WayWiseR: A Rapid Prototyping Platform for Validating Connected and Automated Vehicles

Ramana Reddy Avula, Marvin Damschen, Aria Mirzai, Karl Lundgren, Ashfaq Farooqui, Anders Thorsén
Dependable Transport Systems, RISE Research Institutes of Sweden, 501 15 Borås, Sweden
Email: {ramana.reddy.avula, marvin.damschen, aria.mirzai, karl.lundgren, ashfaq.farooqui, anders.thorsen}@ri.se

***Abstract*—Validating connected and automated vehicles (CAVs), specifically Automated Driving Systems (ADS), remains a challenge, particularly in ensuring safety and reliability across diverse operational scenarios. Before an ADS can be considered safe for deployment, it must be evaluated across a wide range of carefully designed test cases that capture both expected and edge case conditions. As recognized in the UNECE's New Assessment/Test Method for Automated Driving (NATM), testing all such scenarios on a real system is often impractical, making virtual testing an essential complement to physical tests. To enable this, we present WayWiseR, an open-source rapid prototyping platform built on ROS2 that supports researchers in developing and evaluating validation methodologies for CAVs. By integrating modular components, simulation environments such as CARLA, and scaled vehicle hardware, WayWiseR enables reproducible experimentation and flexible orchestration of test scenarios across both virtual and physical platforms. We demonstrate the platform through two representative use cases: autonomous reverse docking in a logistics hub, and human detection and emergency braking in forestry environments. The results demonstrate WayWiseR's ability to bridge simulation-based validation with real-world operational testing, thereby supporting the safer deployment of sufficiently validated CAVs.**

***Index Terms*—Autonomous Driving, CAV Validation, Scenario-Based Testing, Virtual Testing, ROS2**

I. Introduction

As connected and automated vehicles (CAVs) become increasingly complex, ensuring their safe and reliable operation requires systematic evaluation of both nominal and edge-case scenarios. Traditional in-traffic testing cannot feasibly cover the vast number of possible situations, a challenge often referred to as the "billion-mile problem" [1]. A widely accepted solution is a scenario-based approach that combines different validation environments. Building on industrial best practice, UNECE's [2] Informal Working Group on Validation Methods for Automated Driving (VMAD) defined the New Assessment/Test Method for Automated Driving (NATM) [3], a scenario-based multi-pillar framework composed of a scenario catalog and five complementary validation methodologies: simulation, track testing, real-world testing, audit/assessment, and in-service monitoring.

While frameworks like NATM provide guidance on the types of validation activities required, researchers and developers face practical challenges in implementing, coordinating, and systematically evaluating these methodologies. Variations in vehicle platforms, sensor configurations, software stacks, and test environments make it difficult to reproduce experiments, compare results, and iterate on validation strategies efficiently. This highlights the need for flexible, modular tools that enable rapid prototyping and systematic evaluation of validation methods, bridging simulation and physical testing while ensuring consistent behavior across diverse platforms.

To address this need, WayWiseR[1] was developed as an open-source rapid prototyping platform for CAV validation research. Built on ROS 2, WayWiseR integrates modular components, simulation environments such as CARLA, and scaled physical vehicles, enabling researchers to quickly design, test, and iterate validation concepts. Its layered architecture separates high-level automation, safety enforcement, and low-level vehicle interaction, enabling systematic exploration of scenario-based testing, safety assurance, and cybersecurity assessment techniques, while minimizing low-level integration effort and accelerating research workflows.

In this paper, we present WayWiseR and demonstrate its capabilities through two representative use cases. The first, from the SUNRISE project [4, 5], focuses on autonomous reverse docking of a truck with a semitrailer in a logistics hub, illustrating how WayWiseR supports precise maneuvering, safety monitoring, and scenario-based validation in line with the SUNRISE Safety Assurance Framework (SAF) [6]. The second, from the AGRARSENSE project [7], addresses human detection and emergency braking in forestry operations, highlighting the platform's ability to evaluate safety-critical perception and emergency response functions across diverse operational domains. Together, these use cases demonstrate how WayWiseR enables rapid prototyping and rigorous assessment of validation methodologies across simulated and physical environments.

