuvers based on the heuristic finite state machine (FSM) in the mission planning step. Afterward, the corresponding mission plan containing the start-toend position and speed is fitted into a high-resolution motion plan (i.e., executable trajectory) that is ready to be executed by the control module. Notably, the proposed multi-lane platooning algorithm is parallel with other planning applications (i.e., lane cruising, lane change, etc.) and compatible with upstream and downstream ADS modules.

In the remainder of this paper, the multi-lane platooning algorithm is introduced in the methodology section. The performance of the proposed multi-lane platooning algorithm is then evaluated by using SUMO [49] and CARLA [21] simulators. The results are analyzed and further discussed. Finally, conclusions are provided, along with future research directions.

Table 3.1: Differences between platooning and cooperative adaptive cruise control.

Category	Platooning	Cooperative Adaptive Cruise Control
Control hierarchy	Hierarchical control with special responsibilities for platoon leader.	Decentralized control with no special responsibilities for the string leader.
Membership	Coordinated platoon/group membership.	Ad hoc string membership and vehicle behave independently.
Spatial scope	Operates in single or multiple lanes for a platoon lane change.	Operates in a single lane with small following gaps.

3.2 Methodology

This study develops an SOTA ADS platform compatible multi-lane platooning algorithm that allows C-ADS-equipped vehicles to seek platooning opportunities in both the current lane and adjacent lanes. An SOTA ADS platform refers to a software system implemented in a highly automated C-ADS-equipped vehicle which provides and supports basic highly automated functions such as lane following and pure pursuit (i.e., pursuing a set target). It consists of three main levels (i.e., modules): perception level, planning level, and control level. The SOTA ADS platform also supports additional applications to expand its capabilities, such as emergency stop, work zone identification and avoidance, and emergency pull-over. The proposed multi-lane platooning algorithm is developed as one of the applications to perform in parallel with existing basic applications to provide the host C-ADS vehicle with multi-lane cooperative platooning capability.

The multi-lane platooning algorithm consists of two steps: mission planning and motion planning. The mission planning regulates the strategic plans for the host vehicle based on an FSM for organized behaviors. Notably, a genetic fuzzy system (GFS) intelligent controller is trained as a special treatment to handle the complex on-ramp merging scenario. Motion planning is the immediate succeeding step of mission planning, which generates a series of waypoints based on the appropriate strategic plan. We use a basic PID controller in this study to track the planned trajectory as the lower-level control is not the focus of this paper, but any other trajectory tracking method can be applied.

In this section, the two-step framework is introduced first. Then, the platooning behavior is presented. Specifically, a deterministic FSM was developed to guide the platooning operation in the Formation, Operation, and Dissolve superstates and regulate C-ADS-equipped vehicle behavior. Notably, the GFS is also included in this section as a secondary intelligent controller that operates in parallel to the FSM to handle complex scenarios such as the cooperative merge, which the FSM cannot handle well. Lastly, the trajectory generation algorithm is described as generating trajectories for C-ADS-equipped vehicles to complete specific types of behavior planned by the platooning behavior protocol. The proposed multi-lane platooning algorithm distinguishes itself from the previous ones because:

- The multi-lane platooning algorithm abide by the protocols of the modern ADS software stacks, interacting with existing perception and control modules and operating in parallel with other ADS applications to fully deploy ADS functions on C-ADS-equipped vehicles.

- The multi-lane platooning algorithm covers decision-making at different levels. i.e., strategic mission level and tactical motion level.

- The algorithm handles platooning under complex driving scenarios on multi-lane highways by using a combination of rule-based and learning-based methods.

- The algorithm uses trajectory generation and sharing for completing various behavior types such that planned trajectories of other relevant C-ADS-equipped vehicles can be fully considered (i.e., intent sharing, of predictive nature).

3.2.1 Algorithm Framework

As one of the planning applications, the proposed multi-lane platooning algorithm operates in parallel with all other ADS planning applications, e.g., ACC, inline cruising, intersection, and work zone. Specifically, each application follows the two-step framework (i.e., strategic mission and tactical motion planning) in which the mission planning leverages the perception result to find the optimum strategic plan while the motion planning generates a corresponding trajectory as a list of waypoints. The arbitration process of electing the proper planning application is one of the main functionalities of the SOTA ADS platforms. We will use the terms "mission planning" and "motion planning" for simplicity in Figure.3.1 and in the remainder of the paper.

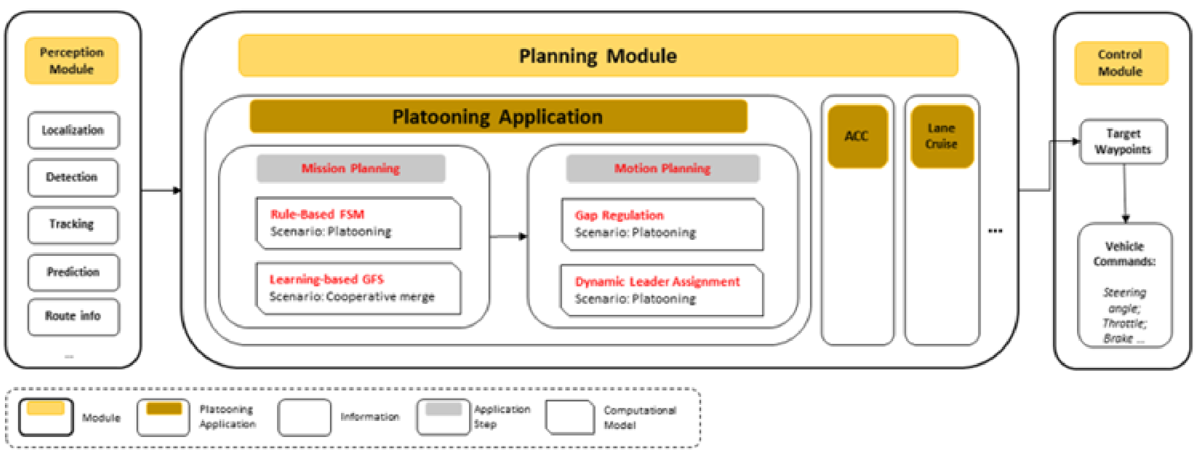


Figure 3.1: Logic and data flow of the platooning algorithm.

Figure.3.1 shows the logic flow of the multi-lane platooning algorithm in the two-step framework. The first step is mission planning, whereby one of the pre-defined semantic mission plans is selected based on the current scenario. The mission plan corresponds to the identified scenario and recommends a proper maneuver based on the deterministic FSM as high-level semantic information (e.g., lane cruise, lane change, on-ramp merge, etc.)

representing the ADS strategy of the next planning horizon. The maneuver plan is categorized into multiple pre-defined plan types, whereby each type contains essential information (e.g., start/target location, start/target speed, and start/target lane, etc.) for the motion planning step. Notably, the principles of the deterministic FSM involved in the mission planning step will be introduced in the next session. Secondly, the motion planning step parses the information from the maneuver plan, generating a detailed trajectory connecting the current position to the desired position following the target speed. The trajectory plan is valid through the current planning window, but it will be updated on a higher frequency to prevent the false maneuvers caused by obsolete information. The trajectory is in the form of a series of connected waypoints, whereby each waypoint consists of the target position, target speed for this specific location, and target angle for this specific location. Such data structure allows the control module to follow each waypoint with the desired speed and angle by manipulating the steering, throttle, and brake. The trajectory generation algorithm, which is presented in detail in the following sections, manages the behavior of the host vehicle by taking as input the desired ADS maneuvers and the desired intra-platoon gap as input, thus providing an executable trajectory plan for the downstream control module.

Additional effort was conducted in terms of algorithm deployment. ADS operation involves transmitting V2V data and processing related modules' results. To this end, we implemented a dedicated messaging system to subscribe to the existing module's result and publish necessary information (i.e., behavioral decisions and planned trajectories) generated by the proposed algorithm. To further improve communication efficiency, we regulated the message as status messages for all platoon members versus request-response messages which are dedicated to related C-ADS-equipped vehicles. This process significantly simplifies the communication overhead. On the other hand, interaction with downstream modules requires accurate and efficient calculation. Most existing vehicle stringing (such as CACC) algorithms focus purely on traffic performance, whereas computational accuracy and efficiency are usually overlooked. We addressed this research gap by utilizing the rule-based FSM strategic planner and the frenet [17] coordination system for behavioral organizations. The rule-based FSM operates with linear complexity that allows efficient strategic decision-making. The frenet [17] coordination system uses the down-track and cross-track distance to express relative position in regard to the host vehicle. The system is constantly rotating as an observer moves along the cure. Hence it is always non-inertial (i.e., a frame system that undergoes acceleration with respect to an inertial frame system) and provides better accuracy than cartesian coordinate system, especially when calculating relative positions.

3.2.2 Behavior Protocol

A deterministic FSM is designed to manage the platooning process with three superstates, as shown in Figure.3.2, such that the platooning behavior is organized. Each of the three superstates—Formation, Operation, and Dissolve and each of these superstates is composed of several states with their internal logic to guide the platooning process.

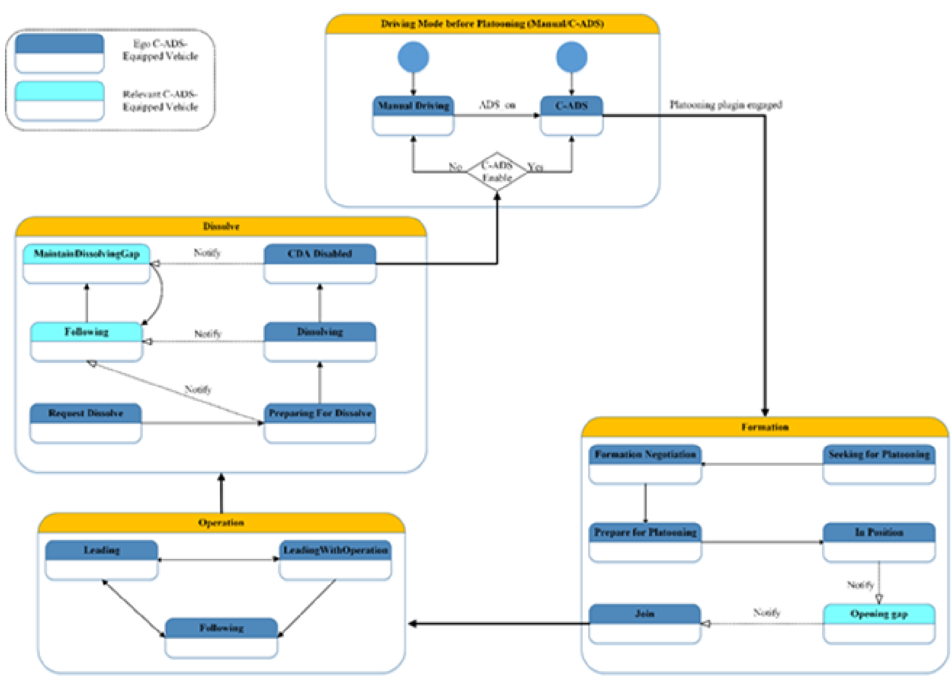


Figure 3.2: The Finite State Machine for platooning.

A C-ADS-equipped vehicle can be either controlled by a human driver (during human operations) or ADS applications (during automated driving). If the platooning plugin of the C-ADS-equipped vehicle is enabled, it will engage in the Formation superstate and seek platooning opportunities by negotiating with surrounding platoon leaders. After joining a platoon, the ego C-ADS-equipped vehicle will switch to the Operation superstate to regulate its speed and keep the desired time gap and/or headway. Once a platoon member intends to leave the platoon, it will request to leave and switch to the Dissolve superstate after getting permission from the platoon leader. A C-ADS-equipped vehicle can keep seeking platooning opportunities after leaving the previous platoon, or it can disable the platooning plugin and switch to manual driving.

In particular, the FSM for the multi-lane platooning algorithm is structured into three main superstates: Formation, Operation, and Dissolve, each containing specific states to manage different phases of the platooning process.

- **Formation Superstate**: This phase manages the process of forming a platoon, consisting of six states: "Seeking for Platooning," "Formation Negotiation," "Preparing for Platooning," "In Position," "Opening Gap," and "Join." During the "Seeking for Platooning" state, a single C-ADS-equipped vehicle communicates its intent to nearby platoon leaders, initiating the formation process. The vehicle will continue transitioning through the subsequent states based on specific events or conditions until it successfully joins the platoon.

- **Operation Superstate**: Once a vehicle joins a platoon, it transitions to the "Operation" phase, which includes three states: "Leading," "Leading with Operation," and "Following." The first vehicle in the platoon assumes the "Leading" state, where it coordinates maneuvers among members and manages interactions with external vehicles using a hierarchical control approach. The platoon leader is solely responsible for the "Leading" state, while other members adopt the "Following" role, maintaining formation and responding to the leader's commands.

- **Dissolve Superstate**: This phase handles the process of disbanding the platoon, consisting of five states: "Request Dissolve," "Preparing for Dissolve," "Dissolving," "Maintain Dissolving Gap," and "CDA Disabled." A vehicle intending to leave the platoon, including the leader, enters the "Dissolve" state and must first send a request to the platoon leader. Upon approval, the vehicle progresses through the remaining states to safely separate from the platoon, ensuring that proper gaps are maintained between vehicles during the dissolution process.

Each superstate is designed to manage a specific aspect of the platooning process, from formation and operation to safe dissolution, allowing for smooth transitions between different phases of platooning as vehicles enter or exit the formation.

3.2.3 Intelligent GFS Controller

Due to the excessive number of traffic merging scenarios, with and without platoons, conventional deterministic rule-based systems may suffer from two perspectives: Rule specification can become extremely difficult and the number of rules will explode. Therefore, a GFS is adopted to handle the complex cooperative merge scenarios. Generally, in a GFS, a Genetic Algorithm (GA) trains a set of parameters of a fuzzy logic system, which includes membership functions of all inputs, outputs, and the rules used to define the input–output relationship [78]. In the proposed algorithm, the inputs information of GFS are the position, speed, and target lane of the surrounding six vehicles (front and rear vehicles from the same lane, left lane, and right lane), whereas the outputs are the desired position, desired speed, and target platoon index for the host vehicle. In V2V on-ramp merging scenarios, the GFS performs as an off-line trained, intelligent controller alongside the existing rule-based FSM. At the same time, the entire system relies mainly on the rule-based controller (i.e., FSM) during all other situations due to its computational efficiency and robustness (i.e., no training is needed and applicable regardless of network geometry). Two cases are considered with the emphasis on maintaining robust cooperation among mainline and merge lane C-ADS vehicles, i.e., merging with and without pre-formed mainline platoons.

Vehicle-Vehicle Merging

As part of the proposed multi-lane platooning algorithm, each C-ADS-equipped vehicle in the network is equipped with the same intelligent GFS controller which is capable of making lane change decisions and controlling its speed, acceleration, and lane change behavior in a decentralized manner. That is, each C-ADS-equipped vehicle only uses local information available from its nearby vehicles (obtained via sensor or V2V communications) to make decisions. This decentralized behavior is trained without the existence of a platoon and serves as the baseline controller to enable individual ADS behaviors, which lays the foundation for vehicle-platoon cooperation. The cooperative single-vehicle ADS GFS is referred to as the GFS-M model, where M stands for "merge" in later sections. Note that during a cooperative merge operation, both the merging vehicle and the mainline gap vehicle are controlled by the GFS model due to the vehicle-vehicle cooperative nature of the algorithm.

The GFS controller uses 21 inputs to manage lane change and acceleration. These inputs include the distance, speed, and signals (blinkers and brake lights) of the closest vehicle ahead and behind the C-ADS-equipped vehicle moving on the left, current and right lanes of the ego C-ADS-equipped vehicle. The inputs to the GFS also include ego vehicle positions (i.e., x, y coordinates) and speed. The recommended acceleration and lane change are the outputs from the GFS. The acceleration is a continuous variable, whereas the lane change is a discrete variable with three classes: (1) move right, (2) stay on the lane, and (3) move left. In terms of training, since the strategies that cause inefficient merge will reduce the average speed, we formalize a fitness function that maximizes the average speed of all vehicles in the network with the constraints of a smooth acceleration and safety distance. Each training episode has a duration of 5 minutes, with actions that control speed and lane change only taken during intervals of 0.5 seconds. This interval is used only during training to keep the training time reasonable and can be reduced during testing. As a result, the GFS was trained to cooperate with other single ADS vehicles with the same model on board. The detailed training method is introduced in the following section.

Training process

The schematic of the training process is shown in Figure.3.3, which correspond to the "Learning and Reasoning Operations" within the parallel framework. We adopt the Fuzzy Bolt framework [78] that trains a single GFS model to reduce the search space effectively. A GA is initialized with random individuals that define the population to train the proposed GFS. The GFS is set up for training with ten triangular membership functions for each input. In terms of output, five triangular membership functions are used to calculate the continuous acceleration output. Three membership functions are adopted to define the three classes of the lane change maneuvers, i.e., lane change to left, lane change to the right, and stay in the current lane. Each GA individual is a vector consisting of the parameters related to the GFS controller, which includes the boundaries of the membership functions and the rules in the rule-base. Thus, each individual in the GA can be converted to a GFS controller, which can then be used to simulate an episode (e.g., in SUMO). An episodic fitness can be evaluated for each such individual in the GA. The fitness would represent the system's ability to achieve the objective of smooth merging. GA individuals with higher fitness values are more likely to be selected for crossover and mutation, thus getting selected for the next generation. In contrast, individuals with lower fitness scores have a higher chance of elimination from the population. This process of evolution continues for a pre-defined number of generations or until convergence, where the fitness score has little to no improvement (i.e., lower than 0.01) for 20 iterations. Each episode is a scenario simulated for a timeframe of 5 minutes with actions taken at intervals of 0.5 seconds. The episode starts at t = 120 seconds and ends at t = 420 seconds, where the initial 120 seconds warm-up period populates the traffic density before training. The GFSs evolve with each generation during training to improve the fitness function. At the end of each generation, the best model of the generation is applied to a

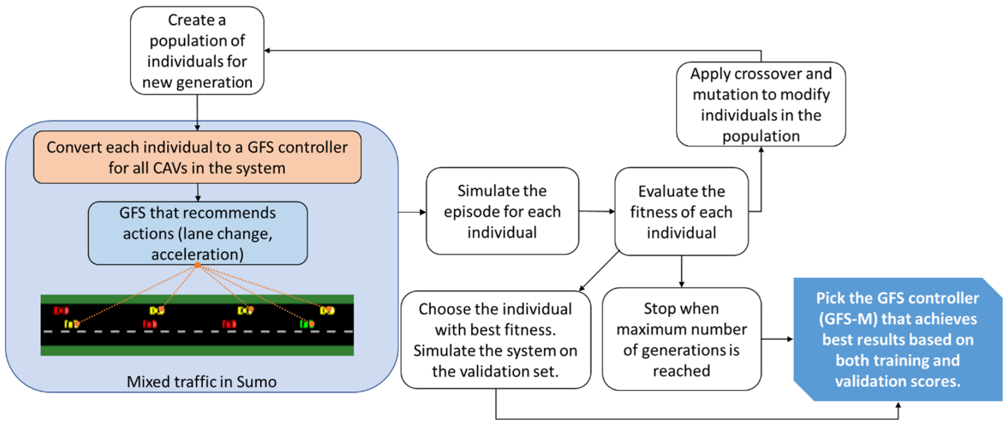


Figure 3.3: Training Process of GFS.

validation episode to check its effectiveness on an unseen scenario.

The training maximizes the fitness function, which captures the average speed of all vehicles across the 5-minute episode. For efficiency, a particular stopping criterion is designed to stop episodes that result in undesired cases, such as stopped vehicles in the network. This function allows early truncation of undesirable episodes, avoiding meaningless episodes in an early stage, thus improving the training time. The GFS controllers that result in such situations are penalized heavily. Thus, the fitness function that is to be maximized by GA algorithm is defined as follows.

$$F = \begin{cases} 0.001\, t_{end} & \text{if } t_{end} < 420\, s \\ \text{mean}(v) & \text{if } t_{end} = 420\, s \end{cases} \tag{3.1}$$

The top portion of equation 3.1 relates to early stoppage scenarios in an episode before the final time t = 420 seconds. These individuals in GA are heavily penalized. Ideally, the simulations should run for the entire time till t = 420 seconds, in which case, the average speed of all the vehicles should be maximized, as shown in the bottom portion of equation 3.1.

Vehicle-Platoon Merge

The GFS also considers cases where C-ADS-equipped vehicles merge onto a mainline containing vehicle platoons. The platoons follow the deterministic FSM mentioned in the previous section. Similar to the single-vehicle case, the GFS for vehicle-platoon cooperation is trained separately in combination with the FSM protocols. Because the platoon establishes hierarchical control architecture, the leaders are trained to allow cooperative merge when a merging ADS vehicle occurs. The behavior of the follower vehicles in a platoon is always coordinated by the platoon leader, as noted in the hierarchical platoon protocol. Other C-ADS-equipped vehicles that engage with platoon leaders to join platoons will also follow platoon protocol. The individual C-ADS-equipped vehicles that are not part of a platoon or have not engaged with any platoons are controlled by the previously trained GFS-M model.

The vehicle-platoon scenario poses some additional challenges to the cooperative merge behavior. The merging vehicle has to work cooperatively with the platoon leader to merge into the platoon in the mainline for different situations, as shown in Figure.3.4. The platoon leader needs to make decisions on speeding up, slowing down, or allowing the merging vehicle to perform a cut-in join at a position as defined in the previous section, platooning protocol.

Similarly, the FuzzyBolt framework [78] is used to train the GFS model that controls the platoon leaders. The GFS model that controls platoon leaders is named GFS-PL. The training schematic is shown in Figure.3.5. The GFS-M model from the previous training controls the individual vehicles that have not engaged with any platoons. We used the same fitness function defined in the previous sections because the objective of a cooperative merge with platoons includes maintaining the smooth flow of traffic.

In this case, only the GFS-PL module is trained. This GFS-PL module contains two GFS sub-models: (1) the GFS-PL-speed submodel for controlling the speed of the platoon leader (GFS-PL-speed) and (2) the GFS-PL-score submodel that predicts a performance score (i. e., considering comfort, safety, and network congestion) for all available merging positions. The two submodels are trained simultaneously. Once the merge position is determined,

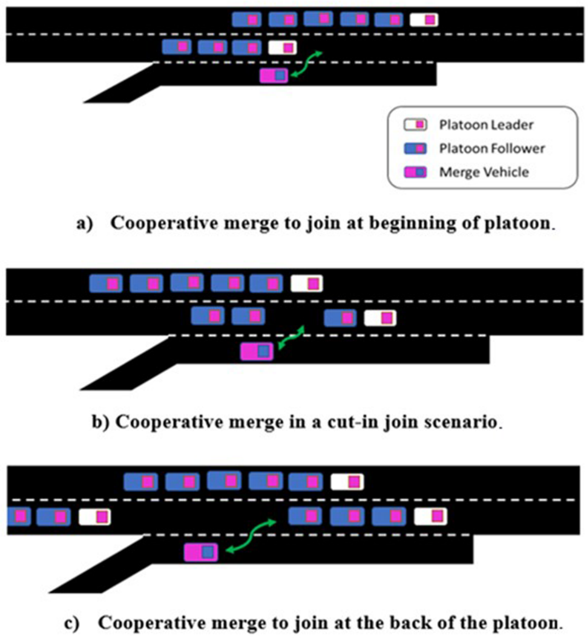


Figure 3.4: Cooperative Merge to join at the back of the platoon.

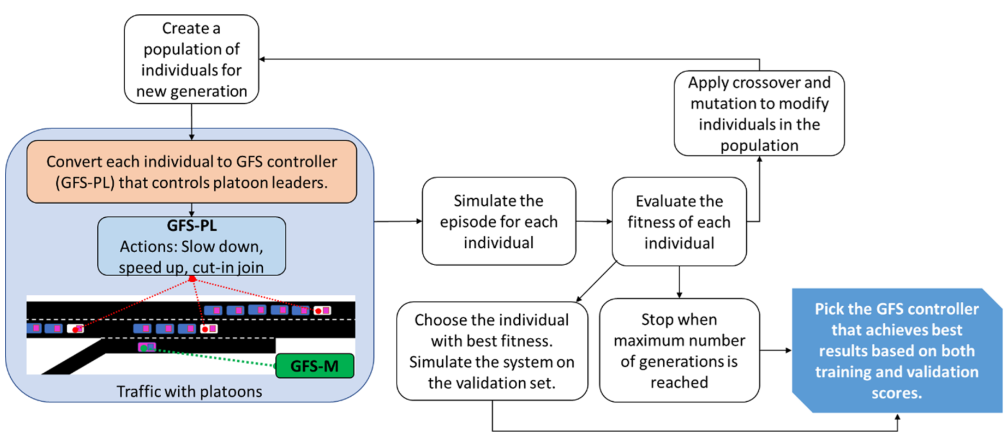


Figure 3.5: Training Process of GFS-PL.

the rear vehicle relative to the merge switches to OpeningGap, and the GFS-M starts to control its speed. Simultaneously, the speed of the merging vehicle is controlled to align it horizontally with the merge position in the platoon. Once the merging vehicle is between the two relative platoon members corresponding to the merge position, the GFS-M controls the speed to allow the host vehicle to merge into the platoon. For instances where an individual C-ADS-equipped vehicle on the mainline encounters a merging vehicle, the GFS-PL will control the mainline vehicle to decide the behavior of the vehicles to allow them to form a two-vehicle platoon.

Single-Vehicle Trajectory Generation

The multi-lane platooning algorithm generates behaviors through the deterministic FSM (i.e., mission planning step), and then trajectories are generated to complete each of the behaviors (i.e., motion planning step). Trajectory generation is a core module connecting high-level decision-making on behavior and lower-level control that outputs throttle and brake levels. Trajectories are usually regenerated based on a real-time dynamic world model and need to be smooth to avoid system disutility, such as excessive jerks to ensure driving comfort. Many types of trajectories (e.g., polynomials, splines, etc.) have been used in real SOTA ADS platforms. In this study, we adopt a widely used method, i.e., the cubic spline interpolation, to generate the trajectories for different maneuvers. Other methods can be used to flexibly replace the spline-based approach.

In a general ADS implementation, when a global route is generated by the global route planner using the A-star algorithm [63] a series of waypoints can be generated to lead the vehicle to the destination, and a pre-specified distance separates each nearby pair of points (e.g., 6 m), as shown in Figure 3.6 where the generated waypoints along the route are highlighted as square blue dots. However, a fixed separation distance between two consecutive waypoints leads to unstable controls due to the various radius of the road curvature. To this end, according to the ego vehicle's location, the proposed algorithm generates a smooth

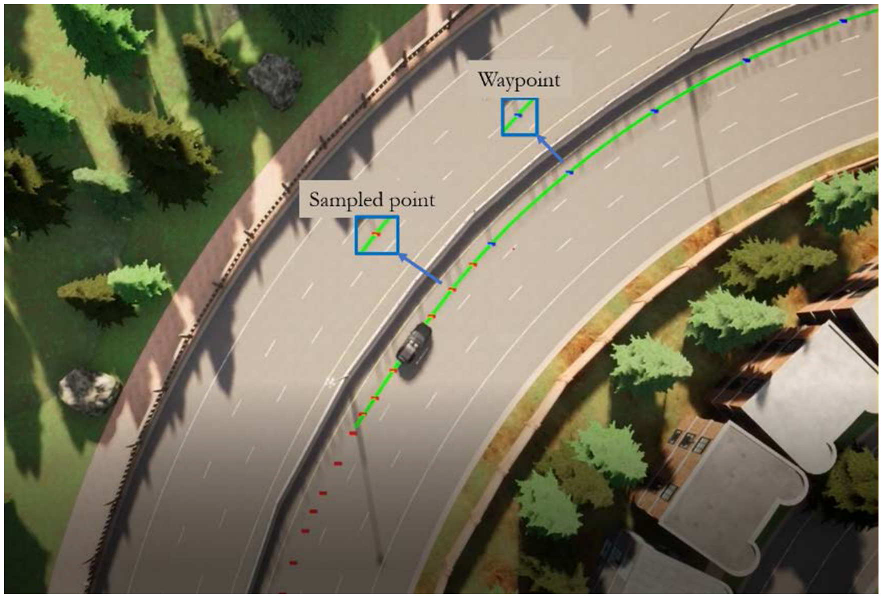


Figure 3.6: Illustration of splines (square red dot), route history (curved green line), waypoints, and sampled points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

curve among the following N waypoints (e.g., N= 8) by using cubic spline interpolation. As shown in Figure 3.6, the highlighted green line represent the generated smooth curve which forms an interim continuous trajectory in-between waypoints, resulting in a smooth trajectory that minimizes the unstable movements and, in contrast to other forms such as polynomials, connects all of the waypoints within the calculation range (i.e., the pre-defined N points). Note that the vehicle will travel along the green line (i.e., cubic spline) as long as the current cubic spline remains valid.

Once the interim continuous cubic spline is generated, it is discretized as a series of candidate sample points with a specific spline resolution (i.e., spl_{res}). In particular, this study assigns the resolution to maintain a 0.1 s time gap between candidate sample points. The ego vehicle then samples discrete trajectory points from the candidate sample points based on the current location, speed, and target speed (i.e., target speed at the next waypoint). The related derivation is shown in equation 3.2 – 3.7, and red dots represent the sampled trajectory points in Figure 3.6. Specifically, equation 3.2 calculates the spline resolution based on the current and target speed; equation 3.3 finds the number of candidate points that satisfies the target spline resolution; equation 3.4 finds the index of the candidate sample points while equation 3.5 finds the x-coordinates of each candidate sample point; equation 3.6 finds the y coordinate of the candidate sampled points; and equation 3.7 defines the cubic spline where A, B, C, and D, are cubic spline parameters to be determined.

$$spl_{res} = avg(spd, spd_{tar}) \times t_{res} \tag{3.2}$$

$$n_{spl} = \left\lceil \frac{dis}{spl_{res}} \right\rceil \tag{3.3}$$

$$idx_i = \left\lfloor \frac{i \times spl_{res}}{dis_{res}} \right\rfloor, \quad i = 1, 2, \ldots, n_{spl} \tag{3.4}$$

$$x_i = X(idx_i) \tag{3.5}$$

$$y_i = Y(idx_i) \tag{3.6}$$

$$Y = AX^3 + BX^2 + CX + D \tag{3.7}$$

Where:

spl_{res} = sample resolution (distance).

spd = current speed of the ego vehicle (distance/time).

spd_{tar} = target speed at the next waypoint (distance/time).

$avg(spd, spd_{tar})$ = average of current speed and target speed at the next waypoint (distance/time).

t_{res} = time resolution (0.1 s in this study).

n_{spl} = number of candidate sample points along the continuous cubic spline.

dis = distance to the next waypoint or to a specified location (which is usually represented by a waypoint).

i = the sequence of the sampled point.

idx_i = index of the sampled point i.

dis_{res} = the distance between two consecutive candidate sample points, i.e., candidate sample point resolution (0.1 m in this study).

X = the x-coordinates of the interpolation points between waypoints, and each of them has a unit distance of 0.1 meters.

Y = y-coordinates of candidate sample points, and A, B, C, and D are determined by cubic spline interpolation method.

x_i = x-coordinate of the sampled point i.

y_i = y-coordinate of the sampled point i.

Specifically, the discretized trajectory only covers the two nearest waypoints (i.e., the current waypoint and the consecutive waypoint). Once the ego vehicle reaches the later waypoint (i.e., distance smaller than 0.5 meters), its interim planned continuous trajectory are updated to cover the upcoming N waypoints (e.g., N= 8) on the route, and the discretized trajectory between the next two consecutive waypoints is re-sampled and updated by repeating the same process. This process guarantees that the vehicle's yaw angle is aligned with the upcoming route, as the discretized trajectory is always a segment of a continuous spline. Note that the number of waypoints used for cubic spline calculation is configurable, while the current setting for the study (i.e., N = 8) follows the default value in CARLA, the simulator used in the case study.

Platoon-Member Trajectory Regulation

The above-mentioned trajectory generation process is the general process that applies to any isolated ADS-equipped vehicles as well as platoon leaders when the leaders are in lane-following mode. The speed and acceleration of the platoon members (i.e., all followers) usually need to be regulated against the immediately preceding vehicles (i.e., gap regulation) and/or the platoon leader (i.e., headway regulation). The gap and regulation processes are usually combined to ensure platooning performance, such as stability. During a gap regulation, a platoon follower needs to consider its desired time gap and the preceding vehicle's position during the trajectory generation, as shown in equation 3.8.

$$pos_j^g(t) = \frac{pos_{j-1}(t) - L_{j-1} + pos_j(t-dt) \times \frac{gap_d}{dt}}{1 + \frac{gap_d}{dt}} \tag{3.8}$$

Where:

$posi(t)$ = the position of the vehicle j at time t.

$pos_{j-1}(t)$ = the position of the preceding vehicle $j-1$ at time t.

L_{j-1} = the length of preceding vehicle $j-1$.

dt = time resolution, i.e., simulation step (0.1s in this study).

$pos_j(t-dt)$ = the position of the vehicle j at last time step $t-dt$.

gap_d = desired time gap of the vehicle j.

During a headway regulation, we can also replace the desired time gap with the time headway by following equation 3.9, but the target vehicle is the platoon leader.

$$pos_j^h(t) = \frac{pos_j(t) + pos_j(t-dt) \times \frac{h_d}{dt}}{1 + \frac{h_d}{dt}} \tag{3.9}$$

Where:

h_d = desired time headway of the ego vehicle j to the platoon leader.

In this study, the ego follower vehicle is regulated by a leader-predecessor scheme by following equation 3.10.

$$pos_j(t) = \mu \cdot pos_j^g(t) + \lambda \cdot pos_j^h(t) \tag{3.10}$$

$$\mu + \lambda = 1, \quad \mu > 0, \quad \lambda > 0 \tag{3.11}$$

Where:

$posj(t)$: the planned position of the Leader-Predecessor regulation.

μ : the gain on the gap regulation value.

λ : the gain on the headway regulation value.

Note that equation 3.10 will be applied to each member/follower of the platoon, starting from the second vehicle, the predecessor of which is the platoon leader. Since the leader's trajectory has been planned through the trajectory generation process by considering the surrounding traffic environment, the trajectory of each of the followers can be calculated subsequently. This process distinguishes this regulation method from other reactive ones (i.e., it is a trajectory-based regulation of a predictive nature) but also seamlessly integrates well with SOTA ADS platforms that require planned trajectories.

3.3 Results

The simulation evaluation in the study requires perspectives from both vehicle/automated driving and traffic management. Regular traffic or automated driving simulators cannot meet this requirement. Therefore, a novel co-simulation tool, OpenCDA, which the authors developed for CDA research, is adopted [103]. OpenCDA is a simulation tool integrated with a prototype cooperative driving automation (CDA; see SAE J3216) [67] pipeline as well as regular automated driving components (e.g., perception, localization, planning, control). The tool integrates automated driving simulation CARLA [21], traffic simulation SUMO [49], and Co-simulation (CARLA + SUMO). Although SUMO can well test the behavior protocol by using simple ADS behavior (i.e., car-following and lane change behavior models), CARLA is needed to test detailed ADS and C-ADS vehicle behavior that is controlled by ADS and C-ADS software stack. One critical aspect of this study is the trajectory generation and control components of C-ADS vehicles and the possibility of using vehicle dynamics models to understand CADS vehicle behavior as controlled by the software pipeline. Therefore, the simulation evaluation is formularized as follows. (See OpenCDA [103] for a detailed discussion of simulator selection and applications):

- SUMO simulation evaluation: SUMO was used to tune the proposed multi-lane platooning algorithm, observe the platooning behaviors in various MPRs, and train the GFS controller in multiple use cases (i.e., vehicle-vehicle merging and vehicle-platoon merging).

- The focus here is to fully consider platooning effects on large-scale traffic. Therefore, a traffic-levels simulator such as SUMO is necessary. The simulator ensures that the algorithmic parameter, particularly those parameters at the mission planning level, is optimized or trained to improve traffic system performance, which exceeds the scope of the conventional platooning studies that only focus on platooned vehicle performance.

