

A Research Testbed for Intelligent and Cooperative Driving in Mixed Traffic

Jiaying Lu¹, *Student Member, IEEE*, Sanzida Hossain², Wakun Lam¹, Weihua Sheng¹, *Senior Member, IEEE*, and He Bai², *Member, IEEE*

Abstract—Autonomous vehicles are gradually entering the transportation system. The traffic will become more heterogeneous since both autonomous and human-driven vehicles will share the roads. Cooperative driving, by promoting synchronized actions and shared situational awareness among vehicles, can significantly enhance driving safety. On the other hand, understanding human drivers is a pivotal step for cooperative driving in such mixed traffic environments, which facilitates effective interaction between human drivers and their vehicles. This paper presents a testbed that can be used to conduct research in intelligent and cooperative driving. The testbed consists of driving simulators, custom-designed copilots with an Artificial Intelligence engine, an optimization server, and a cloud database. The copilot is capable of sensing and understanding the human driver, the vehicle and the traffic. It can assist the driver by providing timely alerts on potential risks. Most importantly, it can communicate with other nearby vehicles for cooperative driving. Two case studies are presented to validate and evaluate the testbed. The first case study demonstrates the performance of the copilot in human distraction detection and driving assistance. The second case study focuses on cooperative driving between one human-driven vehicle and two connected autonomous vehicles in a lane-changing scenario. We expect this research testbed to be used in various research projects that involve human-driven vehicles and connected autonomous vehicles.

Index Terms—Cooperative Driving, Model Predictive Control, Convolutional Neural Network

I. INTRODUCTION

THE autonomous vehicle (AV) market is growing rapidly, with an expected size of 60 billion dollars in 2026 [1]. With the rapid advancement in computer hardware and Artificial Intelligence (AI) technologies [2], AVs are attracting a significant amount of interest from both industry and academia [3], [4]. Because of the prospective huge market of commercial AVs, self-driving technologies have been developed by both traditional car manufacturers and non-traditional technology companies. For example, Google introduced the first self-driving car on real-world roads [5]. Ford Motor Company developed self-driving cars for less friendly environments [6]. As more and more intelligent vehicles are introduced into the transportation system, the traffic will inevitably become more heterogeneous since both emerging vehicles with various levels of driving automation [7] and legacy human-driven vehicles with different Advanced Driver-Assistance Systems (ADAS) will co-exist on the roads.

Ensuring the safety of such a complex mixed transportation system is not trivial, which should leverage the differences between machines and humans in sensing, analytics and control, therefore calling for a human-centered perspective to understand human driving behaviors and their impact on driving safety. To achieve this, there is a need to introduce an intelligent copilot that can observe and understand the human driver, vehicle and traffic while providing driving advice when necessary. Comprehending and modeling human behavior is the initial and most critical step in developing a Human-in-the-Loop (HITL) system, which not only facilitates effective human-vehicle interaction but also enhances vehicle-to-vehicle (V2V) communication. Subsequently, through V2V communication and collaborative decision making, cooperative driving among vehicles can significantly reduce the risk of collision and increase the traffic throughput [8]. However, past research mainly focused on cooperative driving of automated vehicles, such as the PATH project [8] and, most recently, the Cooperative Automation Research Mobility Applications (CARMA) project by Department of Transportation (DoT) Federal Highway Administration (FHWA) [9], among many others [10]–[12]. Several projects approached the cooperative driving problem in mixed traffic from the perspective of Connected Autonomous Vehicles (CAVs) only, such as [13]–[17], which allowed CAVs to accommodate nearby human-driven vehicles but did not actively involve the latter in cooperative driving. It is our belief that the benefit of cooperative driving cannot be fully exploited in mixed traffic if legacy manned vehicles are excluded from cooperative driving. Therefore it would be highly desirable to develop cooperative driving that encompasses both emerging CAVs and legacy human-driven vehicles.

However, conducting research in intelligent and cooperative driving using real vehicles in real traffic is not only dangerous but also costly. Simulation with driving simulators is an alternative to test the algorithms before they are deployed into the real world. With the rapid progress in computer simulation technology, modeling vehicles' dynamics and realizing it in computers is feasible and has gained much attention in recent years.. Although there has been a significant amount of research in intelligent vehicles that use driving simulators, there is very limited work in developing simulation testbeds that can be used for cooperative driving between human-driven vehicles and CAVs. To bridge this gap, the Human-in-the-Loop (HITL) methodology emerges as a critical tool. By monitoring driver behaviors and incorporating their decision making process, HITL provides a more detailed and

¹Jiaying Lu, Wakun Lam, and Weihua Sheng (corresponding author, email: weihua.sheng@okstate.edu) are with Electrical and Computer Engineering, Oklahoma State University, Stillwater, Oklahoma, 74078, USA. ²Sanzida Hossain and He Bai are with Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma, 74078, USA.

accurate assessment of cooperative driving dynamics in mixed traffic environments, surpassing the traditional simulations. Developing such a testbed consists of the following tasks: 1) developing an intelligent copilot, an enhanced ADAS, for monitoring and driving assistance; 2) integrating the copilot into the driver simulator to facilitate cooperative driving with other simulated vehicles; 3) incorporating reliable human driver behavior models to understand human drivers; and 4) developing a reliable agent for real-time optimization and decision-making in cooperative driving.

This paper aims to develop such a simulation testbed. The main contributions of this paper are as follows. First, this research develops a copilot for human-driven vehicles, which collects various data regarding the driver, vehicle, and traffic to evaluate the potential risk during driving. Compared to traditional ADAS, the copilot is able to communicate with adjacent vehicles and facilitate cooperative driving with them. Second, utilizing the AI engine in the copilot, driving behavior monitoring is developed that recognizes eight different driving behaviors. Third, the mathematical formulation for a multi-vehicle lane merging involving both autonomous and human-driven vehicles is proposed. Fourth, the performance of the testbed is experimentally evaluated through two case studies: driver monitoring in standalone driving and 3-vehicle lane merging in cooperative driving.

This paper is organized as follows. Section II introduces the related work. Section III describes the overall hardware architecture of the simulation testbed. The software framework is introduced in Section IV. Section V formulates the methodologies for the two case studies: driver monitoring in standalone driving and 3-vehicle lane merging in cooperative driving. Section VI presents the experiments and results for the above two case studies. We conclude the paper and outline the future research in Section VII.

II. RELATED WORK

A. Cooperative Driving of Automated Vehicles

In recent years, cooperative driving of automated vehicles has been studied in multiple research projects, such as the California PATH project [8] that allowed multiple automated vehicles to conduct platooning in an Intelligent Vehicle Highway System, the CHAUFFEUR project [10] that implemented the Automated Highway Systems on European motorways, the SARTRE project [11] that realized the platooning of trucks and cars, and the Energy ITS project [18] that improved the energy efficiency of fully-automated trucks. With a goal of improving safety, mobility, and environment sustainability through V2V or vehicle to infrastructure (V2I) communication, the CARMA platform enables research on Cooperative Driving Automation (CDA) which leverages emerging capabilities in automation and cooperation to advance Transportation Systems Management and Operations (TSMO) strategies [9]. Jang *et al.* presented a V2I communication framework to improve the decision-making process in autonomous driving through a machine to machine based cooperative driving protocol [19]. Cao *et al.* proposed a consensus framework for cooperative autonomous driving that utilizes V2V communication to enhance

the overall driving performance [20]. Yuan *et al.* introduced a stable and efficient Reinforced Cooperative CAV Collision Avoidance framework, in which a deep learning approach was implemented in order to effectively avoid collision among the CAVs [21]. Liu *et al.* proposed a distributed model predictive control (DMPC) approach for cooperative and flexible vehicle platooning in order to ensure safety and stability of the platoon [22]. Cui *et al.* introduced Coopernaut, an end-to-end learning framework for vision-based cooperative driving that transmitted LiDAR information between vehicles [23]. Jie *et al.* proposed a cooperative lane change control strategy for intelligent connected vehicles to improve traffic flow, reduce energy consumption, and enhance driving safety [24]. They used V2V communication to coordinate lane changes. Simulations show that the strategy can reduce congestion and energy consumption while maintaining safe driving conditions.

These existing projects significantly improved the performance of CAVs in cooperative driving. However, they did not consider the scenarios that involve human-driven vehicles, mainly because cooperation between human-driven vehicles and CAVs is more complicated due to the uncertainty and unpredictability of human driving behaviors.