The remainder of the paper is structured as follows. Section II reviews related work on CAV validation platforms and tools. Section III provides an overview of the WayWiseR architecture and its layered design. Section IV presents the two use cases and experimental results demonstrating the platform's capabilities. Finally, Section V concludes the paper and discusses future development directions for WayWiseR.

This work is carried out within the SUNRISE (Grant Agreement No. 101069573) and AGRARSENSE (Grant Agreement No. 101095835) projects funded by the European Union's Horizon Europe research and innovation programme.

[1]https://github.com/RISE-Dependable-Transport-Systems/WayWiseR

II. Related Work

Bridging the gap between simulation-based development and deployment on physical platforms is a recurring challenge for CAV research. Numerous software frameworks addressing this gap have been proposed, with the Robot Operating System 2 (ROS 2) emerging as a de facto middleware for modular, distributed autonomy stacks [8, 9]. Its tooling, large ecosystem, and industry support make it widely adopted in both full-scale and small-scale CAV testbeds. However, in the context of rapid prototyping and hardware bring-up, researchers have identified the need for complementary tools that simplify integration with low-level hardware interfaces while retaining access to the ROS 2 ecosystem.

Several open-source platforms address parts of this problem space. Autoware [10] and Apollo [11] provide complete ADS stacks on top of ROS 2 or custom middleware, targeting full-scale deployment but with significant complexity and overhead for early-stage proof-of-concepts. Conversely, small-scale research testbeds such as F1TENTH [12], BARC [13], and Duckietown [14] couple ROS or ROS 2 with custom hardware drivers to enable rapid experimentation in constrained environments. These systems typically bind the autonomy logic directly to the robot hardware, limiting flexibility when re-targeting between simulation and real vehicles.

At the simulation level, comprehensive tools like CARLA [15] and Gazebo [16] are widely used for ADS evaluation, and bridges to ROS 2 are common. Yet, as Klüner et al. note [8], many simulators do not natively integrate with low-level control stacks, requiring additional adaptation layers to reconcile simulator I/O with the vehicle's control and perception subsystems. Frameworks such as AutoDRIVE [17] combine simulation, scaled physical platforms, and middleware integration, but are vertically integrated and not primarily aimed at projects needing lightweight, hardware-focused bring-up support.

WayWise [18], on which WayWiseR is built, is a standalone rapid-prototyping platform for CAVs that emphasizes direct interfacing with motor controllers, servos, IMUs, and GNSS hardware. WayWiseR's integration of WayWise with ROS 2 addresses a gap left by the above systems by providing a reusable abstraction layer that can operate in two directions: either exposing WayWise-handled hardware to ROS 2 autonomy stacks (e.g., Nav2, SLAM Toolbox) or augmenting WayWise-centric control with ROS 2-based perception, planning, and simulation modules. Similar bidirectional integration concepts have been explored in mixed-reality CAV testbeds [17] and multi-stack orchestration frameworks [19], but existing approaches are often domain-specific or tied to proprietary components. WayWiseR therefore contributes a flexible, open, and hardware-inclusive integration path that leverages the modularity of ROS 2 while preserving the rapid-bring-up capabilities of WayWise.

III. Architectural Design

The WayWiseR platform follows a modular, ROS2-based architecture designed to support both simulation-based and physical vehicle testing in a unified manner. As illustrated in Figure 1, the main components of the platform are organized into three conceptual layers with distinct responsibilities: a high-level automation layer for mission planning and perception, a safety layer for command arbitration and collision monitoring, and a low-level vehicle interaction layer for hardware abstraction. This separation enables the platform to maintain consistent behavior across different deployment scenarios while ensuring safety-critical functions remain isolated and verifiable.

