Shared Control of Teleoperated Vehicles with Delay-Compensated Safety Filtering

Hang Zhang, Harry Zhang, Yujie Wang, Zhenhao Zhou, Dan Negrut and Xiangru Xu

***Abstract*— This work presents a new shared control framework for teleoperated vehicles, targeting critical safety challenges arising from the control communication latency and the correctness of driver warnings. The proposed delay-compensated shared control architecture integrates two key components: a conformal prediction-based warning system that proactively alerts remote drivers of potential hazards and an onboard safety filter that combines a delay compensator, a disturbance observer, and a control barrier function-based quadratic program. The proposed design framework generates real-time safe control commands at the human-operation level despite delayed human inputs. A high-fidelity simulation platform was developed for semi-autonomous vehicle teleoperation using Chrono, a multi-physics-based simulator. Through extensive experiments in diverse scenarios, the proposed approach demonstrates robust performance and reliable safety maintenance under aggressive maneuvers and communication delays.**

I. Introduction

Autonomous vehicles (AVs) have made significant advancements in recent years. With the increasing commercialization of AV technology, fully autonomous vehicles – operating without an in-vehicle driver – have already been deployed on public roads in various regions. However, ensuring the transferability and adaptability of automated driving solutions across diverse traffic scenarios remains a safety-critical challenge [1], [2]. To bridge the gap to fully autonomous driving without human supervision, teleoperation has emerged as a fallback solution for managing situations beyond AV capabilities, which enables a human driver to assist a semi-autonomous vehicle (SAV) remotely when the automated driving solution encounters some unknown edge cases that may pose safety hazards [3]. Once the SAV negotiates the edge cases, it can switch back to automated driving mode as before. However, teleoperation faces several challenges, such as reduced situational awareness [4], which may lead to improper commands from remote drivers. Moreover, data transmission latency in sensor and control pipelines between remote drivers and SAVs [5], as illustrated in Fig. 1, remains a safety-critical issue, as delayed vehicle information and control commands can hinder timely response. Although various techniques have been developed to mitigate delay effects [6]–[8], latency may not be fully eliminated, which may potentially compromise safety if SAVs are directly controlled by remote drivers with delayed signals.

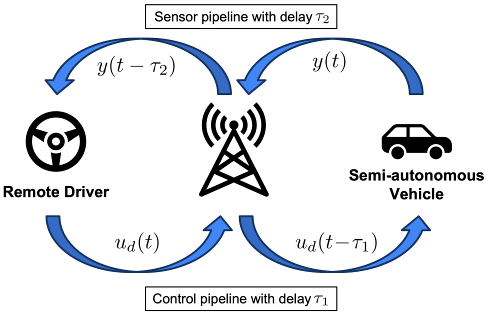


Fig. 1: Framework of teleoperated driving for semi-autonomous vehicles including sensor and control latency.

To that end, *shared control* is developed to address the limitations of direct remote control [9]. In the shared control scheme – particularly the mechanically uncoupled shared control scheme defined in [10] – the driver's control commands are executed only if deemed safe; otherwise, the system intervenes to prevent unsafe scenarios. Various shared control strategies to override driver commands for safety purposes have been developed, including model predictive control (MPC) approaches [11]–[14]. However, these methods operate on high-level control inputs, such as velocity, acceleration, or jerk, assuming that drivers also issue commands at this level. This inconsistency with *human-operated inputs*, such as throttle and braking, potentially reduces the effectiveness of the driver's response. Moreover, MPC may become less effective when system dynamics exhibit strong nonlinearity and safety constraints are highly complex.

This study focuses on the control pipeline in the teleoperation of SAVs, particularly within an uncoupled shared control scheme. We propose a shared control mechanism that integrates a warning system to detect potential safety hazards and an onboard, delay-compensated safety filter. The safety filter processes the human driver's command directly from reception to actuation by mitigating transmission delays with a delay predictor, compensating for acceleration discrepancies with a Disturbance Observer (DOB), and enhancing safety using Control Barrier Functions (CBFs) [15]. Additionally, a high-fidelity simulation platform for SAV teleoperation was established using Chrono, a physics-based dynamics simulation platform [16]. The platform enables the customization of high-fidelity vehicle models, traffic scenarios, and data transmission delays to simulate real-world teleoperation conditions. All the codes for this work are publicly available[1].

This work was supported in part by National Science Foundation under grant CNS-2222541.

The authors are with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI, USA. Email: {hang.zhang, hzhang699, yujie.wang, zzhou292, negrut, xiangru.xu}@wisc.edu.

[1] https://github.com/uwsbel/sbel-reproducibility/tree/master/2025/CCTA-highwayControl

The contributions of this work are twofold: (i) We propose a novel safety filter mechanism that generates safe control commands at the human-operation level, compensating for control pipeline delays and enhancing teleoperated vehicle safety. (ii) We develop a high-fidelity physics-based simulation platform for the shared control of teleoperated SAVs. The remainder of the paper is organized as follows. Section II presents the proposed shared control architecture; Section III introduces the Chrono simulation platform and teleoperation-related modules in Chrono; Section IV introduces the implementation details of the warning system; Section V presents the onboard safety filter; the experimental results and the corresponding analysis are given in Section VI; concluding remarks and directions for future work are provided in Section VII.

II. Overview of Shared Control for Teleoperated Vehicles with Safety Filtering

This section presents an overview of the proposed shared control approach for teleoperated driving in Chrono (see Fig. 2). Note that in this work, we focus on latency in the control pipeline while ignoring the sensor pipeline delay. We will address the sensing delay mitigation using Chrono in our future work.

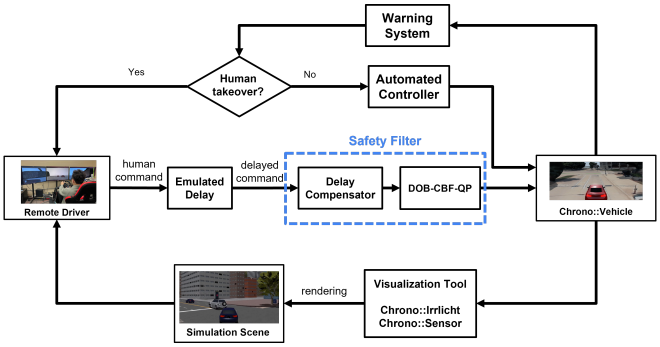


Fig. 2: Shared control of teleoperated driving with delay-compensated safety filtering in Chrono.