- CARLA simulation evaluation: CARLA was used to test detailed vehicle behavior controlled by the proposed platooning protocol becasue CARLA can simulate the full pipeline of an SOTA ADS software platform and some level of vehicle dynamics.

- CARLA + SUMO co-simulation: The co-simulation was adopted to test the algorithm performance comprehensively in a complex multi-lane highway scenario with large-scale background traffic. Specifically, SUMO was adopted to spawn and control the background traffic with realistic behaviors (i.e., NGSIM [73] trajectory or calibrated IDM [38]), while the C-ADS vehicles run the proposed multi-lane platooning algorithm in the CARLA simulator. The two simulators operate in synchronous mode, so vehicles in both simulators can achieve two-way interaction in each step. In this way, we can observe the detailed platooning and merging vehicle behavior (via CARLA) and create realistic traffic scenarios to test the platooning operations (via SUMO).

3.3.1 Platooning Simulation in SUMO

Though the proposed multi-lane platooning algorithm is based on organized behavior, several parameters for both mission planning and motion planning were introduced throughout the pipeline to adjust the algorithmic performance. In addition, the GFS controller needs to be trained before deployment. In this section, we tune and evaluate the multi-lane platooning algorithm in SUMO, demonstrating its effectiveness while the GFS controller is trained in various scenarios in SUMO. The section is divided into four sections: parameter tuning with synthetic trajectory, single-lane platooning simulation in SUMO, capacity analysis at different MPRs, and GFS training.

Parameter Tuning

Before simulating and evaluating the performance of the multi-lane platooning algorithm, several parameters need to be determined. The inter-platoon and intra-platoon were determined based on sensitivity analyses in previous studies [37, 82]. Therefore, the other three parameters, including negotiation distance, time-gap for join, and time-gap for Dissolve are tuned by the GA. Based on the nature of those three parameters, value encoding is used where each gene of the the GA algorithm is represented by a real number with two decimal places. The population size is 50, and the generation is 100 for seeking the globally optimal combination of the three parameters. To prevent any premature convergence, the two-point crossover is applied with a crossover rate of 0.9, and the mutation rate is set to 0.1. The Roulette Wheel selection is utilized to select potential optimal solutions in the parental generation. The total delay is utilized as the fitness value. The searching range of each parameter is as follows:

- Negotiation distance: 0 to 60 meters
- Time-gap for Join: 0.6 to 1.5 seconds
- Time-gap for Dissolve: 0.6 to 1.5 seconds

Convergence of the GA Tunning Process (Figure 3.7) shows the convergence of the GA tuning process. The solid blue line indicates the average fitness of all chromosomes, whereas the solid orange line represents the best fitness of each generation. It can be found that the tuning process is terminated after 72 generations as the fitness score converges. The optimized parameters are listed below:

- Negotiation distance: 53.20 meters

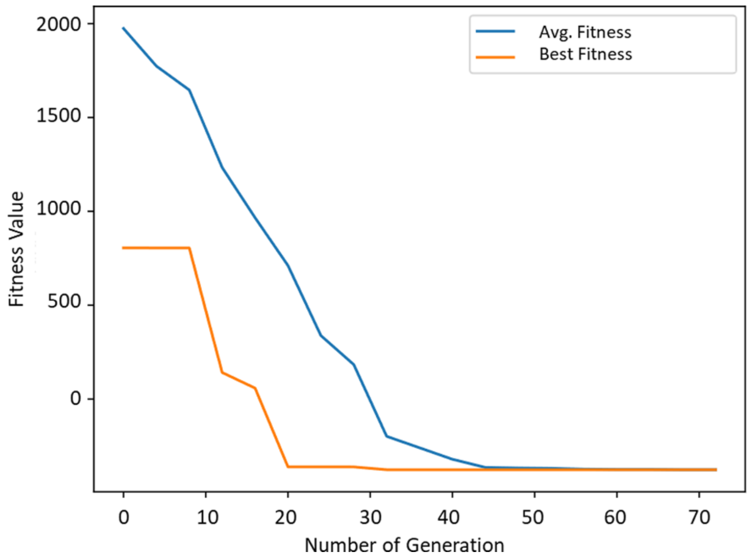


Figure 3.7: Convergence of the GA Tuning Process.

- Time-gap for Join: 1.14 seconds
- Time-gap for Dissolve: 0.96 seconds

Since the simulation resolution in this study is not less than 0.1 s, the value of the two time-gaps are rounded to one decimal, i.e., the time gap for Join is set to 1.1 s, and the time gap for Dissovle is set to 1 s. The control gains of the gap regulation and headway regulation in equation 3.10 are pre-specified in this study and tuned to find a better combination of the μ and λ. Because the μ is defined as $\mu = 1 - \lambda$, the adjustment range of μ is set to 0.6 to 1 using increments of 0.1 to investigate the performance. Other values less than 0.6 will not be investigated based on the initial simulations and evaluations because the platoon followers cannot maintain the desired time gap during the entire trip.

Freeway Capacity at Different MPRs

A synthetic trajectory is generated to simulate stop-and-go traffic. The leading vehicle is an HDV (id: 0), and it follows the given trajectory. A platoon is regulated by the multi-lane platooning algorithm with different values of μ. When $\mu <= 0.8$, it is difficult for the platoon followers to maintain the desired intra-platoon time gap during the operation; when $\mu = 1.0$,

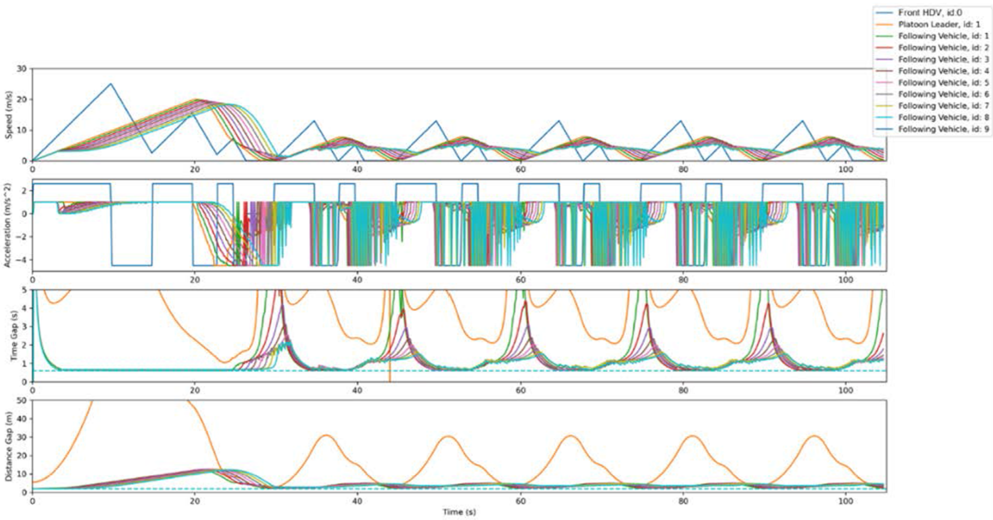


Figure 3.8: Convergence of the GA Tuning Process.

some platoon members cannot maintain the standstill with the stop-and-go traffic. Therefore, the $\mu = 0.9$ (as shown in Figure 3.8) is utilized in this study by considering the performances under different traffic environments.

The pipeline capacities (i.e., the capacity of basic freeway segments) of different C-ADS-equipped vehicle market penetration rates (MPRs) were investigated with the tuned parameters. The MPR varies from 0% to 100% with a 20% increment. The freeway segment is 2800-meters long with two lanes. The first 200 m at the beginning of the freeway only allows the same-lane formation to form stable platoons. The speed limit of this freeway segment is 113 kilometers/hour (70 mph), and the acceleration of C-ADS-equipped vehicles is limited to $1m/s^2$ for comfortable driving behavior. Based on previous studies, the sensor detection range is set to 120 meters.

Both CACC [45, 60] and the proposed multi-lane platooning algorithm are investigated, as shown in Table 2. Generally, the capacity increases with the growing MPR, from 1972 vehicles per hour per lane (vphpl) to 3296 vphpl, by implementing either CACC or the multi-lane platooning algorithm.

As shown in Table 3.2, at 20% and 40% MPRs, the pipeline capacity of CACC is close to (slightly higher than) the platooning. For one, with low MPR, the C-ADS-equipped vehicles are sparsely located in mixed traffic, leading to fewer cooperation opportunities and formation agreements. Therefore, the stop-and-go oscillation is hardly reduced with limited formed platoons under the HDV-dominant traffic flow. In addition, the main reason for the low reduction in oscillation is that the multi-lane platoon formation behavior can slightly impact the upstream traffic at low market penetrations rate due to the lack of cooperation between HDV and C-ADS-equipped vehicles. For example, suppose a single C-ADS-equipped vehicle reaches an agreement with the target platoon leader and implements a cut-in join. In this case, the rear platoon members will slow down to create a safe gap for the single ego C-ADS-equipped vehicle; therefore, the upstream traffic, especially the HDVs, will slow down. Once the single ego C-ADS-equipped vehicle completes the lane change, the rear platoon member will lead all upstream platoon members to accelerate to maintain the desired intra-platoon gap. Though there is no significant delay for C-ADS-equipped vehicles in this process, HDV traffic from upstream may be negatively impacted due to the platoon formation speed changes. Since the HDVs are predominant in the traffic stream at low MPRs, this impact is significant, and therefore the pipeline capacity is negatively affected to a small extent. Note that under the low MPRs (up to 40%), the capacity difference between CACC and the multi-lane platooning algorithm increases with the growing MPR. This increase occurs because more multi-lane formation opportunities will present with the growing MPR, and therefore more oscillations can be generated associated with the multi-lane formations.

Table 3.2: The capacity of CACC and the multi-lane platooning algorithm.

MPR (%)	**CACC (vphpl)**	**Platooning (vphpl)**	**Difference (%)**
0	1972	1972	0.0
20	2030	2063	-0.8
40	2256	2212	-2.0
60	2558	2644	2.2
80	2808	2896	3.1
100	3296	3296	0.0

Under moderate and high MPRs, i.e., when the C-ADS-equipped vehicles are predominant in the traffic stream, the multi-lane platooning algorithm can improve the pipeline capacity by up to 3.1%. As previously mentioned in this section, the C-ADS-equipped vehicle can be well controlled to recover soon from multi-lane join, and the impact will diminish with the increasing MPR. Also, the multi-lane formation provides the opportunity for the single C-ADS-equipped vehicle to utilize the reduced time gap and regulate its speed with the more stable regulation algorithm instead of using ACC [60]. It is worth noting that under 100% MPR, there are no lane changes with high traffic volume inputs because C-ADS-equipped vehicles can form platoons with others in the same lane. Therefore, the capacities of CACC and Platooning are the same under 100% MPR.

Note that even though the capacity enhancements of platooning over CACC are only limited, the multi-lane platooning algorithm can benefit platoon members and the traffic from other aspects. For example, when negotiating with the platoon leader, the single CADS-equipped vehicle will choose the platoon with the same or similar destination. This choice will limit unnecessary formations and dissolves on longer freeway segments with on-ramps and off-ramps, thus reducing traffic disturbances. Reducing this type of human-made disturbance can help to improve capacity and reduce potential safety risks [30]. Notably, the comparison listed in Table 3.2 is conducted only in conventional highway segments that are identical to the CACC experiment environments. In non-basic segments where lane changes are a necessity, the proposed multi-lane platooning algorithm is the superior solution because of its complex cooperation capability and trajectory-sharing (i.e., intent-sharing) nature.

GFS training for cooperative-merge

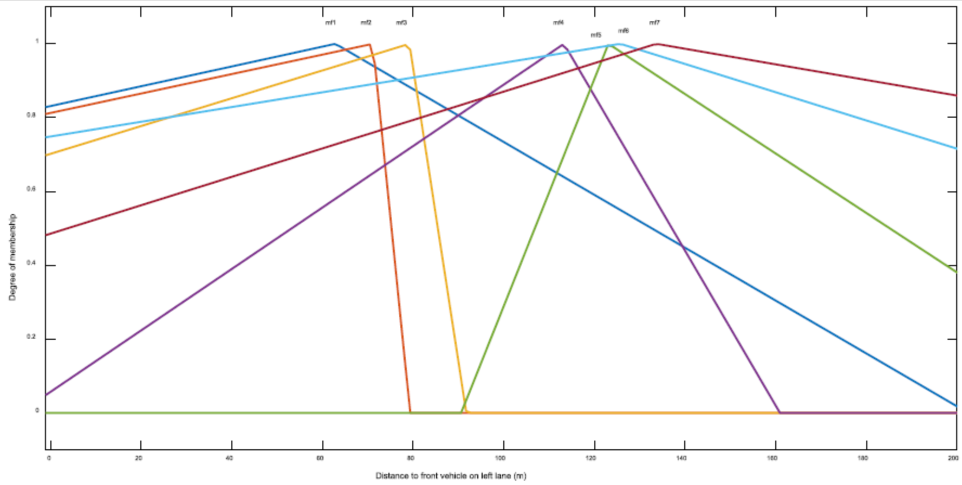
As aforementioned, a GFS controller was adopted to handle the on-ramp merge situation due to the excessive number of scenarios and significant speed difference between mainline and merge lane vehicles. As merge lane C-ADS vehicles approach the merging point, the cooperation may be established as vehicle-vehicle merging or vehicle-platoon merging. Accordingly, we formulate the training in two steps. A GFS-M vehicle-vehicle merging controller was trained during the first step, whereas the vehicle-platoon merging controller, in which the controller from the first step was used during the training, was trained in the second step.

The training involves two scenarios for both steps. To be exact, scenario one contains 2000 vphpl mainline volume and 800 vphpl merge ramp volume, reaching 4800 vph total volume combined for two lanes and one merge lane. In contrast, scenario two populates 1500 vphpl mainline volume, 1500 vphpl merge ramp volume, and the total volume adds up to 4500 vph. Considering the random arrival of vehicles and the usage of different random seeds, the simulation can realize many different combinations of traffic density. As a result, the trained GFS-M model handles traffic scenarios around medium to high densities, allowing efficient vehicle-vehicle cooperative merging that minimizes the total system delay.

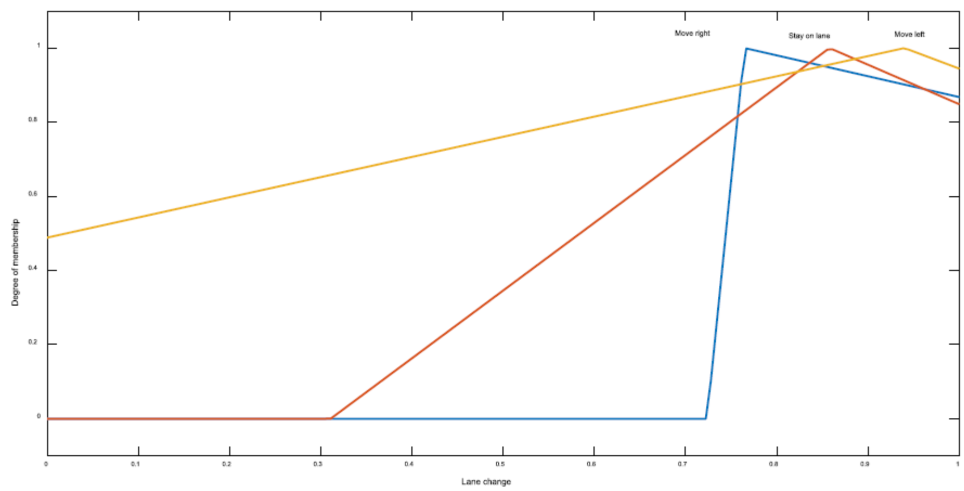
The membership functions of the trained GFS-M model for one of its inputs, distance to the front vehicle on the left lane, is shown in Figure 3.9(a). Note that only the membership functions that were picked in the trained model are shown in the figure. At the start of the training, each variable was assigned 10 random membership functions, while the trained model used only 7 membership functions for this input variable. Similarly, Figure 3.9(b) shows three membership functions that define the acceleration. Since acceleration is a continuous value, centroid defuzzification is used where all the rules are evaluated and aggregated using a weighted average to obtain the final acceleration value. Meanwhile, the discrete lane change output is obtained via mean-of-max defuzzification, from which an output class is obtained based on the output membership function defined for the most significant rule.

The trained GFS-M was evaluated on both scenarios. The improved throughput and delay under different MPRs are shown in Figure 3.10. As more trained C-ADS vehicles are added to the network, the flow of traffic becomes smoother, improving the throughput and reducing the average delay.

Next, the second step of training in which each C-ADS vehicle is simulated using the trained GFS-M model as a baseline was conducted. The GFS-PL model was trained to optimize the speed of the platoon leader and the merge position for vehicle-platoon cooperation. Figure 3.11 shows the validation result of the trained GFS-PL model in both scenarios, where the



(a) Membership function of GFS-M input after training



(b) Membership function of GFS-M model output after training

Figure 3.9: Input and Output Membership functions of the GFS-M model.

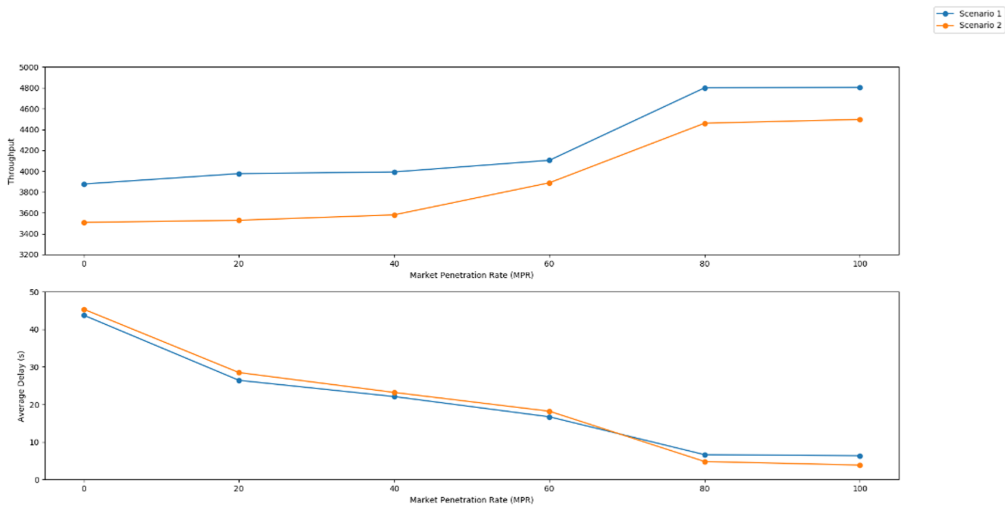


Figure 3.10: GFS-M validation results (throughput and average delay) for different MPRs.

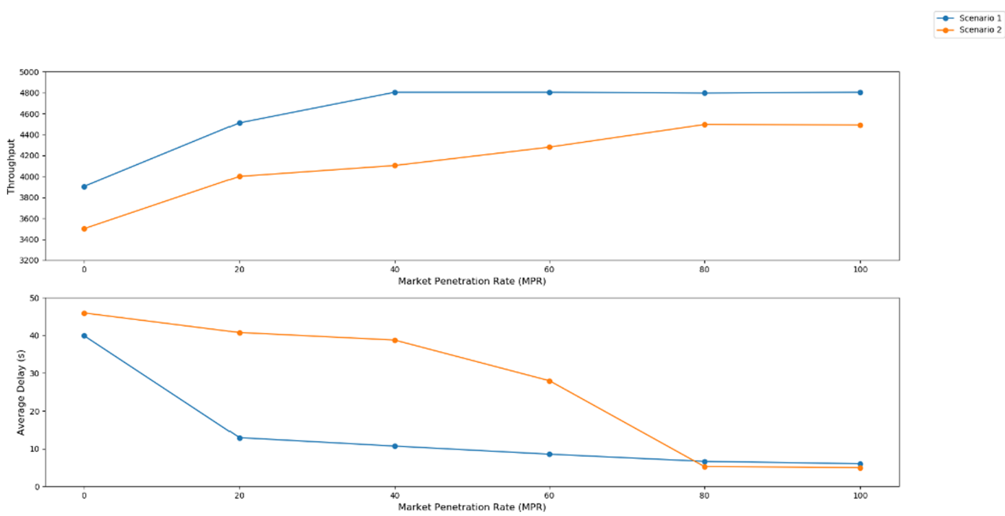


Figure 3.11: GFS-PL validation results (throughput and average delay) for different MPRs.

throughput increases and the average delay decreases with the increase in MPR. Overall, this trend is similar to the cooperative vehicle-vehicle merge (as shown in Figure 3.10). However, the throughput for scenario one reaches the maximum network capacity of 4800 vph much earlier than vehicle-vehicle merging, at an MPR of 40%, because the merge lane volume is low and the disruption to the main traffic can be handled more easily with vehicle-platoon cooperation.

In contrast, the merge density in scenario two was considerably higher (1500 vph). Therefore, for lower MPR, where cooperation by HDVs is not assured, like in the C-ADS vehicles, the throughput is lower, and the time delay is higher. The network throughput with MPR and reaches a maximum when the MPR reaches 80%. The increase in MPR of C-ADS vehicles will coordinate merge volume better with the mainline traffic and gradually increase the capacity.

3.3.2 Platooning simulation in CARLA

Due to its microscopic nature, SUMO cannot simulate vehicular dynamics, including acceleration, deceleration, and jerk. However, as an ADS planning application, it is essential for the proposed algorithm to integrate with downstream control modules. In particular, the proposed algorithm incorporates dynamic vehicle models, physics models, and a trajectory generation model to interact with the downstream control models by providing an executable trajectory for the next planning horizon. Therefore, to demonstrate such integration with downstream control modules, we implemented CARLA as the simulator to conduct vehicle-level simulation evaluations, as described in this subsection. The evaluation is divided into three sections: platooning with synthetic trajectory, platooning with NGSIM trajectory, and cooperative merging.

Platooning simulation in CARLA with synthetic trajectory

This scenario tests the platoon's stability, which is indicated by the degree of amplified oscillations when the leading vehicle changes speed dramatically. In detail, a five-vehicle platoon keeps driving in the same lane while the platoon leader follows a design speed profile to accelerate and decelerate frequently. This scenario was designed because we aimed to evaluate the algorithm's capability to maintain the desired intra-platoon time gap in CARLA. The target intra-platoon time gap was set as 0.6 s in the synthetic cycle testing. As one example in Figure 3.12 demonstrates, the platoon followers were able to keep the designed time gap of 0.6 s during the whole process, even with the leading vehicle dramatically increasing and decreasing speeds. When the platoon leader started to accelerate suddenly, the platoon

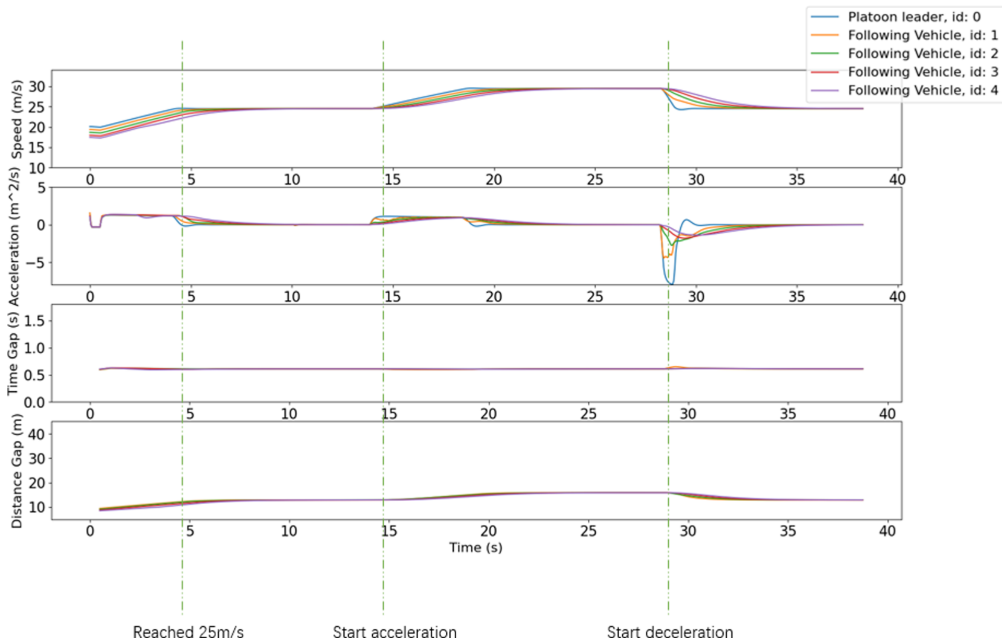


Figure 3.12: Synthetic trajectory results for simulation in CARLA.

members were able to follow it tightly without any speed-overshooting. When the platoon leader rapidly stepped on the brakes, the followers were able to smoothly decelerate at a comfortable rate and stay within constant time gaps between each other, which indicates the stability of the platooning.

Platooning simulation in CARLA with NGSIM trajectories

In the real trajectory testing, we select many challenging trajectories from NGSIM datasets to understand the platoon following behavior. In the examples in Figure 3.13, as vehicles in CARLA are launching from a standstill, it takes about 5 s until the leader is close enough to the HDV in front of it to start using its car following algorithm. Once the leader is engaged in following the HDV, despite the HDV's frequent speed changes, the platoon leader is able to follow it securely and smoothly. In particular, for both NGSIM trajectories, a 1.5 second time gap was successfully maintained throughout the platoon, with damped acceleration disturbances originating from the real-world trajectory. At the same time, the platoon members can keep a constant time gap of 0.6 s even when the front HDV rapidly accelerates or decelerates, indicating the robustness of the proposed algorithm in a real-world setting.

3.3.3 Cooperative merge in CARLA-SUMO co-simulation

As one of the ADS applications, the proposed algorithm provides seamless integration with upstream perception modules and the downstream control modules. The existence of HDV background traffic is crucial to demonstrate such integration. Specifically, the platoon establishes different behaviors when the detected HDV is in front, on the side, or in a different lane. During operation, the platoon leader is expected to maintain a steady gap between the preceding HDV while monitoring the HDVs on the side to avoid crashes. During merging, the C-ADS vehicles should detect mainline vehicle types and select the target platoon accordingly. These organized behaviors constitute crucial testing scenarios for integration validation with upstream perception modules, as they are not only part of the proposed multi-lane platooning algorithm and they depend heavily on the perception results.

CARLA provides vehicle dynamics simulation and roadway environment. However, it falls short when offering large-scale HDV background traffic with realistic (i.e., calibrated IDM [38]) driving profiles, which are essential to test the algorithm's behavior with the existence of the HDV traffic stream. SUMO, on the other hand, supports more realistic and larger-scale background traffic simulations with significantly less computational cost. Therefore, a CARLA-SUMO co-simulation was conducted with maximum C-ADS vehicle dynamics and realistic background traffic behavior for cooperative merge scenarios. Figure 3.14 is a snippet of a platooning test that was performed under the co-simulation setting.

CARLA and SUMO operate in a server-client mechanism where the simulation operates on a local Internet Protocol (IP) address with a dedicated Transmission Control Protocol (TCP) host address. Running both simulators at the same IP address while setting the corresponding port address in both simulators is required to establish communication between the two. Once the connection is established, CARLA-SUMO co-simulation requires an identical simulation map on both simulators. Though CARLA runs in a much more realistic

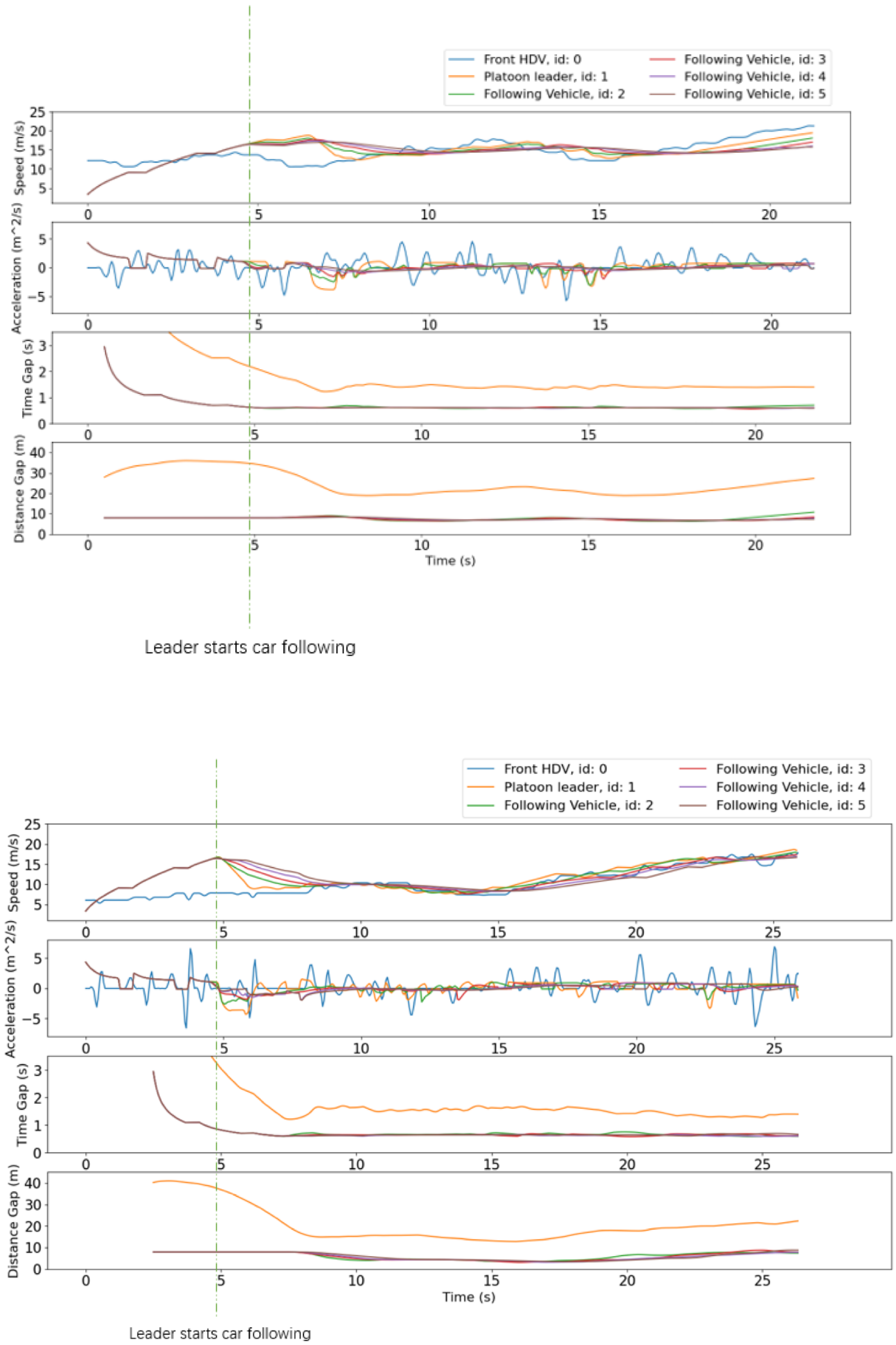


Figure 3.13: NGSIM trajectory results for simulation in CARLA.

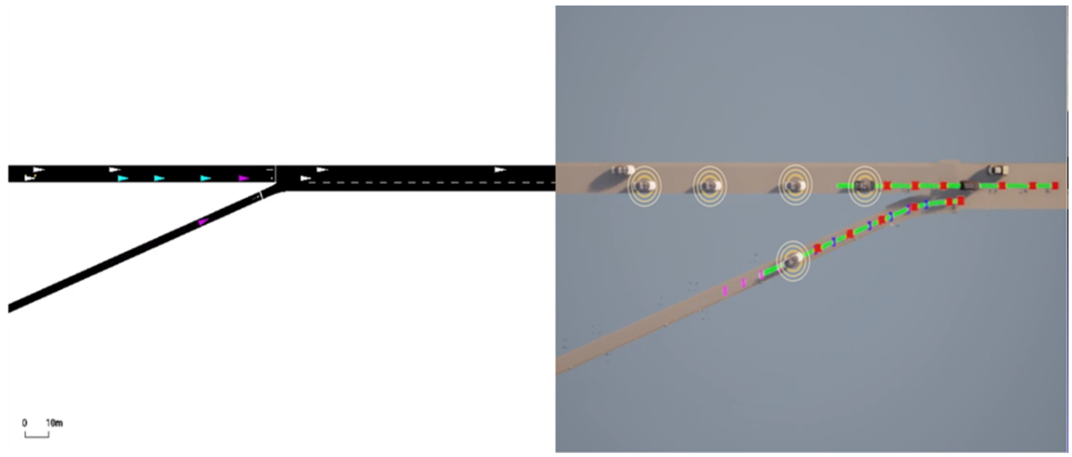


Figure 3.14: view in CARLA where the green lines and red dots represent planned trajectory path and points, and the blue dots and pink dots represent on-route waypoints and historical trajectories, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

environment, the basic traffic network between the two simulators has the same opendrive format [22]. Therefore, it is important to edit the basic traffic network in opendrive format and share this file between the two simulators. With the opendrive file, the CARLA simulator can populate the detailed road surface and environment, whereas SUMO simply interprets the network as a node-edge structure. During testing, the co-simulation requires using synchronous mode for both simulators. This setting limits the simulation clock speed, which is based on the slowest port at each step to ensure the simulation on both simulators is synced. As a result, each vehicle's stepwise information is shared across simulators, though the calculation was performed on different simulation simulators.

We also developed a heuristic joining method (HJM), which serves as the baseline merge algorithm to be compared to the GFS controller. The HJM is a sorting-based method that chooses one platoon member with the shortest Euclidean distance for the merging ends. Unlike a heuristic-based method, GFS considers platoon members' position, speeds, and surrounding human-driven vehicles' information, aiming to accomplish joining maneuvers while maintaining a globally optimized traffic delay. Therefore, the network level performance of GFS is expected to be superior, though the merging sequence may differ from the HJM's

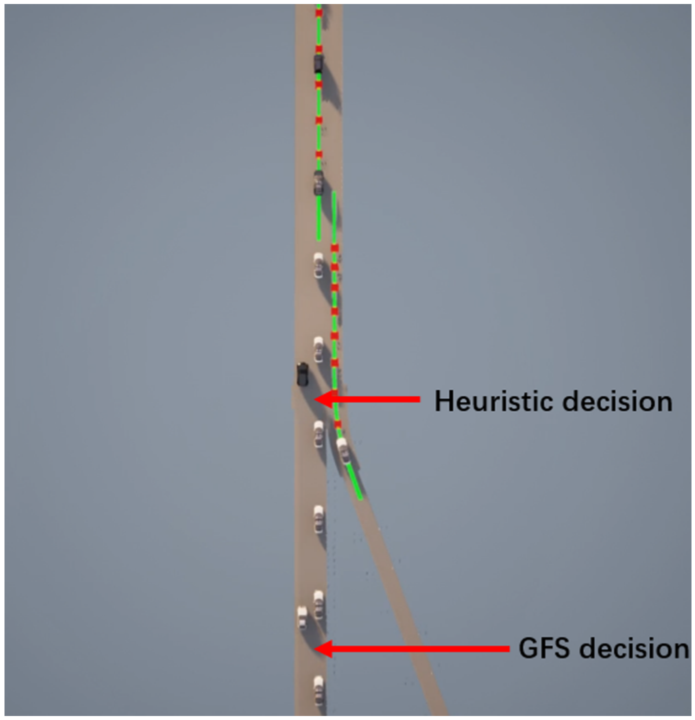


Figure 3.15: Different decision makings between HJM and GFS.

suggested result. Figure 3.15 shows the merging position from both HJM and GFS. HJM chooses the position between the original second and third platoon members as the best merging point, while GFS selects the position between the fifth and sixth platoon members as the merging point. Such difference is generated by the optimization goal of these two methods: HJM tries to reduce the joining time and thus selects the closest position, whereas GFS aims to minimize the traffic delay caused by the joining process and does not seek an immediate completion to the merge operation at the cost of overall traffic performance. Additionally, GFS's strategy reduces the abruptness of lane changes, which may cause upstream oscillations and leaves the vehicle enough space to speed up.