B. Cooperative Driving Among Heterogeneous Vehicles

Cooperative driving among CAVs and human-driven vehicles began to receive attention in recent years. For example, Xie *et al.* developed two cooperative driving strategies for CAVs in heterogeneous traffic to stabilize the traffic flow [25]. They believed that further research regarding developing cooperative driving strategies based on the prediction for human-driven vehicles is needed. Valiente *et al.* developed a Multi-agent Reinforcement Learning (MARL) algorithm with a decentralized framework and a reward function to enable CAVs to learn from human-driven vehicles [26]. In order to tackle the conflicts caused by the difference between the intention of CAVs and human-driven vehicles in mixed traffic, Li *et al.* presented an indirect shared control method to improve the driving performance [27]. Mosharafian *et al.* introduced a Cooperative Adaptive Cruise Control (CACC) system with a hybrid stochastic predictive approach for lane changing in mixed traffic [28] [29]. Liu *et al.* extended the CACC modeling framework to describe interactions between CACC and manually driven vehicles in mixed traffic, which enhanced lane changing rules and implemented automated speed control involving realistic freeway dynamics [30]. A human driver model was used to update the position and speed of the manually driven vehicles in their simulation. Hu *et al.* presented a simulation platform for evaluating CACC in urban settings, focusing on the interaction between human-driven vehicles and CACC systems [31]. In their simulation, they created a model that emulates human control, mimicking how a human driver would maneuver a vehicle. Xiao *et al.* investigated the impact of CACC systems on highway bottleneck areas, focusing on the effects of CACC deactivation and switch to the human-driven mode under certain conditions [32]. A modified version of the Intelligent Driver Model (IDM) was utilized to simulate the manually human driver. Zhou *et*

al. demonstrated the potential of multi-agent reinforcement learning for cooperative lane changing of CAVs in mixed traffic environments, highlighting its ability to improve traffic efficiency and safety [33].

Overall, cooperative driving involving both CAVs and human-driven vehicles is still in its infancy. The existing work mainly focused on the control of the CAVs while the human-driven vehicles were not actively involved in cooperative driving. In addition, most cooperative driving systems for mixed traffic assumed that human-driven vehicles follow a constant driving behavior, which is not realistic [33]. In order to facilitate their active participation, it is necessary to equip the human-driven vehicles with intelligent driving assistance systems and V2V communication capabilities, justifying an HITL-based system instead of only simulated human-driven vehicles.

C. Simulation Testbed

Developing and testing cooperative driving involving multiple vehicles is both risky and costly. Simulations provide an alternative approach. Recent years have seen several simulation testbeds developed for intelligent and autonomous vehicle research. CARLA [34] is an open-source simulator that integrates Unreal Engine [35] to provide a highly realistic 3D environment for simulating a variety of driving scenarios, allowing for the testing of autonomous driving algorithms and related technologies in a safe environment. With a client-server architecture, CARLA allows multiple human-driven vehicles to be controlled by multiple drivers. V2V communication is achievable with a python API interface that CARLA provides. While CARLA is a powerful and flexible platform for research and development in autonomous driving, it has some limitations. CARLA's performance degrades when simulating large-scale scenarios with a large number of vehicles. Its computational requirements also limit its ability to efficiently handle highly dynamic and congested environments. The National Advanced Driving Simulator (NADS) [36] offers a tool for research in transportation safety and driver behavior analysis, with a realistic driving experience for the driver and the ability to collect physiological data. However, the NADS equipment is costly and has very limited scalability for cooperative driving. SimNet [37] is a simulation software developed for training and evaluating military equipment and operations in a distributed fashion. It can simulate different scenarios and equipment and can test the interactions between different vehicles and systems in real-time. However, its high cost and complexity make it difficult to acquire and use for researchers with limited resources.

Therefore it is crucial to develop a driving simulation testbed for cooperative driving in mixed traffic scenarios, which offers a highly customizable and realistic environment for testing and evaluation with low computational demand and good scalability.

III. HARDWARE ARCHITECTURE OF SIMULATION TESTBED

We developed a simulation testbed for cooperative driving that features low computational cost, reconfigurability and

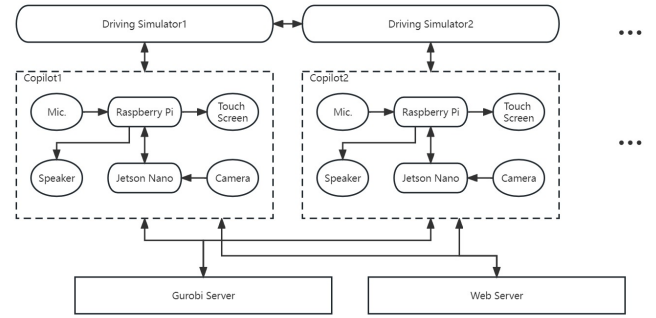


Fig. 1: The architecture of the simulation testbed.

scalability. The overall architecture of the testbed is shown in Figure 1 and the testbed setup is shown in Figure 2. The testbed is developed based on the Carnetsoft driving simulator [38], which offers a realistic driving experience. A copilot is developed to offer driving assistance to human-driven vehicles. The copilot consists of a Raspberry Pi, a Jetson Nano, and various peripheral sensors for driver and vehicle monitoring. In this setup, the Raspberry Pi is responsible for scheduling tasks while the Jetson Nano is utilized for on-demand machine learning tasks, therefore enhancing the modularity and practicality of the copilot. The copilot communicates with the Carnetsoft simulator through the provided APIs, enabling it to receive the vehicle data such as speed, acceleration, and location. A driving simulator equipped with a copilot simulates an intelligent human vehicle (IHV). The testbed supports multiple vehicles, including both IHVs and CAVs, therefore enabling the simulation of various traffic scenarios and a wide range of vehicle interactions. To facilitate cooperative driving control, all copilots are connected to a central Gurobi server [39] for solving the optimization problems. In addition, all copilots are connected to a Web server for data logging and viewing.

A. Carnetsoft Simulator

The Carnetsoft simulation software runs on a desktop Windows computer equipped with an Intel Core-i7 4790 CPU and an NVIDIA GTX 970 GPU. There are a steering wheel, a shifter, pedals, and three monitors that provide a large in-vehicle view. Carnetsoft defines a programmable script language for controlling the objects and the events in the driving scenario. In the script, vehicles are registered as objects with pre-defined attributes, including driving data variable and configurations. Additionally, pedestrians may also be added as objects. The script incorporates multiple “scenario” functions, which serve as interrupt functions and are invoked under user-defined circumstance. In the beginning of the script, a local map of the driving scenario is loaded. Carnetsoft provides a tool for creating maps with various elements, including roads, intersections, buildings, and traffic lights.

To enable scalability and collaboration, each driving simulator can accommodate multiple CAVs while simulating one human-driven vehicle. The simulators exchange the data of vehicles with each other. The data includes information about



Fig. 2: The cooperative driving testbed setup.

the vehicles' position, speed, heading, and lateral position. Upon receiving the information, each simulator can accurately represent and display the other vehicles, thereby simulating the interactions between the ego human-driven vehicle and other vehicles. The interconnected setup and the data exchange mechanism allow the driving simulators to collaboratively simulate scenarios involving both human-driven vehicles and CAVs. With this arrangement, scalability is achieved as multiple simulators can efficiently share data and coordinate their simulations. Consequently, a comprehensive and realistic environment is created for testing and evaluating the performance of the cooperative driving in mixed traffic.

B. Copilot

As shown in Figure 3, the copilot is an in-vehicle intelligent system mounted near the dashboard, serving as a driving assistant and facilitating cooperative driving. It consists of a Raspberry Pi as the main controller, an NVIDIA Jetson Nano as the AI engine, a microphone, a camera, a speaker and a touch screen. To distribute the computational tasks, we separate the software into two parts. The communication/scheduling software runs on the Raspberry Pi. The Jetson Nano, as an AI engine, runs the machine learning services to monitor the driver and the vehicle, which is usually computationally intensive. In the future, a forward facing camera can be used to capture the traffic scenes and the AI engine can analyze the data for traffic monitoring and external risk analysis.

IV. SOFTWARE OF SIMULATION TESTBED

As shown in Figure 4, the software of the cooperative driving testbed runs in a client-server architecture. The communication/scheduling packages run in the Raspberry Pi of the corresponding copilot. The machine learning modules in the Jetson Nano provide driver monitoring services, such as drowsiness detection, distraction detection, etc. The Gurobi server running on the server computer provides mathematical optimization services for cooperative driving to the copilot. Finally, the database stores the real time data sent by the copilots, and the website displays the experimental data to viewers. The Robot Operating System (ROS) [40] is adopted

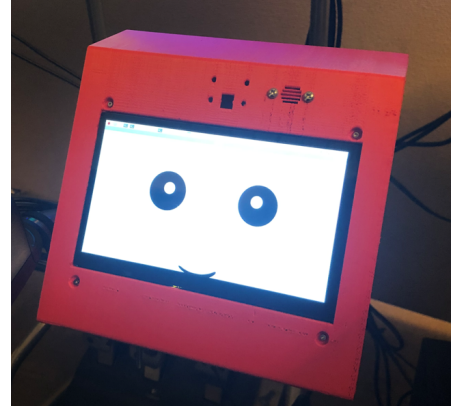


Fig. 3: The prototype of the copilot.

in the design of the software. ROS is a flexible framework for developing robot software that offers tools for hardware abstraction, low-level device control, and message-passing between processes, making it widely popular in robotics and automation. Its modular architecture and open-source nature facilitate easy development and sharing of robotic applications.