In our approach, a vehicle is modeled in Chrono as a virtual replica of a real-world SAV. A warning system, consisting of a neural network-based trajectory predictor and a conformal prediction-based warning algorithm, continuously monitors the vehicle's driving conditions. Under normal conditions, the vehicle operates autonomously using an onboard controller without human intervention. However, when the warning system detects imminent danger, it generates an alert, prompting a remote human driver to take over control. Using simulated scenes in Chrono, generated from remote sensing data, the human driver issues control commands to the vehicle. Due to communication latency in the control pipeline, these delayed commands may compromise safety. To address this, the delayed commands are processed through a safety filter before being applied to the vehicle. The safety filter consists of a delay compensator that mitigates the effects of control delays and a Quadratic Program (QP)-based filter that adjusts the compensated human commands in a minimally invasive manner to enhance safety. Once driving conditions become safe, control can be handed back to the onboard automated controller.

The following sections will provide a detailed explanation of each component of the architecture shown in Fig. 2, including the Chrono simulator (Section III), the driver warning system (Section IV), and the safety filter (Section V).

III. The Chrono Simulator

Chrono is selected for this study due to its optimal balance of accuracy, computational performance, and modeling flexibility. It enables high-fidelity simulation of real-world vehicle dynamics and sensor data collection, while maintaining real-time performance in large-scale traffic scenarios [17]. It is implemented as an open-source project allowing unrestricted use and modification [18], thus allowing for essential customizations in relation to modeling sensor data delays and system command latencies. The users can define a variety of vehicle types using JSON files or through a Python/C++ API, and have the simulator solve the underlying differential-algebraic equations governing the time evolution of the vehicle models. For large-scale traffic scenarios, real-time performance is achievable through its multi-process simulation framework [19]. Additionally, Chrono includes a human-in-the-loop interface and mature sensor support, making it well-suited for AV/SAV testing tasks in this study [20], [21].

Compared to other simulation platforms used in AV research, Chrono stands out for its high-fidelity physics modeling: CARLA [22] and AirSim [23] are two widely used open-source simulators. CARLA is tailored for urban driving with detailed road infrastructure but lacks high-fidelity vehicle dynamics modeling; AirSim, originally developed for drones and later extended to ground vehicles, runs on Unreal Engine or Unity 3D but prioritizes realistic rendering over accurate physics. Chrono's emphasis on accurate physics calculations makes it well-suited for applications requiring precise mechanical system simulations, including vehicle dynamics and large-scale traffic scenarios.

Chrono::Vehicle Module. Chrono::Vehicle is a specialized module within the Chrono framework that provides parameterized templates for modeling various wheeled and tracked vehicle subsystems. It supports vehicle simulations in on- and off-road conditions, while incorporating closed-loop and interactive driver models. Chrono::Vehicle offers a comprehensive suite of subsystem templates for tires, suspensions, steering mechanisms, drivelines, and external systems such as powertrains, drivers, and terrain models. Additionally, it includes utility functions for visualization, monitoring, and data collection. The module supports three categories of tire models: rigid, semi-empirical, and finite element analysis-based models.

Chrono::Sensor Module. Chrono::Sensor is a real-time capable sensor simulation module that supports cameras, LiDARs, SPADs, and GPS/IMU to enable autonomous robot simulations [20]. Chrono::Sensor employs a ray tracing engine that utilizes the NVIDIA OptiX framework [24]. It

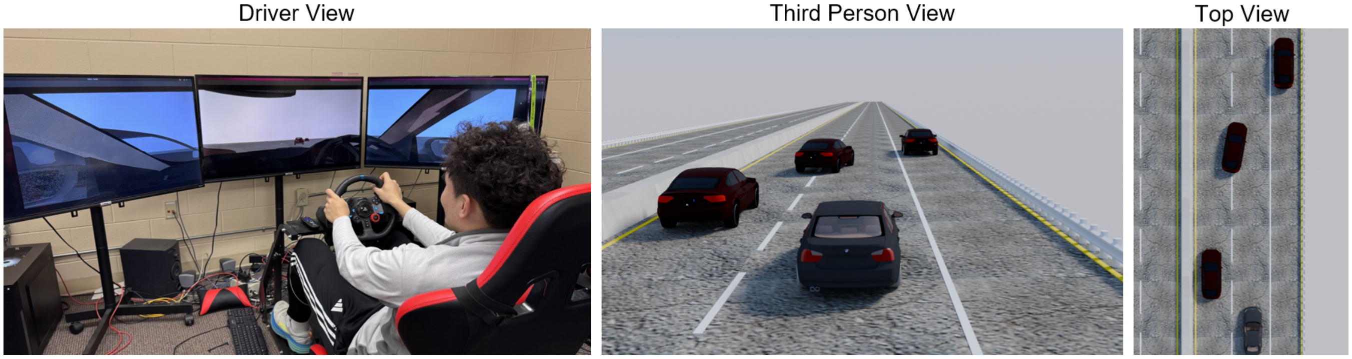


Fig. 3: (Driver View) A driver using the Chrono-based driving simulator developed to control a vehicle by issuing throttle, braking, and steering commands. (Third Person & Top View) Blender rendered snapshots of the third-person view and top view during experiments.

uses path tracing with global illumination and physically-based rendering to simulate the interaction of light with the environment. Chrono::Sensor provides realistic sensor simulations by modeling common artifacts, e.g., lens distortion, depth-of-field, exposure, sensor noise, and sensor lag [25].

Driving Simulator. This study utilized PyChrono [26], a Python interface for accessing the core functionality of the Chrono simulation engine. PyChrono is chosen since it easily interfaces to a variety of control-related packages used in the community. The Chrono::Vehicle module enables the configuration of a driving simulator for human-in-the-loop experiments. The driving simulator setup includes a Logitech G29 driving wheel and pedal, a simplified driving cabin, and a monitor (see Fig. 3). To capture the communication time delay, a buffered driver input is implemented, allowing real-time control and customization of delay magnitude by adjusting the input buffer. This approach provided fine-grained control over delay variations, enabling a detailed assessment of their impact on the system's dynamic response.