The results produced by these two different algorithms are shown in Figure 3.16. As Figure 3.16(a) demonstrates, after the joining request is approved, HJM only takes 7 s to finish both the gap opening and the merging vehicle change lane. In contrast, it takes about13 s for GFS to finish the lane change since the merging vehicle needs to slow down to wait to join the gap behind it, as Figure 3.16(b) shows. Although HJM takes less time to finish the maneuver, potentially, it may affect safety, traffic delay, and energy assumption more negatively because the three platoon members (ID5, ID6, ID7) are required to reduce the speed dramatically. As Figure 3.16(b) demonstrates, compared withHJM, the cooperative merge influences only one platoon member (ID 8) under GFS's strategy. Notably, joining vehicle's following gap decreased in the 30 seconds after the joining started but before lane change occurred. This trend reflects the GFS's preference to speed up the joining vehicle before the lane change and then slow it down afterward to gradually increase the gap (i.e., slow down speed) to the pre-defined steady value. This preference reduces the merging gap created by the mainline platoon members. In comparison, the HJM method handles the lane change with a bigger merging gap but requires all downstream members to accelerate and close the gap after changing lanes, which causes more disturbance (i.e., stop-and-go behaviors, abrupt speed changes, etc.) and fuel use. As a result, the mainline platoon experienced minimal speed change and disturbance, yielding an improved throughput.

Overall, our results indicate that the GFS successfully learned, during the traffic environments via SUMO, to optimize the traffic impact through a complex nonlinear mapping between the inputs (such as the speed differences between the mainline and merging vehicles) and strategy outputs, in a process that is hard to capture by arbitrarily defined heuristic rules. Regarding large-scale traffic, such a joining preference will significantly reduce the cost in terms of traffic delay, safety and energy consumption.

3.4 Conclusion

CDA technologies have the potential to improve roadway capacity, travel reliability, and traffic performance. In the previous studies, a single-lane vehicle stringing algorithm, or CACC, was developed, whereas our study proposes a multi-lane platooning algorithm with organized behavior via a hierarchical control structure for more complicated and practical situations. Based on the simulation results, multiple key observations and implications are

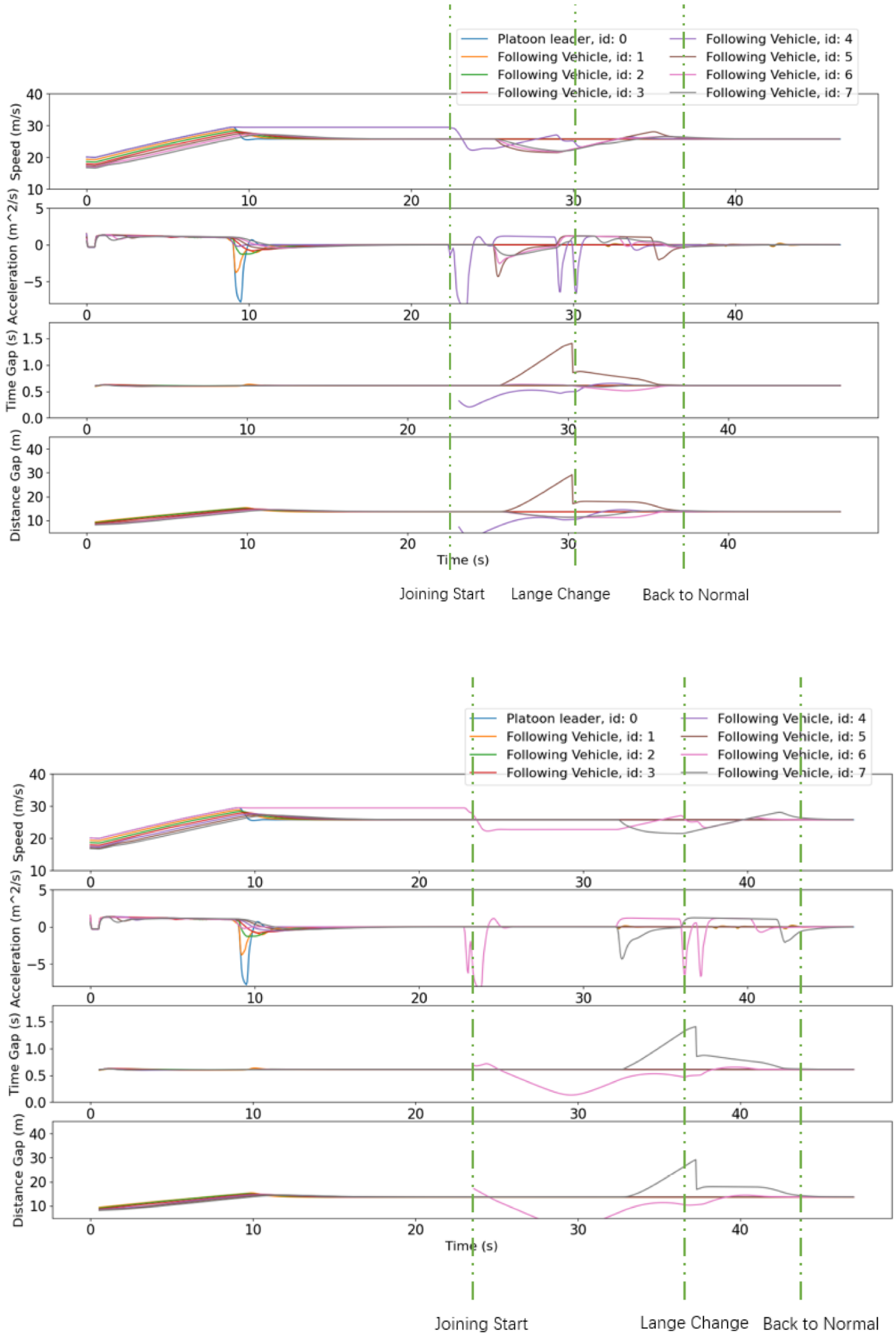


Figure 3.16: Cooperative merging results utilizing different algorithms.

summarized as follows:

- The proposed multi-lane platooning algorithm can efficiently guide the behavior of platoon members and external C-ADS-equipped vehicles aiming to join or leave a platoon in adjacent lanes under each superstate and precisely switch between the superstates and states under different complicated scenarios.

- The combination of gains in gap regulation and headway regulation can significantly impact the capability of maintaining the desired time gap. The results indicate the importance of considering both the immediately preceding vehicle and the platooning leader's trajectories to maintain the desired time gap and standstill distance.

- With intent/trajectory-sharing for CDA, C-ADS-equipped vehicles can be proactive about others' intentions, and the platooning Formation, Operation, and Dissolve superstates are more efficient and effective under increasing MPRs. Therefore, the proposed multi-lane platooning algorithm can improve traffic level performance further.

- The proposed algorithm adopts a GFS algorithm as a secondary controller to handle the more complex and hard-to-define cooperative merge scenarios. This unique structure compromises efficiency in pre-defined scenarios and accuracy in rare corner cases, demonstrating the capability of both rule-based and learning-based solutions.

Moreover, the proposed algorithm adheres to the existing SOTA ADS platforms, subscribing to the perception module as input and publishing the desired trajectory in a data structure compatible with the downstream control module. Within the multi-lane platooning algorithm, the mission planning step leverages the deterministic FSM to propose a suitable strategy for the current situation. The subsequent motion planning step transforms the desired mission plan into a series of waypoints using cubic spline interpolation that calculates a smooth trajectory to maintain vehicle stability. Overall, the proposed algorithm is a planning application that can be deployed in parallel with other planning applications on SOTA ADS platforms.

However, due to the natural construction of the leader–follower structure, accurate, efficient, and secure V2V communication is crucial to such applications. The experiments were conducted assuming that the V2V communication had no noise or delay, and the CADS vehicle has sufficient computation power to handle the communication and reasoning. In more realistic scenarios, algorithm performance under compromised communication capability and computational power are future research topics that need further discussion. Moreover, a stochastic version or real-time optimization and artificial intelligence/machine learning methods can be utilized in the future to improve the performance further by considering the local performance and the corridor traffic performance. On the other hand, the gains in gap regulation and headway regulation are fixed, leaving additional research gaps for dynamic and adaptive control under different scenarios. Lastly, the contribution of this study focuses on the innovations in the planning stack (at both mission and motion planning levels). Future studies can consider integrating the sensing and perception module with planning and control to explore end-to-end or semi-end-to-end solutions for cooperative platooning and merging.

Chapter 4

Strategic and Tactical Decision Making for Multi-lane Cooperative Platooning - Phase Two: Real-world Testings

With advancements in driving automation and vehicle-to-vehicle (V2V) communication, cooperative automated driving systems (C-ADS) have become a key area of research for achieving high standards in transportation safety, efficiency, and sustainability. In a prior study, we proposed and validated a cooperative multi-lane platooning algorithm on a state-of-the-art (SOTA) C-ADS platform within an idealized simulation environment. However, a gap remains between simulated and real-world performance due to system imperfections, such as perception errors, communication lags, and variations in vehicle models. This study extends the evaluation of the multi-lane platooning algorithm in real-world conditions, aiming to bridge this gap by developing a digital twin of the Suntrax testing track in CARLA, based on high-definition sensor data collected on-site in Auburndale, Florida. The digital twin includes a multi-lane highway stretch with an on-ramp merging lane, enabling us to conduct simulated tests and compare them with software-in-the-loop (SIL) and real-world experiments involving up to five C-ADS vehicles. Building on insights gained from simulation tests, real-world experiments were performed at the Suntrax testing facility using the cooperative platooning algorithm integrated with the FHWA CARMA platform, validating its compatibility with C-ADS systems and its ability to establish and maintain multi-lane platoons safely and efficiently.

In alignment with the parallel development and testing framework, the IHP field test in this chapter combines physical (i.e., ADS vehicle platform) and virtual (i.e., test track digital twin) environments to evaluate the cooperative multi-lane platooning algorithm at an advanced stage. Since simulation testing was completed in a previous phase, the focus here shifts to SIL, HIL, and field testing within the parallel validation module, aligning with the primary goal of real-world deployment and multi-vehicle field testing. Scenario engineering within the digital twin enables diverse, platoon-specific highway scenarios that closely replicate real-world conditions, providing a testing environment that mirrors practical challenges. This progression from digital twin simulations to SIL, HIL, and field tests facilitates a robust, scalable assessment of C-ADS capabilities. Parallel operations coordinate all experimental activities, real-time interactions among platoon members, and decision-making processes, enhancing system adaptability and performance. This comprehensive framework demonstrates the algorithm's effectiveness and compatibility across platforms, offering a solid foundation for the technical community and policy-makers in advancing C-ADS-based platooning to improve transportation system performance.

4.1 Introduction

4.1.1 Background

Intelligent transportation systems (ITS) are reshaping transportation technologies by elevating transportation systems management goals such as sustainability, safety, and efficiency. By leveraging cooperative driving automation (CDA) technologies, traffic efficiency, energy consumption, and driving safety can be significantly improved [86] with the potential benefit of significantly enhanced cooperation between vehicles such as platooning (i.e., perception accuracy and perception range) [102]. As one of the most promising CDA technologies, the concept of cooperative adaptive cruise control (CACC) [30,60] was introduced and extensively studied to overcome the lack of peer awareness. Previous researchers [30, 60] reviewed the application of CACC in various scenarios with multiple MPRs, [30] indicating that CACC can significantly enhance the longitudinal behavior of advanced driver-assistance systems (ADAS) and reduce the average following gap by up-to 57%.

4.1.2 Previous efforts in multi-lane platooning and intent-sharing

Though tremendous benefit has been demonstrated, CACC still has a major drawback of lacking lateral consideration. To this end, significant advancements in developing multi-lane platooning have been recently made. A lookup table was utilized to optimize platooning formation across multiple lanes [27]. Simultaneously, the effectiveness of two distinct ad-hoc platoon formation strategies (greedy formation and ordered formation) was evaluated in a multi-lane freeway environment, using comprehensive simulation tests [56]. These studies demonstrated traffic level improvements when using multi-lane platooning strategies compared to the conventional single-lane CACC algorithm. However, a common limitation in these studies is the lack of intent-sharing cooperation and the consequent inability to handle complex maneuvers, such as cooperative merging or adding new members. Therefore, an intent-sharing mechanism, along with longitudinal and lateral planning through V2V communication, are the critical factors for handling complex maneuvers across multiple lanes. Building upon this premise, our preceding research [31] delivered a multi-lane cooperative platooning algorithm, formulating a methodology characterized by organized platooning behaviors. When contrasted against traditional CACC-based methods [30,34,45,60], this algorithm harnesses a hierarchical control structure, assigning specific roles to the platoon leader, which in turn allows each member to share their comprehensive trajectory plan. Both leaders and followers abide by a common set of rules and protocols shared among all members, effectively guiding intricate multi-lane cooperation. Moreover, the multi-lane cooperative platooning algorithm is designed to sync with the real-world ADS vehicle framework, facilitating ease of on-vehicle deployment and promoting excellent compatibility across various vehicle types.

4.1.3 Previous integration and evaluation methods

While our proposed multi-lane platooning algorithm [31] has demonstrated its superiority, the necessity for real-world testing is critical to validate its effectiveness in a real-world setting. The paucity of real-world testing presents a major bottleneck in the advancement of stringing algorithms, given the individual vehicle and traffic-level challenges that must be overcome. These include perception errors, processing delays, vehicle model uncertainties, hardware differences, and the scarcity of physical vehicles and testing resources. Despite these obstacles, real-world experiments are essential for confirming the effectiveness of platooning algorithms. These experiments necessitate dedicated facilities that mirror real-world highway conditions. Despite the significant challenges, [60] conducted on-the-road experiments to test the CACC algorithm on a string of vehicles to compare three longitudinal controllers. This field test proved the string stability of the CACC algorithm and provided valuable data for numerous research aspects. However, because the CACC is a longitudinal-only algorithm, lateral behaviors such as lane change, multi-lane cooperation, and intent-sharing are out of scope. To the best of the authors' knowledge, there is no research in the literature reporting a proper multi-lane platooning field test with C-ADS vehicles. Based on the multi-lane platooning algorithm in our previous study [31], this paper will fill the gap by providing our main findings on the first multi-lane platooning field test.

4.1.4 Key contributions

This study conducts real-world testing of a multi-lane platooning algorithm using the CARMA platform. Our algorithm is implemented as a strategic plug-in on multiple fully automated C-ADS vehicles, enabling cooperative lateral and longitudinal control through a customized

V2V communication protocol. A comprehensive testing process is devised using a digital twin of the Suntrax track, created from high-definition sensor data. This iterative approach progressively introduces more realistic environments, allowing systematic verification of the algorithm's performance while minimizing risk. The testing begins with simulations, followed by software-in-the-loop testing using the Robot Operating System (ROS) [75], and concludes with single and multi-lane testing on the actual track with three to five C-ADS vehicles, demonstrating the algorithm's potential. Reflecting on the test results, the observations and contributions of the field experiment are summarized below:

- To bridge the gap between the simulation and real-world tests, we propose leveraging the digital twin and software in the loop testing pipeline for the multi-lane platooning algorithm. Specifically, we create a digital twin of the real-world testing track in Carla and establish the software-in-the-loop platform with the multi-lane platooning algorithm via the Robot Operating System (ROS). Based on this platform, simulated tests that are similar to the real-world experiment with up to five participated C-ADS vehicles have been conducted. The simulation testing results from the digital twin serve as a baseline for on-the-track tests of C-ADS vehicles. The adaptability of the multi-lane platooning algorithm by seamlessly integrating it with the software-in-the-loop platform has been confirmed. In addition, results confirmed the similar performance of simulation and real-world environments and showed that creating a digital environment for real-world tests provides a general development paradigm for cooperative driving automation algorithm verification.
- After the confirmation of the implementation in the software in the loop tests, based on the CARMA platform, our multi-lane platooning algorithm (Han et al., 2022) was successfully programmed as a strategic plug-in that is compatible with other guidance plug-ins in the CARMA platform. In addition, the multi-lane platooning strategic plug-in has been open-sourced to the community.

- Comprehensive test cases and diverse scenarios with up to five fully automated C-ADS equipped vehicles in the multi-lane platooning have been carried out, and extensive results have been presented and discussed. Real test results have proved that the multi-lane platooning algorithm can work with different numbers of C-ADS vehicles in multiple vehicles with different brands and body types, regardless of the vehicle's physical model uncertainties. Results manifest that the field test experiences are valuable as they validate the applicability of multi-lane platooning, provide field test data for ongoing and future research, and prove that the proposed algorithm can handle multi-lane highway platooning scenarios.

4.2 Multi-lane platooning algorithm and its integration with the CARMA platform

To gain a clearer insight into the experimental outcomes of the first SOTA ADS platform-compatible multi-lane platooning algorithm [31], this section briefly introduces the exposition of the algorithm and its integration into the CARMA platform. The multi-lane platooning algorithm enables vehicles equipped with C-ADS to explore platooning prospects not only in their present lane but also in neighboring lanes. Furthermore, the CARMA platform supplies a C-ADS structure to accommodate the integration of this multi-lane platooning algorithm.

4.2.1 Multi-lane platooning algorithm

As one of the C-ADS applications in the planning module, the multi-lane platooning algorithm follows the two-step framework (i.e., strategic mission and tactical motion planning) in which the mission planning leverages the perception results to find the optimum strategic plan. The consecutive motion planning generates a corresponding trajectory as a list of waypoints. An arbitration process selects the proper planning application on the SOTA C-ADS platform level (i.e., the CARMA platform). For simplicity, in Fig.4.1 and the remainder of the paper, we use the terms "mission planning" and "motion planning."

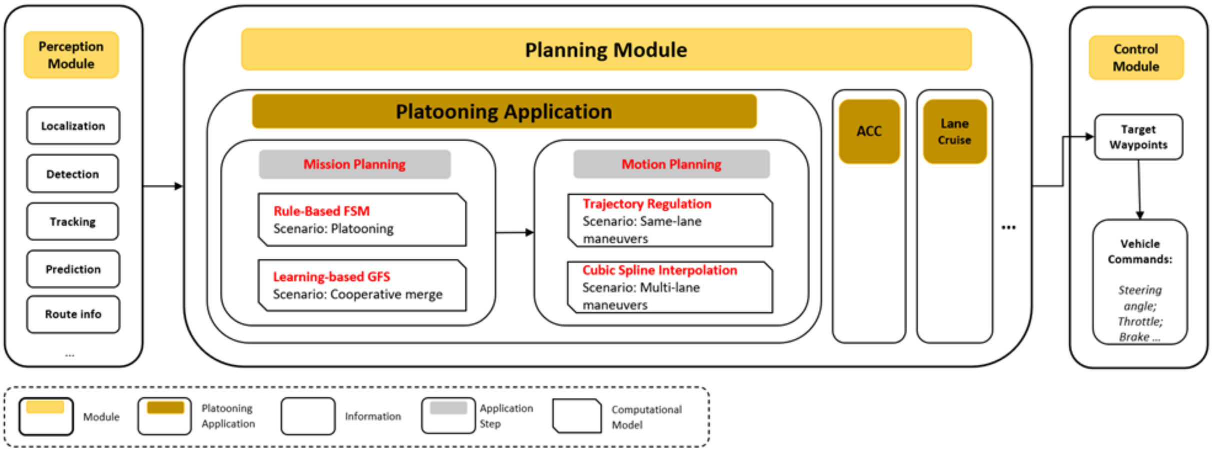


Figure 4.1: Logic and data flow of the platooning algorithm (Han et al., 2022).

Fig.4.1 shows the logic flow of the platooning algorithm in the two-step framework. The first step is mission planning (i.e., strategic planning in the CARMA platform), where one of the pre-defined semantic mission plans is selected based on the current scenario. The mission (i.e., strategic) planning identifies scenarios and recommends proper behavior based on the deterministic finite state machine (FSM). Then, a corresponding maneuver plan generates essential information (e.g., start/target location, start/target speed, start/target lane, etc.) for the motion (i.e., tactical) planning. Secondly, the motion (i.e., tactical) planning step parses the information from the maneuver plan, generating a detailed trajectory connecting the current position to the desired position with planned speed and steering. As this paper focuses on integrating the multi-lane platooning algorithm with the CARMA platform, the principle of the multi-lane platooning algorithm [31] will not be discussed.

4.2.2 Integration with the CARMA platform

The CARMA platform, an open-source C-ADS solution for cooperative driving automation, was used for integrating our multi-lane platooning algorithm due to its extensibility and external software integration capabilities. The CARMA system structures vehicle motion management into three main steps, as shown in Fig.4.2: localization ("SENSE STACK"), planning ("PLAN STACK"), and motion control ("ACT STACK"), providing a comprehensive autonomy framework. This cycle continually refreshes the vehicle's state to ensure precise and smooth operations. Deployed on each testing vehicle, the CARMA platform facilitates Level 3 vehicle automation and V2V communication.

The multi-lane platooning algorithm was integrated as a strategic level plug-in within the "PLAN STACK" of the CARMA platform, which facilitates ROS communication among all modules. In particular, the ROS communication interface, i.e., ROS topic, ROS parameters, and ROS services, will be composed to be compatible with the upstream (sense stack) and downstream (act stack) modules. Adapting to all necessary information from upstream and downstream modules, the algorithm, unlike conventional CACC methods, transcends vehicle brand or body type limitations due to CARMA's compatibility with multiple ADS platforms. This generality emphasizes the algorithm's distinct applicability in diverse real-world testing scenarios.

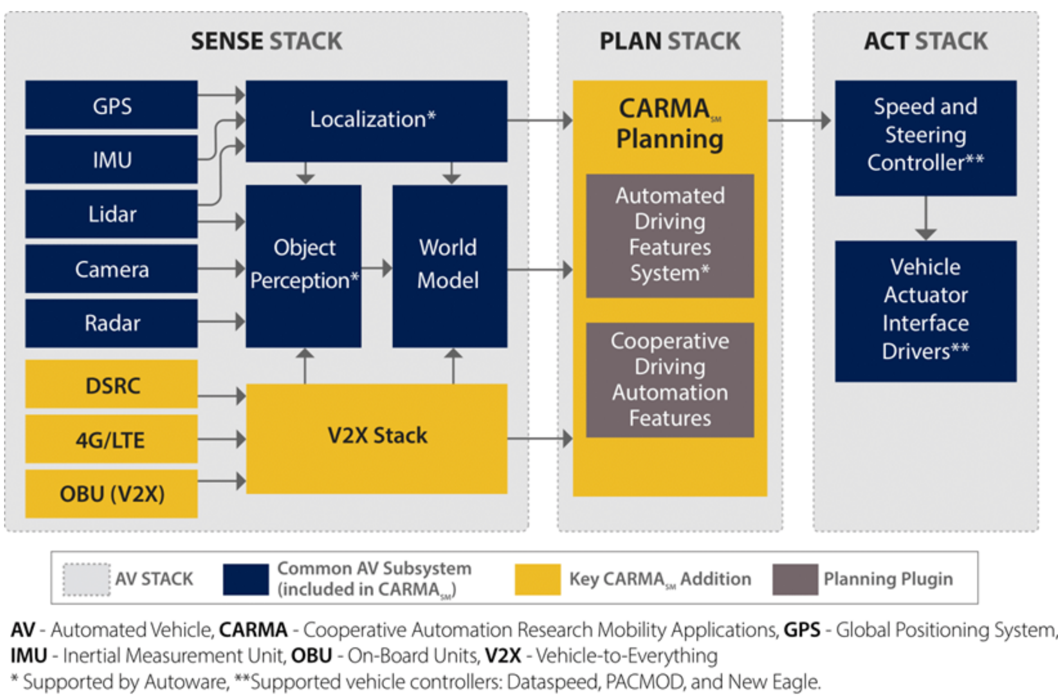


Figure 4.2: The software structure of the CARMA platform.

4.2.3 V2V communication protocol

For real-world deployment of C-ADS software prototypes like platooning, a comprehensive V2V messaging protocol is vital. Our proposed algorithm, based on real-time V2V communication, employs this protocol to enable key functions such as FSM behavior management, gap regulation, cooperative merging, and the APF algorithm [9], making it an integral part of software development compatible with state-of-the-art C-ADS platforms like CARMA.

The CARMA platform supports two types of V2V messaging: directional request-response and broadcast operational messages, which include crucial platooning information like size and ID. This allows all C-ADS vehicles within communication range to understand live platoon status and make decisions. Cooperation messages are structured as mobility requests and responses between any two vehicles within range. These are paired with state transitions in the FSM-based platooning behaviors according to specific circumstances. Table 4.1 summarizes the customized platooning messages.

4.3 Algorithm implementation and testing

Our previous work validated the multi-lane platooning algorithm in the OpenCDA simulation platform [103]. This paper focuses on real-world testing after integrating the algorithm into a pre-existing C-ADS platform. We employed a four-step testing pipeline: simulation, software-in-the-loop (SIL), hardware-in-the-loop (HIL), and close-track testing. This pipeline progressively incorporated increasingly realistic components to ensure algorithm integrity and consistent performance despite different levels of errors and uncertainties. The following sections detail the experiment preparations, testing scenarios, and each step in the testing pipeline.

Table 4.1: The customized platooning message for FSM-based V2V cooperation.

Initial state	Final state	Message Send	Cooperation Type
Mobility Requests			
Single C-ADS	Follower	Mobility join request to the target platoon leader	Rear join (same-lane and cutin)
Single C-ADS	Leader	Mobility join request to the target platoon leader	Frontal join (same-lane and cutin)
Single C-ADS	Prepare to join	Mobility lane change request to the target platoon leader	Cutin join (front and rear)
Mobility Response			
Leader	Leader	Mobility response to accept rear join	Rear join (same-lane and cutin)
Leader	Leader	Mobility response to deny rear join	Rear join (same-lane and cutin)
Leader	Follower	Mobility response to accept frontal join	Frontal join (same-lane and cutin)
Leader	Leader	Mobility response to deny front join	Frontal join (same-lane and cutin)
Leader	Lead with operation	Mobility response to accept lane change	Cutin join (front and rear)
Leader	Leader	Mobility response to deny lane change	Cutin join (front and rear)
Mobility Operation			
Single C-ADS	Single C-ADS	Publish current vehicle status	Constantly publishing
Leader	Leader	Publish current platoon status	Constantly publishing
Follower	Follower	Publish current vehicle status	Constantly publishing

4.3.1 Real-world experimental settings

Various C-ADS vehicles from different manufacturers were chosen for field experiments to validate the multi-lane platooning algorithm, emphasizing its brand and body type independence. The test track and platooning route were also carefully selected to suit each scenario.

Hardware: Sensor Suite, C-ADS vehicles, and test track

Experiments were carried out using multiple C-ADS vehicle platforms (as shown in Fig.4.3) equipped with the lasted version of the CARMA platform [82], where the multi-lane platooning algorithm has been integrated. All vehicles are equipped with an array of sensors, including a 5.9 GHz Cohda DSRC system, a NovAtel SPAN-G320 GNSS/INS integration system, a Velodyne VLP-32C LiDAR, two Mako-G319C front cameras, and a Delphi ESR radar to operate the entire system running on an on-aborad Spectra server to maintain full C-ADS capability. The experiments involved three C-ADS vehicles, two from Chrysler and one from Lexus, all modified by AutonomouStuff.

Figure 4.3: C-ADS vehicles and their specifications for the field experiment.

The Suntrax facility's oval track, the venue for our field experiment, consists of a two-lane mainline and a single-lane on-ramp. The selected test route, covering 971 meters and each lane measuring 3.5 meters in width, starts at the straight segment's beginning and ends right before the curve starts, as indicated in Fig.4.4. The merging area, parallel to the starting point, tapers into the mainline midway as detailed in Fig.4.5.

Figure 4.4: The overall layout of the Suntrax testing facility and the selected route for testing. The rounded green point is the starting position, and the rounded red point is the ending.

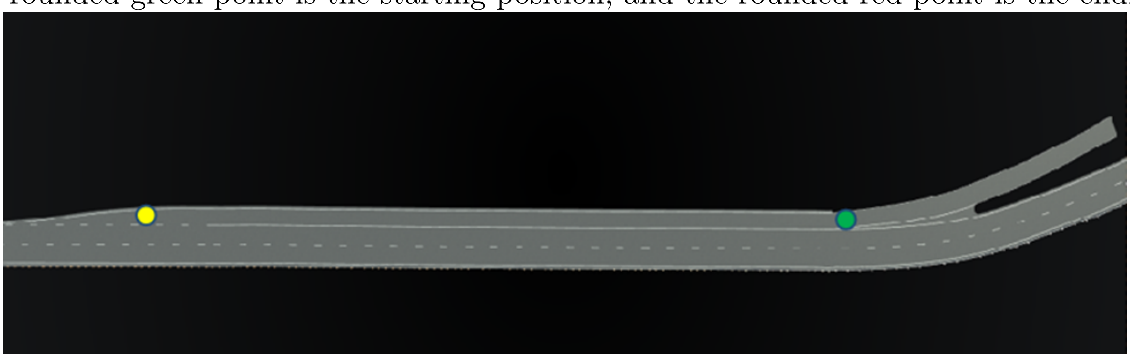


Figure 4.5: The overall layout of the Suntrax testing facility and the selected route for testing. The rounded green point is the starting position, and the rounded red point is the ending.

Software: the Suntrax digital twin

In this work, the multi-lane platooning algorithm was verified using the OpenCDA platform [102] and integrated into the CARMA platform. While there may be variations in sensor noise, communication delay, and vehicle physical model uncertainties between the OpenCDA and CARMA platforms, testing the algorithm in a digital twin of the real-world test track allows for a baseline evaluation of various parameters introduced throughout the pipeline, and can aid in adjusting the algorithm's performance. Additionally, simulation results can serve as a reference for verifying the functionality and proper implementation of the algorithm. The Suntrax testing track makes the digital twin of such a map valuable for future algorithm validation and development.

The digital twin environment was built based on the point cloud map of the test track, generated using LiDAR data and GPS data. The simulated route was identical to the designed route, with the GPS coordinates of the starting point serving as the reference point for the 3D scene reconstruction. A vector map is then fitted to the 3D scene, highlighting the road geometry, network topology relations, and lane information. Further details of the map generation can be found in [100]. The digital twin was subsequently generated based on the vector map with manual adjustments to match the condition of the test track. It should be noted that the term digital twin is used here for three reasons:

- The simulation environment, in terms of road material, lane structure, and traffic regulation such as speed limit, is identical to the real-world environment;
- CAVs configuration, the number and type of testing CAVs are the same as the real-world tests;
- To differentiate the simulation test in this work from that in our previous research.

4.3.2 Simulation Testing

The first stage of the comprehensive ADS testing process, simulation testing, confirms the functionality and behavior of the multi-lane platooning algorithm in a controlled environment. Two scenarios were tested: a merging vehicle joining at the front or rear of the platoon (Fig.4.6)). This stage sets a critical performance baseline but doesn't include factors like communication noise, system delay, and vehicle control differences. For a comprehensive understanding of these factors, larger-scale tests with four and five C-ADS vehicles were performed using real vehicles (section 4.3). The trajectory results from several simulated tests, following procedures from previous work [31], provide crucial insights for actual field tests. The scenarios used here, unlike in the previous work, were designed to mirror real-world conditions and utilize the digital twin's test track.

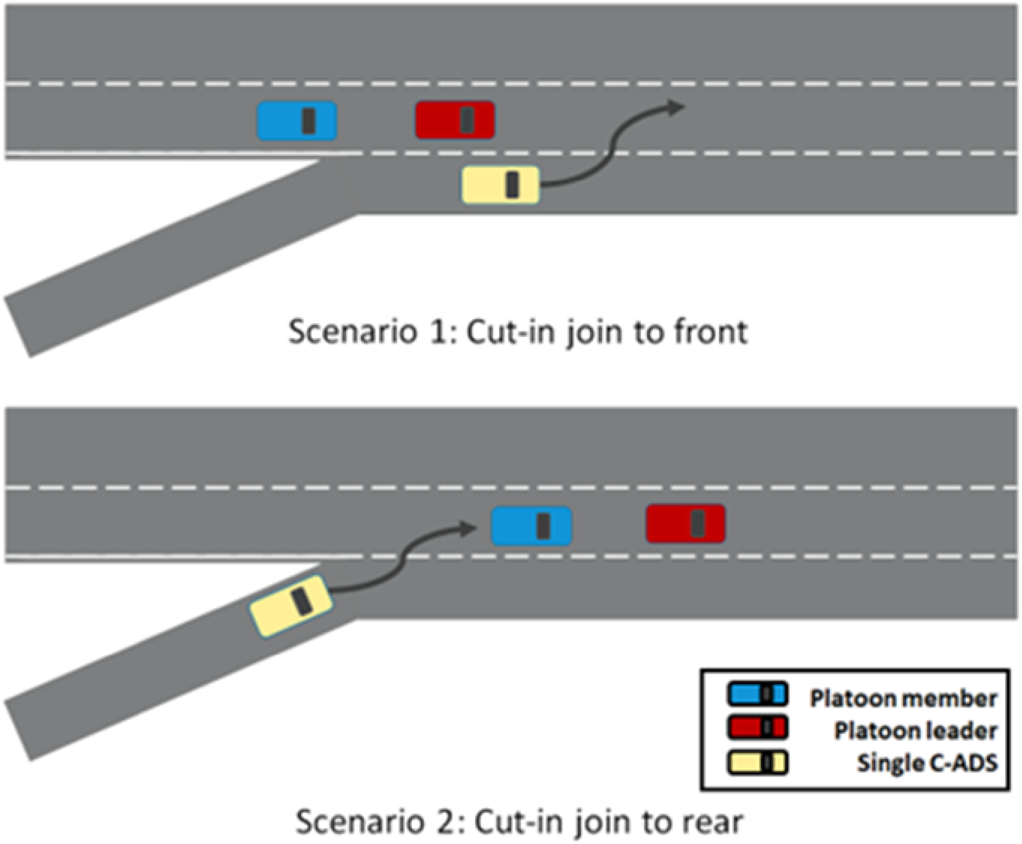


Figure 4.6: Simulation test scenario schematic.

4.3.3 Software-in-the-loop testing

The second stage of ADS testing, software-in-the-loop, evaluates the multi-lane platooning algorithm within a ROS environment that reflects the real-world software structure of the experimental vehicle, hence offering a realistic simulation of communication delays. The algorithm, originally developed within the OpenCDA framework [102], was adjusted to manage the simulation via ROS streams. The ROS-integrated simulation framework, outlined in Fig.4.7, uses ROS topics for environmental perception and vehicle localization. Once the route and control command are determined, they are broadcasted to the vehicle for implementation. This process uses the CARLA ROS bridge [21] and two custom ROS nodes to manage all simulation communication flows, allowing for a thorough validation of the algorithm's performance under software constraints.

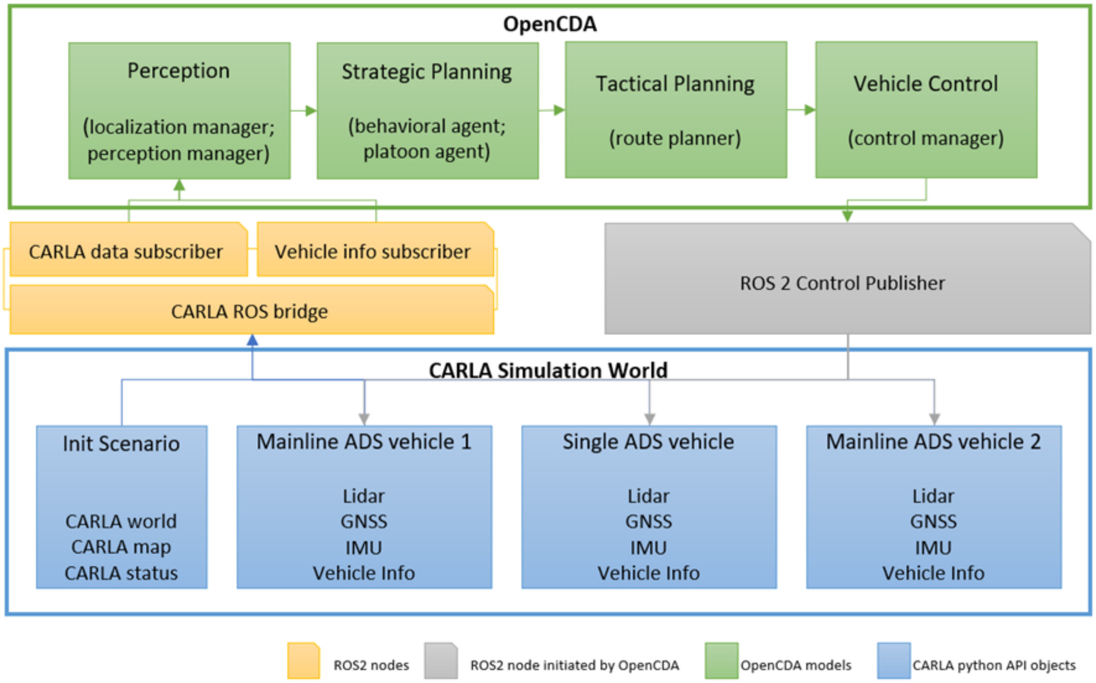


Figure 4.7: Framework of the ROS-integrated simulation test.