A. Copilot software

The copilot software running on the Raspberry Pi consists of a backend and a front end. The backend has several ROS nodes for a variety of tasks, including: 1) a V2V node that communicates with the driving simulator and other vehicles' copilots; 2) a driver understanding node that periodically analyzes the drivers' status with the help of the Jetson Nano; 3) a data logging node that sends vehicle and driver status to the database; 4) a solver request node that communicates with the Gurobi server; and 5) a vehicle interface node that sends vehicle control commands to the simulator.

The front end running on the Raspberry Pi interfaces the speaker, the touch screen, and the microphone. Normally, it displays an animated face until receiving the advice sent by the backend. The advice consists of two parts: voice advice and visual advice. When a visual advice is received, for example, the interface will display a circular chart, similar to the speed gauge on the dashboard, except that it displays both the current speed and the optimal speed. Voice advice will also be announced through the speaker at the same time. In addition, the front end can be easily moved to a mobile device, such as a smart phone or a mobile tablet.

The copilot can run in two different modes: the standalone mode and the cooperative driving mode. In the standalone mode, the driver monitor node periodically receives the driver status data from the Jetson Nano. When the status is abnormal, such as distraction and drowsiness, the driver monitor node will alarm the driver through the front end, typically in the form of a voice reminder. In the cooperative driving mode, the solver request node collects the vehicles' status and sends them to the Gurobi server which analyzes the situation and generates an optimized coordination solution. Upon receiving the solution, the solver request node sends the advice to the driver through the front end. Both the standalone mode and the cooperative driving mode can run at the same time.

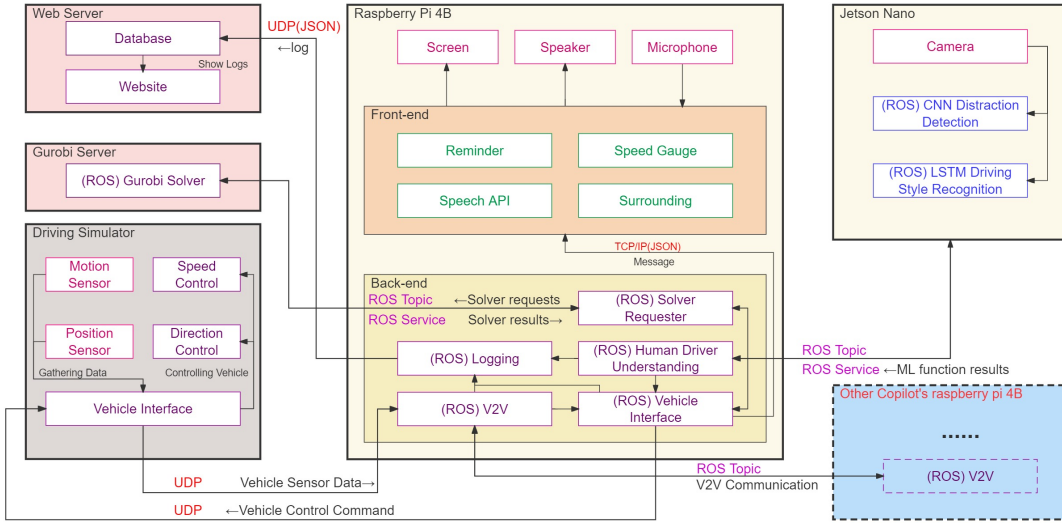


Fig. 4: The system architecture of the simulation testbed.

B. Data Logging on Server

A Flask web server runs on the server computer to collect and display the data generated by the simulated vehicles during the experiments, which makes debugging and testing easier. The logged data consists of the following: 1) vehicle ID; 2) vehicle velocity; 3) vehicle location (longitudinal and lateral position); 4) time; 5) pedal percentage; 6) steering wheel angle; 7) driver drowsiness; 8) driver ID. The stored data can be sorted and displayed in a web browser. The data can also be downloaded as a whole file to any device. A screenshot of the displayed data is shown in Figure 5.

id	time	type	status	intention	velocity	locationX	locationY	distance	latPos	direction
1	5	human-driven	safe	1	15	-1	200.34	200.34	4.5	90
1	5.01	human-driven	safe	1	15	-1	201.12	201.12	4.5	90
1	5.02	human-driven	safe	1	15	-1	201.9	201.9	4.5	90
1	5.03	human-driven	safe	1	15	-1	202.68	202.68	4.5	90
1	5.04	human-driven	safe	1	15	-1	203.46	203.46	4.5	90

Fig. 5: A screen shot of the data display.

V. INTELLIGENT AND COOPERATIVE DRIVING

To evaluate the performance of the testbed for intelligent and cooperative driving, we conduct two case studies: one is standalone driving in which the copilot alerts the driver upon detection of distraction; the other is cooperative driving in which multiple vehicles cooperate to accomplish a lane merging maneuver. The first case study, illustrating intelligent driving, showcases human-vehicle interaction within our simulation testbed, while the second case study focuses on IHV-CAV cooperation.

A. Distraction Detection in Standalone Driving

Distraction detection plays a significant role in ensuring driving safety on the roads. The rise of smartphone usage and other distracting activities while driving has inevitably

led to an increased risk of accidents. The recognition of brake light activation in leading vehicles, indicative of deceleration or braking, has emerged as a prevalent theme within vehicle market [41]. However, for human-driven vehicles, the primary factor leading to accidents after their leading vehicles' sudden braking is the level of driver distraction [42], [43]. Recognizing the need to address this issue, we develop a robust method for real-time distraction detection to mitigate the potential hazards.

Connected with a camera, the Jetson Nano in the copilot is capable of capturing images and processing them in real time. The camera view is mainly focused on the human driver's head. As shown in Figure 6a, the driver is attentively driving, with his eyes focused on the road ahead, exhibiting a responsible and focused driving behavior. Figure 6f shows the driver using his right hand to hold the phone and talking on it, which exemplifies a scenario of driver distraction, as the driver's attention is divided between the conversation and the task of driving. By analyzing such images, the Jetson Nano can accurately identify and classify the driver's state.

A 50-layer convolutional neural network (Residual Network, ResNet-50) is trained with more than 60,000 images collected by the copilot's camera on the simulation testbed. As shown in Table I, the architecture of the ResNet-50 model [44] makes it capable of learning complex patterns in the data efficiently. However, the depth of ResNet-50 does bring about increased computational demand. We found that with the Jetson Nano, the enhanced feature detection capabilities of ResNet-50 justified its selection, despite the higher computational cost.

The image classification algorithm considered 8 behaviors, which consist of 1 normal driving behavior and 7 distracted driving behaviors:

- c0: safe driving
- c1: playing the phone with head down
- c2: talking on the phone using the left hand
- c3: talking on the phone using the right hand



Fig. 6: Original images of the driver for the neural network model before resizing and cropping.

TABLE I: ResNet-50 model architecture

layer name	output size	50-layer	
conv1	112×112	$7 \times 7, 64$	
conv2_x	56×56	$3 \times 3, \text{max pool}$	
		$1 \times 1, 64$	$\times 3$
		$3 \times 3, 64$	
		$1 \times 1, 256$	
conv3_x	28×28	$1 \times 1, 128$	$\times 4$
		$3 \times 3, 128$	
		$1 \times 1, 512$	
		conv4_x	14×14
$3 \times 3, 256$			
$1 \times 1, 1024$			
conv5_x	7×7		
		$3 \times 3, 512$	
		$1 \times 1, 2048$	
			1×1

- c4: drinking/eating
- c5: reaching behind
- c6: talking to the passenger
- c7: doing hair and makeup

During driving, the copilot backend sends a request for the driver's driving status through a ROS message. Upon receiving the request, the AI engine invokes the camera, continuously captures the image, and classifies it with the trained ResNet-50 model. The result is sent back to the copilot backend. Once a distraction behavior is detected, the copilot checks if the vehicle is in a critical situation where a potential collision

could occur and the driver does not pay attention to it. If yes, a reminder is generated by the copilot backend by saying "focus on driving" through the speaker. One example of such critical situations is when the human-driven vehicle is driving behind another vehicle which is quickly slowing down.