IV. Driver Warning System

In this section, we introduce the driver warning system designed for teleoperated driving in the traffic simulator. Our system builds upon the approach in [27], which integrates a trajectory predictor with conformal prediction, a statistical inference technique. The warning system provides a formally guaranteed false negative rate, ensuring that unsafe situations are rarely missed without triggering an alert.

Trajectory Predictor. We employ the Spatio-Temporal Attention Long Short-Term Memory (STA-LSTM) model for vehicle trajectory prediction [28]. Given the historical trajectories of the SAV and neighboring vehicles, the STA-LSTM model encodes spatial relationships using a 3×13 tensor grid and captures temporal dependencies through the attention mechanism. This model achieves high prediction accuracy and enhances interpretability by highlighting how past trajectories and interactions with surrounding vehicles influence the SAV's motion. The STA-LSTM model was trained using a dataset comprising 80% NGSIM data [29] and 20% driving data collected from Chrono simulations.

Warning Algorithm. We employ a conformal prediction-based algorithm [27], outlined in Algorithm 1, to generate alerts s, where $s = 1$ prompts the remote driver to take control of the SAV. The inputs to Algorithm 1 include: a safety threshold $f_0 > 0$; a warning threshold $\epsilon > 0$; an offline-collected calibration dataset $\{(\hat{z}_k, z_k)\}_{i=1}^{K}$ where $\hat{z}_k$ represents the predicted trajectories of surrounding vehicles and z_k denotes the corresponding ground truth trajectories; a new trajectory prediction $\hat{z}_{new}$; and a safety score function f that quantifies the "distance" to unsafe scenarios.

The safety score function f is designed to assess potential safety hazards in the SAV driving scenario. Specifically, f is defined to compute the minimum weighted distance between the SAV and the predicted positions of surrounding vehicles $\hat{z}$ over the prediction length T_f:

$$f(\hat{z}) = \min_{i \in \{1,\cdots,T_f\},\ j \in \mathcal{D}_j} (\hat{z}^{(j,i)})^\top \begin{bmatrix} w_{\text{long}} & 0 \\ 0 & w_{\text{lat}} \end{bmatrix} \hat{z}^{(j,i)} \ , \tag{1}$$

where w_{long} and w_{lat} are weights of the longitudinal and lateral directions, respectively; $\hat{z}^{(j,i)}$ is the predicted coordinates of the j-th surrounding vehicle at time step i relative to the SAV's coordinate frame; $\mathcal{D}_j = \{j \mid \|\hat{z}^{(j,0)}\|_2 \leq r\}$ is the set of surrounding vehicles whose current positions lie within a distance r from the SAV. Note that the safety score function f is also applicable to the ground truth trajectory z for identifying unsafe samples in the calibration data.

When a new trajectory prediction $\hat{z}_{new}$ is obtained based on the current traffic conditions and historical trajecto-

Algorithm 1: Warning Algorithm (Adapted from [27])

```
Input: Exchangeable calibration data $\{(\hat{z}_k, z_k)\}_{i=1}^{K}$,
       safety score function $f$, safety threshold $f_0 > 0$,
       warning threshold $\epsilon > 0$, new prediction $\hat{z}_{new}$.
Output: $s \in \{0, 1\}$
$\mathcal{A} \leftarrow \{f(\hat{z}_k) : f(z_k) < f_0, k = 1, \ldots, K\}$;
sample $U$ uniformly by
    $U \sim \{0, 1, \ldots, |\{a \in \mathcal{A} : a = f(\hat{z}_{new})\}|\}$;
$q \leftarrow \frac{|\{a \in \mathcal{A} : a < f(\hat{z}_{new})\}| + U + 1}{|\mathcal{A}| + 1}$
if $q \leq 1 - \epsilon$ then
    return 1 else return 0
```

ries, Algorithm 1 utilizes the calibration dataset to assess the predictor's inaccuracy and subsequently generates the warning signal s using the safety score function f. If the *exchangeability* assumption holds for the calibration data and the new prediction-ground truth pair $(\hat{z}_{new}, z_{new})$ – meaning that the probability of observing any permutation of the calibration data and the new data point is equally likely – then the warning signals produced by Algorithm 1 have a provable low false negative rate [27]. Specifically, $\Pr(s = 1 \mid f(z_{new}) < f_0) \geq 1 - (\epsilon + 1/(1 + |\mathcal{A}|))$. This ensures that the system reliably issues alerts when a vehicle takeover is necessary, minimizing the risk of missing unsafe situations. A detailed explanation of exchangeability is provided in [27], while the collection of exchangeable calibration data within Chrono simulations is discussed in Section VI-B.

V. Safety Filter

When the warning system issues an alert, the remote driver takes over the vehicle and sends control commands. However, due to delays in the control pipeline, the vehicle may not respond immediately, potentially deviating from the intended trajectory. In this section, we present a safety filter design method to mitigate the impact of control delays while filtering control commands to enhance vehicle safety with minimal modification to the driver's inputs.

As shown in Fig. 2, the safety filter comprises a delay compensator and a DOB-CBF-QP module. The human-issued desired control signals first pass through the delay compensator to counteract control pipeline delays before being DOB-CBF-QP processed to enhance vehicle safety. The DOB is utilized to estimate uncertainties in the engine mapping, ensuring more accurate control commands. Unlike other model-based shared controllers [11]–[14], the proposed safety filter directly produces human-operated level control commands $u \triangleq [\alpha, \beta]^\top$, where $\alpha \in [-1, 1]$ represents the *generalized throttle input* which can be directly controlled via the throttle and braking pedals and is directly applicable to the driving simulator in Chrono, and β denotes the *steering input* of the vehicle. This design enables the SAV to closely replicate the remote driver's intended actions.

For convenience, we define the control command variables as follows: $u_d(t) = [\alpha_d(t), \beta_d(t)]^\top$ is the desired command sent from the remote driver; $u_d(t - \tau_1)$ is the delayed command received by the SAV; $\hat{u}_d(t) = [\hat{\alpha}_d(t), \hat{\beta}_d(t)]^\top$ is the output of the delay compensator; and $u_{\text{CBF}}(t) = [\alpha_{\text{CBF}}(t), \beta_{\text{CBF}}(t)]^\top$ is the filtered control commands through the DOB-CBF-QP.