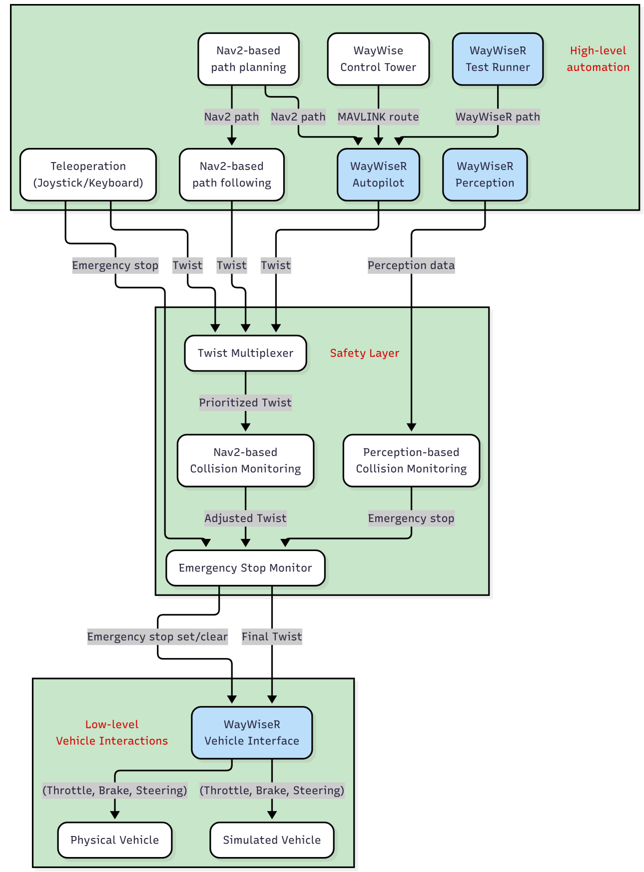


Fig. 1: WayWiseR platform architecture illustrating the hierarchical control flow, supporting both physical and simulated vehicle deployments.

A. High-Level Automation Layer

The high-level automation layer encompasses the primary intelligence components responsible for generating candidate motion commands and providing environmental perception. The *WayWiseR Test Runner* plays a central role in systematic validation by orchestrating predefined test cases, supplying corresponding routes, and coordinating comprehensive data logging through rosbags.

At the core of execution is the *WayWiseR Autopilot*, built on the WayWise library, which implements path-following functionality using a pure-pursuit waypoint follower. This supports both simple Ackermann vehicles and articulated truck–trailer configurations, enabling the platform to handle a wide range of mobility systems with consistent control strategies. In addition to path tracking, the Autopilot integrates other high-level motion planning inputs, including scenario-driven routes supplied by the *WayWiseR Test Runner*, navigation paths generated by Nav2-based planners, and mission-level instructions received via MAVLink-compatible interfaces such as the Control Tower. By combining its path-following functionality with these coordinated planning inputs, the Autopilot produces a unified stream of candidate commands for downstream safety checks.

The *WayWiseR Perception* module provides real-time environmental awareness through computer vision, making detection data available to the safety systems. Human operators can also intervene at any time via *Teleoperation* interfaces using a joystick or keyboard control. These manual overrides are essential for safety-critical testing and, like all other inputs, are subject to the same safety validation before reaching the vehicle interface.

B. Safety Layer

The safety layer acts as a critical intermediary between high-level automation and low-level vehicle actuation, ensuring that all candidate commands are evaluated and filtered before reaching the vehicle interface. Its primary purpose is to enforce safety and prevent hazardous behavior, regardless of the source of motion commands.

The *Twist Multiplexer* serves as the first stage of command arbitration, filtering and prioritizing velocity inputs from the *WayWiseR Autopilot*, *Nav2*, and *Teleoperation* interfaces based on predefined safety hierarchies. Once multiplexed, two parallel collision monitoring systems provide critical redundancy: *Nav2-based collision monitoring* for traditional obstacle avoidance by evaluating potential collisions, while the *perception-based collision monitoring* leverages real-time detection data for taking actions based on identified dynamic obstacles such as pedestrians or vehicles.

The *Emergency Stop Monitor* sits at the top of the safety hierarchy, with overriding authority to halt vehicle motion in the event of imminent collisions, system faults, or explicit operator intervention. It integrates emergency stop requests from multiple sources and ensures that any hazardous command is immediately suppressed. The outputs from this layer consist of the final, safety-validated twist commands and emergency stop signals, which are then forwarded to the low-level vehicle interface for execution.

C. Low-Level Vehicle Interaction Layer

The low-level vehicle interaction layer is responsible for bridging the safety-validated motion commands from the upper layers to the physical or simulated vehicle hardware while abstracting platform-specific differences.