B. Lane Merging in Cooperative Driving

As a case study of cooperative driving, we explore the coordination between CAVs and IHVs for lane merging. The copilot installed in the IHV communicates and collaborates with other vehicles to accomplish cooperative driving. In order to ensure a safe and efficient merge, it's critical to arrive at the necessary safe longitudinal space between the vehicles as soon as possible. The control inputs can directly control the motion of a CAV, while the human driver controls the motion of the IHV. The driver's actions may be affected by the copilot's advisory directives. To coordinate the movements of the vehicles, we formulate a stochastic model predictive control (MPC) problem with state and control constraints. The solution to this problem provides the IHVs and CAVs with optimal advisory instructions and autonomous controls, respectively.

System modeling

We consider the following linear dynamic model of a CAV

$$x_{k+1}^r = A_r x_k^r + B_r u_k^r, \quad (1)$$

where $k \in \mathbb{Z}_+$ is the discrete-time index, $x_k^r \in \mathbb{R}^2$ contains the longitudinal position and velocity with respect to the origin, A_r and B_r are matrices of suitable dimensions that define the vehicle dynamics, and $u^r \in \mathbb{R}$ is the input (acceleration) to the CAV.

For the IHV, the dynamics of the system depends on the driver's compliance with the advised directives. As a result, the dynamics of the IHV alternate between two dynamic systems depending on a binary decision variable $x_k^B \in \{0, 1\}$: (1) following the advisory command when the IHV is under advisory control ($x_k^B = 1$), and (2) not following the advisory command, which denotes that the IHV is under human control ($x_k^B = 0$).

$$\text{IHV under human control: } x_{k+1}^h = A_h x_k^h + B_h f_k^h \quad (2)$$

$$\text{IHV under advisory control: } x_{k+1}^h = A_h x_k^h + B_h u_k^a, \quad (3)$$

where A_h and B_h are matrices defining the IHV dynamics, and we assume that f_k^h is available through a driver monitoring system and $u_k^a \in \mathbb{R}$ is the advisory commands for the IHV. One can deduce the solution to x_k as follows:

$$x_k^r = A_r^k x_0^r + \sum_{j=0}^{k-1} A_r^{k-j-1} B_r u_j^r, \quad (4)$$

$$x_k^h = A_h^k x_0^h + \sum_{j=0}^{k-1} A_h^{k-j-1} B_h f_j^h + \sum_{j=0}^{k-1} A_h^{k-j-1} B_h z_j^u \quad (5)$$

where $z_k^u = x_k^B (u_k^a - f_k^h)$.

We use a stochastic finite state machine (sFSM) to model the stochastic transitions of the binary human state x_k^B following [45]. The detailed formulation is shown in Appendix.

System Constraints

The following four sets of constraints are taken into account for each pair of vehicles while coordinating the movements of the vehicles.

- 1) sFSM transition constraints: The stochastic human state transitions of the IHV formulated by the sFSM are enforced using this set of constraints.
- 2) State constraints: The new state $z_k^u = x_k^B(u_k^a - f_k^h)$ formulates the transition of different inputs to the system based on x_k^B . This transition is modeled in the state constraints. Additionally, the state limits are enforced in these constraints.
- 3) Merging constraints: The longitudinal position difference between a pair of vehicles must be greater than a threshold s_k in order to avoid a collision during lane merging. The value of s_k depends on the relative position of the merging vehicle with respect to the other vehicle. To ensure the longitudinal distance between a CAV and an IHV is larger than s_k , the following constraint is considered:

$$|x_{k,1}^r - x_{k,1}^h| \geq s_k \quad (6)$$

where $x_{k,1}$ denotes the position state.

- 4) Chance constraints: Chance constraints are used to reject trajectories that only occur with a small probability from the set of possible solutions.

All the above constraints can be combined in the form,

$$G_k \theta_k \leq g_k, \quad (7)$$

where $\theta_k \in \mathbb{R}^{n_t}$ is the vector of decision variables, $G_k \in \mathbb{R}^{n_c \times n_t}$ and $g_k \in \mathbb{R}^{n_c \times 1}$ where n_c is the total number of constraints. The total number of decision variables is denoted by n_t .

Cost function

We take into account five goals in the cost function:

- 1) Minimize the control inputs to the CAVs and IHVs based on their respective weights. This is a quadratic function of θ_k ;
- 2) Minimize the time to reach the desired longitudinal merging distance.
- 3) Maximize both vehicles' speeds within a speed limit. This is a linear function of θ_k ;
- 4) Minimize the number of advisory actions so that the merge can occur with fewer advisory actions. This is a linear function of θ_k ;
- 5) Maximize the probability of the stochastic events and human input stochasticity. This is a linear function of θ_k

The objective function of the MPC is the sum of the aforementioned five functions weighted by user preference, which can be represented as

$$J(\theta_k) = \theta_k^\top Q \theta_k + c^\top \theta_k, \quad (8)$$

where $Q \in \mathbb{R}^{n_t \times n_t}$ and $c \in \mathbb{R}^{1 \times n_t}$ are the designed objective weights for the system.

Optimization problem

Assuming the human state x_k^B is a known parameter, the optimization problem can be formulated as

$$\min_{\theta_k} J(\theta_k), \quad s.t. \quad G_k \theta_k \leq g_k. \quad (9)$$

Applying this optimization algorithm in a receding horizon fashion with the constraints and defined objective, we get the MPC solution, which consists of the control inputs applied to the CAVs and the advisory commands communicated to the IHVs at each time step.

VI. CASE STUDY AND RESULTS

This section presents the results of the two case studies: distraction detection in standalone driving and lane merging in cooperative driving.

A. Distraction Detection

The dataset consists of 50,000 images and is divided into training, evaluation, and testing subsets with a 8:1:1 ratio. The data collection involves 14 volunteers. To construct the evaluation subset and the test subset, we select the data from two specific volunteers as the primary components. Subsequently, we randomly sample images from the remaining 12 volunteers and allocate them to both the evaluation and test datasets, ensuring an 8:1:1 ratio. This division allocates 40,000 images for training. The evaluation subset has 5,000 images and is used to examine the performance of the model during training. At the end, the testing subset, also containing 5,000 images, serves as the independent evaluation to measure the model's generalization ability on unseen data. The testing result is shown in the confusion matrix in Figure 7. One can notice that the overall accuracy of the classification reaches 95%.

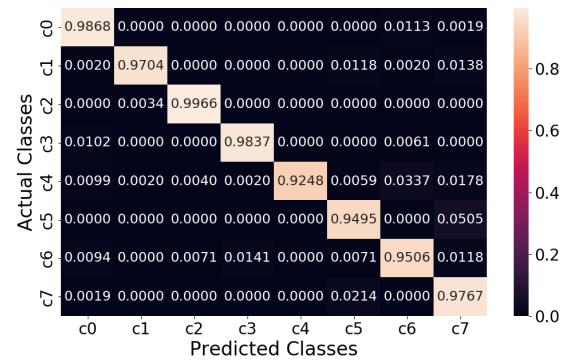


Fig. 7: Confusion matrix.

We also conducted experiments on the simulation testbed to assess the AI engine's ability to detect distractions in real time. In the experiment, a distracted human driver drove an IHV which followed a CAV with an initial distance of 35 meters. Initially, the CAV was driving at the same speed with the IHV before it quickly slowed down to be 5 meters/second slower than the IHV, which invoked the copilot to monitor the potential collision and request the AI engine to detect the driver's status. At that moment, the human driver had a

distracted driving behavior as he was talking to a passenger, without noticing the slowing down of the CAV. The copilot recognized the driver's distraction behavior and was aware of the short distance between the two vehicles, which prompted it to remind the driver to 'focus on driving'. Comparative tests were conducted where the AI engine was not running and the copilot did not remind the driver. 6 volunteers conducted 30 tests with reminding and 20 tests without reminding.

As an example of a test conducted with the AI engine on, Figure 8a shows the speed of both vehicles and Figure 8b shows the distance between the vehicles, with time point 0 in x axis representing the moment when the CAV quickly slowed down. The AI engine alerted the driver by saying "focus on driving" when the distance between the vehicles was less than 30 meters. The human driver reacted and pressed the brake pedal, thereby successfully maintaining the safe distance and avoiding a collision. Conversely, Figure 8c depicts the speed of both vehicles in an experiment without the advising of the copilot, and Figure 8d shows the distance between the two cars. One can notice that without the AI engine advising, the distracted human driver kept the same speed and a collision happened after the 6th second.