A. Delay Compensator

Since modeling the dynamics of the desired control commands $u_d(t)$ is intractable, we employ a signal-level second-order predictor to drive the predictor output $\hat{u}_d(t)$ to converge to $u_d(t)$ [7]. Meanwhile, the predictor error $u_d(t) - \hat{u}_d(t)$ is kept smaller than the coupling error $u_d(t) - u_d(t - \tau_1)$. The dynamics of the second-order predictor are expressed as

$$\begin{aligned}\dot{\hat{u}}_d(t) = \dot{u}_d(t - \tau_1) + k_1(\dot{u}_d(t - \tau_1) - \dot{\hat{u}}_d(t - \tau_1))\\ + k_2(u_d(t - \tau_1) - \hat{u}_d(t - \tau_1))\ ,\end{aligned} \tag{2}$$

where (k_1, k_2) are the PD gains. The guidance for choosing (k_1, k_2) is provided in [7]. Given the delayed desired control commands received by the SAV, $u_d(t - \tau_1)$, the system (2) compensates for the impact of control delays by predicting the current desired control command, $u_d(t)$. Herein, we focus on the case of constant control pipeline delay. The extension to varying delay scenarios is left for future research.

B. DOB-CBF-QP

We assume the dynamics of the SAV are described by

$$\dot{x} = v\cos(\theta), \tag{3a}$$

$$\dot{y} = v\sin(\theta), \tag{3b}$$

$$\dot{\theta} = \frac{v}{L}\tan\beta, \tag{3c}$$

$$\dot{v} = \hat{T}(\alpha, v) + \Delta\ , \tag{3d}$$

where L denotes the length of the SAV, and $x, y, \theta, v \in \mathbb{R}$ represent the longitudinal coordinate, lateral coordinate, orientation, and velocity of the SAV, respectively. Recall that $\alpha \in [-1, 1]$ is the generalized throttle input, and β is the steering input. The fitted engine model is given by $\hat{T}(\alpha, v) = p_0(v) + p_1(v)\alpha$ where $p_0(v)$ and $p_1(v)$ are polynomials (see Section VI-C for more details); the modeling uncertainty/error of the engine mapping is denoted as $\Delta = T(\alpha, v) - \hat{T}(\alpha, v)$, where $T(\alpha, v)$ is the true engine mapping. The mapping T is highly nonlinear in practice, and the uncertainty in T can lead to incorrect and unknown acceleration outputs, which will ultimately compromise safety. To address this issue, we will design a DOB to estimate the uncertainty Δ, a CBF to encode the safety constraints, and a DOB-CBF-QP to compensate for Δ while enforcing safety constraints for the system (3).

DOB Design. Following the procedure developed in [30], we design the following DOB to estimate Δ online:

$$\hat{\Delta} = \xi + \eta v, \tag{4a}$$

$$\dot{\xi} = -\eta(p_0(v) + p_1(v)\alpha + \hat{\Delta})\ , \tag{4b}$$

where ξ is the internal state of the DOB and $\eta > 0$ is the DOB gain. Assume that the derivative of the lumped uncertainty is bounded, that is, $|\dot{\Delta}| \leq \omega$, where $\omega > 0$ is a constant. Then, the disturbance estimation error $e_\Delta = \hat{\Delta} - \Delta$ is uniformly ultimately bounded by $|e_\Delta| \leq \sqrt{|e_\Delta(0)|^2 e^{-2\kappa t} + \frac{\omega^2}{2\nu\kappa}}$, where $0 < \nu < 2\eta$ is a constant and $\kappa = \eta - \frac{\nu}{2}$ [30].

CBF Design. To first introduce CBF, consider a control-affine system $\dot{x} = f(x) + g(x)u$, where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, and $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are known and locally Lipschitz continuous functions. Define a safe set $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}$, where $h : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function. The function h is called a CBF of (input) relative degree 1 if $\sup_{u\in\mathbb{R}^m} [L_f h + L_g h u + \gamma h] \geq 0$ holds for all $x \in \mathbb{R}^n$, where $\gamma > 0$ is a given positive constant, and $L_f h = \frac{\partial h}{\partial x} f$ and $L_g h = \frac{\partial h}{\partial x} g$ are Lie derivatives. When the CBF condition

$L_f h + L_g h u + \gamma h \geq 0$ is incorporated into a QP, the resulting CBF-QP-based controller can formally ensure the safety (i.e., $h(x(t)) \geq 0$ for any $t \geq 0$) of the closed-loop system [15].

Given system (3), we define the safe set $\mathcal{C}$ for the teleoperated driving as follows:

$$\mathcal{C} = \{(x, y, \theta, t) \in \mathbb{R}^3 \times \mathbb{R}_+ : h(x, y, \theta, t) \geq 0\} \ , \quad (5)$$

where the function $h : \mathbb{R}^4 \times \mathbb{R}_+ \to \mathbb{R}$ is given as

$$h(x, y, \theta, t) = (p - p_0(t))^\top R(\theta)^\top \Lambda R(\theta)(p - p_0(t)) - 1 \ . \quad (6)$$

Here, $p = [x \ y]^\top$, $p_0(t) \in \mathbb{R}^2$ denotes the position of the surrounding vehicle, $R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$ denotes the rotation matrix, $\Lambda = \begin{bmatrix} 1/\Lambda_1^2 & 0 \\ 0 & 1/\Lambda_2^2 \end{bmatrix}$, and $\Lambda_1, \Lambda_2 > 0$ are constants. As shown in Fig. 4, the CBF h given in (6) aims to keep a safe distance between the SAV and surrounding vehicles, i.e., the surrounding vehicles are prohibited from entering the ellipsoid region represented by $\mathcal{C}^c$.

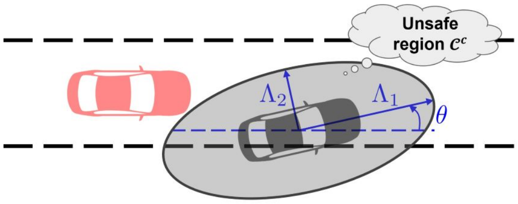


Fig. 4: Illustration of the CBF defined in (6). The safe set $\mathcal{C}$ denotes the exterior of the ellipsoid in grey.