At the core of this layer is the *WayWiseR Vehicle Interface*, which receives the final twist commands and emergency stop signals from the safety layer. For physical vehicles, it manages communication with motor controllers, servos, braking systems, steering actuators, and onboard sensors such as IMUs and GNSS receivers. For simulated deployments, the same commands are translated into the appropriate simulator control messages, allowing identical software behavior to be validated in virtual environments such as CARLA, Unreal Engine-based AGRARSENSE simulator [20], or Gazebo.

The interface also provides low-level feedback to the upper layers, including vehicle state, sensor readings, and actuator status. This feedback is critical for closed-loop control, monitoring, and logging, enabling both the Autopilot and safety layers to make informed decisions. By isolating hardware-specific logic within this layer, WayWiseR achieves modularity, simplifies integration with new vehicle platforms, and ensures consistent control logic across both real and virtual vehicles without requiring modifications to higher layers.

IV. Experimental Validation using WayWiseR

To demonstrate the practical capabilities of the WayWiseR platform, we conducted experimental validation across two representative use cases that reflect real-world operational challenges and safety-critical scenarios. The selected use cases illustrate the platform's ability to bridge simulation-based testing with physical vehicle deployment while ensuring adherence to safety requirements and systematic data collection. The first use case from SUNRISE focuses on autonomous reverse docking in a logistics hub, representing a controlled, low-speed environment for heavy vehicles. The second use case from AGRARSENSE addresses human detection and emergency braking in forestry operations, highlighting dynamic obstacle handling in semi-structured environments.

A. Use Case 1: Autonomous Reverse Docking in Logistics Hub

The use case examines the reverse parking of a truck with a semitrailer within a logistics hub, as illustrated in Figure 2. A logical scenario in this use case comprises the semi-truck starting from a staging area, maneuvering backward at low speed until it reaches the docking area. The reverse parking function is defined to operate within a controlled environment with a bounded Operational Design Domain (ODD). The corresponding scenario space encompasses all plausible starting positions and orientations that a truck with a trailer might assume within the defined square staging area. Environmental parameters such as lighting are initially set to baseline conditions, with potential variations introduced to extend the ODD during further testing. The primary objective of this use case is to validate the safety of the reverse parking function, thereby demonstrating relevant aspects of the Safety Assurance Framework (SAF) developed in the SUNRISE project [6, 21]. To assess safety performance, a hazard analysis and risk assessment (HARA) [22, 23] was conducted, resulting in two Safety Goals (SGs) and three Key Performance Indicators (KPIs), summarized in Figure 2.

To evaluate Safety Goal SG1 (*Vehicle shall not collide*), we used the WayWiseR platform to conduct scenario-based tests of the reverse parking functionality on a physical 1:14

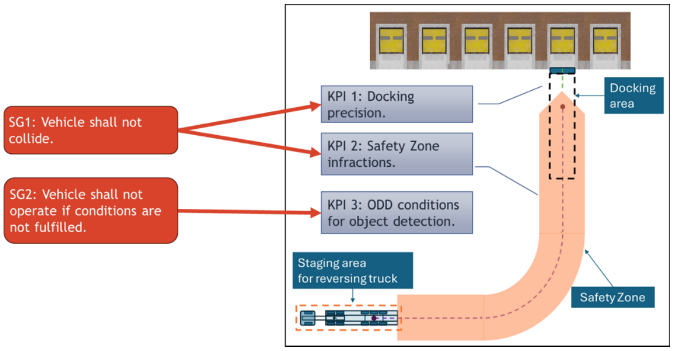


Fig. 2: Overview of the use case in SUNRISE where a truck with a trailer autonomously reverses to a docking station, adhering to two Safety Goals (SGs) which are evaluated using three Key Performance Indicators (KPIs).