4.3.4 Hardware-in-the-loop testing

Hardware-in-the-loop testing replicates the final close-track testing setup using actual hardware, introducing additional complexities like communication and computational costs, sensor noise, and system delays. This phase requires a fully integrated C-ADS platform, including sensor suites, an onboard computer, and the ride-by-wire system. Here, the CARMA platform [82] is implemented as a C-ADS software framework, and the multi-lane platooning algorithm serves as a strategic and tactical plug-in. To simulate real-world operating conditions, the entire system is implemented on C-ADS vehicles that can support all system hardware requirements.

The HIL test is essential for validating both system integrity and the fundamental functions of the platooning algorithm. This testing sequence includes a single-vehicle functionality check followed by a two-vehicle platooning test. Initially, a single vehicle is deployed with a fully operational CARMA system to execute basic autonomous functions via CDA capabilities, ensuring that core autonomy features are functioning as intended. During this phase, the vehicle also continuously sends a joining request, preparing for potential platooning interactions. However, to fully validate the platooning functions, a two-vehicle test is required. The platooning features rely on V2V communication and coordinated vehicle behavior, which cannot be assessed with a single vehicle alone. In the two-vehicle test phase, two C-ADS vehicles form a platoon, establishing V2V communication to exchange real-time vehicle information. During operation, the gap regulation algorithm uses this shared data to continuously adjust the target speed, maintaining a safe and stable distance between vehicles. This coordinated testing setup is essential to confirm that the platooning algorithm's dynamic adjustments and collaborative behaviors perform reliably under real-world conditions.

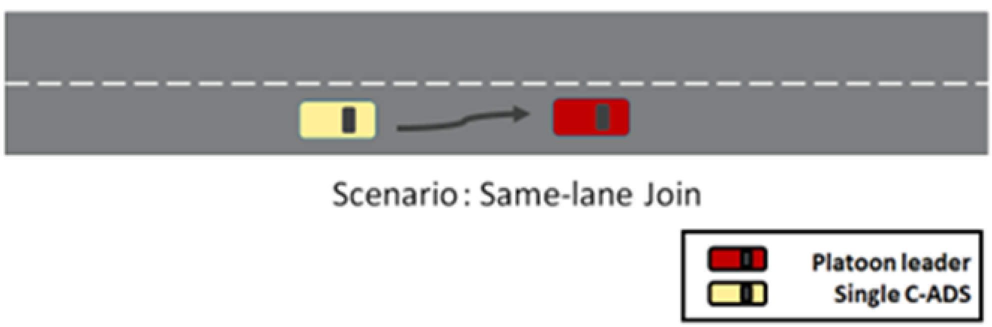


Figure 4.8: Hardware-in-the-loop experiment scenario schematic.

Conducted at the Suntrax testing facility, the hardware-in-the-loop testing, outlined in Fig.4.8, aims to validate the integration of the multi-lane platooning algorithm within C-ADS vehicle platforms and confirm the basic functionalities. By restricting the test to same-lane operation and minimizing the platoon size, the risk is significantly mitigated, paving the way for subsequent close-track tests.

4.3.5 Close-track Testing

The close-track field experiment marks the final phase of the ADS testing pipeline, designed to evaluate the proposed multi-lane platooning algorithm in a real-world environment. This stage scrutinizes multi-lane functionality, organized platooning behavior facilitated by V2V communication, and steady gap regulation dependent on shared vehicle data. Two scenarios are conducted as shown in Fig.4.9: a vehicle joining at the front of the platoon, and a vehicle joining at the rear. Despite the high degree of exposure and risk, these tests occur within a secluded test track to ensure safety and comply with legal limitations.

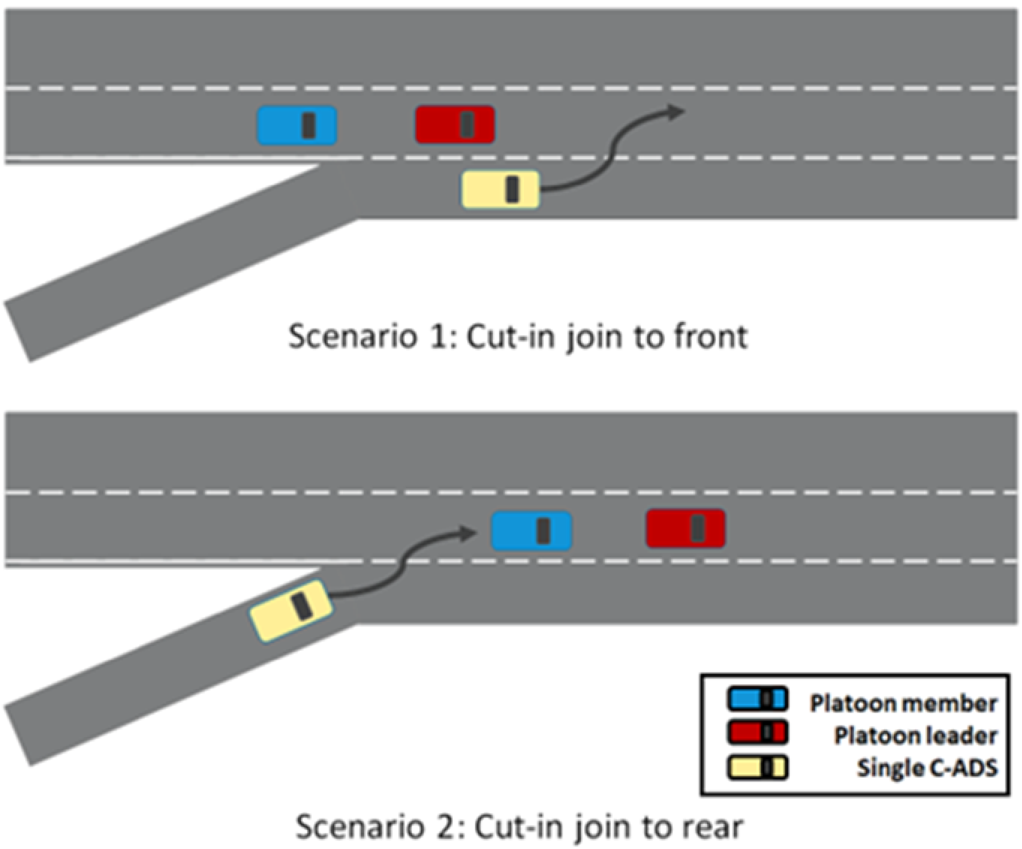


Figure 4.9: Field close-track experiment scenario schematic.

Both tested scenarios involve a C-ADS vehicle merging into a mainline platoon, assessing the algorithm's multi-lane capabilities. These scenarios also test the algorithm's capacity for cooperative maneuvers, necessitating effective cooperation between the merging vehicle and the platoon leader [31]. The leader determines the optimal point for incorporating the merging vehicle based on its relative position and speed and adjusts its speed to aid the merge while the joining vehicle integrates into the designated position concurrently.

To further validate the algorithm's ability to manage gaps within larger platoons, four-vehicle, and five-vehicle platoon tests are conducted. The stability of the algorithm in handling larger platoons can thus be verified, while the joining C-ADS vehicle performs identical cut-in maneuvers regardless of the platoon size.

4.4 Experiment results

This section presents the experiment results for each phase in the comprehensive ADS testing pipeline, signifying the achievement of set goals. The testing begins with the simulation test, using OpenCDA as the ADS platform to centrally control all platoon members. Next, the software-in-the-loop testing phase involves the vehicle status and control communication stream managed within ROS, further challenging the algorithm's stability, efficiency, and integrity.

The third step, hardware-in-the-loop testing, involves two real C-ADS vehicles forming a platoon in a single-lane scenario. With the involvement of hardware (sensors, onboard computer, ride-by-wire system), this phase focuses on a simplified same-lane platoon, validates the integration and functionality of the system, and establishes a strong foundation for the close-track test. The final phase, the close-track test, incorporates multi-lane scenarios, varying platoon sizes, and different C-ADS vehicles. This stage showcases the platooning algorithm's full potential in facilitating cooperation via V2V communication and organized behavior to guide vehicles in complex multi-lane situations.

4.4.1 Simulation evaluation in digital twin

In this subsection, the multi-lane platooning algorithm is performed in the digital twin environment. The simulation test that is identical to the real-world testing is of significant value as it serves as a performance baseline, diminishing tunning and validation effort by providing a contrast dataset under an ideal environment (i.e., no GPS error and delay, no system error and delay) during the on-site testing process.

Join to the front evaluation in digital twin

In this scenario, the joining vehicle starts at the on-ramp merging lane, merging into the mainline, and joins the target two-vehicle platoon to the front as the new leader.

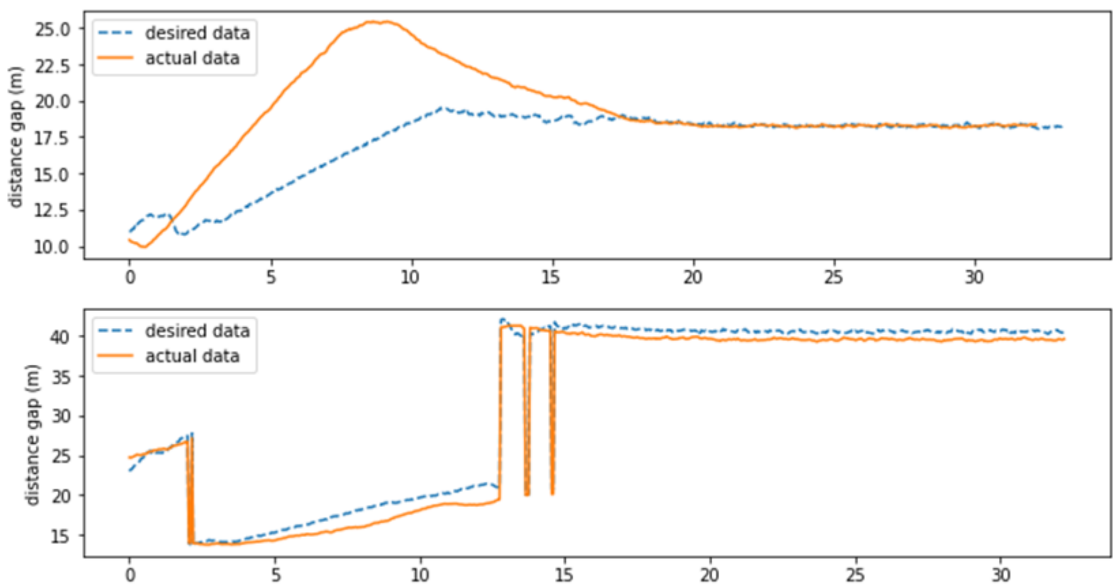


Figure 4.10: Distance gap comparison for joining to the front in simulation testings. The top figure describes the desired gap versus the actual gap for the middle vehicle, and the bottom figure describes the last member of the platoon.

The distance gap data for the two following vehicles in a platoon are illustrated in Fig.4.10. The top figure presents the data for the second vehicle, originally the leader of the platoon, which becomes a follower after the merge. The dotted blue line represents the desired gap, while the solid orange line displays the actual gap. Although the initial actual gap was larger than desired, the vehicle managed to reduce this gap and maintain a consistent distance from the leader.

The lower graph provides data for the third vehicle. As this vehicle was part of the initial platoon formation, its initial gap was close to the desired value. A distinct drop in both the desired and actual gap is observable due to a dynamic leader switch initiated by the APF algorithm. Initially, the third vehicle was following the platoon leader. However, due to proximity to the second vehicle, a "too close" safety disturbance was triggered, which reassigned the third vehicle's dynamic leader to the second vehicle. It's important to note that the second vehicle maintained its dynamic leader (the platoon leader) throughout, as it had only one potential vehicle to follow.

Join to the rear evaluation in digital twin

In this scenario, the joining vehicle starts at the on-ramp merging lane, merging into the mainline, and joins the target two-vehicle platoon to the rear as the new member at the end of the target platoon.

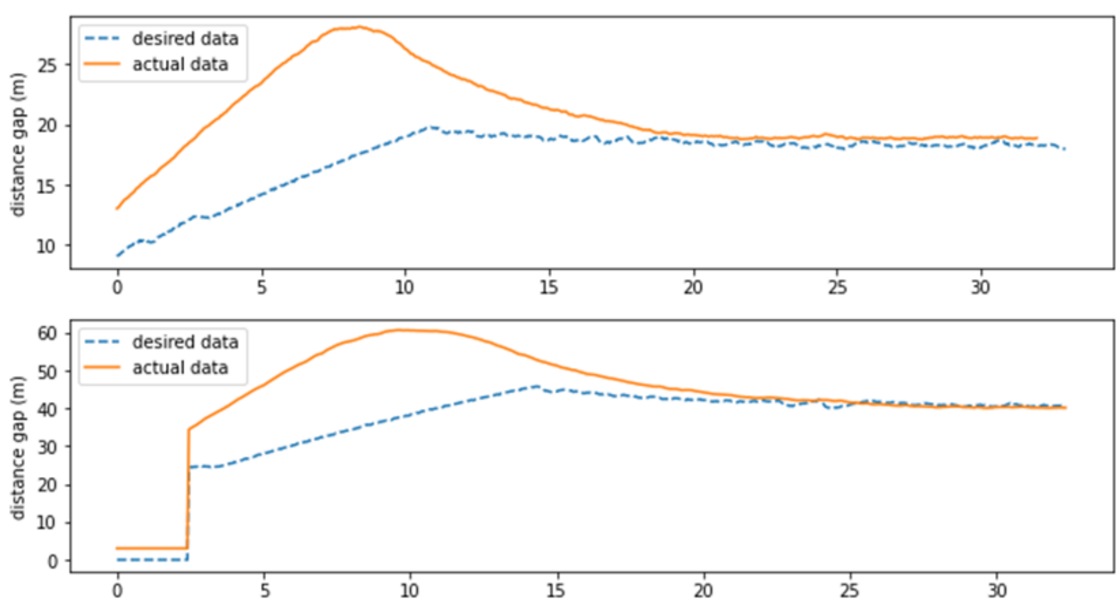


Figure 4.11: Distance gap comparison for joining to the front in simulation testings. The top figure describes the desired gap versus the actual gap for the middle vehicle, and the bottom figure describes the last member of the platoon.

Fig.4.11 illustrates the gap data for the second and third vehicles in the platoon. The upper chart denotes the second vehicle's desired and actual distance gaps, marked by blue dotted lines and solid orange lines, respectively. Even though the initial distance gap exceeded the desired one, the second vehicle was able to adjust and maintain the intended distance. Likewise, the lower chart outlines the third vehicle's progression, which also starts with a larger gap and successfully reduces it to match the desired value. Overall, joining at the rear is a less complicated process with minimum disturbance that doesn't involve APF leadership transition.

Results demonstrate the successful application of the multi-lane platooning algorithm using three C-ADS vehicles in a digital twin environment. This simulation test verifies lane platooning behaviors, gap regulation algorithm, and the V2V cooperation protocol in a setting mirroring real-world conditions. It also serves as a performance baseline, easing the tuning and validation process by offering a comparison data set under ideal (i.e., error-free) circumstances.

4.4.2 Software-in-the-loop experimental results

The software-in-the-loop testing replicates the scenarios of the simulation test. As previously discussed, the aim of this phase is to integrate ROS as the medium for exchanging simulation status. By doing so, the experiments can better emulate real-world conditions.

Join to the front in the software-in-the-loop evaluation

The software-in-the-loop test emulates the scenarios from the simulation test, with a vehicle merging as the new leader into an existing two-vehicle platoon. Fig.4.12 illustrates the distance gap data of the two following vehicles. Results closely mirror the simulation test.

The top graph displays the second vehicle's desired and actual distance gaps. Initially, the platoon leader transitions into a middle-member role following the new leader's entry. Despite a larger initial gap, the second vehicle adjusts to maintain a consistent distance. The

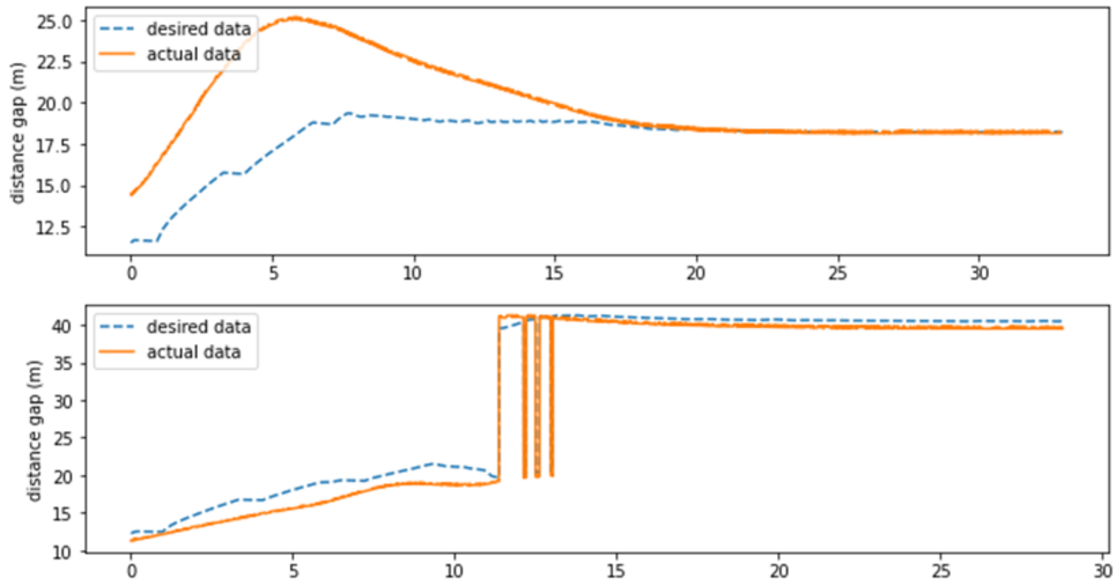


Figure 4.12: Distance gap comparison for joining to the front in the software-in-the-loop testings.

bottom graph presents the distance gap data for the third vehicle. Because it's already a platoon member, its data closely aligns with the desired value. Similar to the simulation test, the APF algorithm reassigned the third vehicle's dynamic leader to the second vehicle after deeming it too close.

Join to the rear in the software-in-the-loop evaluation

In this scenario, the joining vehicle merges into the mainline, becoming the third member. Fig.4.13 displays the distance gap data for the two following members of the platoon. For both the second and third vehicles, the plots show a minor discrepancy between the desired and actual gaps at the start, but they swiftly manage to adjust and adhere to the intended distances.

Overall, the results obtained from the software-in-the-loop test closely match those from the simulation test, indicating that the multi-lane platooning algorithm performs consistently and reliably across various scenarios and platforms. An important observation is the increased variability and instability in the actual gap values, highlighting the differences between direct

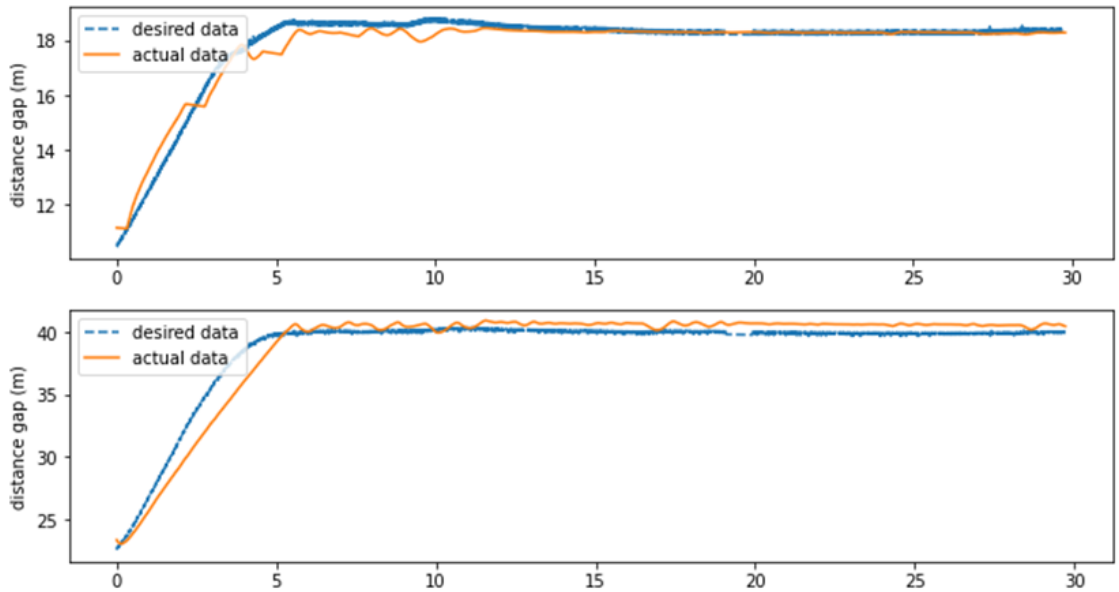


Figure 4.13: Distance gap comparison for joining to the rear in the software-in-the-loop testing.

server data streams and ROS subscriptions. However, despite the additional challenges of communication noises and delays, the algorithm continues to deliver stable and consistent performance. This successful outcome affirms the algorithm's effectiveness, thereby providing valuable insights for subsequent integration and experiments.

4.4.3 Hardware-in-the-loop experimental results

The single-vehicle portion of the HIL test was successful, confirming the system's basic autonomy and stability. During this phase, the C-ADS vehicle, equipped with a fully operational CARMA system, effectively executed core functions like lane-keeping and adaptive speed regulation, demonstrating stable, consistent performance. Additionally, the vehicle's platooning algorithm operated in an idle state, actively broadcasting joining requests to seek potential platooning opportunities, validating its readiness to initiate cooperative driving interactions. These results confirm that the vehicle is both stable in its basic functions and proactively prepared for future multi-vehicle coordination, meeting the objectives of the single-vehicle HIL test.

The platooning algorithm's performance was further validated. The two C-ADS vehicles successfully established V2V communication, sharing real-time vehicle information necessary for cooperative movement. During the platooning operation, the gap regulation algorithm continuously adjusted each vehicle's target speed based on the exchanged data, ensuring stable spacing and synchronized speed adjustments within the platoon. This coordinated behavior confirmed the algorithm's ability to maintain safe inter-vehicle distances and adapt dynamically to changing conditions, validating the core functionality and stability of the platooning system in a multi-vehicle setting. As displayed in Fig.4.14, results validate same-lane platooning and the gap regulation algorithm in a real-world setting. Two C-ADS vehicles were utilized, forming a two-vehicle platoon upon engagement of the CARMA platform and maintaining a consistent intra-platoon gap throughout the test route.

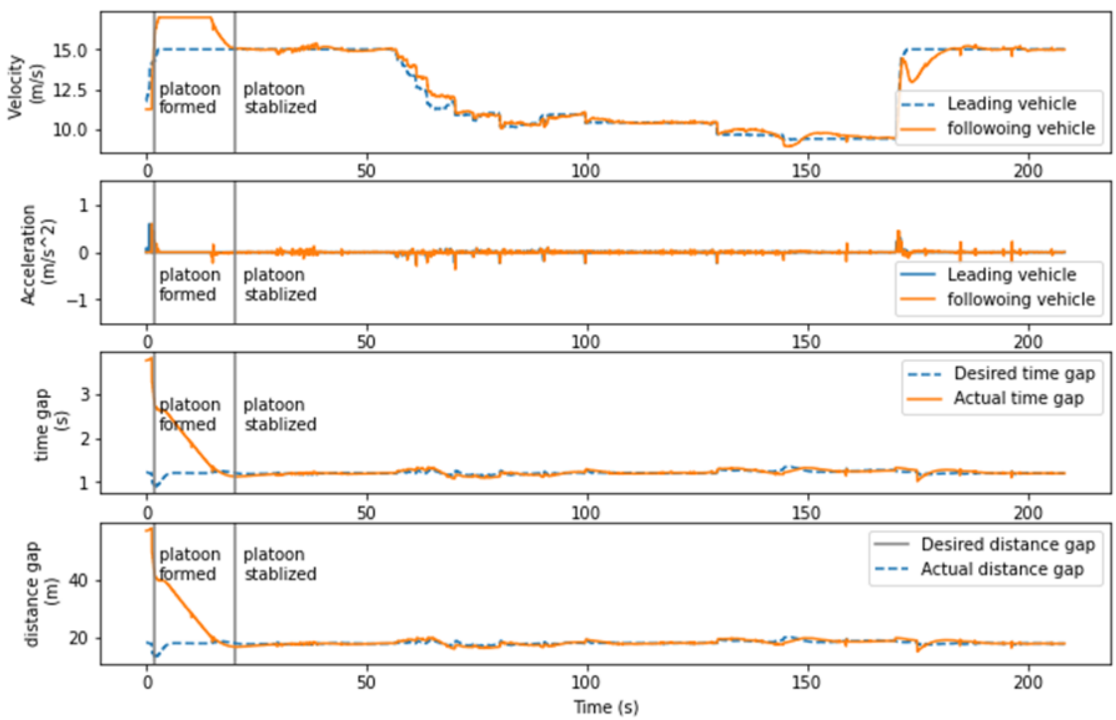


Figure 4.14: Same-lane platooning trajectory results where the detailed trajectory data of both members of the platoon is presented.

The desired time gap was 0.8 seconds, and the gap regulation algorithm took this as input. The desired gap varied as it was based on speed data read from the vehicle odometer, and the initial velocity difference was due to the follower catching up with the leader. While this test focuses on same-lane application, it serves as a foundation for multi-lane scenarios by demonstrating successful platoon formation and leader-follower structure. Fig.4.14 shows that the platooning algorithm can consistently uphold a steady platoon.

Crucially, the hardware-in-the-loop test provides a realistic environment that incorporates real-world communication costs and system delays. It confirms the algorithm's efficacy on actual vehicle platforms, paving the way for multi-lane experiments and field tests.

4.4.4 Real-world experimental results

Closed track tests under the described scenarios were carried out in the aforementioned environment, where the multi-lane platooning algorithm [31] was tested multiple times. The trajectory figure presented for each scenario is randomly selected among all testing results, and a statistical comparison with all testing data is presented.

Cutin join to the front

Fig.4.15 depicts the trajectory of a cut-in join to the front, aimed at validating multi-lane platooning functions. Here, a joining vehicle merges into the mainline becoming the new leader of the mainline platoon. To enable this, the joining vehicle starts moving before the platoon reaches the merging area's end.

The joining vehicle integrates into the platoon at the 50-second mark, as the other two vehicles hit top speed to close the enlarged gap with their new leader - a gap deliberately created for lane changes. The three-vehicle platoon stabilizes after 10 seconds, maintaining their desired gaps hereafter. Notably, vehicle acceleration fluctuates, within a safe range, at this stage due to the complexities of the cut-in-join process.

Fig.4.16 presents the gap comparison of trailing two members. The last vehicle's desired gap experiences several spikes due to the APF dynamic leader assignment to enlarge abnormally small gaps. This reassignment is crucial when the joining vehicle becomes the new leader.

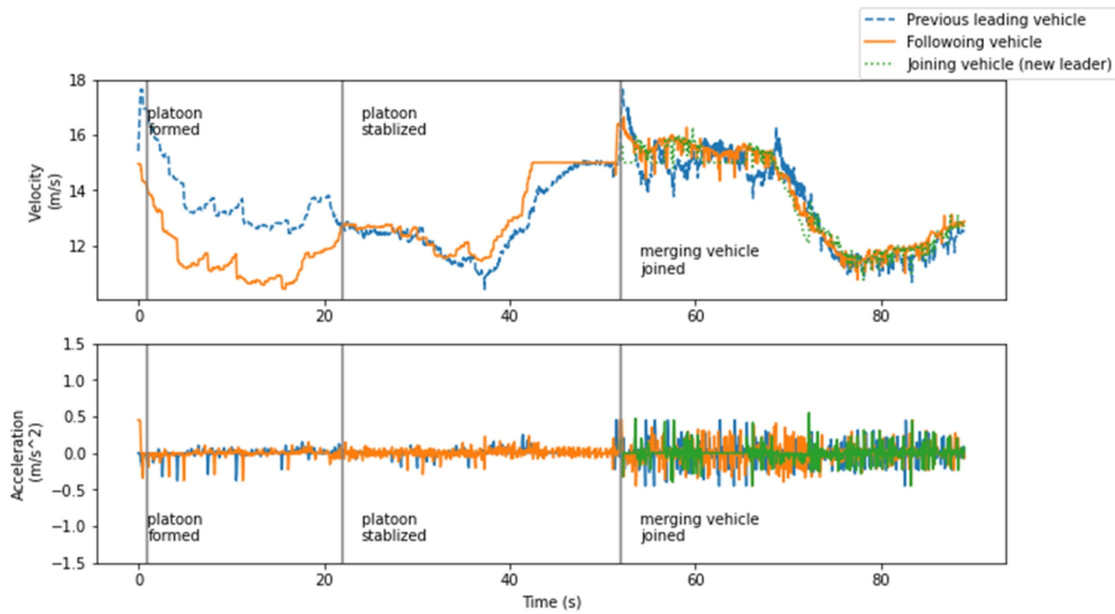


Figure 4.15: Cut-in join to the front scenario trajectory results. The velocity of all three platoon members is presented in the top figure, and the acceleration of all three members is presented in the bottom figure.

As this process creates a large cut-in gap between the first two members and a small gap between the last two members.

The cut-in join to the front scenario is the first multi-lane test. Despite some fluctuations in the intra-platoon gap and differences between the simulation and the real world, the algorithm successfully manages C-ADS integration, V2V communication, gap regulation, and platooning behaviors. Considering the challenges of the field tests, the experiment is deemed successful, with the vehicles behaving similarly to the simulation tests.

Cut-in join to the rear

Fig.4.17 illustrates the results of the cut-in join to the rear scenario. This scenario tests the multi-lane platooning capabilities by having the joining vehicle merge into the platoon from behind. The CARMA platform is engaged by the joining vehicle as the mainline platoon passes by, simulating a realistic rear-joining situation. Compared to the cut-in-front scenario, this situation is less complex as the platoon leader remains unchanged, which eliminates the

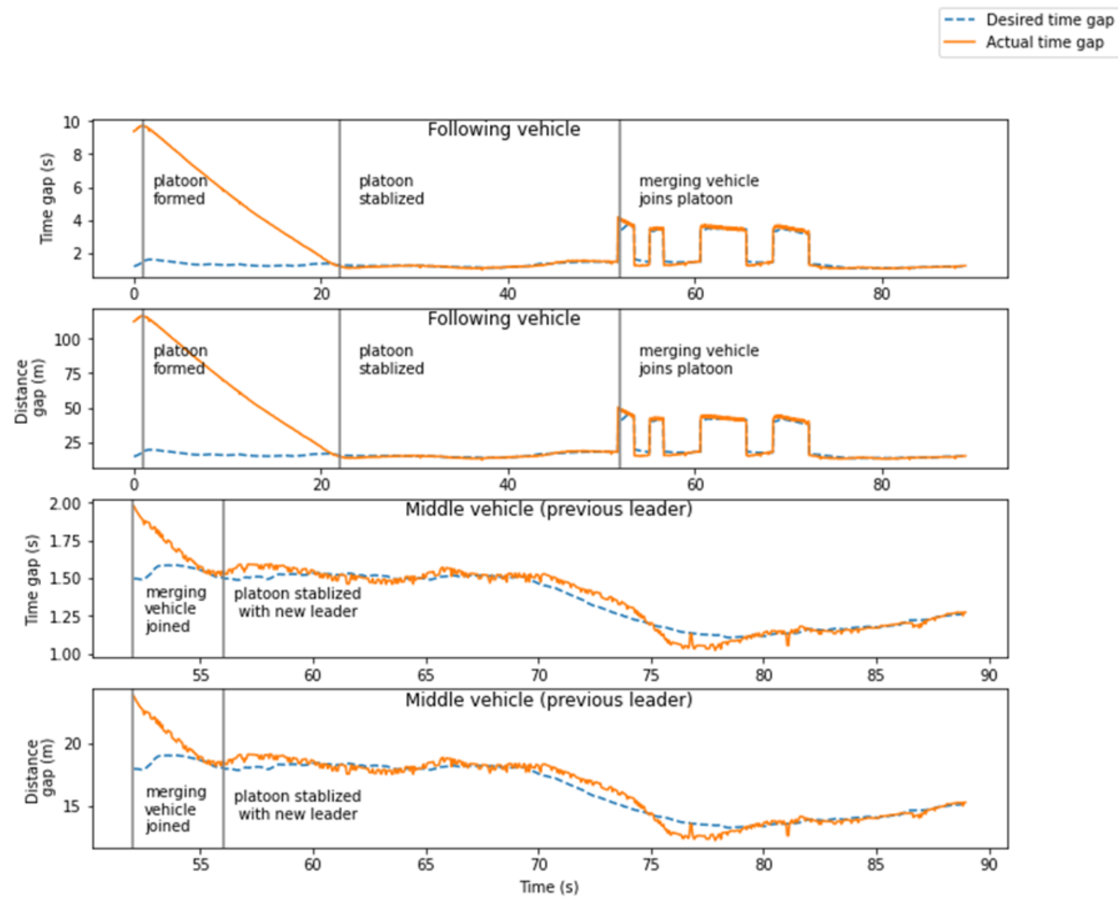


Figure 4.16: Cutin join to the front scenario gap comparison where the gap data of the second and third members of the platoon is presented.

leadership exchange process. The primary challenge here lies with the joining vehicle's ability to close the intra-platoon gap. Compare to the cut-in front, this joining sequence commences earlier and unfolds more smoothly. As shown in Fig.4.17 where the merge is completed within 6 seconds with minimal disturbance.

The time gap comparison plot, presented in Fig.4.18, provides additional insights. The last two members successfully achieve and maintain the desired gap, causing minimal disturbance to the platoon's operation. Importantly, because there is no occurrence of abnormal gaps, there is no APF leader adjustments. This further highlights the relative simplicity and efficiency of the rear-join scenario compared to its front-join counterpart.