DOB-CBF-QP Design. It is evident that h is a CBF of a relative degree of 2 w.r.t. the control input α and 1 w.r.t. β for the system (3), which makes the CBF-QP design difficult. To achieve a uniform input relative degree for h, we augment the original system (3) with an additional integrator:

$$\dot{x} = v\cos(\theta), \quad (7a)$$
$$\dot{y} = v\sin(\theta), \quad (7b)$$
$$\dot{\theta} = \frac{v}{L}\tan\beta, \quad (7c)$$
$$\dot{v} = p_0(v) + p_1(v)\alpha + \Delta, \quad (7d)$$
$$\dot{\beta} = \chi \ , \quad (7e)$$

where χ denotes the auxiliary control input to be designed and β is now considered as a state variable. Define the state vector of (7) as $X = [x, y, \theta, v, \beta]^\top$. Note that h has a uniform relative degree of 2 with respect to α and χ.

We define an exponential CBF [31] $\bar{h}$ as $\bar{h} = \dot{h} + \lambda h$ with $\lambda > 0$ a constant. The function $\bar{h}$ can be expressed as

$$\bar{h}(X, t) = \frac{\partial h}{\partial x} v\cos\theta + \frac{\partial h}{\partial y} v\sin\theta + \frac{\partial h}{\partial \theta}\frac{v\tan\beta}{L} + \frac{\partial h}{\partial t} + \lambda h. \quad (8)$$

Following [32, Theorem 2], we have the following result, which introduces a DOB-CBF-QP that formally guarantees the safety of system (7), ensuring that $h(x(t), y(t), \theta(t), t) \geq 0$ for any $t \geq 0$.

Theorem 1: Consider the augmented system (7) and the safe set $\mathcal{C}$ defined in (5). Suppose that there exist $\lambda, \zeta, \gamma, \eta$ satisfying $\bar{h}(X(0), 0) > 0$, $\eta > \frac{\gamma + \nu}{2}$ and $\zeta > \frac{e_\Delta^2(0)}{2\bar{h}(X(0),0)}$. Then, any Lipschitz continuous controller $(\alpha, \chi) \in K_{BF}(X, \hat{\Delta}) \triangleq \{(a, b) \in \mathbb{R}^2 : \psi_0 + \psi_1 a + \psi_2 b \geq 0\}$ will guarantee $h(X(t), t) \geq 0$ for any $t \geq 0$ where

$$\psi_0 = \zeta\left[\frac{\partial \bar{h}}{\partial x} v\cos\theta + \frac{\partial \bar{h}}{\partial y} v\sin\theta + \frac{\partial \bar{h}}{\partial v}(p_0 + \hat{\Delta}) - \frac{\left(\frac{\partial \bar{h}}{\partial v}\right)^2}{4\kappa - 2\gamma}\right] + \zeta\frac{\partial h}{\partial t} - \frac{\omega^2}{2\zeta\nu} + \zeta\gamma h,$$

$\psi_1 = \frac{\partial \bar{h}}{\partial \alpha} p_1(v)$, and $\psi_2 = \frac{\partial \bar{h}}{\partial \beta}$.

Given the output of the delay compensator $(\hat{\alpha}_d, \hat{\beta}_d)$, the safe controller in Theorem 1, α and χ, can be obtained by solving the following convex QP in real-time:

$$(\alpha^*, \chi^*) = \underset{\alpha, \chi}{\arg\min} \quad \rho_\alpha \|\alpha - \hat{\alpha}_d\|^2 + \rho_\chi \|\chi - \hat{\chi}_d\|^2 \quad (9)$$
$$\text{s.t.} \quad \psi_0 + \psi_1 \alpha + \psi_2 \chi \geq 0 \ ,$$

where ψ_0, ψ_1, ψ_2 are given in Theorem 1, $\rho_\alpha, \rho_\chi > 0$ are positive constants, and $\hat{\chi}_d$ is obtained via numerically differentiating $\hat{\beta}_d$. The filtered throttle input is designed as $\alpha_{\text{CBF}} = \alpha^*$, and the filtered steering input β_{CBF} is obtained via integrating χ^*. Finally, $(\alpha_{\text{CBF}}, \beta_{\text{CBF}})$ are applied to the SAV to ensure safe operation.

The parameters (ρ_α, ρ_χ) determine which desired control input, α_d or χ_d, should be better preserved by the safety filter. For example, if $\rho_\chi \gg \rho_\alpha$, the DOB-CBF-QP primarily modifies the throttle while keeping the steering as close as possible to the input from the remote driver. In practice, ρ_α and ρ_χ should be carefully tuned to maintain the performance of the control inputs generated by the remote driver.

VI. Experiments

In this section, we present the simulation environment setup, the data collection process for engine mapping and the warning system, and the teleoperation experiments for two different driving scenarios. The Chrono simulations were run on a workstation equipped with an AMD Ryzen 9 3900X 12-Core processor, 32GB RAM, and an NVIDIA GeForce RTX 2070 GPU.

A. Simulation Environment Setup

The simulation environment is designed to replicate a three-lane highway driving scenario. During the simulation experiment, the Chrono::Vehicle agent, representing the SAV, is operated by a human driver as shown in Fig. 3. The steering angle mapping is set to be a linear function of the driver's steering wheel rotation. While the SAV employs a high-fidelity physics-based model, the surrounding vehicles utilize a lower-fidelity dynamics model to optimize computational efficiency. These surrounding vehicles exhibit three behaviors relative to the SAV: (i) maintaining parallel motion in the adjacent right or left lanes, (ii) merging from the right or left lanes into the lane ahead of the SAV, and (iii) moving forward directly in front of the SAV.

To introduce variability and realism into the simulation, the speeds of the surrounding vehicles, along with the occurrence of each behavior, are randomly generated. Once the speed and behavior of a surrounding vehicle are determined, a corresponding reference trajectory is created. A simple PID controller is then employed to regulate the surrounding vehicle's motion, ensuring it adheres to the desired speed and driving behavior.

B. Calibration Data Collection for Warning System

As stated in Section IV, Algorithm 1 guarantees a low false negative rate if the calibration data points are exchangeable. Similar to [27], we generated diverse scenes in Chrono by randomizing driveway shapes, the surrounding vehicles' behaviors, and controller parameters. These different scenes can be considered exchangeable. From each scene, a single trajectory was randomly sampled, and its prediction and ground truth were collected to construct a dataset of exchangeable trajectories.