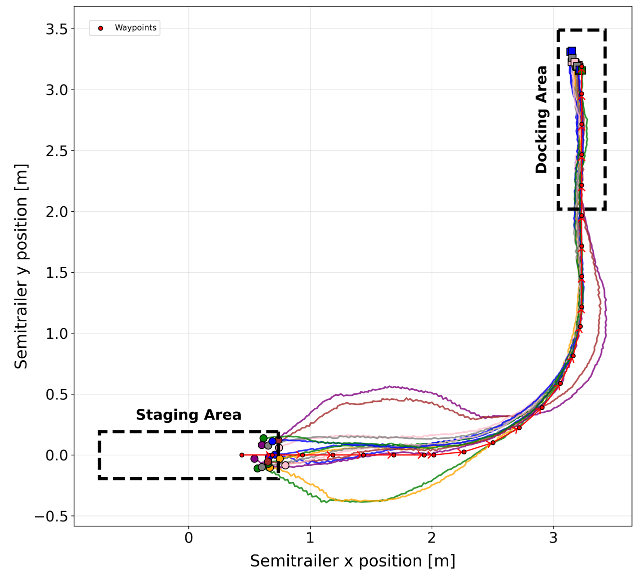


Fig. 3: Trajectories of the scaled semitrailer in physical tests across 16 edge-case scenarios.

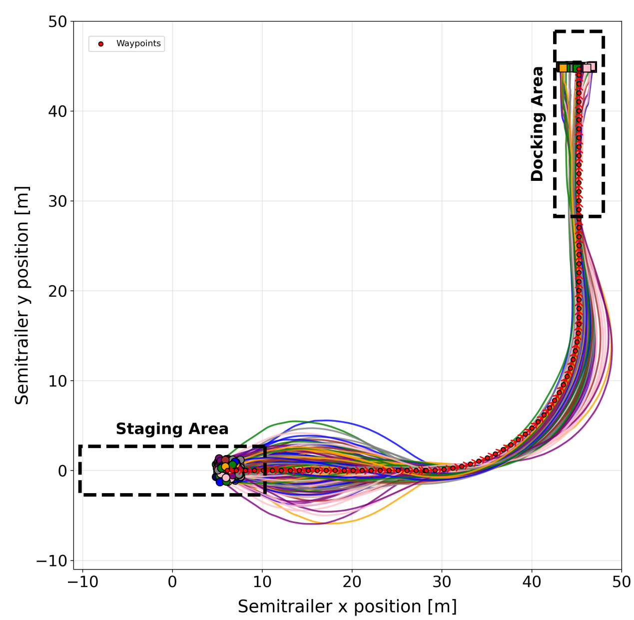


Fig. 4: Trajectories of the semitrailer in CARLA simulation across 200 sampled scenarios, including 16 edge-cases.

scaled model semi-truck and a representative full-scale virtual model developed in the CARLA simulator. The platform enabled automated orchestration and data logging of test cases through the *WayWiseR Test Runner*. In total, 16 edge-case scenarios were defined to represent challenging starting positions, orientations, and environmental conditions within the staging area. These edge-case scenarios were executed on the physical scaled model to assess the reverse parking behaviour in a realistic environment. For broader coverage, including both nominal and challenging situations, a larger set of 200 sampled scenarios, including the same 16 edge cases, was executed in the CARLA simulator.

Figures 3 and 4 show representative trajectories of the semitrailer rear axle across the two test environments. In each scenario, the semitrailer successfully navigates from the staging area to the docking station without leaving the predefined safety zone or colliding with the docking station, demonstrating compliance with SG1. The first KPI (*Docking Precision*) is evaluated by the final position and orientation of the trailer relative to the docking station, while the second KPI (*Safety Zone Infractions*) is assessed by monitoring deviations from the allowed maneuvering area throughout the trajectory. Both KPIs indicate that the reverse parking function maintains safety-critical performance under the tested operational conditions.

B. Use Case 2: Human Detection in Forestry Operations

The second use case originates from the AGRARSENSE project and focuses on the validation of human detection and emergency braking functions for an autonomous forestry shuttle, illustrated in Figure 5. The shuttle is conceived as an autonomous alternative to a conventional forwarder, transporting cut logs from the harvester to a landing area along pre-mapped forest roads. Its platform consists of a tracked chassis, onboard compute hardware, and a suite of perception sensors. Although still in the prototype stage, its intended operational profile is well defined: after startup checks, the shuttle departs under remote supervision, follows a geofenced corridor at low speed while carrying its payload, and executes a controlled stop at the designated drop-off point.