In the third scenario, the cut-in join to the rear proves to be smoother and faster than the

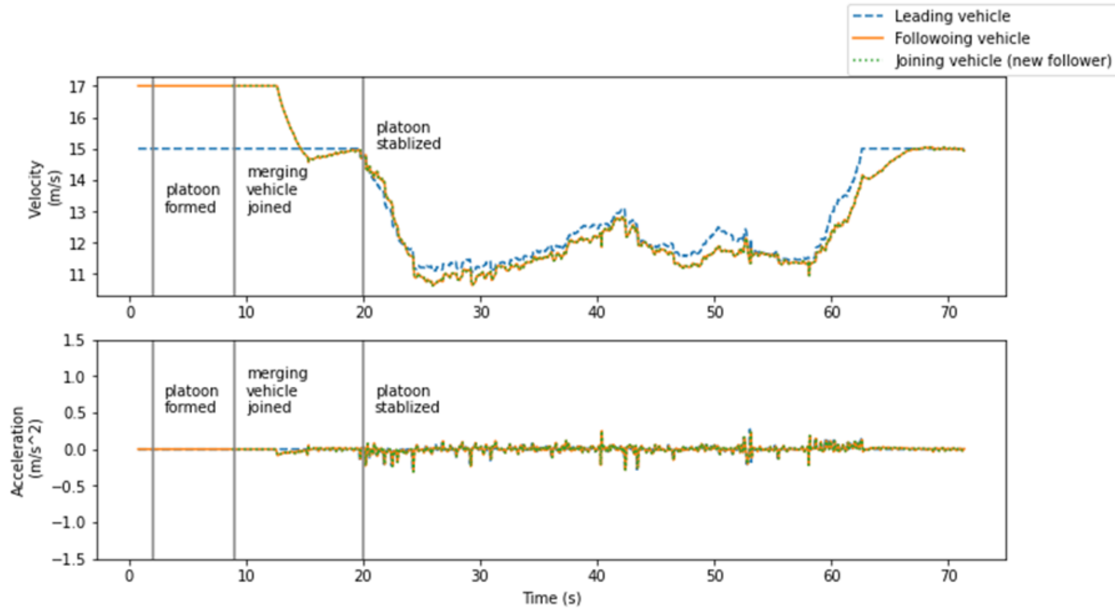


Figure 4.17: Cut-in join to the rear scenario trajectory results. The velocity of all three platoon members is presented in the top figure, and the acceleration of all three members is presented in the bottom figure.

cut-in join to the front, in line with the results from the digital twin simulation tests. Despite minor delays in the front-join scenario due to the leadership exchange process, the overall experiments were deemed successful. There is, however, a need for careful management of the computational resources for optimal performance during parallel tasks such as updating the platoon leader and regulating the intra-platoon gaps. The disparity in system performance under certain conditions, especially when compared with the simulation results, opens up avenues for future research, with a more detailed system-level analysis presented in section 4.3.

Scaled platooning tests

Drawing upon the prior study [31], which stated that the efficacy of the multi-lane platooning algorithm corresponds with the Market Penetration Rate (MPR). It is safe to conclude that a higher MPR often corresponds to larger platoon sizes and more efficient traffic flows. Therefore, the significance of larger platoon tests, specifically those involving four and five

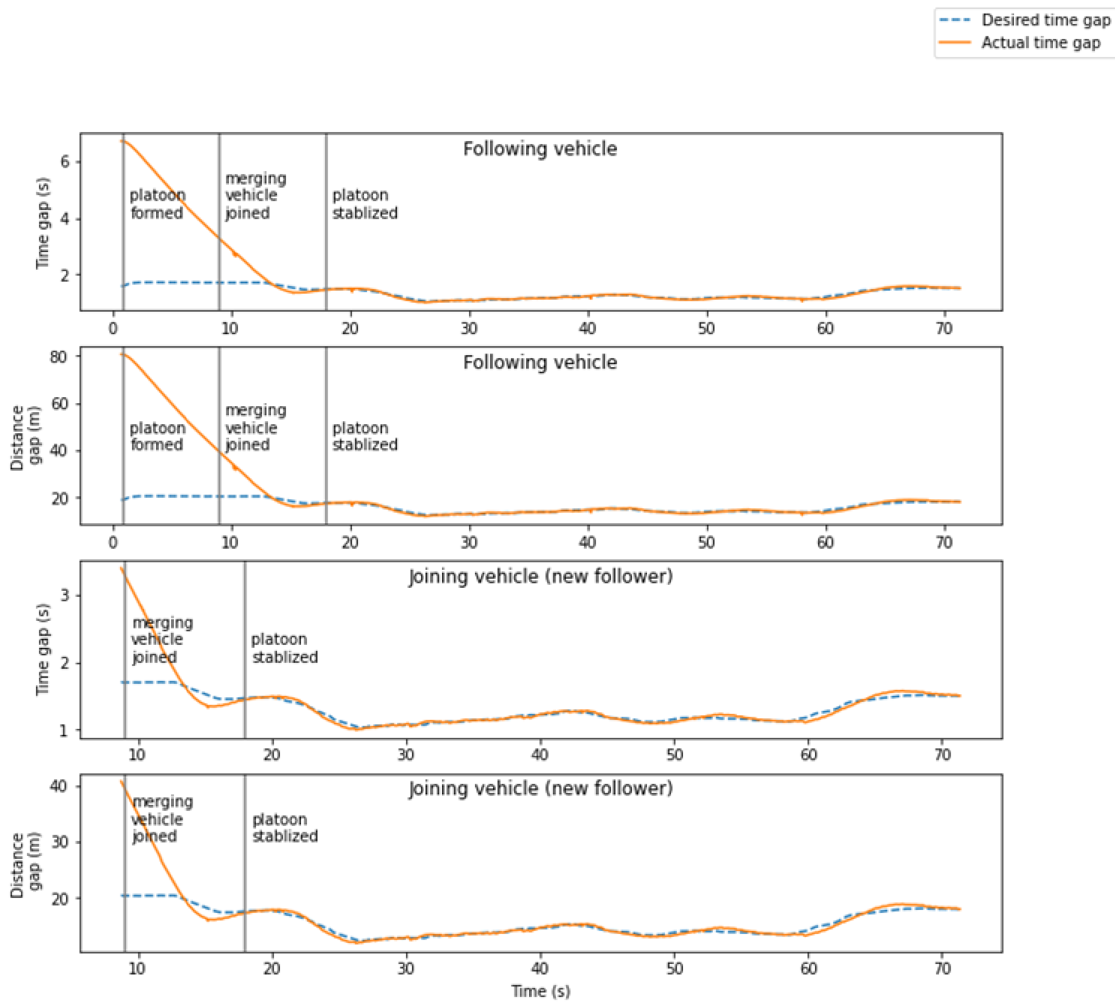


Figure 4.18: Cut-in join to the rear scenario gap comparison where the detailed trajectory data of the second and third members of the platoon is presented.

vehicles, is underlined by the complex practical conditions they simulate. These tests offer crucial insights into the algorithm's capacity to handle high MPR scenarios and facilitate smoother traffic flows. The selected scenarios for the four-vehicle and five-vehicle tests presented in Fig.4.19 and Fig.4.20, respectively, involve a cut-in join to the front, recognized as one of the most complex scenarios due to the leadership transition it necessitates.

The conclusions drawn from these tests are encouraging. Although the desired distance gap was purposely increased to account for potential oscillations and to grant a longer reaction time, the larger platoons successfully maintained consistent intra-platoon gaps. The vehicles displayed a capacity to adjust to these increased gaps, indicating a level of control

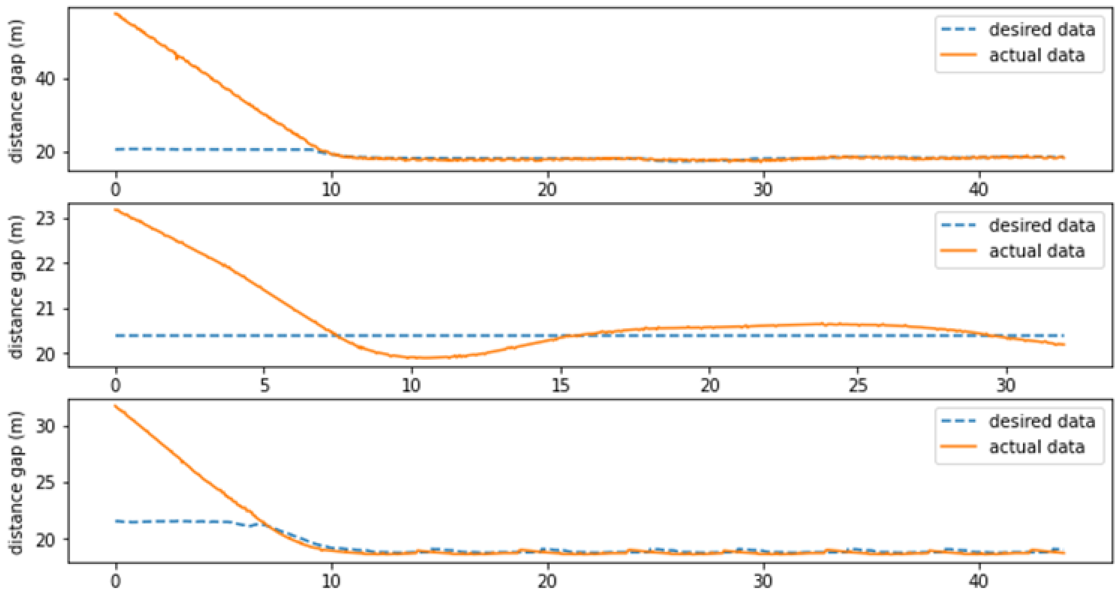


Figure 4.19: Cut-in join to the front scenario with four vehicles gap comparison where the gap data of all three platoon members (i.e., following vehicles) is presented.

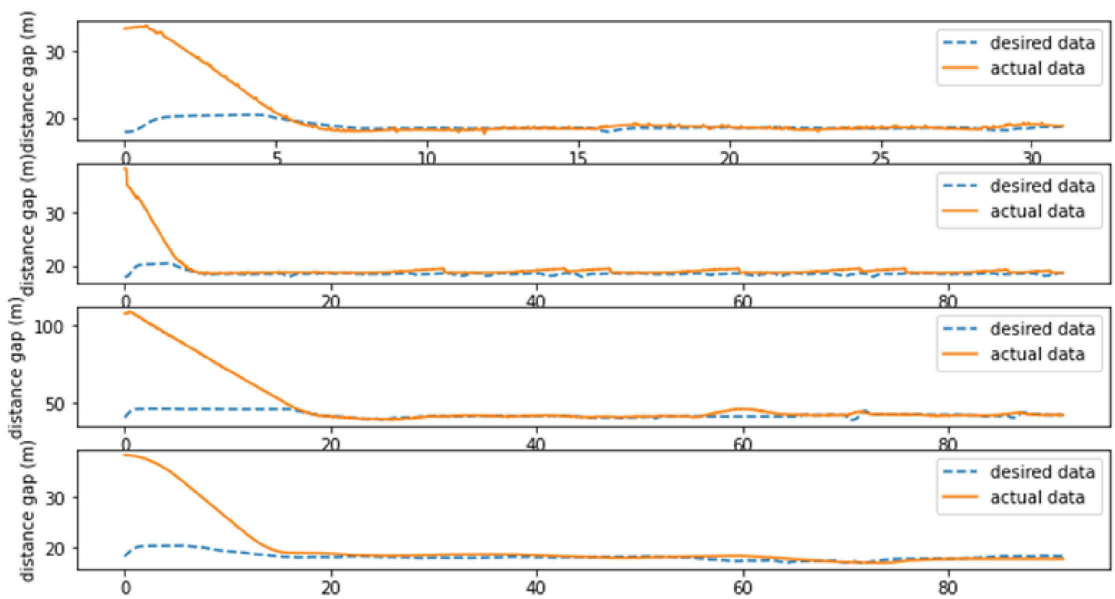


Figure 4.20: Cut-in join to the front scenario with five vehicles gap comparison where the gap data of all four platoon members (i.e., following vehicles) is presented.

over their individual drive-by-wire systems and an ability to navigate mechanical differences. Despite the augmented complexity and size of these larger platoons, the platooning field tests show performance comparable to previous tests with smaller platoons. In all scenarios, the cooperative behavior in Vehicle-to-Vehicle (V2V) intent sharing adhered to the algorithm's design, signaling the successful execution of the field experiment.

Repeated test results

Statistical analysis was performed to validate the algorithm's performance in repeated field tests, encompassing various leading and following vehicles across all scenarios. Safety aspects were closely examined, including merge time, maximum and minimum accelerations, and distance gap range. Additionally, the algorithm's compatibility with different vehicle brands was evaluated through tests with diverse lead and follow vehicles in the same scenarios. The summarized outcomes of the comprehensive statistical analysis are presented in Table 4.2.

Stability analysis

The algorithm's performance stability is a critical performance measure. It directly relates to the safety and usability of the multi-lane platooning algorithm. Though the FSM faced with state various switching challenges, results show no instance of failure. Meanwhile, statistical results can also capture the motion planning module's effectiveness. As shown in Table 4.2, the two-vehicle platooning scenario yields a constant average gap error of around $+2$ m (i.e., maintaining larger gaps), indicating a consistent gap regulation performance. Notably, the gap error value for two-vehicle tests is higher compared to the three-vehicle scenarios. This is due to the same parameter that was used for regulating gaps with different desired gaps. However, this tuning imperfection doesn't affect gap regulation's stability, as the behavior remains consistent throughout the tests.

For both three-vehicle tests (i.e., cut-in front and rear), the middle vehicle can follow the desired gap better than the trailing vehicle, as indicated by the max (min) gap error. As it is the direct follower of the leader with the most instant communication and the least amount of error. In addition, the time taken to merge is comparable for the middle vehicles in all three-vehicle tests, as they always start as platoon members. Besides, the third vehicle also performs similarly in terms of time gap error. These observations indicate that the gap

Table 4.2: Statistical Analysis Results

Following vehicle (for 3-vehicle platoons, with Lexus SUV and Chrysler minivan as followers)										
Scenario	**Time taken to merge (s)**	**Acceleration (m/s^2)**				**Distance Gap Error (m)**				**Desired time gap (s)**
		min	**max**	**var**	**Mean error**	**min**	**max**	**var**	**Mean error**	
Same-lane platoon	20.350	-0.450	0.450	0.004	0.008	-1.450	3.880	0.350	2.450	0.8
	51.350	-0.450	0.450	0.006	-0.035	0.830	3.790	0.380	2.250	
	55.850	-0.450	0.450	0.006	0.000	-1.070	5.630	0.820	2.480	
	31.050	-0.600	0.550	0.017	-0.001	0.060	4.006	0.420	2.450	
Cutin front	26.450	-0.460	0.450	0.005	-0.003	-0.580	0.710	0.116	-0.290	1.2
	28.500	-0.450	0.450	0.008	-0.003	-1.600	1.600	0.410	-0.100	
	25.350	-0.600	0.600	0.056	-0.005	-1.200	0.940	0.170	-0.032	
	24.100	-0.450	0.450	0.028	-0.005	-0.820	1.480	0.200	-0.050	
Cutin rear (joiner)	46.100	-0.450	0.450	0.024	0.001	-1.680	0.820	0.510	-0.290	1.2
	35.600	-0.450	0.450	0.007	0.007	-1.080	1.250	0.150	-0.034	
	45.750	-0.450	0.450	0.014	0.009	-1.490	2.118	0.520	-0.130	
	33.900	-0.450	0.450	0.004	0.001	-1.000	0.740	0.220	-0.120	
	40.100	-0.450	0.450	0.007	-0.006	-1.900	0.800	0.320	-0.120	
Middle Vehicle (for 3-vehicle platoons, with Lexus SUV and Chrysler minivan as followers)										
Cutin front	20.850	-0.450	0.450	0.004	-0.002	-1.660	0.600	0.210	-0.110	1.2
	18.150	-0.450	0.450	0.003	-0.002	-1.840	0.640	0.360	-0.120	
	16.250	-0.450	0.450	0.005	-0.001	-0.960	0.740	0.140	0.070	
	14.500	-0.450	0.450	0.029	-0.001	-0.780	0.590	0.130	-0.040	
	21.550	-0.450	0.450	0.006	-0.001	-1.700	0.700	0.160	-0.077	
Cutin rear	15.450	-0.450	0.450	0.002	-0.005	-1.770	0.900	0.180	-0.050	1.2
	19.300	-0.450	0.450	0.007	0.001	-0.930	0.930	0.140	0.043	
	18.850	-0.450	0.450	0.050	0.001	-1.300	0.920	0.150	-0.060	
	14.750	-0.450	0.450	0.003	0.001	-0.990	0.940	0.100	-0.021	
	24.600	-0.450	0.450	0.002	-0.002	-2.200	0.780	0.320	-0.130	
Platoon Leader (with Lexus SUV and Chrysler minivan as followers)										
Same-lane platoon		-0.450	0.450	0.002	-0.001					0.8
		-0.450	0.200	0.002	-0.003					
		-0.400	0.600	0.002	-0.001					
		-0.440	0.300	0.001	-0.001					
Cutin front (joiner)		-0.300	0.300	0.002	-0.003					1.2
		-0.440	0.310	0.001	0.002					
		-0.440	0.250	0.002	0.001					
		-0.440	0.290	0.001	0.002					
		-0.450	0.260	0.002	-0.002					
Cutin rear		-0.420	0.300	0.001	0.000					1.2
		-0.450	0.450	0.002	0.001					
		-0.450	0.390	0.001	0.000					
		-0.440	0.370	0.001	0.001					
		-0.450	0.150	0.001	0.000					

regulation is operating as expected in all three-vehicle tests.

Safety analysis

Acceleration directly reflects harshness, safety, and riding comfort. Among all the test data, the maximum acceleration is 0.6 m/s2, which only happens several times. Most acceleration and deceleration values are within 0.45 m/s2, which is significantly below the uncomfortable value by most studies [6, 51]. Meanwhile, though the CARMA platform allows emergency braking, the maximum deceleration is below -0.6 m/s2, far from invoking safety concerns.

In terms of distance gap, statistical data indicates that the mean error is under $+1$ meter in all three-vehicle scenarios. The one-meter average error between the desired gap and the actual gap is considerably smaller than the desired distance gap of 15 meters. The maximum gap error is mostly within $+2$ meters. Therefore, results indicate the algorithm can maintain an accurate and consistent gap during platooning. The major contributing factor is sharing planning trajectory information with other members, which is a key advantage of the multi-lane platooning algorithm. Therefore, it is safe to conclude that sharing the intend-embedded trajectory is crucial and effective when regulating a gap repeatedly within a platoon.

4.4.5 Closed-track test overview

Overall, the algorithm is accountable across all vehicles. In particular, two different vehicles were used as the leader during the cut-in-front and the cut-in-rear experiments. For all three-vehicle tests, we swapped follower sequences and vehicles. Notably, no data shows abnormal behavior even with repeated vehicle role exchange. Therefore, it is safe to conclude that the functionality of the platooning algorithm and the testing vehicles are stable. In addition, results show the multi-lane platooning algorithm can operate across two vehicle brands (i.e., Chrysler and Lexus) and two vehicle types (i.e., Minivan and SUV), validating its compatibility and versatility.

In terms of the experiment, the success of comprehensive close-track and scaled platooning tests not only marks the successful integration of the multi-lane platooning algorithm but also showcases the algorithm's robustness, adaptability, and consistent performance in real-world scenarios. Additionally, the achievement under moderate speed with four and five participating vehicles underscores the algorithm's promising potential for scalable usage, demonstrating a promising advancement in the realm of automated driving systems.

On the other hand, the comprehensive testing pipeline utilized here, encompassing digital twin simulation testing, software-in-the-loop testing, hardware-in-the-loop testing, and close-track field experiments, offers a rigorous and robust evaluation of the multi-lane platooning algorithm. This multistage approach ensures the algorithm's performance and stability are thoroughly scrutinized under varying conditions, systematically progressing from controlled virtual environments to real-world settings. By tackling the challenges at each stage, the algorithm's performance, efficiency, and reliability are validated. This comprehensive testing pipeline showcases the commitment to ensuring safety, dependability, and excellence in the field of automated driving systems.

4.5 Conclusion and future research

Cooperative Driving Automation (CDA) technologies hold the promise of enhancing road capacity, consistency in travel times, and overall traffic performance. In contrast to the existing ad hoc CACC algorithms, this study presents a comprehensive testing pipeline including the digital twin simulation testing, software-in-the-loop testing, hardware-in-the-loop testing, and close-track field testing as well as real-world experimental outcomes of a multi-lane platooning algorithm [31] that employs organized behavior through a hierarchical control structure designed for complex scenarios. The experiments were carried out via a comprehensive testing approach, where each stage incrementally presents more realistic challenges and risk factors. Numerous repeated tests across each stage ultimately verify the algorithm's functionality and stability in a controlled setting that simulates real-life highway infrastructures. Based on the experiment outcomes, several vital observations and implications can be distilled as follows:

- The multi-lane platooning algorithm has been integrated with an existing C-ADS successfully and verified by the simulation results under digital twin settings and the extensive real-world experimental tests.

- Throughout the comprehensive testing pipeline, the previously proposed multi-lane platooning algorithm can efficiently guide the behavior of platoon members and external C-ADS vehicles (to join a platoon) under each superstate and precisely switch between the FSM under different scenarios.

- The gap regulation algorithm can effectively maintain the desired time gap. Moreover, the results of all tests among the producer indicate that it is necessary to consider both the immediately preceding vehicle and the platoon leader's (i.e., APF algorithm) during the platooning process.

- With intent/trajectory-sharing for cooperative driving automation, C-ADS-equipped vehicles can be proactive about others' intentions, conducting safe and efficient multi-lane operations at a moderate speed (i.e., up to 35 mph).

- The proposed algorithm adopts the C-ADS software platform and successfully operates in parallel with other planning applications. Deployment experiences indicate that the tested algorithm consistently performs across multiple C-ADS-ready vehicle platforms from different manufacturers.

- The proposed algorithm successfully operates with various platoon sizes. Scaled test results indicate that the proposed algorithm is effective and efficient even when the platoon size reaches four or five vehicles. This further demonstrates the algorithm's capability to improve the traffic system's performance under higher MPRs.

- The comprehensive testing pipeline, marked by its thoroughness and adaptability, has proven its value in assessing the functionality and integration of the proposed multi-lane platooning algorithm, establishing its potential as a robust framework for verifying and validating future ADS vehicle technologies.

Chapter 5

Traffic Regulation-aware Path Planning with Regulation Databases and Vision-Language Models

This chapter presents a parallel development and testing framework that integrates traffic regulation compliance into automated driving systems (ADS), enabling these systems to interpret and follow traffic laws within a real-time decision-making context. The framework combines physical and virtual environments, scenario engineering, and coordinated operations to provide comprehensive validation of ADS under diverse conditions. A key feature is the regulation-aware path planning framework, which uses RGB camera inputs alongside a vision-language model (VLM) to generate descriptive text about the driving environment. This information, paired with a machine-readable regulation database, guides ADS in making lawful and contextually appropriate decisions, thus enhancing driving safety and compliance.

This ADS regulation project involves simulation testing and SIL within the parallel validation process. Note that while experiments were conducted using the ADS vehicle, due to safety and legal constraints, the vehicle was fully controlled by a human driver, and regulation-related outcomes were displayed on a webUI. As such, this phase qualifies as SIL rather than HIL, since automation functions were not engaged during testing. Validation of this regulation-aware framework occurs across both simulated and controlled real-world settings. Simulated scenarios, carefully engineered to reflect complex, multi-agent traffic conditions, test the ADS's compliance with various legal challenges, such as speed limits, pedestrian zones, and yielding requirements. Real-world tests further evaluate the framework's practical effectiveness, demonstrating how vision-language models (VLMs) support integrated detection, reasoning, and planning within a legal context. Through a combination of regulation-aware path planning, scenario engineering, and SIL testing, this approach illustrates the framework's scalability and adaptability, marking a significant step toward reliable, socially responsible autonomous driving systems that enhance public trust and safety.

5.1 INTRODUCTION

With the continuous advancements in Automated Driving Systems (ADS) technology, the goal of achieving fully automated mobility is becoming increasingly attainable. Recent developments in ADS research and testing have showcased substantial improvements in sensor-based perception, vehicle control, and decision-making algorithms [32]. As these technologies evolve, it is important to ensure that autonomous vehicles (AVs) comply with laws and regulations for their safe and efficient operation. Many developers recognize that legal compliance plays a key role in the public acceptance and operational success of AVs. However, despite the growing focus on accountability, there has been a noticeable gap in current regulatory efforts, particularly in the absence of a machine-readable database designed specifically for ADS software and the lack of comprehensive ADS-related regulations within different jurisdictions.

In response to this gap, the US Federal Highway Administration (FHWA) initiated the development of a prototype data framework for traffic regulations in 2021 [24]. This framework aimed to bring together various stakeholders involved in traffic regulation to start discussions on developing voluntary specifications. These voluntary specifications are vital for ensuring that infrastructure owners and operators, ADS developers, and technology providers operate on a standardized foundation. On the research front, recent efforts have been made to integrate legal constraints into the ADS decision-making and path-planning process. Building on these efforts, our proposed regulation-aware path-planning framework integrates a comprehensive ADS traffic law database with a novel planning process that uses a finite state machine and cost functions to evaluate multiple plans based on safety, comfort, and legal compliance. Moreover, a Vision-Language Model (VLM) is integrated to directly interpret driving conditions, eliminating the need for specialized models like object detection or trajectory prediction, which commonly rely on domain-specific knowledge and may struggle in certain conditions. This enhances the framework's robustness and adaptability to complex traffic situations.

This paper's contribution is two-fold: (1) proposing a practical path planning framework that feasibly integrates with VLMs for traffic regulation-aware planning (as compared to direct VLM- or LLM-based driving which has been proved as unreliable); and (2) perform simulation and real-world vehicle experiments to assess the framework's performance and understand the potential and challenges of VLM in this critical use case for future reference.

5.2 RELATED WORKS

Recent research highlights the need for ADS systems to comply with traffic laws, recognizing the critical role of regulations in ensuring the safe and effective operation of autonomous vehicles. Ilková et al. [35] provide an overview of AV legal frameworks in Europe and the United States, emphasizing the importance of understanding how legal provisions apply to ADS and the need for harmonizing regulations across jurisdictions. Bakar et al. [2] similarly stress the necessity of unified global guidelines to enhance roadway safety, as inconsistent traffic laws pose challenges for AV developers. While these studies emphasize legal accountability in

ADS operations, current regulatory efforts have yet to develop machine-readable databases to support ADS decision-making fully. LEE and Hess [40] also highlight progress in addressing national regulations but note the complexities in adapting AV software to comply with varying legal requirements across regions.

Efforts to integrate legal constraints into ADS decision-making and path planning are gaining momentum. Zhang et al. [106] introduced a framework using formal methods to detect violations of scenario-based driving rules, providing a rigorous approach for ADS to comply with traffic regulations. Cho et al. [15] proposed a deep learning model that predicts vehicle paths while assessing regulatory compliance. De Vries et al. [19] incorporated traffic regulations into a cost function within a local model predictive contouring control (LMPCC) system for real-time motion planning. Despite these advances, many methods are limited by focusing on a narrow set of regulations and lacking jurisdictional awareness. Additionally, frameworks like AVChecker and LMPCC often depend on specialized detection models and act on pre-selected sets of traffic regulation, reducing their real-world applicability. It is necessary to utilize the comprehension and reasoning capabilities for such complex tasks.

5.3 METHODOLOGY

5.3.1 Framework Overview

The proposed framework, illustrated in Fig.5.1 follows a structured approach to integrating traffic laws into real-time ADS decision-making through both mission (strategic) and motion (tactical) planning stages. In the mission planning phase, the Finite State Machine(FSM) [57] evaluates potential driving plans based on real-time perception data. To enhance regulation compliance, a VLM named LLaVA [46], is integrated to work interactively with the FSM, interpreting the driving scene and advising on relevant traffic regulations based on FSM status, the interactive process between the FSM and VLM, highlighted in orange in Fig.5.1.

After available plans are generated in the mission planning stage, the process moves to

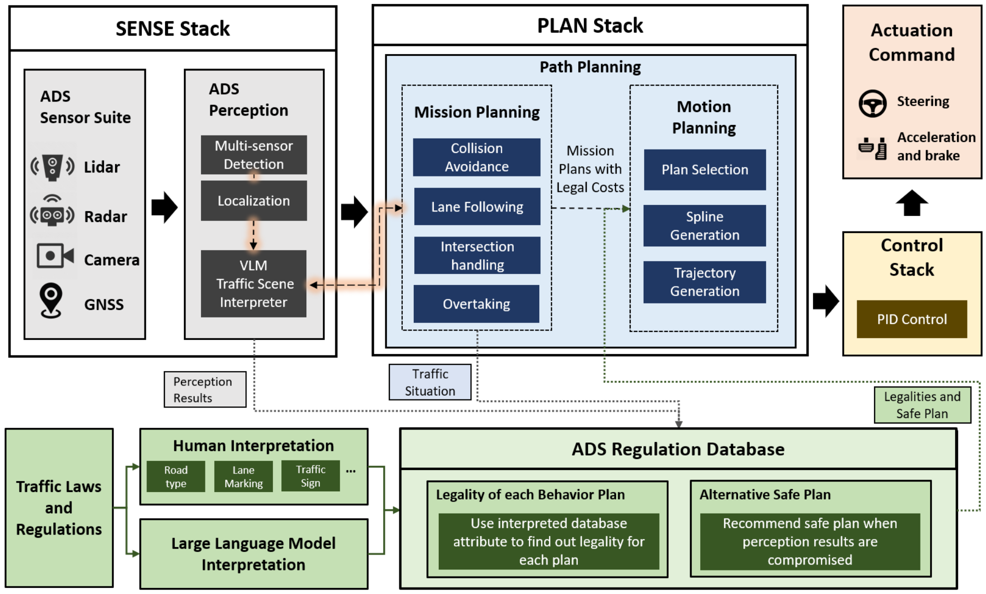


Figure 5.1: The overview of the path planning framework where the path planning modules are highlighted in blue, the interaction between VLM and mission planning is highlighted in orange, and the ADS regulation database is highlighted in green.

motion planning, where a detailed trajectory is created from key waypoints using cubic spline interpolation [58]. A cost function is then applied, taking into account factors such as comfort, economy, safety, and legality. This ensures that the selected plan not only complies with traffic laws but also optimizes overall driving performance, providing a scalable solution for real-time ADS planning.

5.3.2 ADS Regulation Database

Traffic regulations in the U.S. vary by state and locality, posing challenges for Automated Driving Systems (ADS). In the initial phase of this study, common elements from the Uniform Vehicle Code (UVC) [68], California Vehicle Code (CVC) [84], Los Angeles County Vehicle Code (LACVC) [50], and City of Los Angeles Vehicle Code (CLAVC) [16] were analyzed to create a foundational ADS database. This database transitions traffic legislation into a machine-readable format, facilitating the accommodation of diverse regulations across different jurisdictions.

To further this effort, the study develops a state-level ADS regulation database adaptable to local entities. Vehicle codes are analyzed to extract relevant metadata, including effective dates, legislative locations, reference numbers, and descriptions. Each code is assigned an ID and categorized as "Condition" or "Result," with exceptions where necessary, converting them into machine-readable commands. During the database generation, traffic codes are broken down into attributes, as shown in Table 5.1, allowing the perception module to assess the legality of current driving conditions based on these attributes. Table 5.1 illustrates the workflow, highlighting how attributes are derived from the raw regulation text.

Table 5.1: An example regulation of the traffic regulation database

Code Text	**Condition**	**Result**	**Legality**	**Attribute: Road Type**	**Attribute: Max Speed**
"A person who drives a vehicle upon a highway at a speed greater than 100 miles per hour is guilty of an infraction punishable, as follows:..."	"A person who drives a vehicle upon a highway at a speed greater than 100 miles per hour."	"is guilty of an infraction."	FALSE	Highway	100 mph

5.3.3 VLM Integration

Planning is a critical component of ADS software architecture, particularly regarding legal regulations and safety. Most conventional ADS frameworks [21, 94, 104] use a two-stage planning strategy: strategic or mission planning, followed by tactical or motion planning. A key limitation of these frameworks is their reliance on multiple specialized models to analyze regulation-related traffic conditions, requiring various detection models to interpret the traffic scene. The proposed work addresses this by introducing a VLM to provide a high-level description of the traffic environment, including key elements like intersections and traffic lights, offering a generalized, one-step solution for regulation-oriented environment assessment and subsequent planning decisions.

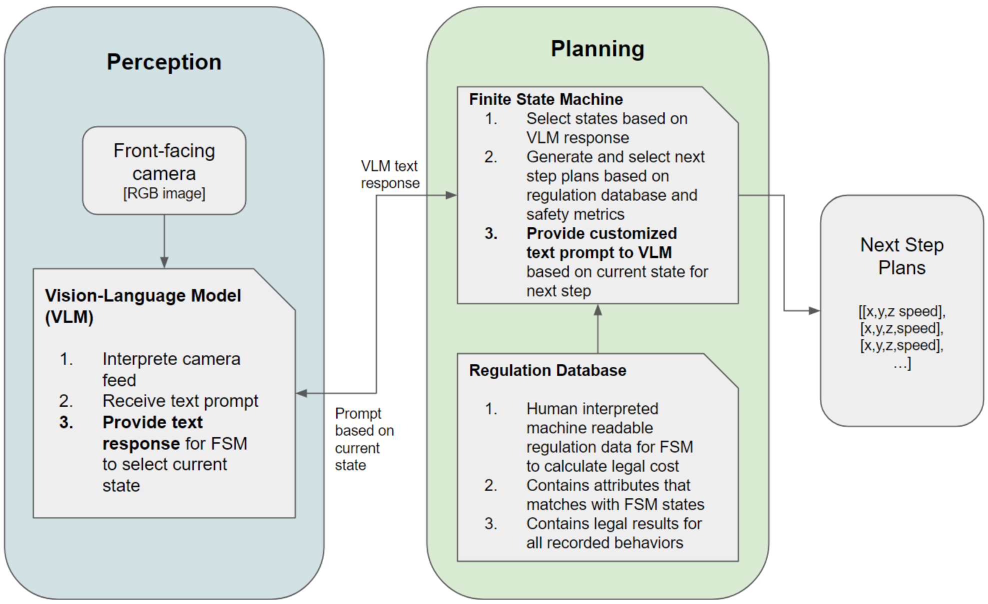
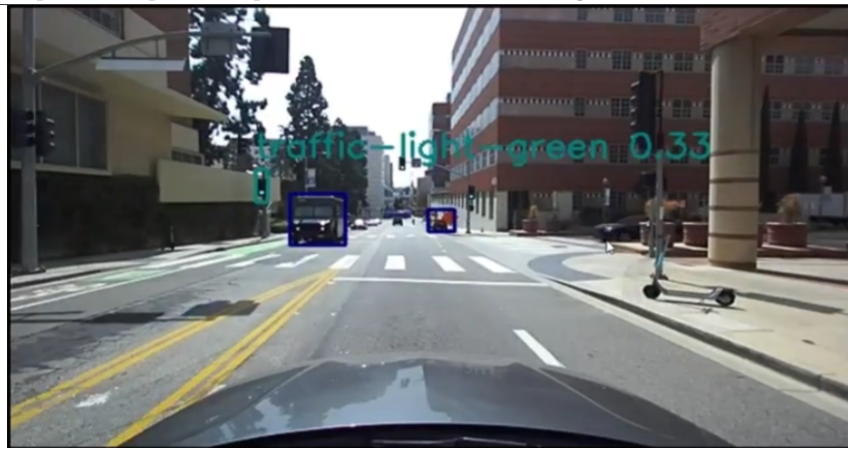


Figure 5.2: Framework structure of the VLM integrated path planning framework.

To integrate the VLM into the perception system, it is crucial to identify its upstream and downstream modules. As shown in Fig.5.2, the VLM is incorporated into the ADS perception process, receiving camera frames to assess the driving environment. The resulting text summaries of regulation-related traffic conditions are sent to the FSM, the ADS planning module responsible for decision-making. This process, commonly named “instruction running”, uses instruction-tuned large language models (LLMs) like LLaVA, a multimodal model combining a vision encoder and LLM for enhanced visual and language understanding [46]. LLaVA leverages GPT-4 [69] to generate multimodal instruction-following data and has shown strong conversational abilities, particularly in synthetic multimodal tasks. An input-output example of the LLaVA is summarized in Table 5.2.

The integration of LLaVA into the ADS framework aims to enhance the system's understanding of the driving environment by using LLaVA's text responses, based on front-facing camera input, to select relevant traffic regulations from the ADS regulation database. This

Table 5.2: An example of LLaVA Inferring an Image with Text Prompt

Visual input example: Sample Frame from Front-facing Camera on the ADS Vehicle	
	
Text Prompt	**LLaVA Response**
what traffic situation do you observe from this image?	The image depicts a vehicle approaching an urban intersection where the traffic light is green, indicating it can proceed. Several vehicles ahead are detected and outlined in blue boxes. The scene includes a pedestrian crosswalk with no pedestrians present, a bicycle lane on the left, and a sidewalk with a scooter on the right. The surrounding environment suggests a typical city street setting.

database classifies regulations by attributes, with values representing specific regulation content. When LLaVA generates keywords such as “intersection” or “incoming vehicles,” these are compared against the regulation “condition” keys in the database, ensuring the appropriate regulations are selected.