C. Engine Mapping Fitting

In the experiments, we use a virtual BMW E90 sedan as the SAV controlled by the remote driver. Chrono::Vehicle provides high-fidelity engine modeling through an empirical RPM-to-torque mapping, which drives the power transmission to the vehicle's axles and wheels. Due to the non-differentiable nature and complexity of this modeling pipeline, directly integrating it into the vehicle dynamics formulation is impractical. To address this challenge, as discussed previously in Section V-B, a polynomial engine model $\hat{T}(\alpha, v)$ defined in (7) is calibrated using data from Chrono simulations, where the vehicle randomly undergoes cycles of acceleration to target speeds (20-30 m/s) followed by variable braking (10-60%). The fitted engine model's performance, trained using recorded speed, acceleration, and throttle data, is validated as shown in Fig. 5.

D. Testing Experiments

Experiment I: Straight Lane Scenario. In this experiment, a straight-lane driving scenario with a low-speed leading vehicle is simulated in Chrono to evaluate the performance of the proposed DOB-CBF filter and how the choice of weight parameters (ρ_α, ρ_χ) influences the filtered commands. For all the experiments in this scenario, the motion of the leading vehicle is set the same, and the remote driver aggressively approaches the leading vehicle with full throttle and no steering actions. Since this experiment uses constant driver inputs to evaluate the DOB-CBF-QP module, both the warning system and delay compensator are not central to this scenario, and their evaluation is omitted.

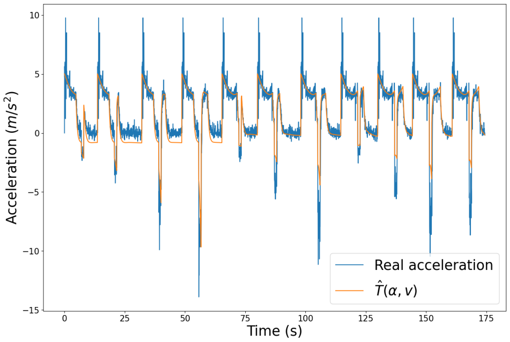


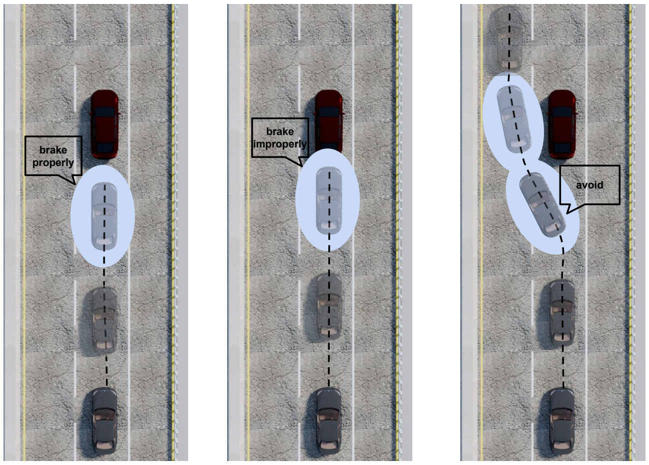
Fig. 5: Comparison between the real acceleration and the output from the fitted engine mapping model in a validation.

Figures 6a and 6b compare vehicle behavior with and without a DOB when the safety filter is activated, using weight parameters $(\rho_\alpha, \rho_\chi) = (1, 500)$. Since ρ_χ is much larger than ρ_α, the safety filter primarily modifies throttle input rather than steering to enhance safety. It can be observed that both safety filters override the improper driver commands and take braking to avoid a collision. However, the safety filter without DOB fails to ensure safety due to inaccurate acceleration output from the fitted engine mapping, as shown in Fig. 7a. In contrast, Fig. 7b demonstrates that the safety filter with DOB successfully maintains vehicle safety, indicating the importance of the DOB in estimating the engine mapping uncertainty.

Figure 6c shows the vehicle behavior when the safety filter is activated with a DOB and $(\rho_\alpha, \rho_\chi) = (1, 1)$. It can be observed that the safety filter takes over the steering commands to avoid the collision, though the remote driver does not issue any steering commands. As a result, the CBF values remain above the threshold as shown in Fig. 7c, implying safety satisfaction.

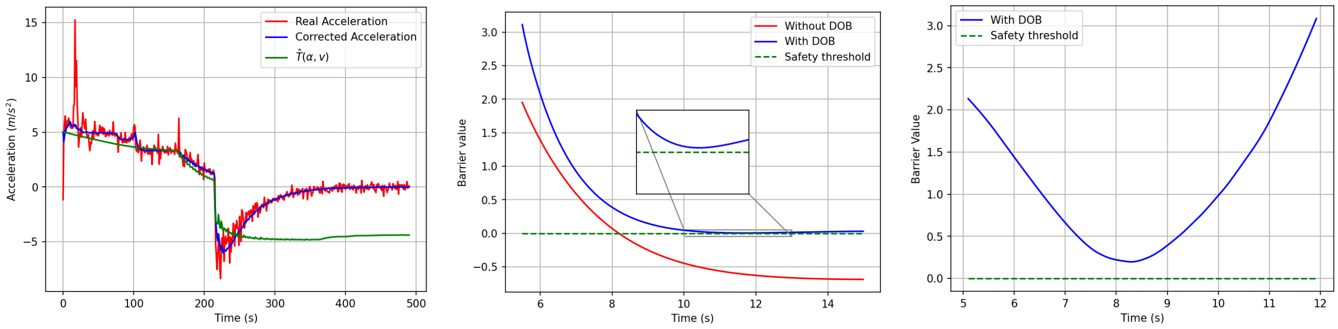
E. Driving Experiment II: Complex Traffic Scenario

Experiment II: Complex Traffic Scenario. In this experiment, a driving scenario with a curved driveway is constructed to evaluate the performance of the proposed shared control scheme. To mimic real-world driving scenarios, the surrounding vehicles with the behaviors defined in Section VI-A are randomly generated around the SAV. In the experiment, a control delay $\tau_1 = 200$ ms is injected in the delay buffer to mimic the data-transmission delay in the teleoperation. A lane-keeping MPC controller is employed



(a) (b) (c)

Fig. 6: Visualization of Experiment I in Chrono. (a) $(\rho_\alpha, \rho_\chi) = (500, 1)$ with DOB, the safety filter ensures safety. (b) $(\rho_\alpha, \rho_\chi) = (500, 1)$ without DOB, the safety filter fails to ensure safety. (c) $(\rho_\alpha, \rho_\chi) = (1, 1)$ with DOB, the safety filter ensures safety by modifying steering.