To systematically address safety risks, a preliminary hazard analysis following ISO 12100 identified the most critical hazard as a person entering the shuttle's forward hazard zone and being struck. Based on this, a primary safety goal SG1 (*Vehicle shall prevent collision with humans in its forward hazard zone*) was defined. The hazard zone is defined as a forward arc sector, characterized by a radius equal to the maximum stopping distance plus a safety margin, and by a sector angle

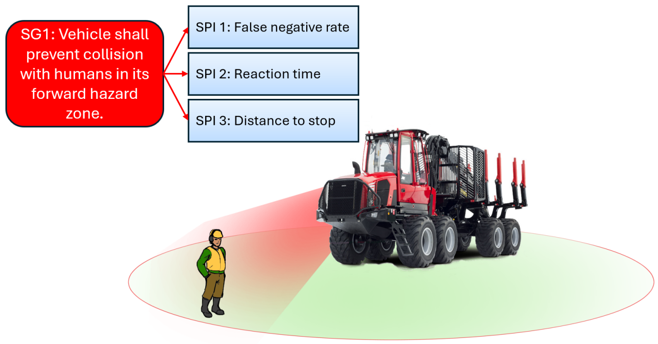
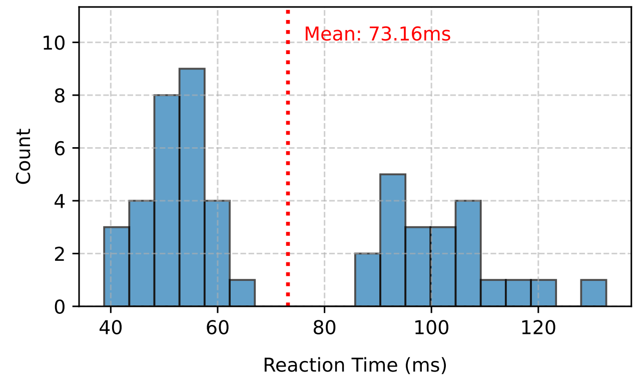


Fig. 5: Overview of the AGRARSENSE forestry shuttle in its operational context, highlighting the primary safety goal (SG1) and three associated Safety Performance Indicators (SPIs).

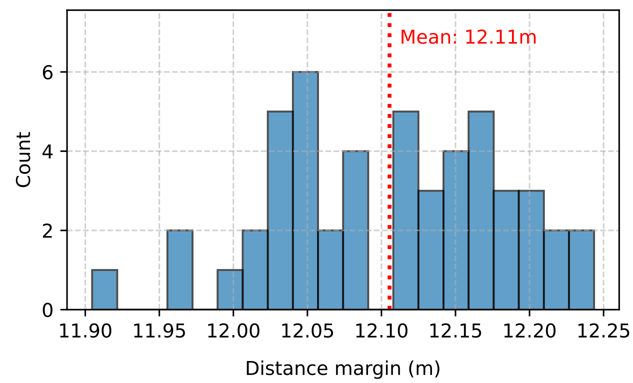
corresponding to the lateral field of view to be monitored ahead of the vehicle. To mitigate this hazard, the shuttle employs a combination of passive and active safety measures. Passive measures include audible and visual warnings during startup and motion, geofenced route enforcement, and remote-supervision heartbeats that trigger an emergency stop on communication loss. In addition, the shuttle integrates an active safety system for detecting people, which continuously monitors the hazard zone and commands an immediate stop if a person is detected or if perception quality falls below acceptable thresholds. When a trigger occurs, whether human presence, loss of perception data, or operator command, the vehicle transitions to a safe state by engaging the service and parking brakes and cutting drive power.

To validate the defined safety goal (SG1), three Safety Performance Indicators (SPIs) were introduced, as summarized in Figure 5. The validation scenario represents a typical segment of a forest road where the autonomous shuttle transports cut logs from the harvester to a landing area. The shuttle starts from a stationary position, accelerates to a low operational speed along a geofenced corridor, and is expected to stop at the designated drop-off point. During the test, a human subject enters the shuttle's forward hazard zone along its path. As in Use Case 1, environmental and operational baseline parameters such as vehicle speed, payload, road segment, and lighting conditions are kept constant to systematically assess the shuttle's performance in a single scenario.