As illustrated in Fig.5.2 and Table 5.2, LLaVA requires both a front-facing camera frame and a text prompt to focus on the most relevant features of the scene. The prompt is crucial for directing LLaVA’s attention to the key aspects of the current driving scenario, and it must be contextually related to the situation. To achieve this, the FSM, the core of the strategic planning phase, is coupled with LLaVA prompts. The FSM defines various states and prescribes actions based on the environment, guiding LLaVA’s inference. For instance, in the “Lane-Following” superstate, prompts differentiate between states like “Intersection Handling” or “Overtaking,” with instructions such as “Examine the current driving scenario, look out for intersections or obstacle vehicles” guiding LLaVA’s focus. Table 5.3 lists text prompts for each superstate and their transitions. Additionally, an “Emergency Handling” superstate, triggered by safety-critical situations, overrides other states but operates independently from

LLaVA, relying on distance measurements rather than visual inference.

In summary, integrating LLaVA with the regulation database enables ADS to process front-facing camera frames and use the FSM to guide LLaVA's inference based on driving scenarios. This approach ensures accurate identification of relevant traffic regulations for path planning. By using FSM-driven prompts at each state, LLaVA helps the ADS maintain awareness of the current driving environment, effectively summarizing complex real-world scenarios for regulation compliance.

Table 5.3: Superstate Corresponding Text Prompts and Transitions

SuperStates	**LLaVA Text Prompt**	**Possible Next Superstates**
Lane Following	"Examine the current driving scenario, look out for intersections or obstacle vehicles."	Lane Following; Intersection Handling; Overtaking
Intersection Handling	"Examine the current driving scenario, check of the ego vehicle is still facing an intersection."	Intersection Handling; Lane Following
Overtaking	"Examine the current driving scenario, check nearby lane occupation conditions, and look out for intersection."	Overtaking; Lane Following; Intersection Handling

5.3.4 Cost Function

The cost function consists of mathematical formulas used to measure the attractiveness of various potential travel plans, considering the vehicle's current state and planned trajectory. Utilized in decision-making procedures, this function evaluates different plans according to a uniform standard, selecting the most suitable plans with the lowest cost for the next step. The cost function takes into account factors including the plan's legality, safety and comfort, and the distance to the global navigation goal following the execution of the plan, where each cost is calculated based on the current vehicle driving status, such as location, speed, acceleration, jerk, and legality. The final cost score is the sum of the scores across all these factors. It is worth noting that each plan has a final cost score once the detailed trajectory is generated, but legal constraints are also considered during the mission planning process, assigning a legal cost to mission plans.

5.4 EXPERIMENTS

The experiments in this study consist of two phases: simulation and real-world tests. In the simulation phase, the goal is to validate the framework's functionality with LLaVA as a perception tool to ensure regulation compliance in ADS vehicles. These tests assess the entire ADS cycle, from perception to actuation, focusing on the legality and safety of the system's trajectories. The real-world tests, on the other hand, evaluate system performance under real driving conditions, where LLaVA's inference efficiency is crucial for real-time decision-making.

5.4.1 Simulation Test

The simulation tests in this study use UCLA's OpenCDA platform [102], an open-source and full-stack simulation framework integrating key ADS components like perception, localization, planning, control, and V2X communication. The scenario tests multiple traffic regulations at once, reflecting real-world complexity. The ego vehicle starts in the rightmost lane of an intersection with a 35 mph speed limit, encounters a cyclist, makes a right turn, and continues through a school zone with a reduced 25 mph limit, requiring it to navigate various challenges and comply with traffic regulations. In the simulated scenario, three traffic regulations co-exist:

- **Cyclist avoidance:** A driver of a motor vehicle shall not overtake or pass a bicycle proceeding in the same direction on a highway at a distance of less than three feet between any part of the motor vehicle and any part of the bicycle or its operator. California Vehicle Code Section 21760 [65].

- **Right turn on red:** Except when a sign is in place prohibiting a turn, a driver, after stopping as required by subdivision (a), facing a steady circular red signal, may turn right. California Vehicle Code Section 21453 [64].

- **School Zone:** A 25 miles per hour prima facie limit in a residence district, on a highway with a posted speed limit of 30 miles per hour or slower, when approaching, at a distance of 500 to 1,000 feet from, a school building. California Vehicle Code Section 22358 [66].

5.4.2 Real-World (HIL) Test

Real-world testing evaluated the regulatory data framework's performance, focusing on its adaptability and compliance with traffic regulations in dynamic conditions. The test scenario occurred at a UCLA campus intersection, where the ego vehicle, approaching from the eastbound direction, made a right turn when conditions allowed. Along the route, the ADS system identified key traffic elements, including signs, vehicles, and vulnerable road users, all tied to regulations. The VLM's ability to recognize lanes and speed limits was tested to ensure compliance and contribute to a legal driving plan. Due to regulation and safety constraints, a human driver was operating the vehicle while the perception and planning modules remained fully functional. The ADS vehicle processed input from a front-facing camera and GNSS unit, with the VLM providing real-time text-based inferences to guide planning. A WebUI displayed FSM information, detection outputs, and VLM responses, with static planning not visualized due to the human driver. The ADS vehicle and the WebUI interface are presented in Figure 5.3.

5.5 RESULTS

This section presents results from simulation and real-world testing of the proposed framework for ADS regulation compliance. Simulation tests validate overall functionality, while real-world

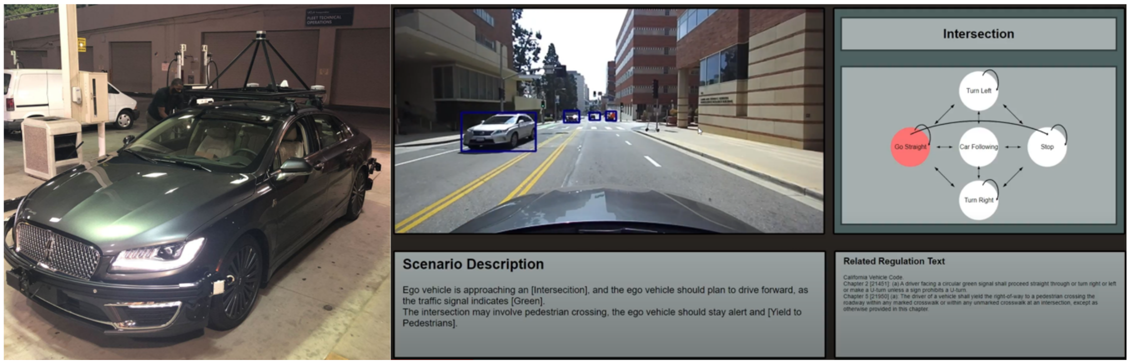


Figure 5.3: ADS vehicle and the real-time WebUI interface for VLM and FSM visualization.

tests focus on the system's ability to handle complex traffic scenarios and make real-time, regulation-based decisions. The framework's efficiency is also assessed, confirming its ability to manage real-world driving tasks while ensuring legal and safe operation.

5.5.1 Simulation Testing

The simulation results demonstrate the robustness of the proposed path planning framework. In a complex use case involving three traffic regulation scenarios, the ego vehicle successfully overtakes a cyclist, stops at a red light before turning, and adjusts speed in a school zone. The VLM's inference and the framework's path planning were also tested against simulated human-driven traffic, showing effective performance.

Overtaking Cyclist

Two key events occur in this scenario: the ego vehicle detects the cyclist ahead and initiates overtaking, marked by the first moment, and completes the maneuver by returning to its original lane. This scenario presents a dual-layered challenge in planning. The corresponding simulation snap is presented here in Figure 5.4.

In this scenario, the ego vehicle decides to overtake a cyclist ahead, taking into account traffic regulations and the speed of surrounding vehicles. It verifies the legality of overtaking based on dashed road markings and ensures the lane is wide enough to maintain the required

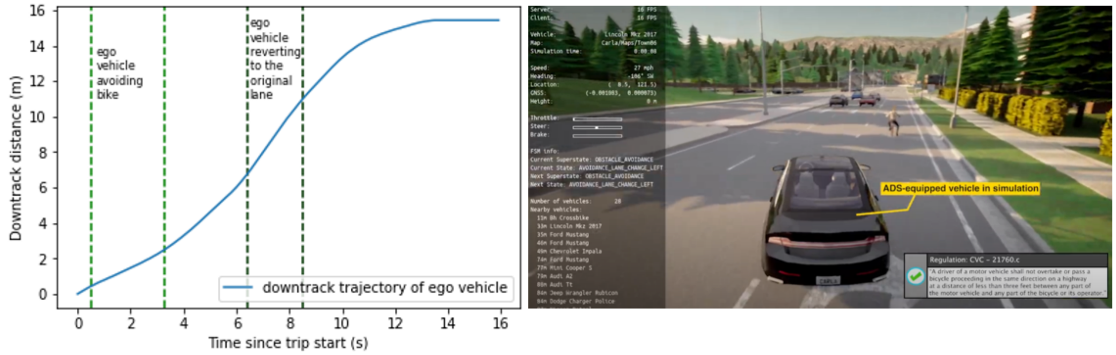


Figure 5.4: Trajectory plot and the simulation screenshot when ego ADS vehicle overtaking a cyclist.

3-foot clearance. Once these conditions are met, the framework transitions from the lane-following state to the overtaking state. During the maneuver, the vehicle adheres to speed limits and maintains a safe distance. After completing the overtaking, the vehicle returns to the lane-following state. This complex maneuver demonstrates that the VLM accurately recognizes the cyclist, and the FSM effectively guides the VLM to assess the target lane conditions, showcasing the framework's coherent functionality across the VLM and FSM.

Right Turn on Red

Making a right turn on red poses a complex challenge for automated driving. The ego vehicle must detect the red light, stop at the line, and then proceed with the turn. This stop-and-check process is marked by two red dashed lines, as shown in Figure 5.5.

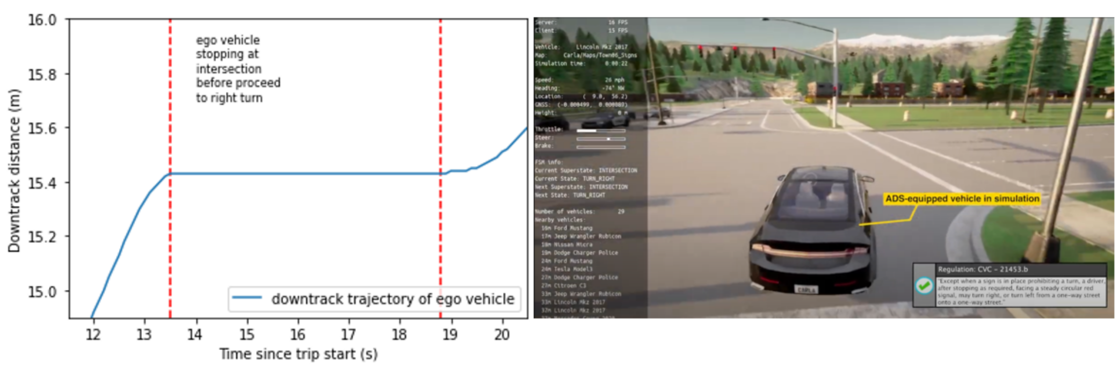


Figure 5.5: Trajectory plot and the simulation screenshot when ego ADS vehicle making a right turn during red light.

The ego vehicle must first confirm the legality of a right turn on red, as regulations vary by region. This requires the automated driving system to reference a comprehensive traffic rule database for the jurisdiction. Once legality is established, the vehicle ensures safety by adhering to the California Vehicle Code (CVC) and coming to a complete stop before turning. Using its sensors, the vehicle assesses the environment for oncoming traffic and proceeds only when safe. The process transitions from lane-following to intersection-handling, with the path planning framework checking for prohibitive signs before authorizing the turn if conditions allow. Compared to previous regulations, this scenario involves recognizing multiple traffic signs, further validating the VLM's capability in understanding complex environments, generalizing its ability to comprehend widely different objects, and demonstrating its seamless integration with the FSM.

Variable Speed Limit

In this scenario, managing variable speed limits in school zones poses a significant challenge for automated driving systems, as they must anticipate upcoming zones. The ego vehicle, operating in the lane-following superstate, detects the school zone and begins decelerating early, marked by a red vertical line. The accurate detection of such zones—either through visual recognition or HD-map annotation—is crucial to ensure pedestrian safety, particularly for children. School zones require a controlled reduction in speed, and the vehicle must reach the correct speed limit by the time it enters the zone, ensuring smooth driving without abrupt changes. The yellow dashed vertical line shows the vehicle maintaining reduced speed through the zone, with the framework calculating proximity and speed limits accurately despite the regulation having a vagueness score of one. The system effectively adjusts the vehicle's speed and continues the route while complying with the regulation.

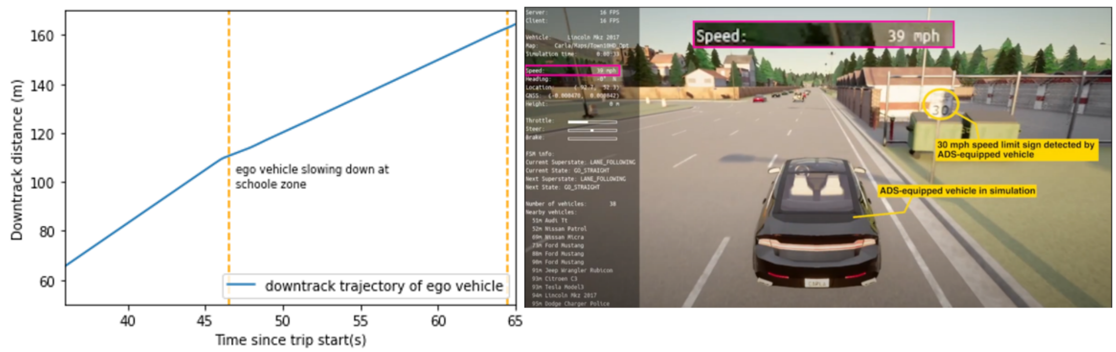


Figure 5.6: Trajectory plot and the simulation screenshot when ego ADS vehicle adjusting speed for areas with different speed limit regulations.

5.5.2 Real-World Testing

Real-world testing introduces greater complexity by incorporating actual traffic signs and dynamic traffic, including vulnerable road users (VRUs). Unlike simulated scenarios, this environment is more unpredictable, requiring the VLM to identify and interpret key traffic elements accurately. The VLM must process real-time data and provide precise inferences for the ADS to generate lawful and feasible driving plans. The system's ability to safely and legally navigate real-world conditions depends on the VLM's capacity to capture and understand the relevant traffic environment. To demonstrate LLaVA's effectiveness in identifying critical traffic scenarios, Table 5.4 lists the key elements detected and their corresponding inference responses.

As shown above, Table 5.4 summarizes the VLM's performance in identifying key traffic scenarios, which is crucial for the framework's compliance with traffic regulations. Results show the VLM successfully detected most scenarios except for the "end road work" and "stop here on red" signs. Overall, it correctly identified 7 out of 9 key scenarios, demonstrating promising performance for most driving conditions. In addition, the tested VLM excels in recognizing dynamic elements like vehicles, pedestrians, cyclists, and intersections but struggles with less common or text-heavy signs, especially in zero-shot inference. Despite these limitations, results indicate strong potential for real-time traffic scene comprehension through VLM.

Table 5.4: VLM Detection and Response in Key Traffic Scenarios

Meaningful Traffic Scenario	**VLM Detection Result**	**VLM Response**
Nearby intersection	Detected	"Ego vehicle is approaching an intersection…"
Yield for pedestrian	Detected	"The intersection appears to have pedestrian crossing, the ego vehicle should stay alert and yield to pedestrians."
Road work warning sign	Detected	"There is a visible warning sign: Road Work Ahead."
Speed limit sign	Detected	"The speed limit sign indicates 20 mph maximum speed."
Traffic light - red	Detected	"There is a visible red traffic light in sight."
Traffic light - green	Detected	"There is a visible green traffic light in sight."
Bicycle lane	Detected	"There is a visible bicycle lane."
End road work warning sign	Miss Detected as road work sign.	"There is a visible warning sign: Road Work Ahead."
Stop here on red sign	Miss Detected as stop sign.	"There is a visible stop sign."

VLM Efficiency

The inference speed of the tested VLM is crucial for real-time ADS planning, as it directly affects the system's responsiveness and overall performance. As part of the perception module, the VLM's speed dictates the frequency at which the path planning framework operates, with other ADS modules running at a minimum of 5Hz. Ensuring the VLM processes data quickly enough is essential for maintaining system reliability and safety. Image resolution impacts VLM performance, with higher resolutions offering more detail but requiring more processing time. Additionally, different model backbones present trade-offs between accuracy and computational efficiency. A comparison of these models, balancing inference capability and speed, is shown in Table 5.5, highlighting the key factors for real-time ADS applications.

The above Table 5.5 indicates inference frequencies for different LLaVA [46] model quantizations using 1080p frames from both simulation and real-world tests. The LLaVA-

Table 5.5: LLaVA [46] inference frequency with different models and different quantization

Model	Quantization	Inference Frequency
LLaVA-1.5-7B	4 bit	2 Hz
LLaVA-1.5-7B	8 bit	0.5 1 Hz
LLaVA-1.5-13B	4 bit	lower than 2 Hz
LLaVA-1.5-13B	8 bit	lower than 0.5 Hz

1.5-7B model with 4-bit quantization consistently performs best at 2 Hz, while the 8-bit version ranges between 0.5 and 1 Hz. The LLaVA-1.5-13B model with 4-bit quantization occasionally reaches 2 Hz but typically runs slower, with its 8-bit version falling below 0.5 Hz. Based on these results, the LLaVA-1.5-7B with 4-bit quantization is preferred for its reliable speed. Although lowering image resolution could increase speed, it sacrifices critical details like traffic light recognition, making high-resolution input with efficient quantization essential for real-time ADS planning. Overall, the VLM's inference speed meets the 2 Hz requirement for strategic planning tasks using the onboard computer with NVIDIA RTX 4070 GPU.

5.6 CONCLUSION

In conclusion, this study demonstrates the effectiveness of a novel, generalized ADS planning framework designed for traffic regulation compliance. Key advancements, tested in both simulated and real-world environments, include the integration of a VLM, customized prompt texts structured with the FSM, and a comprehensive regulation database. These elements work together to ensure the system accurately interprets and adheres to traffic regulations, enhancing the overall reliability and safety of ADS planning. Overall, the proposed framework addresses existing limitations in ADS traffic law and regulation awareness. Testing results confirm the effectiveness of the proposed framework and indicate the VLM has promising potential in real-time strategic planning tasks. In terms of future work, efforts will focus on fine-tuning VLMs to achieve more accurate scene interoperability and specialized domain expertise, ensuring the capability of handling a more comprehensive range of traffic scenarios with greater precision. Additionally, refining the zero-shot mechanism to a multi-shot prompting method will help the model better utilize situational prior knowledge and previous prompts, thereby enhancing its understanding of specific scenes and overall performance.

Chapter 6

CDA Distributed Testings with Heterogeneous Agents: Distributed Testings using VOICES System

Distributed testing is a comprehensive evaluation method for Cooperative Driving Automation (CDA) that enables multiple, independently operated agents—vehicles, roadside units, and infrastructure elements—to interact and coordinate in real-time across both physical and virtual environments. This testing approach is essential for accurately assessing CDA systems, as it mirrors the complex, multi-agent ecosystem in which CDA applications must operate. Distributed testing involves simultaneous testing across different locations and setups, allowing for interaction between diverse CDA agents, each with unique control systems and capabilities, which closely reflect the real-world CDA environment. To support this testing need, a full-stack CDA framework, OpenCDA, was utilized to establish a local testing environment, enabling participation in a joint distributed effort through the VOICES system. This setup incorporates heterogeneity—where each CDA agent operates independently—ensuring that the CDA system's performance, interoperability, and scalability can be assessed in conditions that resemble real-world complexity.

Aligned with the parallel framework, distributed testing integrates several key sections to provide a robust evaluation environment. Both virtual and physical parallel environments are leveraged, combining assets like real vehicles, roadside sensors, and infrastructure with virtual elements such as digital twins, SUMO, and CARLA. Scenario engineering is applied to create diverse and challenging testing conditions that mirror real-world situations. In the parallel validation component, testing progresses through multiple stages, including SIL, HIL, and distributed testing, each introducing higher levels of real-world complexity and disturbance. Finally, parallel operations incorporate coordination and control operations as well as experiment and evaluation operations, ensuring that all agents interact dynamically and that data is systematically collected and analyzed. Through this comprehensive setup, partners such as Mcity, Argonne National Laboratory (ANL), Oak Ridge National Laboratory (ORNL), and the Federal Highway Administration (FHWA) conduct coordinated, nationwide tests, validating CDA systems' readiness for complex real-world deployments.

6.1 Introduction

The surface transportation ecosystem is highly complex, with a wide range of independent organizations developing and optimizing their systems according to unique objectives and priorities. Since these systems are typically operated in isolation, the surrounding systems with which they interact effectively become "black boxes," leading each organization to make assumptions—often imprecise—about the behaviors of other systems. To support the research and development of a cohesive, distributed transportation system, the U.S. Department of Transportation (USDOT) funded two projects under the Virtual Open Innovation Collaborative Environment for Safety (VOICES) initiative, led by the Federal Highway Administration's (FHWA) Saxton Transportation Operations Laboratory and the MITRE Corporation. The goal of these projects is to create and validate a secure, mixed-reality test environment where multiple organizations can integrate their existing simulated and physical test assets and engage with other research platforms in real-time. This type of environment allows for early detection and resolution of integration issues across systems, providing a safe testing space before physical trials. Moreover, this approach enables developers to utilize their own models without needing to re-code for a single, unified simulation platform, making collaborative testing more efficient and accessible. This type of testing is highly suited for the integration of heterogeneous advanced intelligent systems such as CDA and smart infrastructure in which development is happening across multiple sectors. These systems are potentially transformative but necessitate careful integration with the existing transportation ecosystem, as well as emerging technologies.

As CDA technologies evolve, extensive testing is required to ensure their reliability, safety, and performance. Waymo has driven more than 20 billion miles in simulation to identify the most challenging situations and run approximately 40,000 unique scenarios in closed-course environments to evaluate their technologies [92]. California reported that approximately 5.7 million test miles were driven cumulatively by CDAs produced by a variety of companies from December 1, 2021 to November 30, 2022 [29]. Kalra and Paddock mentioned that it may require 5 billion miles to be driven to demonstrate autonomous vehicles' failure rate is statistically significantly lower than the human driver failure rate [36]. Further, even for a small change in a single line of code of an autonomous driving system, the counter of miles driven resets back to zero and the entire test must be repeated. Another limitation is that there is a substantial difference between miles driven on an empty stretch of road versus miles driven in heavy traffic.

A simulation platform provides a promising approach for evaluating CDA systems by enabling various driving scenarios without real-world risks [81]. Commonly, simulation testing involves virtual CDA systems interacting within a modeled traffic network using platforms like CARLA [21], LGSVL [77], CarMaker [83], and CarSim [12], which validate autonomous systems on simulated or real-world maps with vehicle dynamics and sensors such as cameras, LIDAR, and RADAR. However, simulations alone require extensive calibration and may miss edge scenarios that real environments may occasionally present. Therefore, mixed traffic testing is essential as unique behaviors of different manufacturers' algorithms and intelligence levels across systems impact CDA system performance. Advanced testing approaches like hardware-in-the-loop (HIL) integrate real vehicle hardware in virtual setups to identify issues early and reduce on-road testing risks, with research by Cantas et al. [11] and Jiaqi et al. [54] utilizing HIL setups combining traffic signal controllers and DSRC devices with simulated networks for V2X testing. Vehicle-in-the-loop (VIL) testing, bridging HIL and real-world environments, allows real vehicles in virtual spaces for precise energy consumption data and accurate vehicle dynamics representation [12]. Rengarajan et al. [76] and Miriam et al. [20] utilized VIL with chassis dynamometers for CDA energy analysis, and Shao et al. [81] used a hydrostatic dynamometer to test CDA robustness and real-time capabilities.

However, none of the SOTA validation methods have been able to deliver a method that integrates multiple heterogeneous agents to interact in the same environment. The integration of heterogeneous platforms, such as simulation, on-road, and on-dynamometer testing platforms in a distributed testing environment has potential to address all these research questions. Therefore, distributive testing has emerged as a pivotal approach in this context, leveraging the strengths of multiple stakeholders, including automotive manufacturers, technology developers, regulatory bodies, and research institutions. Distributive testing enables all participants to communicate among themselves through a common network and exchange simple and complex data among themselves. Distributive testing is widely used in many industries, such as healthcare, for collaborative learning to train interprofessional teams [1, 44] military for training teams in emergency situations [39, 70]. In the automotive industry, distributed testing offers numerous benefits, such as enabling large-scale testing of new driving features across a wide range of complex scenarios. It also supports early-stage testing of critical hardware components, including ECUs (Electronic Control Units), batteries, and other vehicle subsystems, under various conditions. To the best of our knowledge, there is no established distributed testing architecture that has been used in the automotive industry to understand the behavior of different entities when they operate collaboratively.

This paper presents a distributed test for CDA applications that involve multiple heterogeneous agents operating in individual and mixed-reality environments. This distributed testing addressed multiple crucial CDA challenges, such as identifying if multiple geographically distributed sites simultaneously co-simulate and interact with micro traffic across a cloud-based network, understanding the impact of surrounding traffic when vehicles demonstrate cooperative driving while interacting with smart infrastructure systems, etc. The distributed test is conducted as a collaborative effort among multiple entities, including UCLA, and is officially designated as the Pilot 2 test.

6.2 Methdology

In order to validate OpenCDA's ability to interact with heterogeneous agents within a distributed testing framework, UCLA participated as a major entity in the Pilot 2 distributed test. The test has two primary objectives: ensuring system functionality and connectivity through SAE standard messages for interoperability and evaluating interactions among diverse heterogeneous agents within a shared environment. During the test, UCLA utilizes OpenCDA within a local environment that engages seamlessly with other agents in the distributed network, with the VOICES system facilitating data exchange via SAE messages to ensure real-time interoperability and data synchronization across test sites. This section elaborates on the UCLA-OpenCDA testing environment, provides an overview of the distributed systems for other entities, and introduces the detailed distributed test plan.

6.2.1 UCLA-OpenCDA Testing Environments

UCLA is a key participant in the distributed experiments within the VOICES framework, focusing on testing OpenCDA's ability to interact effectively with heterogeneous agents and contribute to broader traffic system objectives. In the co-simulation experiments, the UCLA vehicle, controlled by OpenCDA, will operate under human input, providing human-in-the-loop testing to assess adaptive responses in a mixed-reality environment. For application deployment, OpenCDA will decode SAE J2735 messages in real-time and apply an eco-approach algorithm designed to optimize vehicle trajectory as it approaches signalized intersections. This approach aims to improve traffic flow efficiency and reduce emissions by smoothing vehicle movements. Through these experiments, UCLA is testing OpenCDA's interoperability and scalability in diverse scenarios, supporting cooperative driving automation advancements and demonstrating the framework's potential in real-world-like applications.

Local System Overview

As shown in Figure 6.3, the VOICES system integrates four potential testing environments, with UCLA participating as one of the simulated testing environments in the distributed test. In UCLA's local setup, a virtual testing environment projects the data and status of all other agents in real time, allowing interaction within the shared simulation. The simulated world is established and managed by VOICES, while UCLA's ego vehicle is locally controlled by OpenCDA—a CDA system equipped with perception and control capabilities to interact dynamically with other agents. Heterogeneity is a central aspect of the distributed test, as each entity operates independently and maintains individual control over its systems. Within this setup, UCLA's OpenCDA directs the ego vehicle to respond to and interact with other individually controlled agents, creating a realistic simulation of diverse agent behaviors and interactions in the shared testing framework.

The simulation is powered by the CARLA platform, where multiple entities interact in real-time. Within this environment, the UCLA vehicle is fully controlled by OpenCDA and operates under dynamic conditions. As shown in Figure 6.1, OpenCDA is an open co-simulation-based research/engineering framework integrated with prototype cooperative driving automation (CDA) pipelines as well as regular automated driving components (e.g., perception, localization, planning, control). Unlike traditional simulations where OpenCDA

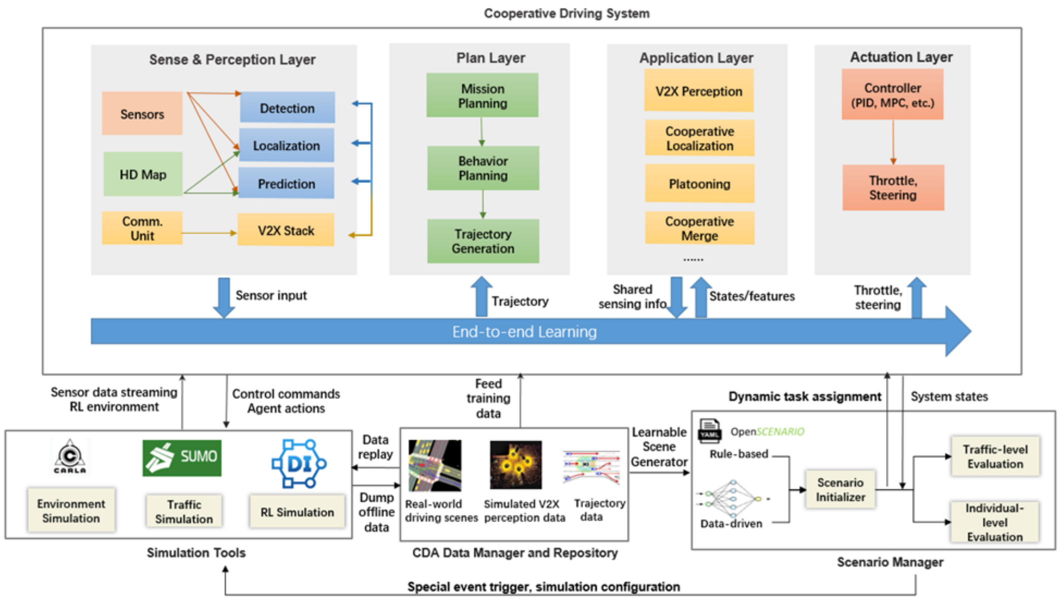


Figure 6.1: OpenCDA System Diagram.

initiates the simulation by spawning vehicles from a static configuration file, the VOICES system manages the entire simulation environment, including receiving test data from all participating entities and managing the local CARLA environment configurations. OpenCDA adapts by spawning the UCLA vehicle at a location specified by VOICES, which ensures flexibility and integration across the distributed system. Importantly, each entity runs CARLA locally to accommodate the high computational demands and to interface with its unique systems and hardware, which differ in communication methods. This setup allows each entity to utilize the CARLA server within its local network, ensuring efficient integration with its specific hardware and software configurations. The VOICES system also manages the data streams between entities. It controls the overall simulation environment, enabling flexible distributed testing where each entity operates independently while interacting within the shared simulation data stream.

Simulated (Virtual) Testing Environment

A fundamental building block for a distributed test is a mutual testing environment accessible across all platforms. By establishing this mutual environment, each agent can locate itself,

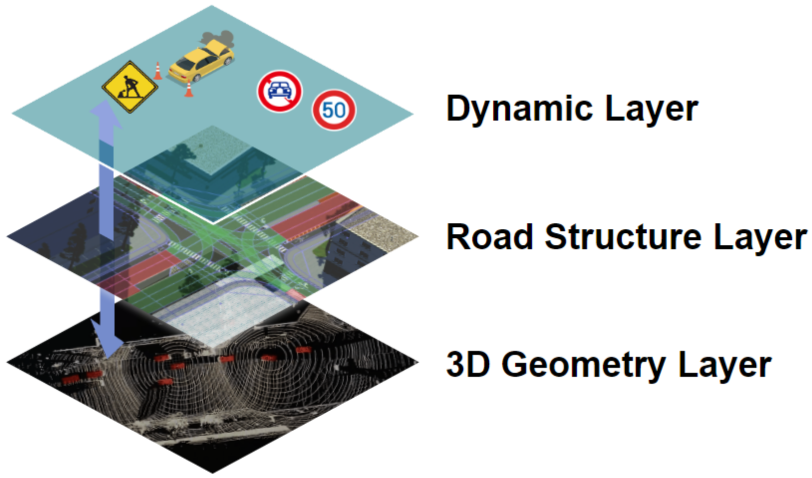


Figure 6.2: High definition map structure.

perceive the shared surroundings consistently, and identify the locations and conditions of other agents. This mutual understanding is essential for any meaningful interaction and potential cooperation, as it guarantees geometric consistency and provides a common foundation for all traffic movements. In the distributed test, the testing environment spans both a virtual and a physical test track, also known as the digital twin, centered around the Mcity testing facility.

Mcity's digital twin is a multi-layered virtual representation of the real-world Mcity test facility, built using a range of digital assets within the open-source CARLA simulator. As shown in Figure 6.2, three primary layers define the environment: the 3D geometry layer, the road structure layer, and the dynamic layer. The 3D geometry layer provides a detailed model of road surfaces and surrounding areas, with realistic textures and high-precision positioning data accurate to within 1-3 centimeters. The road structure layer represents the layout of roads, lanes, intersections, traffic signals, and signs across Mcity, formatted in OpenDRIVE for seamless integration with CARLA. The dynamic layer includes active traffic movements during the simulation, such as traffic signals, nearby vehicles, and evolving environmental conditions, capturing the dynamics of the testing environment in both virtual and physical aspects.

In the road structure layer, the road network map, formatted in OpenDRIVE, provides essential functionality for all participating agents. With the topology map in OpenDRIVE and the digital twin integrated within CARLA, a comprehensive set of APIs enables the determination of road connectivity, lane markings, and access to critical regulatory data. This information is crucial for automated driving systems, including OpenCDA, to support route planning (using topology maps like lanelet2) and trajectory regulation. For example, OpenCDA identifies the current and target lanelets based on Cartesian coordinates and calculates a global route from the starting point to the destination. Along this route, OpenCDA updates a detailed path every 5 seconds, adjusting to the current speed limit obtained from the network map. For maneuvers such as stopping at traffic lights and lane following, precise lane marking data—including lane lines and stop bars—is extracted directly from the map, guiding accurate vehicle movements within the simulation.

In the 3D geometry layer, the digital twin captures precise elevation and surface details that extend beyond standard GPS accuracy. Additional methods, such as fixed-wing flyover LiDAR, photogrammetry, and terrestrial-based LiDAR, supplement GPS data, allowing the digital twin to accurately reflect natural terrain features like grass-covered hills and dirt patches, as well as constructed elements like curbs and sidewalks. This detailed elevation mapping ensures that Mcity's topography is faithfully represented in the simulation, supporting realistic interactions with diverse surfaces, from gravel to pavement to sidewalks. Additionally, the 3D geometry layer incorporates accurately aligned and complex road geometries calculated with RoadRunner, including transitions such as highway segments leading into traffic circles and roundabouts adjacent to city blocks. This precise alignment creates a comprehensive and lifelike testing environment for a variety of driving scenarios.

Deployed CDA Application

The CDA application deployed for the distributed test is the eco-approach algorithm, an advanced method designed to optimize fuel consumption and driving efficiency by dynamically adjusting speed and energy use in real time [52]. Based on the Relaxed Pontryagin's Minimum Principle (RPMP), this algorithm addresses the computational challenges of traditional optimal control by simplifying certain constraints, making it feasible for real-time application in dynamic environments. Through RPMP, the eco-approach algorithm calculates near-optimal speed trajectories, balancing kinetic energy with throttle and braking inputs to minimize fuel consumption as vehicles approach intersections. By continuously monitoring the vehicle's speed, intersection distance, and current signal phase, the algorithm adjusts the speed profile to align with green lights, aiming to reduce unnecessary stops and sudden decelerations. This smooth trajectory minimizes the energy lost during braking and acceleration, allowing the vehicle to "glide" through intersections whenever possible. Unlike fixed-parameter models, the eco-approach algorithm adapts to real-time inputs, adjusting its decisions based on real-time signal data and the vehicle's immediate environment. This adaptability allows it to optimize fuel efficiency dynamically, promoting smoother, more energy-efficient driving patterns in response to changing conditions. The eco-approach algorithm not only improves the lead vehicle's efficiency but also sets a smoother traffic flow that aligns well with cooperative driving applications, showcasing its effectiveness in real-world traffic scenarios.