(a)

(b)

(c)

Fig. 7: Profiles of accelerations and CBF values in Experiment I: (a) Accelerations of the SAV with $(\rho_\alpha, \rho_\chi) = (1, 500)$. The true acceleration is extracted from the SAV via Chrono::Vehicle module. The DOB compensates for the acceleration obtained from the engine model $\hat{T}(\alpha, v)$. (b) CBF values with $(\rho_\alpha, \rho_\chi) = (1, 500)$. Safety is satisfied with DOB, but violated without DOB. (c) CBF values with $(\rho_\alpha, \rho_\chi) = (1, 1)$. Safety is satisfied with the filtered steering commands.

as the onboard automated driving controller of the SAV. The weights for the CBF defined in (6) are set to $\Lambda_1 = 10$ and $\Lambda_2 = 4$. The weights for the safety score defined in (1) for the warning system are chosen as $w_{\text{long}} = 1/400$, $w_{\text{lat}} = 1$, and $\rho_\alpha = \rho_\chi = 1$ for the DOB-CBF-QP in (9). The PD gain for the delay compensator is chosen as $k_1 = 0.6$ and $k_2 = 2$.

Fig. 8 shows the driver's control commands and the minimal CBF values of the surrounding vehicles within a distance r of the SAV in a segment of the entire simulation. Recall the notations defined in Section V: $\alpha_d(t)$ and $\beta_d(t)$ are the desired throttle and steering commands, respectively; $\hat{\alpha}_d(t)$ and $\hat{\beta}_d(t)$ are the output of the delay compensator, respectively; and $\alpha_{\text{CBF}}(t)$ and $\beta_{\text{CBF}}(t)$ are the output of DOB-CBF-QP which are applied to the SAV directly. Initially, the driver takes control of the SAV but issues some improper steering and throttle commands, which makes the safety filter intervene and mitigate the associated risk. The warning system stops alerting around $t = 19.8$ s, and the vehicle is switched to the automated driving controller once the traffic scenario is deemed safe from the driver's judgment at $t = 21.3$ s. Around $t = 22.7$ s, the warning system detects a potential safety risk and issues an alert in advance. The driver perceives the alert and then takes over the vehicle from the automated lane keeping controller at $t = 22.9$ s. Due to the reaction delay, the driver does not brake promptly. The safety filter hence intervenes by overriding the driver's commands to enhance safety. The driver continues to operate the vehicle until the safety is reaffirmed by both the warning system and the driver's judgment.

By comparing the desired control input $u_d(t) = [\alpha_d(t), \beta_d(t)]^\top$ issued by the remote driver and the output of the delay compensator $\hat{u}_d(t) = [\hat{\alpha}_d(t), \hat{\beta}_d(t)]^\top$ in Fig. 8, one can see that the predictions $\hat{u}_d(t)$ generated by the delay compensator are phase-synchronized with the driver's original inputs $u_d(t)$, which demonstrates the desired performance of the delay compensator. Additionally, by comparing the final output of the safety filter $u_{\text{CBF}}(t) = [\alpha_{\text{CBF}}(t), \beta_{\text{CBF}}(t)]^\top$ and the output of the delay compensator $\hat{u}_d(t)$, one can observe that the safety filter preserves the delay-compensated driver inputs $\hat{u}_d(t)$ when they are appropriate, while intervening to modify control inputs (shown in cyan and red blocks in Fig. 8) when the driver fails to respond promptly to risks. As can also be observed from Fig. 8, this intervention maintains CBF values above the safety threshold, thus maintaining vehicle safety.

VII. Conclusion

In this work, a shared control architecture with delay-compensated safety filtering for teleoperated SAVs was designed and demonstrated in a driving simulator. A conformal prediction-based warning system prompts remote drivers to take control when unsafe conditions arise. To address control delays and engine model uncertainty, an onboard safety filter combining a delay compensator with DOB-CBF-QP generates safe control commands at human-operated levels. Experimental results show that the warning system provides timely alerts, and the safety filter effectively mitigates delays, enhancing SAV safety.

References

[1] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 683–711, 2019.

[2] K. Wang, T. Zhou, X. Li, and F. Ren, "Performance and challenges of 3d object detection methods in complex scenes for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1699–1716, 2022.

[3] L. Zhao, M. Nybacka, M. Aramrattana, M. Rothhämel, A. Habibovic, L. Drugge, and F. Jiang, "Remote driving of road vehicles: A survey of driving feedback, latency, support control, and real applications," *IEEE Transactions on Intelligent Vehicles*, 2024.

[4] C. Mutzenich, S. Durant, S. Helman, and P. Dalton, "Updating our understanding of situation awareness in relation to remote operators of autonomous vehicles," *Cognitive Research: Principles and Implications*, vol. 6, pp. 1–17, 2021.

[5] J.-M. Georg, J. Feiler, S. Hoffmann, and F. Diermeyer, "Sensor and actuator latency during teleoperation of automated vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 760–766, IEEE, 2020.

[6] L. Zhao, M. Nybacka, M. Rothhämel, and J. Mårtensson, "Enhanced model-free predictor for latency compensation in remote driving systems," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 51–56, IEEE, 2024.