Experiments were carried out using both a 1:10 scaled physical prototype and a full-scale model in the AGRARSENSE simulator. The scaled prototype shuttle was equipped with a Raspberry Pi 5 and a Luxonis OAK-D Pro W RGB-D camera running a lightweight YOLOv3-Tiny network for person detection. In a single preliminary physical trial with a hazard zone radius of 1.32 m, a scaled human model was placed along the shuttle's path while it moved at a low speed of 0.5 m/s. The onboard perception system successfully detected the human model and issued an emergency stop command. In this test, the first two SPIs were estimated as 0% false negative rate and 15.63 ms reaction time from the image



(a) Distribution of reaction times from human detection to emergency stop command.



(b) Distribution of distance between the detected human model and the shuttle after stopping.

Fig. 6: Simulation results over 50 runs in the AGRARSENSE simulator: (a) reaction times and (b) distance margins.

capture moment to the emergency stop request received by the vehicle interface (for brake actuation) over a 3 s interval after the first trigger, when the human model entered the hazard zone. The third SPI (distance margin) was measured as 1.15 m, representing the distance between the detected object and the vehicle after stopping. As this was only a first trial, additional repeated tests are planned to fully assess performance under varied human entry positions, approach angles, and environmental conditions such as time of day and lighting, providing a more comprehensive evaluation of the system's robustness and reliability in real-world conditions.

In parallel, the same scenario was reproduced in the AGRARSENSE simulator [20] under clear sunny day ODD conditions for 50 independent runs. Across these simulation runs with a hazard zone radius of 13.2 m, a full-scale human model was placed along the shuttle's path while it moved at a low speed of 1.39 m/s (≈5 km/h). The perception system implemented in WayWiseR, using the same lightweight YOLOv3-Tiny network as in physical trials, consistently detected the human model, resulting in a 0% false negative rate over a 10 s interval after the first trigger. The mean reaction time from human detection to emergency stop command was measured as 73.16 ms, and the mean distance margin after stopping was 12.11 m. The distribution of reaction times and distance margins across all runs is shown separately in Figures 6a and 6b. These simulation results, together with the preliminary physical trial, provide an initial assessment of the shuttle's ability to achieve SG1, while further trials under varied environmental conditions are planned to fully evaluate system robustness and safety performance in realistic operational scenarios.

V. Conclusions and Future Directions

In this work, we introduced WayWiseR, an open-source platform designed to support rapid prototyping of models and methods for connected and automated vehicle (CAV) validation research. By integrating modular components from ROS 2, simulation environments such as CARLA, and scaled vehicle hardware, the platform enables validation concepts to be developed, tested, and iterated rapidly through reproducible experimentation. The layered architecture of WayWiseR, which separates high-level automation, safety enforcement, and low-level vehicle interaction, provides a consistent basis for evaluating validation methodologies across both simulated and physical environments. Its modularity, flexible integration, and consistent execution across platforms make it well-suited for systematic scenario-based testing, safety assurance, and cybersecurity assessment.

The platform's capabilities were demonstrated through two representative use cases: autonomous reverse docking in a logistics hub and human detection with emergency braking in forestry operations. These use cases highlight how WayWiseR enables the same validation methods and underlying software to be applied seamlessly across both virtual and physical testing environments, supporting reproducibility and efficient iteration in research workflows. By providing a structured, modular framework, WayWiseR allows researchers to focus on designing and evaluating validation strategies rather than low-level integration challenges, accelerating the exploration of novel safety and performance assessment techniques.

While the current implementation demonstrates its capabilities, WayWiseR remains under active development, with planned enhancements to support experimentation using standard scenario descriptions such as OpenSCENARIO and to expand the range of supported vehicle types and operational domains, including aerial and maritime systems. These extensions will further enhance the platform's utility as a versatile research instrument, enabling more comprehensive studies of CAV behavior across diverse scenarios and contributing to the development of robust autonomous systems.

References

[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, Dec. 2016, ISSN: 0965-8564. DOI: 10.1016/j.tra.2016.09.010.

[2] "Working party on automated/autonomous and connected vehicles - introduction — UNECE." (), [Online]. Available: https://unece.org/transport/road-transport/working-party-automatedautonomous-and-connected-vehicles-introduction (visited on Aug. 15, 2025).