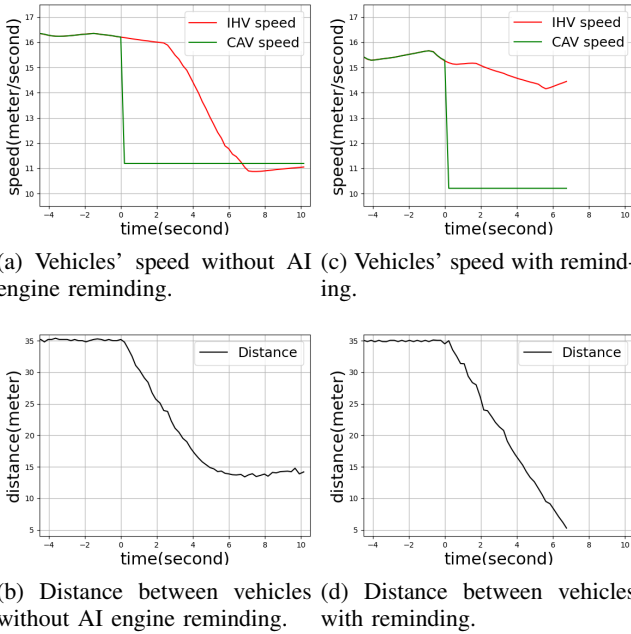


Fig. 8: Vehicles' speed and distance between vehicles.

Based on the analysis of the experimental results, we learned that when reminders were provided, the average shortest distance between the IHV and the CAV was 11.045 meters. In contrast, in tests without reminders, the average shortest distance between them reduced to 5.023 meters, leading to collisions. Furthermore, in tests with reminders, the average reaction time, measured from the moment when the distance between the CAV and the IHV fell below 30 meters and the copilot detected driver distraction, to the point when the human driver successfully followed the advice and decelerated the IHV to a speed slower than the CAV to ensure safety, was found to be 4.133 seconds. In contrast, in tests without reminders, the average reaction time, beginning at the same

time point but ending with the occurrence of a collision, was 4.938 seconds.

B. 3-vehicle Lane Merging

To showcase the effectiveness of the proposed cooperative driving testbed, experiments were conducted involving 3-vehicle lane-merging scenarios. Four scenarios were considered, each featuring distinct initial location settings. In all scenarios, a human-driven vehicle Va located in the right lane was required to merge into the left lane, where CAVs Vb and Vc were present. Figure 11 shows the in-vehicle view of the IHV during the 3-vehicle lane merging. The top view window, situated at the bottom left on the central monitor, provides the locations of the CAVs Vb and Vc.

With the connection between the copilots and the simulator, the copilot backend publishes vehicle data as a ROS message to all vehicles immediately after receiving the corresponding vehicle data from the driving simulator. Once the merging intention is detected, the backend requests the Gurobi server to optimize the cooperative driving as an mixed integer optimization problem. For each time step of 0.8 seconds, the backend transmits its vehicle data, including speed, location, and acceleration, to the Gurobi server, which is deployed on a remote powerful computer. After receiving data, the Gurobi server needs approximately 0.3 to 0.5 seconds to compute the optimal vehicle acceleration control commands. The optimized accelerations are then transmitted to each corresponding copilot. For the copilot connecting with the CAV, the acceleration is conveyed to the simulator as the control command. On the human-driven vehicle, the copilot utilizes voice and visual cues to remind the driver to either speed up or brake as necessary.

For the first scenario, as shown in Figure 9a, initially the human-driven vehicle is in front of both CAVs regarding longitudinal location. As an example of the first scenario, Figure 10a shows three vehicles' speed and their relative location, assuming the human-driven vehicle's location is 0m at each time step. In the figure, the green curve represents data related to the human-driven vehicle Va, while the blue and the red curves correspond to data related to the CAVs Vb and Vc, respectively. Notably, time 0 is the time when the merging intention is detected, and all curves conclude at the point when a safe distance is established for the merging to start.

It is evident that the Gurobi server obtained the optimal solution in which the human-driven vehicle speeds up and both CAVs slow down. As depicted in Figure 12, which illustrates the advised and actual acceleration of the human-driven vehicle, each data point represents the average acceleration over the past 0.8 sec. Although the maneuver control conducted by the human driver was not as proficient as that of the CAVs, which followed commands 100 percent of the time, the "speed up" maneuver control was performed correctly.

As shown in Figure 9b, the human-driven vehicle Va is behind both CAVs in the second scenario. Similarly, Figure 10b demonstrates speed for the three vehicles and their relative location. In this scenario, since the human-driven vehicle is in behind, one can simply imagine the optimal solution

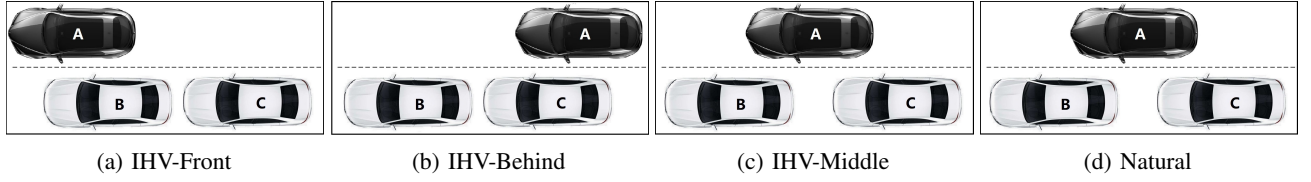


Fig. 9: Different scenarios of lane merging. Vehicle A is an IHV. Vehicle B and Vehicle C are CAVs.

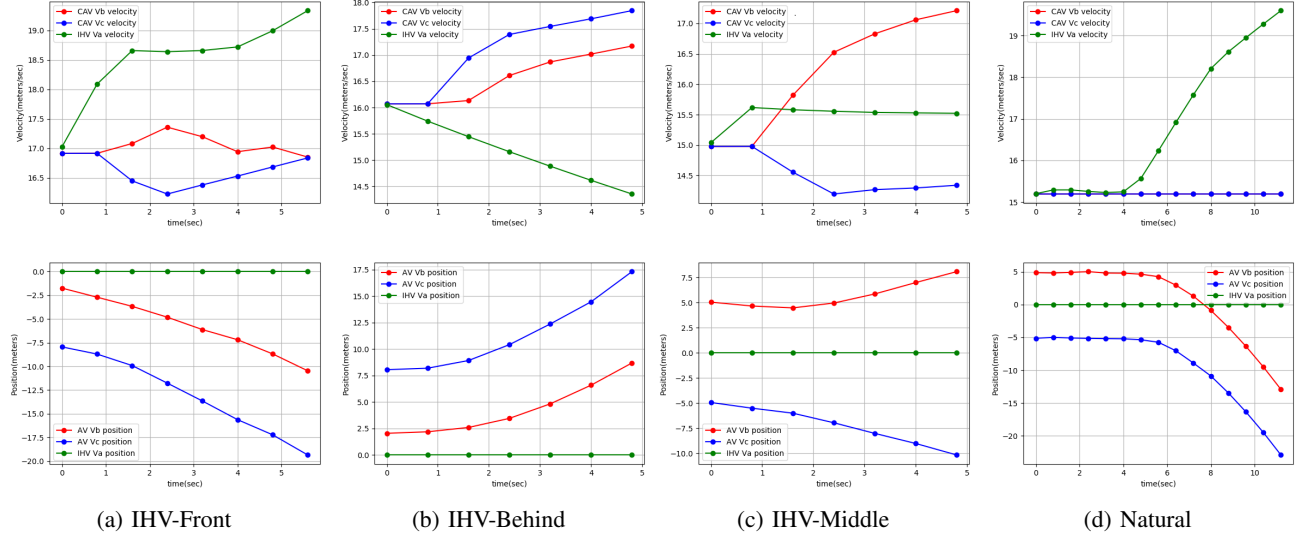


Fig. 10: Vehicles' speed and relative location.



Fig. 11: In-vehicle perspective of IHV's 3-monitor view during the 3-vehicle lane merging.

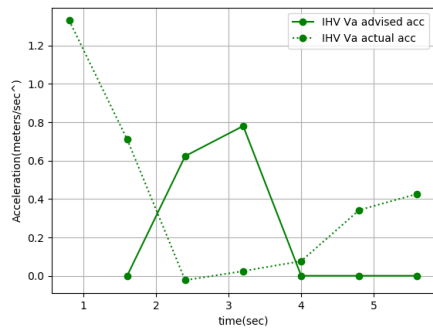


Fig. 12: Acceleration for the human-driven vehicle in the first scenario.