In the context of OpenCDA, the eco-approach algorithm leverages the full perception stack, which includes localization and perception modules to determine the precise location of the next signal and its distance from the vehicle's current position. OpenCDA also receives J2735 data to assess the current signal phase, the timing of upcoming phases, and the duration of each phase. Using this information, OpenCDA's trajectory optimization and downstream planning and actuation modules work in unison to implement a smooth, continuous trajectory that allows the vehicle to pass through intersections without unnecessary stops. This planned trajectory aligns the vehicle's speed with the timing of the green light, thus maintaining the flow and achieving the eco-approach objective.

Testing heterogeneous interactions based on the eco-approach algorithm involves observing how OpenCDA, leading a stream of vehicles on a single-lane road, influences the entire traffic

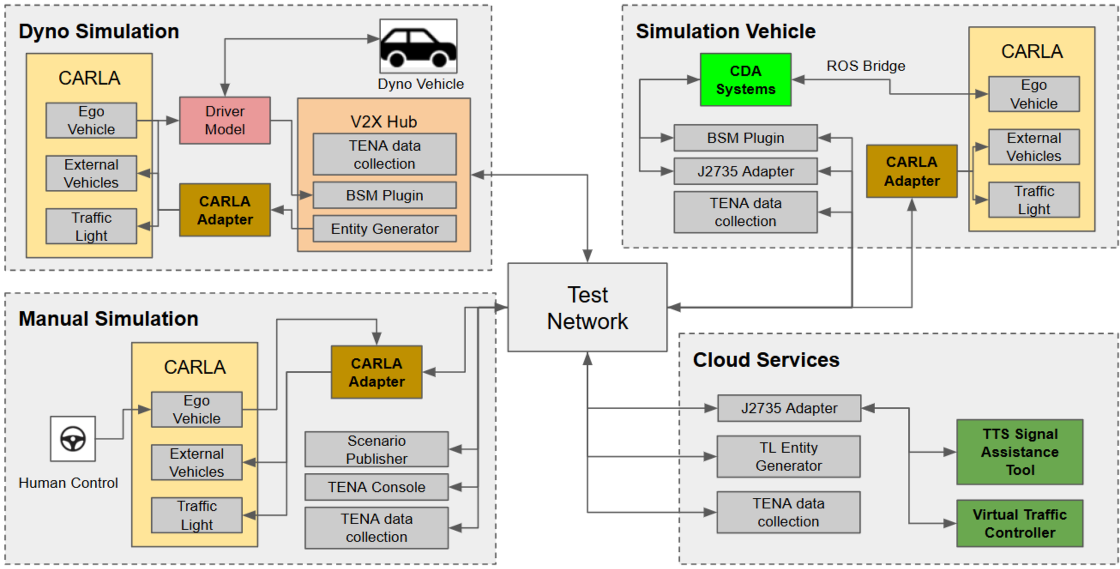


Figure 6.3: VOICES system architecture. BSM = basic safety message; TENA = test and training enabling architecture; ROS = robot operating system; TL = traffic light; V2X = vehicle to everything; TTS = traffic technology services Inc

flow at an intersection. OpenCDA can regulate the stream's pace by slowing down slightly to align all vehicles with the intersection's green phase, allowing each vehicle to pass through without stopping. This interaction becomes more complex as different participants in the stream are controlled by separate agents, each potentially using different driving algorithms. Some may lack eco-approach capabilities and follow basic car-following models, while others may recognize OpenCDA's approach and adjust their speed accordingly. For instance, a conventional vehicle would maintain a fixed following distance, while an eco-approach-equipped agent would observe OpenCDA's behavior and further adjust its speed to match. Since each agent is autonomous and varies in behavior, OpenCDA relies on real-time observations to adjust its own trajectory, while other agents do the same. If the eco-approach strategy succeeds and different behaviors are observed among agents, it demonstrates effective interaction and validates OpenCDA's eco-approach algorithm, even without shared planning principles or intentions, thereby validating the distributed test.

6.2.2 Distributed Testing Environments

The VOICES system is a comprehensive, distributed testing network specifically designed to evaluate Cooperative Driving Automation (CDA) across heterogeneous agents using a variety of testing assets, including both physical devices and systems as well as virtual systems. This system integrates simulations, real-time communications, and various physical and virtual test assets, enabling coordinated testing across multiple types of CDA entities within a shared framework. VOICES is structured into distinct testing environments, each representing a unique component of the transportation ecosystem and accommodating various CDA assets such as simulators, dynamometers, physical vehicles, and roadside units.

These testing environments are interconnected through a central test network that synchronizes all activities and manages data flow, ensuring cohesive operation across distributed test sites. This setup allows VOICES to comprehensively assess CDA functionality with heterogeneous agents, as each environment can host a customized combination of assets to simulate complex, real-world interactions. As shown in Figure 6.3, the VOICES system includes four potential testing environments linked by a unified test network, providing a robust platform for analyzing CDA performance in distributed, mixed-reality settings:

- The Test Network is the communication backbone of the VOICES system, enabling real-time, synchronized interactions across simulations, cloud services, and hardware. Standardized protocols like J2735 UDP, V2X, and TENA ensure compatibility across heterogeneous agents, creating a unified framework crucial for testing CDA in complex scenarios.

- The dyno simulation integrates a physical dynamometer vehicle into the virtual environment via a V2X Hub, transmitting vehicle state data using J2735 BSM. Controlled by an ADS agent, the dyno interacts dynamically with other virtual entities, supporting hardware-in-the-loop testing. This setup enables realistic interactions between physical and simulated systems, facilitating comprehensive CDA functionality assessment in a distributed testing environment.

- The simulated vehicle environment uses CARLA and OpenCDA to replicate automated driving, with CARLA creating digital replicas in real-time and sharing live data through a CARLA Adapter. OpenCDA, compatible with CARLA and SUMO, enables vehicles to interpret and respond to dynamic conditions. Supported by ROS Bridge and a J2735 Adapter, this setup ensures interoperability for engaging with virtual signals and interacting across simulated and real-world environments.

- The manual simulation section enables human-in-the-loop control of virtual vehicles within CARLA, connected to the main test network via the CARLA Adapter. This setup adds real-time human input to interactions with automated systems, supporting co-simulation by linking multiple test sites and integrating mixed-reality ADS platforms. This framework enables distributed testing to evaluate coordinated CDA performance across diverse scenarios.

- The VOICES system uses remote traffic signals and cloud-based services to enhance CDA testing flexibility and realism. Key components include a traffic light generator and virtual controller, integrated via J2735 Adapters and TENA Console for real-time synchronization. The cloud-based TTS PSA Tool provides dynamic, centralized control, allowing flexible traffic signal management and comprehensive CDA evaluation across varied scenarios.

Overall, the VOICES system also aligns with the parallel development and testing framework to create a distributed test that is part of the parallel validation module. Within the distributed environment, heterogeneous agents—both virtual and physical—operate and interact within a synchronized network. This setup coordinates various parallel systems, such as dyno simulations, simulated vehicles, human-in-the-loop simulations, and remote traffic signals, to support diverse traffic and environmental scenarios. By facilitating coordinated operations like experimentation and iterative learning, VOICES enables realistic multi-agent interactions, advancing the interoperability, reliability, and scalability of CDA systems in a controlled yet dynamic testbed. Meanwhile, the distributed test places a strong emphasis on cooperation and agent interaction, enabling testing scenarios that would not be feasible without multiple participants. The rest of this section will provide a brief overview of each participant's local testing environment.

Argonne: VIL Dynamometer Set-Up

Argonne National Laboratory supported the VOICES Pilot 2 program with a research vehicle, a 2020 Chevrolet Bolt, equipped with advanced sensors and hardware, installed at the four-wheel-drive (4WD) chassis dynamometer facility of the Advanced Mobility Technology Laboratory (AMTL) at Argonne [87]. The 4WD test cell accommodates light-to-medium duty vehicles with up to 373 kW of power per axle and can adjust ambient temperatures from -20°C to +35°C, with simulated solar loading of 850 W/m^2 and a vehicle speed-match fan reaching 100 km/h. An in-house adaptive cruise control (ACC) system was used to operate the research vehicle, which lacks factory-installed ACC, by translating acceleration requests from the controller into pedal actions via a robot driver (Figure 6.4). The setup's flexible architecture allows seamless replacement with advanced control systems. To integrate Argonne's vehicle-in-loop (VIL) workflow into the VOICES system, researchers developed a software component (BSM-Generator) to update the vehicle's location on a CARLA map based on its speed signal and predefined waypoints.

Argonne researchers leveraged VOICES data collection system and Argonne's existing custom-built modular data acquisition (DAQ) capability to capture high fidelity data during the Event 2 test to evaluate the performance of the system and vehicle's energy consumption. During the VOICE Pilot 2 Event 2, an Intrepid Control System logger (NeoVi Fire3) was also installed in the vehicle to capture the vehicle's speed signal from CAN interface and a dSPACE MABX II collected controller data in real time. Argonne's DAQ system enabled integrating all these CAN, analog, digital and facility data into a single time-aligned output

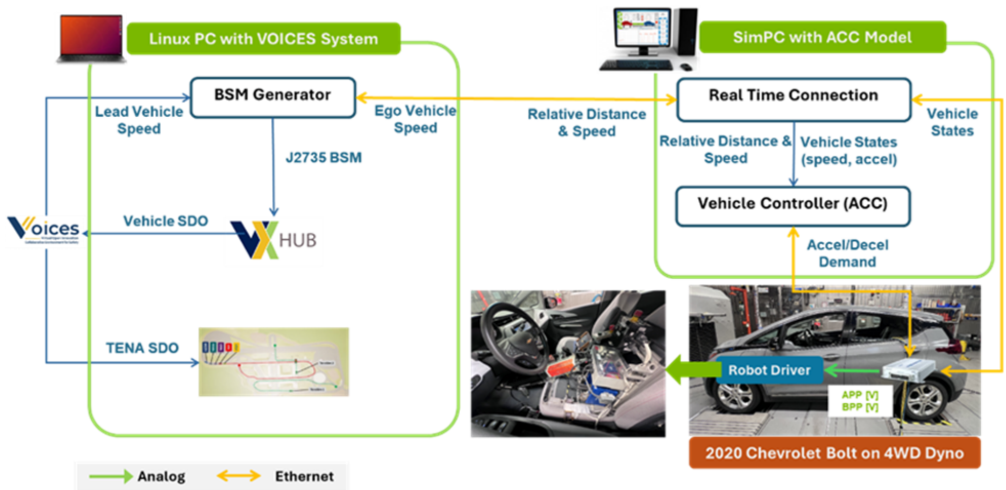


Figure 6.4: ANL VIL workflow with VOICES system.

data file for analysis. In addition, one in-house data-collection system is developed to capture ego vehicle (ANL dyno vehicle) and lead vehicle (UCLA Open-CDA) BSMs and speed data in real-time.

FHWA: CARMA Platform Setup

The CARMA CDA simulation environment facilitates controlled testing of cooperative driving automation by integrating a vehicle simulator within the CARLA platform. In this setup, CARMA controls a single simulated vehicle locally within CARLA, while other ego vehicles from entities such as ANL, Mcity, and FHWA are continuously updated through the VOICES platform. This setup allows real-time coordination, as VOICES shares data on the CARMA vehicle's conditions with these entities, ensuring synchronized interaction across multiple simulation and physical test sites.

The CARMA simulation framework rigorously tests CDA capabilities under realistic conditions by integrating tools like CARMA Platform, CARMA Streets, SUMO, and NS-3. CARMA Platform handles vehicle control and cooperative maneuvers, while CARMA Streets manages infrastructure interactions, such as traffic signals. SUMO simulates larger traffic flows, and NS-3 provides realistic V2X communication modeling by simulating network interactions between vehicles and infrastructure [14]. Together, these components enable

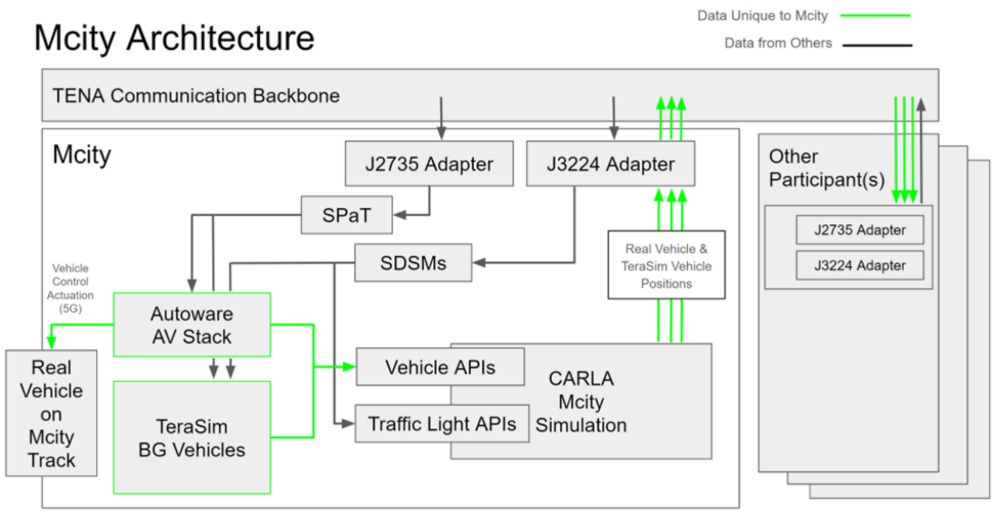


Figure 6.5: Mcity mixed reality architecture.

CARMA to evaluate CDA applications like V2X communication, cooperative perception, and shared decision-making, within a virtual environment that mirrors real-world conditions. The simulation's integration with live traffic data and hardware inputs allows CARMA-equipped vehicles to interact with other CDA agents, creating a multi-agent environment where CDA systems can be tested, refined, and validated for real-world application.

Mcity: Mixed-Reality Set-Up

Mcity utilized a mixed-reality setup, with their contribution being a real vehicle controlled by an Autoware AV stack hosted on an edge computer located near the physical test facility. The detailed system architecture is presented in FIgure 6.5. The vehicle, a Hybrid Lincoln MKZ, was equipped with an onboard computer, an onboard localization system (featuring an Oxford RT3000 RTK GPS and inertial navigation system), and a Dataspeed drive-by-wire system. Localization data from the vehicle's RTK (Real-Time Kinematic) system was shared and transmitted to the Autoware stack at the edge. Additionally, Autoware received the positions of other test vehicles from the VOICES system. The Autoware stack then handled navigation and generated control commands, which were sent back to the real vehicle, where they were executed by the drive-by-wire system.

Mcity was also responsible for controlling the background vehicles (shown above with green paths) remotely through TeraSim, Mcity's cloud-based traffic behavior simulator built on SUMO. TeraSim is trained using thousands of hours of real-world driving data, including scenarios involving conflicts (near misses and crashes). This training allows users of TeraSim to adjust how much adversarial behavior the background vehicles exhibit. However, because this test focused on demonstrating distributed testing capability and gathering econometric data, TeraSim's adversarial behavior was set to zero, ensuring that the background vehicles did not interfere with the AV string.

6.2.3 Experiment Design

The experiment design is structured to validate OpenCDA's ability to interact seamlessly with heterogeneous agents in a shared testing environment, a core objective for advancing Cooperative Driving Automation (CDA). Following the parallel validation approach, three key tests are organized: a simulation test to verify connectivity, a software-in-the-loop (SIL) test to assess OpenCDA's capability to handle SAE-standard messages, and a full distributed test that integrates mixed-reality elements, combining virtual and physical agents in real-time interactions. The corresponding experiments are named Pilot 2 Event 0, Pilot 2 Event 1, and Pilot 2 Event 2, respectively.

Adhering to scenario engineering principles within the parallel framework, tailored testing scenarios are developed for each of these experiments, crafted to mirror complex, real-world conditions. A central focus of the scenarios is to test agent cooperation under diverse configurations by implementing the eco-approach algorithm. The objective is for all vehicles to pass through a signalized intersection without stopping, requiring smooth speed adjustments and coordination. However, each agent functions independently, without shared knowledge of other agents' specific decision-making strategies, meaning that some agents may employ eco-approach while others may not. This setup provides a realistic assessment of OpenCDA's interoperability and adaptability, testing its ability to coordinate with diverse agents in a mixed environment and evaluate its real-world readiness within a distributed multi-agent CDA system. This section will introduce all experiments in detail.

Pilot 2 Event 0: System Connectivity Test (Simulation Test)

Event 0 was designed to integrate multiple entities into a single, distributed testing system and to validate the VOICES platform and its wireless (internet) connectivity across diverse sites. The primary goal was to assess whether multiple sites could simultaneously co-simulate and interact with micro-traffic across a cloud-based network. By using only human-controlled vehicles and removing CDA algorithms and specialized messaging protocols from the equation, Event 0 minimized complexity, creating an ideal setup for VOICES to establish a true distributed test across geographically separated entities with varied equipment types. Following the parallel development and testing framework, Event 0 incorporated both virtual and physical environments, including simulations, dyno setups, and actual vehicles. The scenario involved straightforward vehicle interactions within an urban setting modeled as the Mcity digital twin, providing a simple yet effective environment for evaluating VOICES system coordination.

The event consisted of two parts: Part 1 involved two vehicles controlled by SUMO and one UCLA OpenCDA vehicle navigating a signalized intersection. The SUMO vehicles entered the intersection from opposite directions on a green light, while the OpenCDA vehicle approached on red and proceeded once the light turned green (see Figure 6.6 for starting positions and paths). Part 2 expanded the setup to eight participants—Mcity, TTS, ANL, ORNL, UCLA, Econolite, FHWA, and VW—each controlling a manually operated vehicle. All vehicles gathered at a central location before completing a lap around the Mcity test track, with Figure 6.7 depicting each site's starting positions.

Despite the simplified CDA objectives, Event 0 achieved major milestones in distributed testing. It marked the first use of the Mcity digital twin, the first simultaneous integration of multiple simulation platforms (CARLA and SUMO), and the highest number of participants

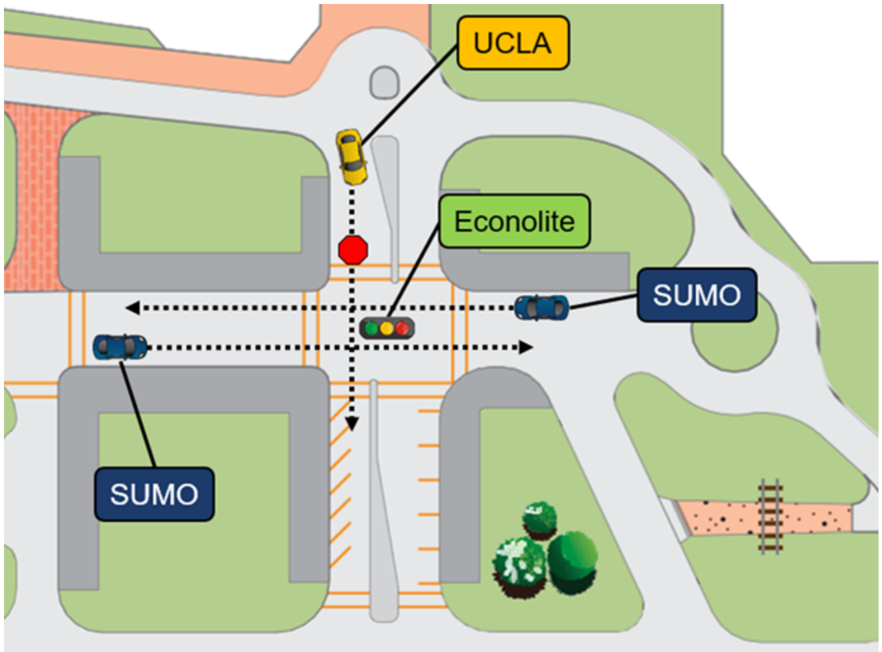


Figure 6.6: © 2024 Mcity. Modified by FHWA. Pilot2 Event0 Part 1 Scenario Diagram

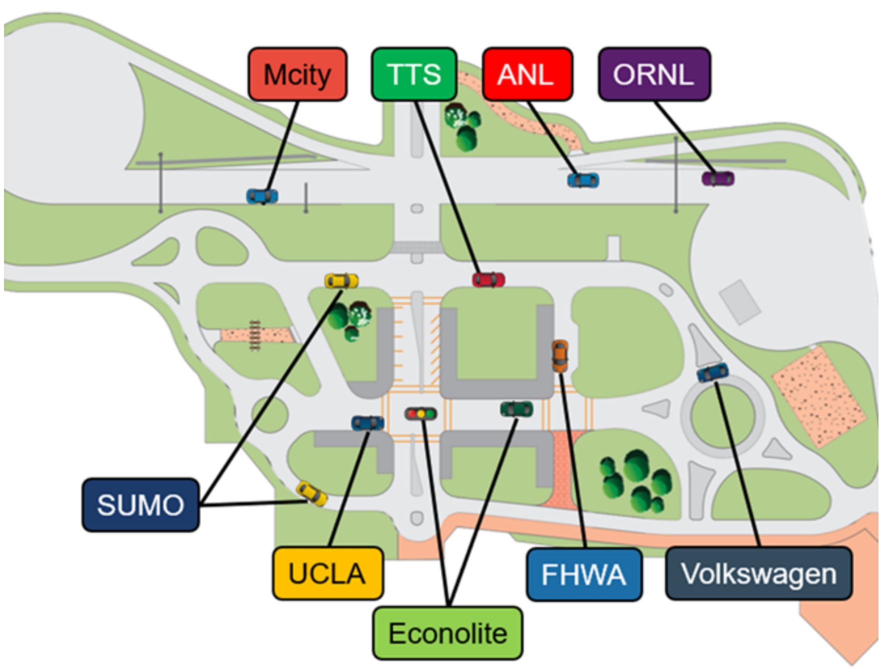


Figure 6.7: © 2024 Mcity. Modified by FHWA. Pilot2 Event0 Part 2 Scenario Diagram.

to date (previously capped at four). The preparation and execution process helped develop platform enhancements and validated the VOICES system's ability to manage more complex scenarios, setting a strong foundation for the Pilot2 test campaign.

Pilot 2 Event 1: Interoperability Compatibility Test (SIL Test)

Event 1 of Pilot 2 focused on testing interoperability across multiple entities by utilizing SAE standard messages, aiming to ensure compatibility in communication and message processing between different systems. This test introduced an additional layer of complexity by incorporating SAE-standard message encoding and decoding, with VOICES coordinating encoded messages across sites. Each participating entity implemented encoding and decoding processes, allowing VOICES to assess the system's resilience to communication noise and delays, as well as the efficiency of message handling within the distributed network. In terms of parallel systems, the environment setup remained the same, including simulations, dynamo setups, and real vehicles; however, the scenario expanded to include the transmission of safety-critical messages for VRUs using SAE standards.

A key focus in Event 1 was the successful exchange of the SAE J3224 Standard Dynamic Safety Message (SDSM), which required new TENA J3224 messages, J2735 adapters, and encoding methods to support the communication framework. Additionally, other standardized messages such as SAE J2735 MAP (providing roadway and ITS device location information) and SAE J2735 SPaT (Signal Phase and Timing, detailing traffic signal states) were incorporated into the test. Participants in Event 1 included FHWA, Mcity, Traffic Technology Services (TTS), and Volkswagen (VW). In this setup, Mcity generated SDSM, SPaT, and MAP messages and distributed them to all sites. FHWA received SDSM data and displayed pedestrian locations in the CARLA environment; TTS integrated SPaT data into its Personal Signal Assistant (PSA) Tool, allowing users to access current and predicted traffic signal states; VW collected SDSM messages to compare against the European Telecommunications Standards Institute (ETSI) Collective Perception Message (CPM), which serves a similar

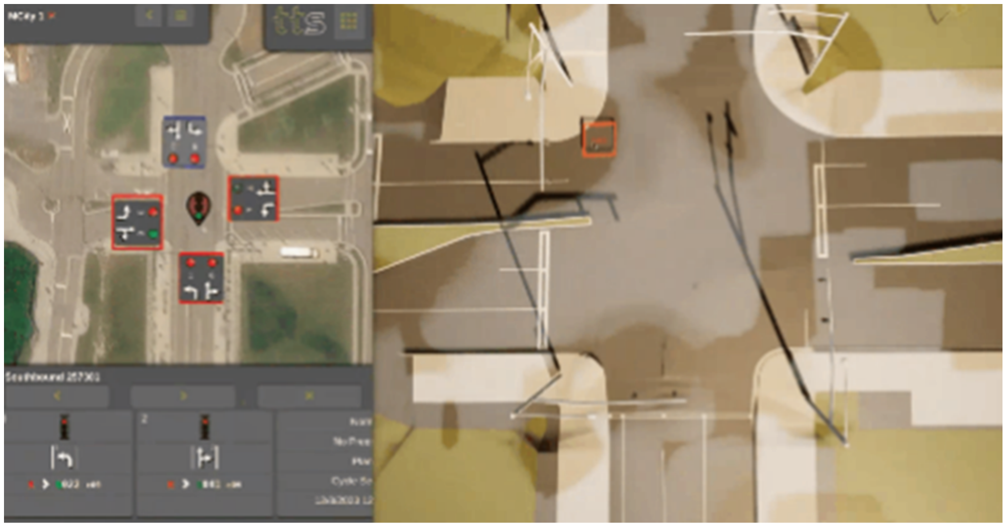


Figure 6.8: Event 1 PSA Tool (left) and SDSM Visualization (right).

function for sensor-based perception data. A sample experiment scene of the PSA tool and the SDSM visualization is presented in Figure 6.8.

Through Event 1, the test demonstrated that distributed testing is an effective approach for exchanging standardized messages, providing a valuable framework for organizations to test applications against real-world data without the need to co-locate hardware. This structure facilitates the adoption of standardized messages and enables entities to validate their applications against real-world interoperability requirements, ensuring message compatibility and correct usage during development.

Pilot 2 Event 2: Comprehensive Distributed Test

Event 2 marked the final and most complex test in the Pilot 2 distributed testing series, aiming to conduct a fully distributed test with heterogeneous agents. Each agent operated independently from unique locations, employing specialized CDA assets and independently developed CDA algorithms. Within a shared virtual environment, each ego vehicle could observe other agents in real-time and interact with them, allowing each agent's behavior to influence the overall traffic stream. The event focused on assessing how surrounding AV traffic impacts the eco-driving assumptions made by each vehicle, measuring system-level

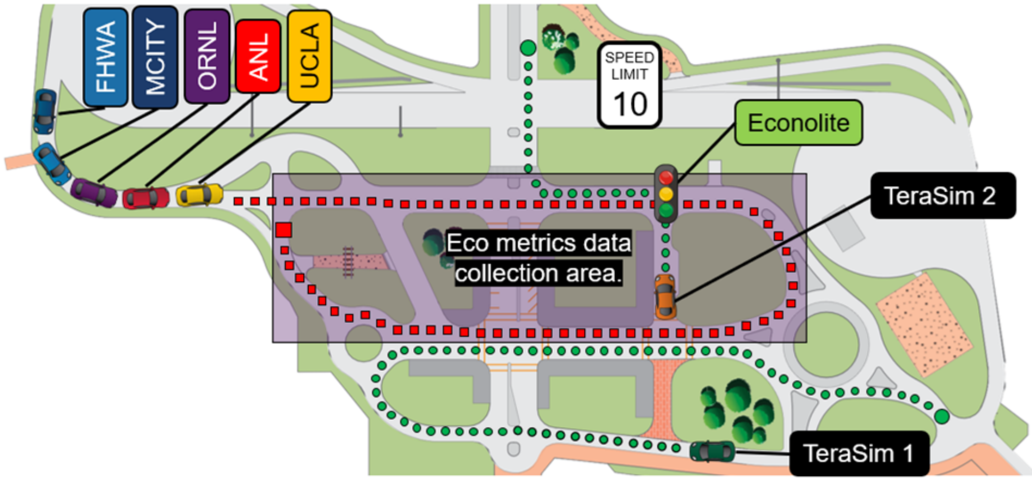


Figure 6.9: Pilot2 Event2 Scenario Diagram.

improvements in metrics such as energy consumption, traffic efficiency, and roadway safety. Following the parallel framework, the environmental setup remained the same, encompassing simulations, dyno setups, and live vehicles. The scenario base was structured as an eco-approach scenario, where all entities participated in a vehicle stream led by UCLA's OpenCDA vehicle.

A comparison study on energy consumption was a central aspect of Event 2, evaluating five distinct automated driving platforms when operating in a vehicle string versus driving independently. For each run, vehicles completed a clockwise loop around the Mcity test track, stopping at a traffic signal midway. Participating vehicles included FHWA's CARMA Platform, UCLA's OpenCDA, a driving model developed by ORNL, a live dynamometer vehicle at ANL, and a teleoperated vehicle at Mcity using Autoware. Additionally, Mcity contributed two background vehicles from their TeraSim naturalistic driving platform, and Econolite provided a virtual traffic signal controller. Starting positions and routes can be found in Figure 6.9. Energy consumption data was obtained through two methods: for the ANL dynamometer vehicle, data was collected directly using onboard sensors, while all other vehicles' energy data was calculated using ANL's Autonomie tool, based on vehicle type, speed, location, and road grade. The ENERGY ANALYSIS section contains further details on the Autonomie algorithm and results.

Beyond energy analysis, Event 2 underscored the advancements made in distributed testing. It marked the first FHWA-sponsored event to feature five unique automated driving platforms collaborating within a cooperative driving setup. This milestone demonstrated that distributed testing could significantly streamline CDA development by enabling academia, industry, and government entities to connect and test applications digitally, allowing earlier and easier integration with their existing development platforms. By coordinating complex multi-agent interactions through the VOICES platform, Event 2 illustrated the feasibility and benefits of a distributed framework in advancing CDA research and deployment.

6.3 Experiment Results

The experiment results section presents an in-depth analysis of outcomes and insights from the three Pilot 2 events: Event 0, Event 1, and Event 2. Each event focuses on specific aspects of the distributed testing framework, progressively increasing in complexity to evaluate the VOICES platform's capabilities. Event 0 assesses the foundational functionality of the distributed system, confirming that it can support simultaneous operations across multiple entities, validating network stability and core communication channels. Event 1 evaluates system interoperability using SAE standard messages, specifically J2735 and J3224, to ensure seamless communication between entities, establish cooperative interactions, and validate message exchange through the VOICES platform. Lastly, Event 2 represents the most advanced test, with fully independent entities—each at a unique location and running its own CDA system—interacting to achieve a shared objective, such as an eco-approach strategy, while impacting traffic stream dynamics collaboratively. This event demonstrates the system's ability to handle complex, multi-agent interactions across diverse platforms and locations, showcasing VOICES' capacity to manage large-scale, distributed environments effectively.

Pilot 2 Event 0: System Connectivity Test (Simulation Test) Result

The network latency and jitter results from Event 0, as presented in Table 6.1, provide a detailed understanding of the distributed test's performance across multiple sites, highlighting both the strengths of the VOICES system and areas for potential enhancement. First, the results underscore the variability in network delay between sites, largely driven by local network conditions. For instance, transmissions from FHWA to other sites, such as ANL and Mcity, exhibit higher latency values, whereas transmissions involving ANL and ORNL demonstrate relatively lower latency. This variability points to the influence of factors like geographic distance, local internet infrastructure, and network congestion at each site, underscoring the importance of localized conditions in distributed testing scenarios.

Sender	**Receiver**				
	FHWA	**ORNL**	**ANL**	**UCLA**	**Mcity**
FHWA	-	305.9 $(+/-52.4)$	381.4 $(+/-85.4)$	189.3 $(+/-7.6)$	242.5 $(+/-23)$
ORNL	165.5 $(+/-4.4)$	-	65.8 $(+/-5.3)$	115.7 $(+/-8)$	80.93 $(+/-6.7)$
ANL	151.1 $(+/-12.72)$	66.0 $(+/-6.1)$	-	94.8 $(+/-5.9)$	61.6 $(+/-5.1)$
UCLA	187.2 $(+/-4.1)$	116.4 $(+/-7.1)$	95.7 $(+/-2.8)$	-	112.7 $(+/-9.3)$
Mcity	164.7 $(+/-6.3)$	78.5 $(+/-6.2)$	62.48 $(+/-5.8)$	120.6 $(+/-11.7)$	-

Table 6.1: Network Latency and Jitter Results (ms).

In terms of system stability, the relatively low jitter values across all site interactions reveal that, while network delays fluctuate, the VOICES system itself is capable of maintaining steady communication without introducing additional disturbances. Low jitter means that packet transmission remains consistent and predictable, which is crucial for time-sensitive applications in CDA testing, where actions between agents are required to be broadcast in real time. This stability demonstrates that the VOICES platform is robust enough to support complex, real-time coordination among multiple entities, thereby affirming the system's capability to support distributed testing at scale. The consistent jitter performance, even in the presence of variable network latency, indicates that the VOICES system's internal processing and data handling efficiently manages diverse network conditions without impacting real-time testing fidelity.

An interesting observation is that certain entities, such as ORNL and ANL, maintain relatively low latency and jitter in most scenarios, regardless of whether they are sending or receiving data. This consistency could be attributed to the quality of their network setup, possibly indicating the use of advanced internet services or lower latency connectivity. Future research could benefit from standardizing network quality across sites or utilizing high-performance network services (such as dedicated fiber lines or low-latency connections) to minimize latency discrepancies. Doing so would not only streamline communication across entities but also enable smoother interactions in distributed testing, which is essential for real-time CDA experiments. In this context, enhancing network stability across sites could make distributed testing even more reliable and further optimize the VOICES system for complex, multi-agent experiments.

Pilot 2 Event 1: Interoperability Compatibility Test (SIL Test) Result

Event 1 in the distributed test series was designed to integrate the SAE Basic Safety Message (BSM) protocol across multiple sites, aiming to assess the system's capability to handle standardized communication in a distributed environment. As shown in Table , the results reveal that while the inclusion of BSM messages introduced a slight increase in jitter, this did not translate into significant transmission delays. This minimal impact on latency suggests that the local decoding processes at each participating site were sufficiently efficient, managing the added workload from BSM processing effectively. The ability to integrate BSMs without disrupting real-time performance demonstrates the robustness of the VOICES system in managing complex data exchanges across geographically separated entities.

Despite the efficient handling of the added encoding and decoding processes at each

Site	Mean (ms)	Min (ms)	Max (ms)	Jitter (ms)
FHWA	155.952	138.610	169.784	44.963
ORNL	114.615	37.091	314.512	20.971
ANL	52.595	35.917	88.875	10.740
UCLA	121.824	48.637	326.353	19.373
Mcity	151.234	67.892	361.859	17.032

Table 6.2: Data Transmission Time (ms) with Basic Safety Messages (BSM).

site, network conditions emerged as a key determinant of overall system performance. The additional transmission load generated by the BSMs, combined with the frequent data exchange through the VOICES system, highlighted the variability in network conditions across sites. Sites with stable network connections experienced less delay, while those with less reliable connections saw higher variability. This observation suggests that network stability could be a limiting factor in distributed testing and emphasizes the importance of high-quality network infrastructure. The VOICES system proved capable of handling real-time communication loads effectively, yet it remains partially dependent on the local network conditions at each site.

Lastly, while the VOICES system effectively supported SAE BSM messaging under the current test's controlled traffic conditions, scaling to higher traffic volumes presents additional challenges. As traffic volume and complexity increase, so will the number and size of messages, potentially adding strain to both the VOICES platform and the network infrastructure. This test validated the feasibility of distributed testing in a low-traffic setting, but it also underscores the importance of optimizing message selection and prioritizing relevant data exchanges to manage network and computational loads in high-traffic scenarios. Future research should focus on strategies for selective communication, ensuring the most impactful messages are shared among entities to maintain system efficiency as the network scales up.