[3] ECE/TRANS/WP.29/2023/44/Rev.1, *New assessment/test method for automated driving (NATM) guidelines for validating automated driving system (ADS)*, Jun. 7, 2023. [Online]. Available: https://unece.org/sites/default/files/2023-06/ECE-TRANS-WP.29-2023-44-r.1e%20.pdf (visited on Aug. 14, 2025).

[4] "Safety assUraNce fRamework for connected, automated mobIlity SystEms — SUNRISE — project — fact sheet — HORIZON," CORDIS — European Commission. (), [Online]. Available: https://cordis.europa.eu/project/id/101069573 (visited on Aug. 13, 2025).

[5] "Sunrise project — developing and providing a harmonized and scalable CCAM safety assurance framework." (), [Online]. Available: https://ccam-sunrise-project.eu/ (visited on Aug. 15, 2025).

[6] "Performance assurance — sunrise project." (), [Online]. Available: https://ccam-sunrise-project.eu/performance-assurance/ (visited on Aug. 15, 2025).

[7] CORDIS, *AGRARSENSE - Smart, digitalized components and systems for data-based Agriculture and Forestry*. [Online]. Available: https://cordis.europa.eu/project/id/101095835.

[8] D. Klüner, S. Schäfer, L. Hegerath, J. Xu, J. Kahle, H. Ibrahim, A. Kampmann, and B. Alrifaee, *Simulation to Reality: Testbeds and Architectures for Connected and Automated Vehicles*, 2025. arXiv: 2505.03472 [cs.MA]. [Online]. Available: https://arxiv.org/abs/2505.03472.

[9] 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, eabm6074, 2022. DOI: 10.1126/scirobotics.abm6074. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[10] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi, "Autoware on board: Enabling autonomous vehicles with embedded systems," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, IEEE, 2018, pp. 287–296.

[11] *Baidu Apollo team (2017), Apollo: Open Source Autonomous Driving*, https://github.com/ApolloAuto/apollo, Accessed: 2025-08-15.

[12] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," *Proceedings of Machine Learning Research*, vol. 123, 2020.

[13] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: A predictive control approach," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2019.

[14] L. Paull *et al.*, "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," in *Int. Conf. on Rob. & Aut. (ICRA)*, IEEE, 2017, pp. 1497–1504.

[15] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on robot learning*, PMLR, 2017, pp. 1–16.

[16] "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Int. conf. on intel. robots & sys. (IROS)*, Ieee, vol. 3, 2004, pp. 2149–2154.

[17] T. Samak, C. Samak, S. Kandhasamy, V. Krovi, and M. Xie, "AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education," *Robotics*, vol. 12, no. 3, 2023, ISSN: 2218-6581. DOI: 10.3390/robotics12030077. [Online]. Available: https://www.mdpi.com/2218-6581/12/3/77.

[18] M. Damschen, R. Häll, and A. Mirzai, "Waywise: A rapid prototyping library for connected, autonomous vehicles," *Software Impacts*, vol. 21, p. 100 682, 2024. DOI: https://doi.org/10.1016/j.simpa.2024.100682.

[19] J.-V. Zacchi, E. Clementi, and N. Mata, "APIKS: A Modular ROS2 Framework for Rapid Prototyping and Validation of Automated Driving Systems," *arXiv preprint arXiv:2502.20507*, 2025.

[20] FrostBit Software Lab (Lapland UAS), *Agrarsense simulator*. [Online]. Available: https://dev.azure.com/AMKFrostBit/AGRARSENSE (visited on Aug. 15, 2025).

[21] Bernhard Hillbrand, Gerhard Benedikt Weiss, Daniel Hassler, Sarp Yetkin, Bryan Bourauel, Jose Miguel Torres Camara, *et al.*, "SUNRISE D7.2 safety assurance framework demonstration instances design," SUNRISE project, Deliverable D7.2 V1.0, Nov. 22, 2024.

[22] ISO, *ISO 26262:2018 Road Vehicles : Functional Safety*. ISO, 2018. [Online]. Available: www.iso.org.

[23] ISO, *ISO 21448:2022 - Road vehicles – Safety of the intended functionality*. Jun. 30, 2022. [Online]. Available: https://www.iso.org/standard/77490.html.