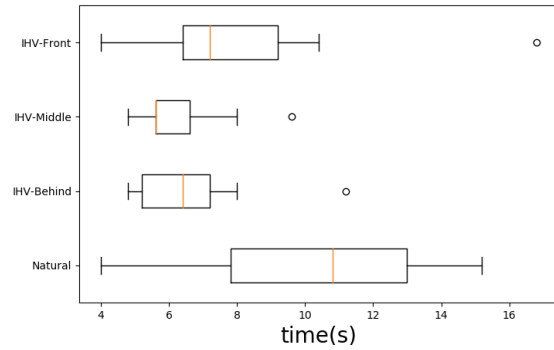


Fig. 13: Comparison of completion times across scenarios. The completion time is the duration from detecting the merging intention to establishing a safe merging distance.

would be the human-driven vehicle slowing down while CAVs accelerate. The Gurobi server obtained a similar optimal solution for the IHV, and then successfully controlled both CAVs and advised the human driver to complete the process of establishing the safety distance.

In the third scenario, the human-driven vehicle is in the middle of the CAVs regarding longitudinal distance, as shown in Figure 9c. According to Figure 10c, which shows the three vehicles' speed and relative location, we observe that the Gurobi server decides to let the CAV Vb speed up and let the CAV Vc slow down, while the human driver is advised to speed up slightly. This approach enabled a safe distance to be promptly established between vehicles Vb and Vc.

As a comparison, in the fourth scenario that featured the same initial location setting as the third scenario, the Gurobi server is turning off, resulting in both CAVs maintaining the same speed while the human driver took charge of the merging process. As indicated by the experimental results shown in Figure 10d, the process took a longer duration of time and potentially made more impact on the surrounding traffic system, given that the human-driven vehicle accelerated substantially.

Six volunteers, possessing recent and proficient driving experience, conducted the tests. For every volunteer, all four scenarios were performed 5 times in a randomized order. Figure 13 illustrates the comparative analysis of completion times across different scenarios. The completion time refers to the duration from detecting the merging intention to establishing a safe merging distance. In the figure, the boxes indicate the interquartile range of the data, with the median represented by a line inside each box. It can be observed that the test is completed notably faster with the model assisting the human driver and controlling the CAV.

VII. CONCLUSION AND FUTURE WORK

In order to address the cooperative driving problem in mixed traffic of CAVs and human-driven vehicles, this paper introduced a simulation testbed that features a copilot to assist human drivers. Two case studies were conducted to assess the performance of the testbed in intelligent and cooperative driving. The first case study focused on validating the effectiveness of the AI engine in the context of intelligent driving. The specific scenario addressed involved detecting driver distraction and alerting the driver to reduce the risk of collision with adjacent vehicles. Experimental data indicates that with copilot assistance, the average time from the initiation of a hazardous situation to its resolution is about 4.13 seconds. The second case study tested the performance of an MPC algorithm in facilitating cooperation between three vehicles, thereby enabling effective coordination and cooperation among these vehicles. Experimental results show that with copilot assistance, the time required to establish a safe distance and complete lane merging is significantly decreased. Therefore, experimental results demonstrate that the proposed testbed is effective as a platform for developing and testing algorithms related to intelligent and cooperative driving.

Further research will be carried out to explore the challenges of human driver modeling, aiming to thoroughly investigate

the HITL problem within the context of cooperative driving in mixed traffic. Particularly we will improve the performance of human-vehicle interaction by better understanding human driving behaviors and utilizing reinforcement learning for effective advising. Additionally, we will address the issues related to V2V communication, with emphasis on reducing data latency and minimizing redundant data transmissions. More comprehensive driving scenarios will also be considered, including the study of cooperative driving that involves interactions between multiple IHVs and CAVs. We will also investigate the modeling of the adjacent traditional human-driven vehicles that have no AI assistance and incorporating them into the cooperative driving system. Finally, we will extend the research to study the cooperative driving that involves the interactions between IHVs and CAVs themselves.

VIII. ACKNOWLEDGEMENT

This work is supported by the National Science Foundation (NSF) Grants CISE/IIS 1910933, EHR/DUE 1928711, CPS 2212582 and TI 2329852.

APPENDIX

A. State constraints

The new state equation $z_k^u = x_k^B(u_k^a - f_k^h)$ is enforced as constraints in the following form:

$$z_k^u \leq (M_u - f_k^h)x_k^B, \quad z_k^u \geq (m_u - f_k^h)x_k^B \quad (10)$$

$$z_k^u \leq (u_k^a - f_k^h) - (m_u - f_k^h)(1 - x_k^B) \quad (11)$$

$$z_k^u \geq (u_k^a - f_k^h) - (M_u - f_k^h)(1 - x_k^B) \quad (12)$$

where M_u and m_u are the upper and lower limit of the input.

The following constraints can be used to enforce the state limits of CAVs and IHVs:

$$x_k^r \leq M, \quad x_k^r \geq m, \quad (13)$$

$$x_k^h \leq M, \quad x_k^h \geq m, \quad (14)$$

for some upper limit M and lower limit m .

B. sFSM transition constraints

Let $u_k^B \in \{0, 1\}$ denote the on/off of an advisory control at time step k . Based on the first-order Markov assumption, we prescribe the transition probability of x_{k+1}^B given x_k^B and u_k^B . Therefore, there are 8 different possibilities for transitions [46]. We introduce a binary variable $w^i \in \{0, 1\}$ (an uncontrollable event) for each transition and constrain $w^i = 1$ if and only if the i th transition occurs. The 8 transitions and the definitions of w_k^i are given below:

$$\begin{aligned} (x_{k+1}^B = 0 | x_k^B = 0, u_k^B = 1) &\leftrightarrow w_k^1 = 1 \\ (x_{k+1}^B = 1 | x_k^B = 0, u_k^B = 1) &\leftrightarrow w_k^2 = 1 \\ (x_{k+1}^B = 0 | x_k^B = 0, u_k^B = 0) &\leftrightarrow w_k^3 = 1 \\ (x_{k+1}^B = 1 | x_k^B = 0, u_k^B = 0) &\leftrightarrow w_k^4 = 1 \\ (x_{k+1}^B = 0 | x_k^B = 1, u_k^B = 1) &\leftrightarrow w_k^5 = 1 \\ (x_{k+1}^B = 1 | x_k^B = 1, u_k^B = 1) &\leftrightarrow w_k^6 = 1 \\ (x_{k+1}^B = 0 | x_k^B = 1, u_k^B = 0) &\leftrightarrow w_k^7 = 1 \\ (x_{k+1}^B = 1 | x_k^B = 1, u_k^B = 0) &\leftrightarrow w_k^8 = 1. \end{aligned} \quad (15)$$

The probability of each event $p[w_k^i]$ needs to be specified. In particular, we let $p[w_k^2] = p_t$, which is the probability of transitioning to an advisory action. We also let $p[w_k^6] = p_f$ which is the probability of continuously following the advisory control. We set $p[w_k^3] = 1$, $p[w_k^4] = 0$, $p[w_k^7] = 1$, $p[w_k^8] = 0$. Note that $p[w_k^i] + p[w_k^{i+1}] = 1$, $i = 1, 3, 5, 7$. The p_t and p_f may be learned from human-in-the-loop experiments. Let $\delta_k^1 = x_k^B u_k^B$. Then the possible transitions of the sFSM can be enforced using the following constraints:

$$w_k^1 + w_k^2 \leq u_k^B - \delta_k^1, \quad w_k^1 + w_k^2 \geq u_k^B - \delta_k^1, \quad (16)$$

$$w_k^5 + w_k^6 \leq \delta_k^1, \quad w_k^5 + w_k^6 \geq \delta_k^1, \quad (17)$$

$$-x_k^B + \delta_k^1 \leq 0, \quad -u_k^B + \delta_k^1 \leq 0, \quad (18)$$

$$u_k^B + x_k^B \leq 1 + \delta_k^1, \quad (19)$$

$$x_k^B = \sum_{j=0}^{k-1} u_j^B + C \sum_{j=0}^{k-1} w_j, \forall k \geq 1 \quad (20)$$

where $C = [-1 \ 0 \ -1 \ 0]$.

C. Merging constraints

Let binary variables f_k^{rh} and f_k^{hr} denote the relative position of a vehicle with respect to another vehicle. If CAV is in front of IHV, then $f_k^{rh} = 1$ and $f_k^{hr} = 0$. Otherwise $f_k^{rh} = 0$ and $f_k^{hr} = 1$. This relationship forms the following inequalities.

$$x_{k,1}^r - x_{k,1}^h \geq -\bar{M} f_k^{hr}, \quad (21)$$

$$x_{k,1}^r - x_{k,1}^h \leq -\epsilon + (\bar{M} + \epsilon) f_k^{rh}, \quad (22)$$

$$f_k^{rh} + f_k^{hr} = 1, \quad (23)$$

where $\bar{M}$ is a sufficiently large positive number and ϵ is a small positive number.