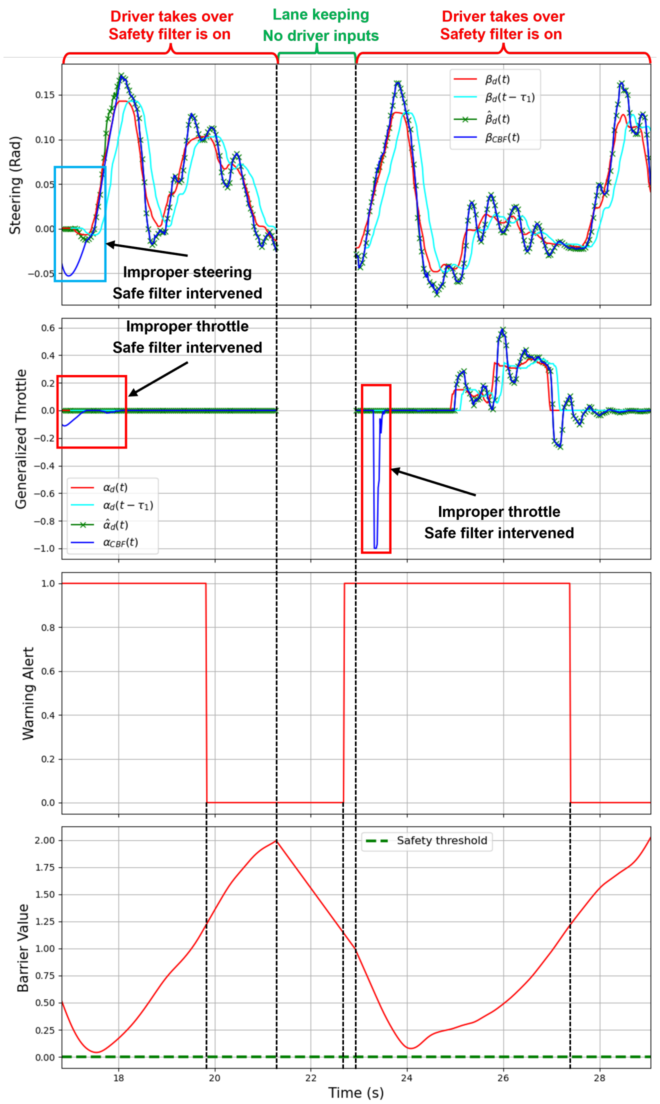


Fig. 8: Profiles of the driver's inputs and the minimal CBF values in a segment of Experiment II. The compensated inputs $\hat{u}_d(t)$ computed by the delay compensator have the same phase as the driver's inputs $u_d(t)$. The safety filter retains the compensated inputs when the driver behaves properly and modifies them when the driver's commands are deemed unsafe.

[7] S. Guo, Y. Liu, Y. Zheng, and T. Ersal, "A delay compensation framework for connected testbeds," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 7, pp. 4163–4176, 2021.

[8] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "A predictor-based framework for delay compensation in networked closed-loop systems," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2482–2493, 2018.

[9] D. Majstorović, S. Hoffmann, F. Pfab, A. Schimpe, M.-M. Wolf, and F. Diermeyer, "Survey on teleoperation concepts for automated vehicles," in *International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1290–1296, IEEE, 2022.

[10] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 475–491, 2020.

[11] D. Schitz, G. Graf, D. Rieth, and H. Aschemann, "Model-predictive cruise control for direct teleoperated driving tasks," in *IEEE European Control Conference (ECC)*, pp. 1808–1813, 2021.

[12] J. Storms, K. Chen, and D. Tilbury, "A shared control method for obstacle avoidance with mobile robots and its interaction with communication delay," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 820–839, 2017.

[13] S. Saparia, A. Schimpe, and L. Ferranti, "Active safety system for semi-autonomous teleoperated vehicles," in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pp. 141–147, 2021.

[14] A. Schimpe, D. Majstorovic, and F. Diermeyer, "Steering action-aware adaptive cruise control for teleoperated driving," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 988–993, IEEE, 2022.

[15] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[16] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science* (T. Kozubek, ed.), pp. 19–49, Springer International Publishing, 2016.

[17] X. Zhong, Y. Zhou, A. V. Kamaraj, Z. Zhou, W. Kontar, D. Negrut, J. D. Lee, and S. Ahn, "Human-automated vehicle interactions: Voluntary driver intervention in car-following," *Transportation Research Part C: Emerging Technologies*, vol. 171, p. 104969, 2025.

[18] Project Chrono, "Chrono Projects web page." `https://github.com/projectchrono/chrono-projects`. Accessed: 2024-04-07.

[19] J. Taves, A. Elmquist, A. Young, R. Serban, and D. Negrut, "Synchrono: A scalable, physics-based simulation platform for testing groups of autonomous vehicles and/or robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2251–2256, IEEE, 2020.

[20] A. Elmquist and D. Negrut, "Methods and models for simulating autonomous vehicle sensors," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 684–692, 2020.

[21] A. Elmquist and D. Negrut, "Modeling cameras for autonomous vehicle and robot simulation: An overview," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25547–25560, 2021.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[23] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*, pp. 621–635, Springer, 2018.

[24] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich, "Optix: A general purpose ray tracing engine," *ACM Transactions on Graphics*, vol. 29, pp. 1–13, August 2010.

[25] N. M. Batagoda, B.-H. Chen, H. Zhang, R. Serban, and D. Negrut, "A physics-based sensor simulation environment for lunar ground operations," in *IEEE Aerospace Conference, Big Sky, MT, USA*, 2025.

[26] S. Benatti, A. Young, A. Elmquist, J. Taves, R. Serban, D. Mangoni, A. Tasora, and D. Negrut, "Pychrono and Gym-Chrono: A deep reinforcement learning framework leveraging multibody dynamics to control autonomous vehicles and robots," in *Advances in Nonlinear Dynamics*, pp. 573–584, Springer, 2022.

[27] R. Luo, S. Zhao, J. Kuck, B. Ivanovic, S. Savarese, E. Schmerling, and M. Pavone, "Sample-efficient safety assurances using conformal prediction," in *International Workshop on the Algorithmic Foundations of Robotics*, pp. 149–169, Springer, 2022.

[28] L. Lin, W. Li, H. Bi, and L. Qin, "Vehicle trajectory prediction using LSTMs with spatial–temporal attention mechanisms," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 197–208, 2021.

[29] Federal Highway Administration, "Next generation simulation (NGSIM)." `https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm`.

[30] W.-H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-observer-based control and related methods—An overview," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1083–1095, 2015.

[31] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, pp. 322–328, IEEE, 2016.

[32] Y. Wang and X. Xu, "Disturbance observer-based robust control barrier functions," in *American Control Conference*, pp. 3681–3687, IEEE, 2023.