Pilot 2 Event 2: Comprehensive Distributed Test Result

The distributed test data section presents the performance data of all participating vehicles during the distributed test, including metrics such as distance, speed, acceleration, and energy consumption. These tests aim to validate that all agents can effectively interact with each other, even when operating on different systems and running different algorithms. The primary focus of validation lies in comparing the performance of vehicle strings operating with and without the eco-approach algorithm. A positive impact from the eco-approach algorithm, despite system noises and delays, would indicate that the agents are successfully interacting and that the individually operating algorithms are contributing to overall system performance. This validation would confirm that the VOICES platform can establish a distributed environment where different systems and algorithms can effectively cooperate and enhance overall performance.

Downtrack Distance The downtrack distances over time for all vehicles are shown in Figure 6.10. These plots display the distance progression for five vehicles—'FHWA-CARMA,' 'ANL-DYNO,' 'ORNL-Auto,' 'Mcity-CAV-01,' and 'UCLA-OpenCDA'—comparing their performance without the eco-approach algorithm (left) and with the eco-approach algorithm (right). In the plot without the eco-approach, the vehicles demonstrate less consistent progressions, with abrupt changes and pauses, particularly visible in 'ORNL-Auto' and 'Mcity-CAV-01,' which show noticeable slowdowns around the 600-time mark. This indicates inefficiencies in maintaining coordinated distances and potential delays in response.

In contrast, the plot with the eco-approach algorithm shows smoother and more consistent distance profiles for all vehicles. There is better spacing between vehicles, especially towards the later stages of the test, indicating improved coordination and efficiency. The more gradual and balanced changes in distance over time suggest that the eco-approach algorithm facilitates better cooperation among vehicles, optimizing their movement and likely reducing energy consumption and unnecessary acceleration or deceleration. This highlights the positive impact

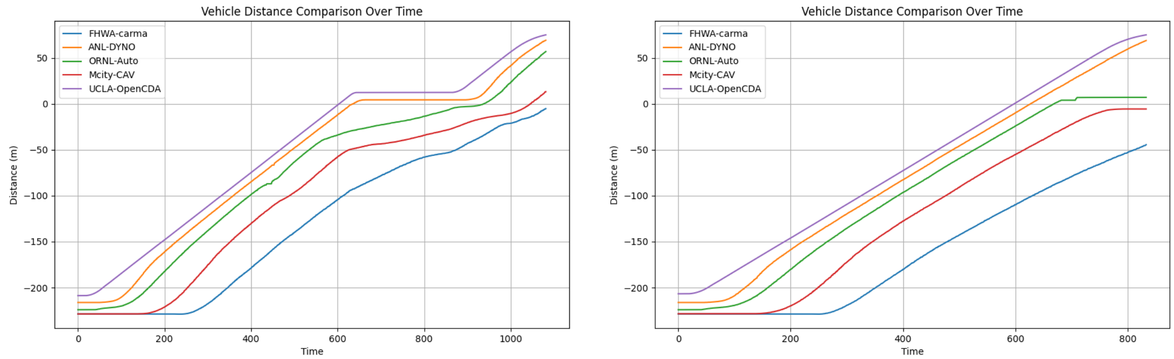


Figure 6.10: Downtrack distance over time plot for all vehicles. The left figure presents the result without the eco-approach algorithm, and the right figure presents the result with the eco-approach algorithm.

of the eco-approach algorithm on system performance.

Speed over Time Profiles The speed plots in Figure 6.11 compare the performance of five vehicles—'FHWA-CARMA,' 'ANL-DYNO,' 'ORNL-Auto,' 'Mcity-CAV-01,' and 'UCLA-OpenCDA'—without the eco-approach algorithm (left) and with the eco-approach algorithm (right). In the plot without the eco-approach algorithm, the vehicles display erratic speed patterns, with frequent spikes and fluctuations, particularly between time 200 and 600. Vehicles like 'ANL-DYNO' and 'ORNL-Auto' exhibit sharp increases and decreases in speed, indicating inconsistent driving behavior, which likely results in inefficient energy usage and less coordinated vehicle movement. In contrast, the plot with the eco-approach algorithm shows a more stable and controlled speed pattern, especially after time 200. While there are still some variations, they are noticeably less pronounced, and the vehicles generally maintain more consistent speeds. This smoother driving behavior indicates improved coordination and interaction between the vehicles, leading to more efficient overall system performance. The reduced volatility in speed changes highlights the effectiveness of the eco-approach algorithm in promoting steady, cooperative driving, contributing to reduced energy consumption and enhanced stability across the distributed test environment.

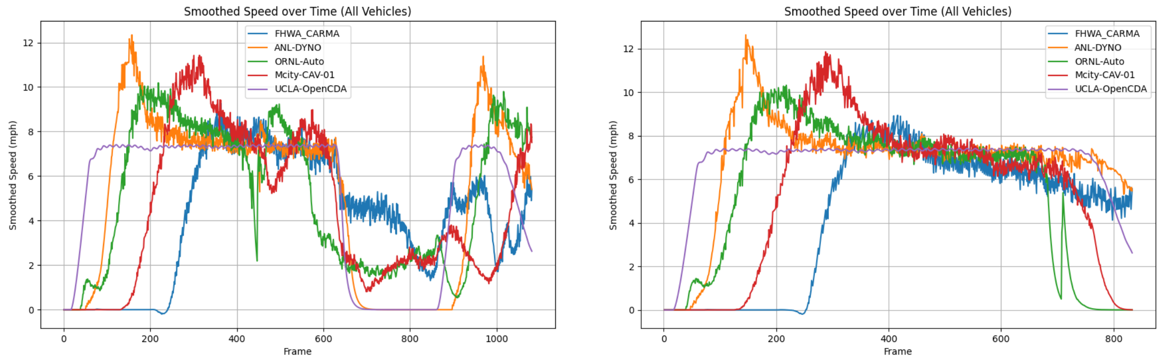


Figure 6.11: Velocity (magnitude, mph) over time plot for all vehicles. The left figure presents the result without the eco-approach algorithm, and the right figure presents the result with the eco-approach algorithm.

Acceleration over Time Profiles The acceleration plots, as shown in Figure 6.12, compare the performance of five vehicles—'FHWA-CARMA,' 'ANL-DYNO,' 'ORNL-Auto,' 'Mcity-CAV-01,' and 'UCLA-OpenCDA'—without the eco-approach algorithm (left) and with the eco-approach algorithm (right). In the plot without the eco-approach algorithm, the vehicles exhibit more erratic acceleration and deceleration patterns, with frequent spikes and fluctuations across all vehicles, particularly between 200 and 600 seconds. These sharp changes in acceleration indicate higher energy consumption due to constant speed adjustments, leading to reduced system efficiency and increased wear on vehicle components. In contrast, the plot with the eco-approach algorithm displays smoother and more stable acceleration profiles. The reduced fluctuations suggest that vehicles are operating in a more energy-efficient manner, with fewer drastic speed changes required. This stability reflects enhanced system performance as vehicles interact more harmoniously, reducing unnecessary energy expenditure. The eco-approach algorithm significantly improves energy efficiency by promoting smoother acceleration patterns, thus enhancing both vehicle and system performance within the distributed test environment.

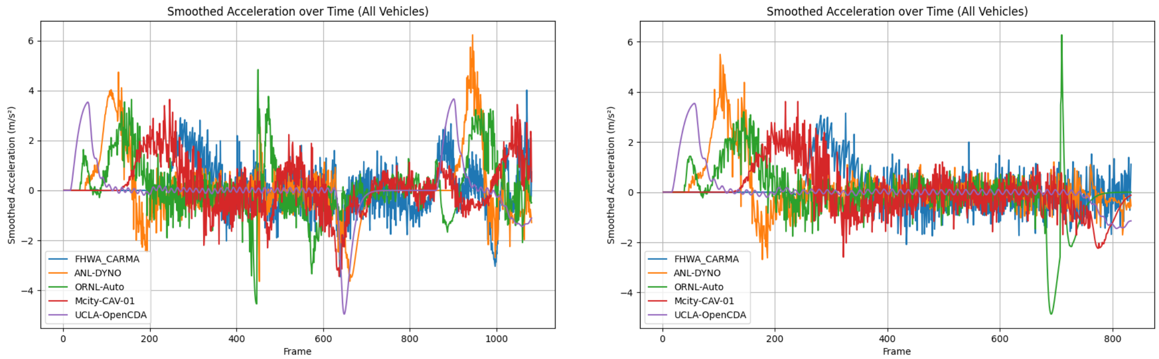


Figure 6.12: Acceleration over time plot for all vehicles. The left figure presents the result without the eco-approach algorithm, and the right figure presents the result with the eco-approach algorithm.

Energy Consumption In Pilot 2 Event 2, energy consumption was measured to evaluate the impact of the eco-approach strategy on various automated driving systems. This event included multiple runs to analyze how surrounding traffic influences different vehicle control algorithms under eco-driving conditions. To ensure accuracy, the data shown in the Table 6.3 reflects averaged values across five runs, providing reliable insights into the energy efficiency of each system. The results, recorded in watt-hours per mile (Wh/mi), highlight the effectiveness of the eco-approach (left column) compared to driving without it (right column). Across all test sites, the vehicles displayed a notable increase in energy consumption during Run 2, where the eco-approach was disabled. For instance, the energy usage for ANL's vehicle rose from 359.5 Wh/mi in Run 1 to 393.8 Wh/mi in Run 2. Similarly, Mcity's vehicle increased from 432.8 Wh/mi to 467.7 Wh/mi, while UCLA's went from 381.1 Wh/mi to 400 Wh/mi. ORNL's vehicle saw an increase from 335.3 Wh/mi to 348.4 Wh/mi, indicating that the eco-approach strategy provided energy savings across different automated driving algorithms.

For more accurate energy estimates, the powertrain and component specifications of the ANL research vehicle (a 2020 Chevrolet Bolt) were modeled in detail using the Autonomie tool, which allowed for specific drive cycles to be assigned to each vehicle configuration. This approach ensured that the recorded energy consumption accurately reflected each vehicle's

Site	Energy Consumption with Eco-Approach (Wh/mi)	Energy Consumption without Eco-Approach (Wh/mi)
ANL	359.5	393.8
Mcity	432.8	467.7
UCLA	381.1	400.0
ORNL	335.3	348.4
FHWA	377.2	400.7

Table 6.3: Averaged Energy Consumption (Wh/mi) for Pilot 2 Event 2 with and without Eco-Approach.

performance under diverse traffic scenarios, isolating the benefits of eco-driving strategies in a distributed, multi-vehicle testing environment. The results underscore the potential of the eco-approach strategy to promote energy efficiency in cooperative driving applications, with implications for future CDA deployments in mixed-traffic environments.

Insights and Potential Improvement of the VOICES System

Integrating real-world agents with simulations introduces inherent synchronization challenges, making it difficult to achieve the precise alignment seen in purely simulation-based tests. In a simulation-only environment, entities pause until the slowest one completes its computations, maintaining timing alignment across the board. However, with real-world assets like L3-capable ADS vehicles and dynamometer machines, pausing physical systems to wait for simulations is not feasible. This real-time dependency makes the system vulnerable to internet delays and network latency fluctuations, as even a slight delay from one site can disrupt overall timing and potentially compromise the experiment's integrity. To mitigate these challenges, the VOICES system uses a multi-threaded message-handling mechanism to process simulation data from all entities in parallel. This approach maximizes efficiency by handling each data stream independently, allowing simulations to advance quickly while minimizing idle time for physical assets. While this solution partially addresses network delay issues in the current tests, further advancements are needed to scale up. As more sites participate, the risk of synchronization issues grows, highlighting the importance of advanced strategies to manage network-induced disruptions in larger, distributed setups.

Another important consideration is the limited traffic volume involved in the current test. This phase of testing primarily aimed to validate the concept of distributed testing and to demonstrate the functionality and reliability of the VOICES system under relatively controlled conditions. As such, handling high traffic volumes was not within the scope of this trial. However, for large-scale deployment, the ability to support increased traffic density becomes critical. With a higher volume of vehicles, both the number of messages exchanged and the data size transmitted across the network would likely grow exponentially. This surge could potentially strain the system's communication capacity and impact performance. Accordingly, the next phase should focus on optimizing communication by identifying and transmitting only the most meaningful messages necessary for maintaining cooperative behaviors, such as collision avoidance or energy-efficient driving. Additionally, careful selection of communication participants, or determining which vehicles or infrastructure elements need to share data in real time, will be essential. This approach could prevent message overload and ensure efficient network usage, paving the way for scalable and resilient CDA implementations.

6.4 Conclusion

The distributed test conducted under Pilot 2 was a success, effectively validating the concept of hosting a joint experiment with heterogeneous CDA agents interacting within a mutual environment. This test demonstrated that diverse systems, operating independently yet collaboratively, could successfully exchange information and influence one another's actions in real time. The proposed parallel development and testing framework effectively realized this distributed test. The parallel environment enabled the integration of a mixture of CDA systems, combining real-world assets like L3-capable ADS vehicles and dynamometers with simulation-based entities, ensuring that both physical and virtual agents could coexist within the same testbed. Scenario engineering played a crucial role in designing three progressively challenging events, each targeting specific testing goals—from basic network functionality to interoperability with SAE standard messaging and, finally, to a fully distributed test of eco-approach strategies. The use of parallel operations facilitated the coordination and control of each heterogeneous agent, allowing them to manage ego-vehicle behaviors and make informed decisions. Altogether, the framework provided a robust foundation that not only validated the viability of distributed testing with multiple CDA systems but also set a precedent for future deployments.

Chapter 7

Conclusion and Future Works

Automated driving and Cooperative Driving Automation (CDA) are transforming transportation by enabling vehicles to communicate, coordinate, and operate with unprecedented levels of safety and efficiency. This dissertation presents a comprehensive framework for advancing CDA through a parallel development and testing approach designed to bridge the gap between prototype testing and real-world deployment. The parallel framework forms the foundation for scalable and reliable CDA applications by integrating parallel systems, scenario engineering, and coordinated operations. Detailed within this dissertation, the framework's structured approach has proven essential for simulating, testing, and deploying advanced CDA systems across varied environments and with diverse agents, addressing both the technical and logistical complexities of large-scale CDA implementation.

The work first demonstrates the framework's application in developing a multi-lane platooning algorithm. Through simulation testing and field experiments, the platooning algorithm proved its potential for enhancing traffic flow and vehicle coordination across multi-lane highway environments, highlighting the framework's ability to facilitate complex cooperative maneuvers in controlled and real-world settings. This work then extends into the VOICES distributed testing initiative, where the framework was instrumental in coordinating multiple entities from different sites, each running unique CDA systems. The distributed testing environment validated the capacity of CDA agents to interact across a shared virtual and physical landscape, establishing the framework's effectiveness for multi-agent, multi-site CDA validation.

Finally, the dissertation explores a regulation-aware path planning algorithm for intersection scenarios, addressing one of the most challenging aspects of CDA deployment—adherence to dynamic traffic regulations. This application underscores the framework's flexibility, demonstrating how CDA systems can be designed to interpret and respond to complex regulatory scenarios reliably. Overall, the parallel development and testing framework presented in this dissertation sets a new standard for scalable CDA testing, ensuring that prototypes can be rigorously evaluated and confidently deployed in real-world conditions. This work contributes both foundational knowledge and practical insights, positioning the field for further innovation and paving the way for safer, more efficient automated transportation systems. Future research should focus on broadening CDA validation and deployment, emphasizing advanced digital twin environments to enhance testing fidelity and integrate virtual and physical settings seamlessly. Key areas include cooperative perception for real-time sensor data sharing, algorithms for work zone detection and safe navigation, and emergency vehicle detection and response systems to prioritize safety and compliance. Additionally, validating CDA applications under complex, dynamic traffic scenarios will ensure scalability and reliability. These advancements will pave the way for efficient, robust autonomous transportation systems and their integration into multi-modal mobility solutions.

Bibliography

[1] Samir M Abdelrazek, Hazem M El-Bakry, NE Mastorakis, SM Abd El-Razek, WF Abd El-Wahed, and N Mastorakis. Collaborative virtual environment model for medical e-learning. In *Proceedings of the 9th WSEAS international conference on applied computer and applied computational science*, pages 191–195, 2010.

[2] Amirul Ibrahim Abu Bakar, Mohd Azman Abas, Mohd Farid Muhamad Said, and Tengku Azrul Tengku Azhar. Synthesis of autonomous vehicle guideline for public road-testing sustainability. *Sustainability*, 14(3), 2022.

[3] Adekunle Adebisi, Yan Liu, Bastian Schroeder, Jiaqi Ma, Burak Cesme, Anxi Jia, and Abby Morgan. Developing highway capacity manual capacity adjustment factors for connected and automated traffic on freeway segments. *Transportation Research Record*, 2674(10):401–415, 2020.

[4] Sadeq Almeaibed, Saba Al-Rubaye, Antonios Tsourdos, and Nicolas P Avdelidis. Digital twin analysis to promote safety and security in autonomous vehicles. *IEEE Communications Standards Magazine*, 5(1):40–46, 2021.

[5] Ramin Arvin, Asad J Khattak, Mohsen Kamrani, and Jackeline Rio-Torres. Safety evaluation of connected and automated vehicles in mixed traffic with conventional vehicles at intersections. *Journal of Intelligent Transportation Systems*, 25(2):170–187, 2020.

[6] Jonas Bärgman, Kip Smith, and Julia Werneke. Quantifying drivers' comfort-zone and dread-zone boundaries in left turn across path/opposite direction (ltap/od) scenarios. *Transportation Research Part F: Traffic Psychology and Behaviour*, 35:170–184, 2015.

[7] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo–simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.

[8] Ghanishtha Bhatti, Harshit Mohan, and R Raja Singh. Towards the future of smart electric vehicles: Digital twin technology. *Renewable and Sustainable Energy Reviews*, 141:110801, 2021.

[9] Pavle Bujanovic, Taylor Lochrane, Jia Hu, Tomislav Bujanovic, and C Michael Walton. Cooperative adaptive cruise control algorithm with priority weights assigned to downstream vehicles for increased safety. Technical report, FHWA, 2018.

[10] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[11] Mustafa Ridvan Cantas, Ozgenur Kavas, Santhosh Tamilarasan, Sukru Yaren Gelbal, and Levent Guvenc. Use of hardware in the loop (hil) simulation for developing connected autonomous vehicle (cav) applications. Technical report, SAE Technical Paper, 2019.

[12] Riccardo Cespi, Renato Galluzzi, Ricardo A Ramirez-Mendoza, and Stefano Di Gennaro. Artificial intelligence for stability control of actuated in–wheel electric vehicles with carsim® validation. *Mathematics*, 9(23):3120, 2021.

[13] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse:

3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019.

[14] Anton Chistyakov. A software architecture for large multi-simulation experiments over ad hoc networks using ns-3 discrete-event network simulator. In *2014 European Modelling Symposium*, pages 403–408. IEEE, 2014.

[15] Kyunghoon Cho, Timothy Ha, Gunmin Lee, and Songhwai Oh. Deep predictive autonomous driving using multi-agent joint trajectory prediction and traffic rules. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2076–2081, 2019.

[16] City of Los Angeles. City of Los Angeles Vehicle Code, 2023.

[17] Hugh C Crenshaw and Leah Edelstein-Keshet. Orientation by helical motion—ii. changing the direction of the axis of motion. *Bulletin of mathematical biology*, 55:213–230, 1993.

[18] Erwin de Gelder and Jan-Pieter Paardekooper. Assessment of automated driving systems using real-life scenarios. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 589–594. IEEE, 2017.

[19] Jitske de Vries, Elia Trevisan, Jules van der Toorn, Tuhin Das, Bruno Brito, and Javier Alonso-Mora. Regulations aware motion planning for autonomous surface vessels in urban canals. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3291–3297. IEEE, 2022.

[20] Miriam Di Russo, Simeon Iliev, Kevin Stutenberg, and Trevor Crain. Microsimulation-based evaluation of an eco-approach strategy for automated vehicles using vehicle-in-the-loop. Technical report, SAE Technical Paper, 2021.

[21] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[22] M Dupuis. Opendrive® format specification rev. 1.3, vires simulationstechnologie gmbh. *Bad Aibling*, 2010.

[23] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv preprint arXiv:1807.08048*, 2018.

[24] Federal Highway Administration. Collaborative research framework for automated driving system developers and infrastructure owners and operators. Report FHWA-HOP-21-012, U.S. Department of Transportation, 2021.

[25] Martin Fellendorf and Peter Vortisch. Microscopic traffic flow simulator vissim. *Fundamentals of traffic simulation*, pages 63–93, 2010.

[26] Shuo Feng, Yiheng Feng, Xintao Yan, Shengyin Shen, Shaobing Xu, and Henry X Liu. Safety assessment of highly automated driving systems in test tracks: A new framework. *Accident Analysis & Prevention*, 144:105664, 2020.

[27] Roya Firoozi, Xiaojing Zhang, and Francesco Borrelli. Formation and reconfiguration of tight multi-lane platoons. *Control Engineering Practice*, 108:104714, 2021.

[28] Daniel J Fremont, Edward Kim, Yash Vardhan Pant, Sanjit A Seshia, Atul Acharya, Xantha Bruso, Paul Wells, Steve Lemke, Qiang Lu, and Shalin Mehta. Formal scenario-based testing of autonomous vehicles: From simulation to the real world. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[29] Noah Goodall. Comparability of automated vehicle crash databases. *arXiv preprint arXiv:2308.00645*, 2023.

[30] Yi Guo and Jiaqi Ma. Leveraging existing high-occupancy vehicle lanes for mixed-autonomy traffic management with emerging connected automated vehicle applications. *Transportmetrica A: Transport Science*, 16(3):1375–1399, 2020.

[31] Xu Han, Runsheng Xu, Xin Xia, Anoop Sathyan, Yi Guo, Pavle Bujanović, Ed Leslie, Mohammad Goli, and Jiaqi Ma. Strategic and tactical decision-making for cooperative vehicle platooning with organized behavior on multi-lane highways. *Transportation Research Part C: Emerging Technologies*, 145:103952, 2022.

[32] Xu Han, Runsheng Xu, Xin Xia, Anoop Sathyan, Yi Guo, Pavle Bujanović, Ed Leslie, Mohammad Goli, and Jiaqi Ma. Strategic and tactical decision-making for cooperative vehicle platooning with organized behavior on multi-lane highways. *Transportation Research Part C: Emerging Technologies*, 145:103952, 2022.

[33] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*, 2023.

[34] Carlos Hidalgo, Ray Lattarulo, Carlos Flores, and Joshué Pérez Rastelli. Platoon merging approach based on hybrid trajectory planning and cacc strategies. *Sensors*, 21(8):2626, 2021.

[35] Viktória Ilková and Adrian Ilka. Legal aspects of autonomous vehicles — an overview. In *2017 21st International Conference on Process Control (PC)*, pages 428–433, 2017.

[36] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[37] Shinpei Kato, Shota Tokunaga, Yuya Maruyama, Seiya Maeda, Manato Hirabayashi, Yuki Kitsukawa, Abraham Monrroy, Tomohito Ando, Yusuke Fujii, and Takuya Azumi. Autoware on board: Enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 287–296. IEEE, 2018.

[38] Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

[39] BaekGyu Kim and Eunsuk Kang. Toward large scale test for autonomous driving software in collaborative virtual environment. *Ieee Access*, 2023.

[40] Dasom Lee and David J. Hess. Regulations for on-road testing of connected and automated vehicles: Assessing the potential for global safety harmonization. *Transportation Research Part A: Policy and Practice*, 136:85–98, 2020.

[41] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3461–3475, 2022.

[42] Xuan Li, Peijun Ye, Juanjuan Li, Zhongmin Liu, Longbing Cao, and Fei-Yue Wang. From features engineering to scenarios engineering for trustworthy ai: I&i, c&c, and v&v. *IEEE Intelligent Systems*, 37(4):18–26, 2022.

[43] Yiming Li, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: A virtual collaborative perception dataset for autonomous driving. *arXiv preprint arXiv:2202.08449*, 2022.

[44] Sok Ying Liaw, Shawn Leng-Hsien Soh, Khoon Kiat Tan, Ling Ting Wu, John Yap, Yeow Leng Chow, Tang Ching Lau, Wee Shiong Lim, Seng Chee Tan, Hyekyung Choo,

et al. Design and evaluation of a 3d virtual environment for collaborative learning in interprofessional team care delivery. *Nurse education today*, 81:64–71, 2019.

[45] Hao Liu, Lin Xiao, Xingan David Kan, Steven E Shladover, Xiao-Yun Lu, Meng Wang, Wouter Schakel, and Bart van Arem. Using cooperative adaptive cruise control (cacc) to form high-performance vehicle streams. final report. 2018.

[46] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[47] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal*, 8(8):6469–6486, 2020.

[48] Yiliang Liu, Zhou Su, and Yuntao Wang. Artificial noise-assisted beamforming and power allocation for secure d2d-enabled v2v communications. In *2021 IEEE 94th vehicular technology conference (VTC2021-Fall)*, pages 01–05. IEEE, 2021.

[49] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE, 2018.

[50] Los Angeles County. Los Angeles County Vehicle Code, 2023.

[51] Nils Lubbe and Erik Rosén. Pedestrian crossing situations: Quantification of comfort boundaries to guide intervention timing. *Accident Analysis & Prevention*, 71:261–266, 2014.

[52] Jiaqi Ma, Jia Hu, Ed Leslie, Fang Zhou, Peter Huang, and Joe Bared. An eco-drive experiment on rolling terrains for fuel consumption optimization with connected

automated vehicles. *Transportation Research Part C: Emerging Technologies*, 100:125–141, 2019.

[53] Jiaqi Ma and Yisheng Lv. The ucla mobility labs [its research lab]. *IEEE Intelligent Transportation Systems Magazine*, 16(1):203–208, 2024.

[54] Jiaqi Ma, Fang Zhou, Zhitong Huang, and Rachel James. Hardware-in-the-loop testing of connected and automated vehicle applications: a use case for cooperative adaptive cruise control. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2878–2883. IEEE, 2018.

[55] Yining Ma, Chen Sun, Junyi Chen, Dongpu Cao, and Lu Xiong. Verification and validation methods for decision-making and planning of automated vehicles: A review. *IEEE Transactions on Intelligent Vehicles*, 7(3):480–498, 2022.

[56] Santa Maiti, Stephan Winter, Lars Kulik, and Sudeshna Sarkar. Ad-hoc platoon formation and dissolution strategies for multi-lane highways. *Journal of Intelligent Transportation Systems*, 27(2):161–173, 2023.

[57] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[58] Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 45(1):1049–1060, 1998.

[59] Jesús Mena-Oreja and Javier Gozalvez. Permit-a sumo simulator for platooning maneuvers in mixed traffic scenarios. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3445–3450. IEEE, 2018.

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

[61] Demin Nalic, Hexuan Li, Arno Eichberger, Christoph Wellershaus, Aleksa Pandurevic, and Branko Rogic. Stress testing method for scenario-based testing of automated driving systems. *IEEE Access*, 8:224974–224984, 2020.

[62] Ilja Nastjuk, Bernd Herrenkind, Mauricio Marrone, Alfred Benedikt Brendel, and Lutz M Kolbe. What drives the acceptance of autonomous driving? an investigation of acceptance factors from an end-user's perspective. *Technological Forecasting and Social Change*, 161:120319, 2020.

[63] P Norving and SJ Russell. Artificial intelligence a modern approach: Vol, 2010.

[64] State of California. California vehicle code, § 21453, 2024. Retrieved from https://leginfo.legislature.ca.gov/faces/codes.xhtml.

[65] State of California. California vehicle code, § 21760, 2024. Retrieved from https://leginfo.legislature.ca.gov/faces/codes.xhtml.

[66] State of California. California vehicle code, § 22358, 2024. Retrieved from https://leginfo.legislature.ca.gov/faces/codes.xhtml.

[67] On-Road Automated Driving (ORAD) committee. Sae j3216 standard: Taxonomy and definitions for terms related to cooperative driving automation for on-road motor vehicles. In *SAE International*, 2020.

[68] National Committee on Uniform Traffic Laws. *Uniform vehicle code*, volume 5. Department of Commerce, Bureau of Public Roads, 1952.

[69] OpenAI. ChatGPT, 2021.

[70] C Passos, S Nazir, AC Mol, and PV Carvalho. Collaborative virtual environment for training teams in emergency situations. *Chem Eng*, 53:217–222, 2016.

[71] Germán Pizarro and Felipe Núñez. Graph-based distributed lane-change in tight multi-lane platoons. In *2021 ieee conference on control technology and applications (ccta)*, pages 1031–1036. IEEE, 2021.

[72] Kallirroi N Porfyri, Evangelos Mintsis, and Evangelos Mitsakis. Assessment of acc and cacc systems using sumo. *EPiC Series in Engineering*, 2:82–93, 2018.

[73] Vincenzo Punzo, Maria Teresa Borzacchiello, and Biagio Ciuffo. On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (ngsim) program data. *Transportation Research Part C: Emerging Technologies*, 19(6):1243–1262, 2011.

[74] Rui Qian, Xin Lai, and Xirong Li. 3d object detection for autonomous driving: A survey. *Pattern Recognition*, 130:108796, 2022.

[75] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[76] Sankar B Rengarajan, Scott Hotz, Charles Hirsch, Peter Lobato, Michael Gross, Purser Sturgeon, and Jayant Sarlashkar. Test methodology to quantify and analyze energy consumption of connected and automated vehicles. Technical report, SAE Technical Paper, 2019.

[77] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020.

[78] Anoop Sathyan, Jiaqi Ma, and Kelly Cohen. Decentralized cooperative driving automation: a reinforcement learning framework using genetic fuzzy systems. *Transportmetrica B: Transport Dynamics*, 9(1):775–797, 2021.

[79] Chris Schwarz and Ziran Wang. The role of digital twins in connected and automated vehicles. *IEEE Intelligent Transportation Systems Magazine*, 14(6):41–51, 2022.

[80] Rana Shaaban and Saleh Faruque. Cyber security vulnerabilities for outdoor vehicular visible light communication in secure platoon network: Review, power distribution, and signal to noise ratio analysis. *Physical communication*, 40:101094, 2020.

[81] Yunli Shao, Mohd Azrin Mohd Zulkefli, Zongxuan Sun, and Peter Huang. Evaluating connected and autonomous vehicles using a hardware-in-the-loop testbed and a living lab. *Transportation Research Part C: Emerging Technologies*, 102:121–135, 2019.

[82] Saeid Soleimaniamiri, Xiaopeng Shaw Li, Handong Yao, Amir Ghiasi, Govind Vadakpat, Pavle Bujanovic, Taylor Lochrane, John Stark, David Hale, Sujith Racha, et al. Cooperative automation research: Carma proof-of-concept transportation system management and operations use case 4-dynamic lane assignment. Technical report, United States. Federal Highway Administration, 2021.

[83] Burak C Soydas. The impact of programming language choice on execution time when performing virtual simulation with a driver model: A comparison of c++ and python performance using the open simulation interface (osi) in esmini. 2024.

[84] State of California. California vehicle code, 2024. Retrieved from https://leginfo.legislature.ca.gov/faces/codes.xhtml.

[85] A. Stevens and J. Hopkin. Benefits and deployment opportunities for vehicle/roadside cooperative its. In *IET and ITS Conference on Road Transport Information and Control (RTIC 2012)*, pages 1–6, 2012.

[86] A Stevens and J Hopkin. Benefits and deployment opportunities for vehicle/roadside cooperative its. 2012.

[87] Kevin Stutenberg, Henning Lohse-Busch, Michael Duoba, Simeon Iliev, Forrest Jehlik, and Miriam Di Russo. An overview of argonne's advanced mobility technology laboratory vehicle systems instrumentation and evaluation methodology. 2021.

[88] Jian Sun, He Zhang, Huajun Zhou, Rongjie Yu, and Ye Tian. Scenario-based test automation for highly automated vehicles: A review and paving the way for systematic safety assurance. *IEEE transactions on intelligent transportation systems*, 23(9):14088–14103, 2021.

[89] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[90] Zsolt Szalay. Next generation x-in-the-loop validation methodology for automated vehicle systems. *IEEE Access*, 9:35616–35632, 2021.

[91] Curtis R Taylor, Jason M Carter, Shean Huff, Eric Nafziger, Jackeline Rios-Torres, Bob Zhang, and Joseph Turcotte. Evaluating efficiency and security of connected and autonomous vehicle applications. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 236–239. IEEE, 2022.

[92] Waymo Team. Self-driving car technology for a reliable ride. 2023.

[93] Oguz Tengilimoglu, Oliver Carsten, and Zia Wadud. Implications of automated vehicles for physical road environment: A comprehensive review. *Transportation research part E: logistics and transportation review*, 169:102989, 2023.

[94] Tim Tiernan, Pavle Bujanovic, Philip Azeredo, Wassim G Najm, Taylor Lochrane, et al. Carma testing and evaluation of research mobility applications. Technical report, John A. Volpe National Transportation Systems Center (US), 2019.

[95] Atsuya Uno, Takeshi Sakaguchi, and Sadayuki Tsugawa. A merging control algorithm based on inter-vehicle communication. In *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No. 99TH8383)*, pages 783–787. IEEE, 1999.

[96] Fei-Yue Wang, Yilun Lin, Petros A Ioannou, Ljubo Vlacic, Xiaoming Liu, Azim Eskandarian, Yisheng Lv, Xiaoxiang Na, David Cebon, Jiaqi Ma, et al. Transportation 5.0: The dao to safe, secure, and sustainable intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[97] Ziran Wang, Kyungtae Han, and Prashant Tiwari. Digital twin-assisted cooperative driving at non-signalized intersections. *IEEE Transactions on Intelligent Vehicles*, 7(2):198–209, 2021.

[98] Ziran Wang, Xishun Liao, Xuanpeng Zhao, Kyungtae Han, Prashant Tiwari, Matthew J Barth, and Guoyuan Wu. A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–6. IEEE, 2020.

[99] Ziran Wang, Guoyuan Wu, and Matthew Barth. Distributed consensus-based cooperative highway on-ramp merging using v2x communications. Technical report, SAE Technical Paper, 2018.

[100] Xin Xia, Zonglin Meng, Xu Han, Hanzhao Li, Takahiro Tsukiji, Runsheng Xu, Zhaoliang Zheng, and Jiaqi Ma. An automated driving systems data acquisition and analytics platform. *Transportation research part C: emerging technologies*, 151:104120, 2023.

[101] Hao Xiang, Runsheng Xu, Xin Xia, Zhaoliang Zheng, Bolei Zhou, and Jiaqi Ma. V2xp-asg: Generating adversarial scenes for vehicle-to-everything perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3584–3591. IEEE, 2023.

[102] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162. IEEE, 2021.

[103] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE, 2022.

[104] Xin Xia Jiaqi Ma Xu Han, Matt Huffman. Regulation-aware path planning framework for automated driving systems. In *TRB Annual Meeting 2024 Posters*, January 2024. Poster presentation at the Transportation Research Board 102nd Annual Meeting, Washington, DC.

[105] Stanley E Young, Erik A Bensen, Lei Zhu, Christopher Day, J Sam Lott, Rimple Sandhu, Charles Tripp, and Peter Graf. Concept of operations of next-generation traffic control utilizing infrastructure-based cooperative perception. In *International Conference on Transportation and Development 2022*, pages 93–104, 2022.

[106] Qingzhao Zhang, David Ke Hong, Ze Zhang, Qi Alfred Chen, Scott Mahlke, and Z. Morley Mao. A systematic framework to identify violations of scenario-dependent driving rules in autonomous vehicle software. *Proc. ACM Meas. Anal. Comput. Syst.*, 5(2), jun 2021.

[107] Fangfang Zheng, Can Liu, Xiaobo Liu, Saif Eddin Jabari, and Liang Lu. Analyzing the impact of automated vehicles on uncertainty and stability of the mixed traffic flow. *Transportation research part C: emerging technologies*, 112:203–219, 2020.