Define binary variables m_k^r and m_k^h to indicate the merging vehicle. If the CAV is merging, $m_k^r = 1$ and $m_k^h = 0$. If the IHV is merging, then $m_k^r = 0$ and $m_k^h = 1$. We also define binary variables $l_k^{rh} \in \{0, 1\}$ to indicate if the CAV and IHV are in the same lane. If they are in the same lane then $l_k^{rh} = 1$ otherwise $l_k^{rh} = 0$. Let the safe following distance be d_f . For safe merging, we also consider additional rear clearance d_r while merging. The safe merging distance s_k can be calculated using the following equation:

$$s_k = (1 - l_k^{rh})(d_f + m_k^r f_k^{rh} d_r + m_k^h f_k^{hr} d_r) + l_k^{rh} d_f. \quad (24)$$

Here, s_k is the longitudinal distance threshold that the two vehicles should maintain right before a safe merging. If the CAV is merging from behind, we have $m_k^r = 1$, $m_k^h = 0$, $l_k^{rh} = 0$, $f_k^{rh} = 0$, and $f_k^{hr} = 1$. From (24), $s_k = (1 - 0)(d_f + 1 \times 0 \times d_r) + 0 \times d_f = d_f$. If the CAV is merging to the front of the IHV, $f_k^{rh} = 1$, $f_k^{hr} = 0$, and the rest would be the same. Then $s_k = d_f + d_r$. Similarly, if the IHV is merging from behind, we obtain from (24) $s_k = d_f$, and if the IHV is merging from the front, $s_k = d_f + d_r$. Thus, (24) allows incorporation of an additional clearance d_r while merging in front of a vehicle. If the two vehicles are in the same lane, they will keep a following distance as $s_k = d_f$.

Following (6), the safe distance while merging can be enforced using the following inequalities:

$$x_{k,1}^r - x_{k,1}^h \leq -s_k + \bar{M} b_{1,k}, \quad (25)$$

$$x_{k,1}^r - x_{k,1}^h \geq s_k - \bar{M} b_{2,k}, \quad (26)$$

where $x_{k,1}$ denotes the position state. We introduced two binary variables $b_{1,k}$ and $b_{2,k}$ with a sufficiently large $\bar{M}$. When $b_{1,k} = 0$ and $b_{2,k} = 1$, (25) becomes $x_{k,1}^r - x_{k,1}^h \leq -s_k$ and (26) becomes $x_{k,1}^r - x_{k,1}^h \geq s_k - \bar{M}$, which holds trivially. Similarly, when $b_{1,k} = 1$ and $b_{2,k} = 0$, (26) becomes $x_{k,1}^r - x_{k,1}^h \geq s_k$ and (25) becomes $x_{k,1}^r - x_{k,1}^h \leq -s_k + \bar{M}$, which holds trivially. Thus, $b_{1,k} + b_{2,k} = 1$ ensures that the safe distance is satisfied at each time step k .

To reduce the time to reach the condition in (6), we introduce a constraint

$$b_{1,k} + b_{2,k} \geq 1, \quad (27)$$

and minimize $b_{1,k} + b_{2,k}$ (among other objectives) in the objective function.

D. Chance constraint

In our Discrete Hybrid Stochastic Automata (DHSA) formulation, the possible human state transition events are given by $w_k = [w_k^1 \ w_k^2 \ w_k^5 \ w_k^6]^\top$ and the transition probabilities are $p = [p[w_k^1] \ p[w_k^2] \ p[w_k^5] \ p[w_k^6]]^\top$. Following [45], the probability of the state trajectory can be computed as

$$\begin{bmatrix} \pi_0 \\ \vdots \\ \pi_{K-1} \end{bmatrix} = \begin{bmatrix} w_0^\top \\ \vdots \\ w_{K-1}^\top \end{bmatrix} p, \quad (28)$$

where K is the look-ahead window in the MPC. At step k , the π_k indicates the likelihood of taking the transition described by w_k . The probability of the whole w trajectory, $\pi(w)$, is given by

$$\pi(w) = \pi(w_0, w_1, \dots, w_k) = \prod_{k=0}^{K-1} \pi_k. \quad (29)$$

Then the chance constraint can be defined as

$$\pi(w) \geq \tilde{p}_w \quad (30)$$

with $\tilde{p}_w \in [0, 1]$ being a probability bound. This chance constraint (30) enforces that w realizes with at least $\tilde{p}_w$ probability. From (30), for our system, we can compute the constraint as

$$\ln \pi(w) = \sum_{k=0}^{K-1} \sum_{i=1,2,5,6} w_k^i \ln(p[w_k^i]) \geq \ln(\tilde{p}_w). \quad (31)$$

E. Decision variables

For a look ahead window of K , the decision variables are summarized as $\theta_k \in \mathbb{R}^{n_t}$ in the form $\theta_k = [\mathbf{u}_k^r \ \mathbf{u}_k^a \ \bar{\mathbf{z}}_k \ \mathbf{u}_k^B \ \mathbf{w}_k \ \delta_k^1 \ \mathbf{b}_k \ \mathbf{f}_k^{rh} \ \mathbf{f}_k^{hr}]$ where $\mathbf{u}_k^r = [u_k^r, u_{k+1}^r, \dots, u_{k+K-1}^r]$ and all the other decision variables are defined similarly. The total number of decision variables is denoted by n_t . The continuous variables are $\mathbf{u}_k^r$,

u_k^a , and z_k^u while the rest are binary. Note that among all these decision variables, the CAV input u_k^r and the IHV advisory commands u_k^a along with the advisory state u_k^B are the control inputs entering the system dynamics.

REFERENCES

- [1] Statista Inc., "Size of the global autonomous car market in 2021, with a forecast through 2026," 2021. [Online]. Available: <https://www.statista.com/statistics/428692/projected-size-of-global-autonomous-vehicle-market-by-vehicle-type/>
- [2] S. J. Russell, P. Norvig, and E. Davis, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.
- [3] E. Dickmanns and A. Zapp, "Autonomous high speed road vehicle guidance by computer vision," *IFAC Proceedings Volumes*, vol. 20, no. 5, Part 4, pp. 221–226, 1987, 10th Triennial IFAC Congress on Automatic Control - 1987 Volume IV, Munich, Germany, 27–31 July. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017553203>
- [4] S. RUSSELL, "Darpa grand challenge winner: Stanley the robot!" 2006. [Online]. Available: <http://www.popularmechanics.com/technology/robots/a393/2169012/>
- [5] J. Markoff, "Google cars drive themselves, in traffic," 2010. [Online]. Available: <https://www.nytimes.com/2010/10/10/science/10google.html>
- [6] G. Rapier, "Ford is going all-in on self-driving cars," 2015. [Online]. Available: <http://www.businessinsider.com/ford-joins-rivals-developing-self-driving-cars-2015-6>
- [7] Society of Automotive Engineers, "Levels of driving automation," 2019. [Online]. Available: <https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic>
- [8] S. Shladover, "The california path program of ivhs research and its approach to vehicle-highway automation," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992, pp. 347–352.
- [9] Department of Transportation, "Cooperative automation research mobility applications (carma) overview," 2019. [Online]. Available: <https://highways.dot.gov/research/research-programs/operations/CARMA>
- [10] M. Schulze, "Chauffeur-the european way towards an automated highway system," in *Mobility for everyone. 4th World Congress on intelligent Transport Systems*, 1997.
- [11] E. Chan, *SARTRE Automated Platooning Vehicles*. John Wiley Sons, Ltd, 2016, ch. 10, pp. 137–150. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119307785.ch10>
- [12] R. Kunze, R. Ramakers, K. Henning, and S. Jeschke, "Organization and operation of electronically coupled truck platoons on german motorways," in *Intelligent Robotics and Applications*, M. Xie, Y. Xiong, C. Xiong, H. Liu, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 135–146.
- [13] X. Huang, A. Jasour, M. Deyo, A. Hofmann, and B. C. Williams, "Hybrid risk-aware conditional planning with applications in autonomous vehicles," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 3608–3614.
- [14] D. Sadigh, S. Sastry, and S. Seshia, "Verifying robustness of human-aware autonomous cars," *IFAC-PapersOnLine*, vol. 51, pp. 131–138, 01 2019.
- [15] K. Hawkins and P. Tsiotras, "Anticipating human collision avoidance behavior for safe robot reaction," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 6301–6306.
- [16] F. Fabiani and S. Grammatico, "A mixed-logical-dynamical model for automated driving on highways," in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, pp. 1011–1015.
- [17] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1671–1678.
- [18] S. Tsugawa, "An overview on an automated truck platoon within the energy its project," *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 41–46, 2013, 7th IFAC Symposium on Advances in Automotive Control. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016383409>
- [19] J. Jang, J. Baek, K. Lim, Y. Ro, S. Yoon, and S. Jang, "A study on v2i based cooperative autonomous driving," in *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, 2023, pp. 1–3.
- [20] J. Cao, S. Leng, L. Zhang, M. Imran, and H. Chai, "A v2v empowered consensus framework for cooperative autonomous driving," in *GLOBE-COM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 5729–5734.
- [21] Y. Yuan, R. Tasik, S. S. Adhatarao, Y. Yuan, Z. Liu, and X. Fu, "Race: Reinforced cooperative autonomous vehicle collision avoidance," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9279–9291, 2020.
- [22] P. Liu, A. Kurt, and U. Ozguner, "Distributed model predictive control for cooperative and flexible vehicle platooning," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1115–1128, 2019.
- [23] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2022. [Online]. Available: <https://doi.org/10.1109/2FCvpr52688.2022.01674>
- [24] J. Ni, J. Han, and F. Dong, "Multivehicle cooperative lane change control strategy for intelligent connected vehicle," *Journal of Advanced Transportation*, vol. 2020, pp. 1–10, 02 2020.
- [25] D. Xie, Y.-Q. Wen, X.-M. Zhao, X. li, and Z. He, "Cooperative driving strategies of connected vehicles for stabilizing traffic flow," *Transportmetrica B*, 02 2020.
- [26] R. Valiente, B. Toghi, R. Pedarsani, and Y. P. Fallah, "Robustness and adaptability of reinforcement learning based cooperative autonomous driving in mixed-autonomy traffic," *CoRR*, vol. abs/2202.00881, 2022. [Online]. Available: <https://arxiv.org/abs/2202.00881>
- [27] W. Li, Q. Li, S. E. Li, R. Li, Y. Ren, and W. Wang, "Indirect shared control through non-zero sum differential game for cooperative automated driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 980–15 992, 2022.
- [28] S. Mosharafian and J. M. Velni, "Cooperative adaptive cruise control in a mixed-autonomy traffic system: A hybrid stochastic predictive approach incorporating lane change," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 136–148, 2023.
- [29] S. Mosharafian and J. Mohammadpour Velni, "A hybrid stochastic model predictive design approach for cooperative adaptive cruise control in connected vehicle applications," *Control Engineering Practice*, vol. 130, p. 105383, 01 2023.
- [30] H. Liu, X. D. Kan, S. E. Shladover, X.-Y. Lu, and R. E. Ferlis, "Modeling impacts of cooperative adaptive cruise control on mixed traffic flow in multi-lane freeway facilities," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 261–279, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X18310313>
- [31] J. Hu, S. Sun, J. Lai, S. Wang, Z. Chen, and T. Liu, "Cacc simulation platform designed for urban scenes," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2857–2874, 2023.
- [32] L. Xiao, M. Wang, W. Schakel, and B. van Arem, "Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 380–397, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0968090X1830528X>
- [33] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, and W. Ge, "Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic," *CoRR*, vol. abs/2111.06318, 2021. [Online]. Available: <https://arxiv.org/abs/2111.06318>
- [34] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "CARLA: an open urban driving simulator," *CoRR*, vol. abs/1711.03938, 2017. [Online]. Available: <http://arxiv.org/abs/1711.03938>
- [35] Epic Games, "Unreal engine." [Online]. Available: <https://www.unrealengine.com>
- [36] M. Wilkinson, T. Brown, and O. Ahmad, "The national advanced driving simulator (nads) description and capabilities in vision-related research," *Optometry - Journal of the American Optometric Association*, vol. 83, pp. 79–84, 06 2012.
- [37] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. D. Pero, B. Osinski, H. Grimmer, and P. Ondruska, "Simnet: Learning reactive self-driving simulations from real-world observations," *CoRR*, vol. abs/2105.12332, 2021. [Online]. Available: <https://arxiv.org/abs/2105.12332>
- [38] Carnetsoft B.V., "Carnetsoft driving simulator for research and testing," 2016. [Online]. Available: <https://cs-driving-simulator.com/>
- [39] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: <https://www.gurobi.com>
- [40] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>

- [41] H.-T. Chen, Y.-C. Wu, and C.-C. Hsu, "Daytime preceding vehicle brake light detection using monocular vision," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 120–131, 2016.
- [42] J. Pirhonen, R. Ojala, K. Kivekäs, and K. Tammi, "Predictive braking with brake light detection—field test," *IEEE Access*, vol. 10, pp. 49 771–49 780, 2022.
- [43] M. Svård, J. Bärman, and T. Victor, "Detection and response to critical lead vehicle deceleration events with peripheral vision: Glance response times are independent of visual eccentricity," *Accident Analysis Prevention*, vol. 150, p. 105853, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0001457520316730>
- [44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [45] A. Bemporad and S. Di Cairano, "Model-predictive control of discrete hybrid stochastic automata," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1307–1321, 2011.
- [46] S. Hossain, J. Lu, H. Bai, and W. Sheng, "Stochastic model predictive control for coordination of autonomous and human-driven vehicles," *IFAC-PapersOnLine*, vol. 55, no. 41, pp. 142–147, 2022, 4th IFAC Workshop on Cyber-Physical and Human Systems CPHS 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896323001246>



Jiaxing Lu (Student Member, IEEE) received his M.S degree in Computer Science from Stevens Institute of Technology in 2019, and B.S degree in Computer Science from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016. He is currently pursuing the doctoral degree in Electrical Engineering with the Laboratory for Advanced Sensing, Computation and Control (ASCC), Oklahoma State University. His research interest includes cooperative driving in mixed traffic

vehicles, and modeling human drivers in such HITL systems.



Sanzida Hossain completed her B.Sc. in Mechanical Engineering in 2016 from the Military Institute of Science and Technology, Bangladesh and her M.Sc. in Mechanical Engineering from Kansas State University, USA in 2020. She is currently pursuing her doctoral degree in Mechanical Engineering at the Mechanical and Aerospace Engineering Department of Oklahoma State University, USA. Her research is focused on cooperative driving between autonomous vehicles and human-driven vehicles in a connected environment considering the stochasticity of human

driver behavior.



Wa Kun Lam received his B.S. and M.S. degrees in the field of theoretical Condensed Matter Physics from the Jinan University, Guangzhou, China P. R., in 2001 and 2004. He received a Ph.D. degree from Oklahoma State University in both theoretical and experimental optics, majoring in Bose-Einstein Condensation, U.S.A., in 2016. He is currently working as an independent researcher collaborating with the School of Electrical Computer Engineering department at Oklahoma State University and a mathematical modeling consultant for commercial

companies. His interest includes the application of A.I. techniques and/or physical models for product prototyping in gaming and engineering; applying machine learning techniques in graphical design, analysis of player experience, risk analysis for new policy and strategy applied for marketing, etc.



Weihua Sheng (Senior Member, IEEE) is currently a professor at the School of Electrical and Computer Engineering, Oklahoma State University (OSU), USA. He is the Director of the Laboratory for Advanced Sensing, Computation and Control (ASCC Lab, <http://ascclab.org>) at OSU. Dr. Sheng received his Ph.D degree in Electrical and Computer Engineering from Michigan State University in May 2002. He obtained his M.S and B.S. degrees in Electrical Engineering from Zhejiang University, China in 1997 and 1994, respectively. He is the author

of more than 250 peer-reviewed papers in major journals and international conferences. Eight of them have won best paper or best student paper awards in major international conferences. His current research interests include social robotics, wearable computing, human robot interaction and intelligent transportation systems. His research has been supported by US National Science Foundation (NSF), Oklahoma Transportation Center (OTC) /Department of Transportation (DoT), etc. Dr. Sheng served as an Associate Editor for IEEE Transactions on Automation Science and Engineering from 2013 to 2019. He also served as an Associate Editor for IEEE Robotics and Automation Magazine.



He Bai (Member, IEEE) received his B.Eng. degree from the Department of Automation at the University of Science and Technology of China, Hefei, China, in 2005, and the M.S. and Ph.D. degrees in Electrical Engineering from Rensselaer Polytechnic Institute in 2007 and 2009, respectively. From 2009 to 2010, he was a Post-doctoral Researcher at the Northwestern University, Evanston, IL. From 2010 to 2015, he was a Senior Research and Development Scientist at UtopiaCompression Corporation. He was the Principal Investigator for a number of research

projects on sense-and-avoid, cooperative target tracking, and target handoff in GPS-denied environments. In 2015, he joined the Mechanical and Aerospace Engineering Department at Oklahoma State University, where he is currently an associate professor. He has published over 100 peer-reviewed journal and conference papers related to control, estimation, learning, and robotics and a research monograph "Cooperative control design: a systematic passivity-based approach" in Springer. He holds one patent on monocular passive ranging. His research interests include multi-agent systems, nonlinear estimation and sensor fusion, path planning, intelligent control, and GPS-denied navigation.