POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Practical Implementation of Day-3 Cooperative Intersection with Automated Connected Mini-Cars

Original
A Practical Implementation of Day-3 Cooperative Intersection with Automated Connected Mini-Cars / Farina, L., Todisco, V., Gavioli, F., Iandolo, S., Moretti, F., Perrone, G., Piccoli, M., Raviglione, F., Rapelli, M., Solida, A., Casetti, C., Burgio, P., Grazia, C.A., Bazzi, A.. - (In corso di stampa). (2026 IEEE 29th International Conference on Intelligent Transportation Systems (ITSC 2026) Naples, Italy September 15-18, 2026).

Availability:
This version is available at: 11583/3016332 since: 2026-10-05T12:49:17Z

Publisher:
IEEE

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

19 November 2026

arXiv:2606.30838v1 [cs.NI] 29 Jun 2026

A Practical Implementation of Day-3 Cooperative Intersection with Automated Connected Mini-Cars

Lorenzo Farina*†, Vittorio Todisco*†, Federico Gavioli‡, Salvatore Iandolo§, Francesco Moretti‡, Giuseppe Perrone‖, Matteo Piccoli†, Francesco Raviglione¶, Marco Rapelli‖, Antonio Solida§, Claudio Casetti‖, Paolo Burgio‡, Carlo Augusto Grazia§, and Alessandro Bazzi*†

*DEI, Università di Bologna, 40136 Bologna, Italy
†National Laboratory of Wireless Communications (WiLab), CNIT, 40136 Bologna, Italy
‡FIM, University of Modena and Reggio Emilia, 41125 Modena, Italy
§DIEF, University of Modena and Reggio Emilia, 41125 Modena, Italy
¶Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, 10129 Torino, Italy
‖Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy

***Abstract*—Cooperative driving enabled by connected and automated vehicles is expected to improve traffic efficiency and safety, particularly at intersections where traditional control mechanisms such as traffic lights introduce delays and unnecessary stops. Although cooperative intersection management algorithms have been widely studied, experimental demonstrations remain limited. This paper presents a real-time demonstration of cooperative intersection management using connected autonomous mini-cars. The testbed consists of multiple 1:10 scale vehicles equipped with autonomous driving capabilities and wireless communication modules that interact with a centralized controller responsible for scheduling their crossing of the intersection. Vehicles approaching the intersection exchange messages with the controller to set the appropriate mobility profile to traverse the intersection without stopping. The demonstration integrates autonomous driving, wireless communication, and cooperative control in a single experimental platform, providing a practical environment for validating cooperative intersection management concepts for future intelligent transportation systems.**

I. Introduction

Recent advancements in sensing, automation, and wireless communication are enabling connected and autonomous vehicles (CAVs) to exchange information with other vehicles and roadside infrastructure through vehicle-to-everything (V2X) technologies. These capabilities support cooperative driving strategies in which vehicles adapt their behavior based on shared information. The development of cooperative-intelligent transport systems (C-ITS) services is often described through a progressive roadmap referred to as Day 1, Day 2, and Day 3. Day 1 applications focus on cooperative awareness through the periodic exchange of vehicle status information. Day 2 services extend this paradigm through collective perception, where vehicles share information about objects detected by their onboard sensors. Day 3 introduces cooperative maneuver coordination, where vehicles exchange information about their intended actions and coordinate their trajectories to perform complex maneuvers. In Europe, such coordination is expected to be supported by Maneuver Coordination Messages (MCM), currently under standardization within ETSI.

In the context of Day-3 services, coordinated maneuvers will be a cornerstone to improve road efficiency, especially in cities, where cooperation at intersection can increase road capacity significantly [1]. Road intersections therefore represent one of the most relevant yet challenging scenarios for cooperative driving. At these locations, multiple traffic flows interact, and traditional control mechanisms such as traffic lights or right-of-way rules are designed primarily to guarantee safety rather than to optimize traffic efficiency. As a result, vehicles are often required to stop or wait even when no immediate conflicts are present.

Recently, several studies have investigated cooperative approaches to intersection management, where CAVs exchange information about their planned trajectories and adapt their motion accordingly [2]. Proposed solutions include reservation-based schemes [3], trajectory optimization methods [4], and centralized coordination strategies [5]. However, most of these studies rely on simulations or analytical models, while experimental validation remains comparatively limited due to the complexity of integrating autonomous driving, communication technologies, and coordination algorithms in real systems. Although some advanced field trials have been reported, for example in [6], [7], these experiments typically involve only two or three vehicles and consider a single crossing event under highly controlled conditions.

To mitigate the cost and safety challenges associated with real-world testing, experimental platforms based on scaled autonomous vehicles provide a practical way to investigate cooperative driving concepts. Such platforms, which have been primarily used to study autonomous driving functionalities [8], allow researchers to integrate sensing, communication, and control components within a manageable and reproducible setup.

In this paper, we present an experimental demonstration of cooperative intersection management using connected autonomous mini-cars. The demonstration involves multiple

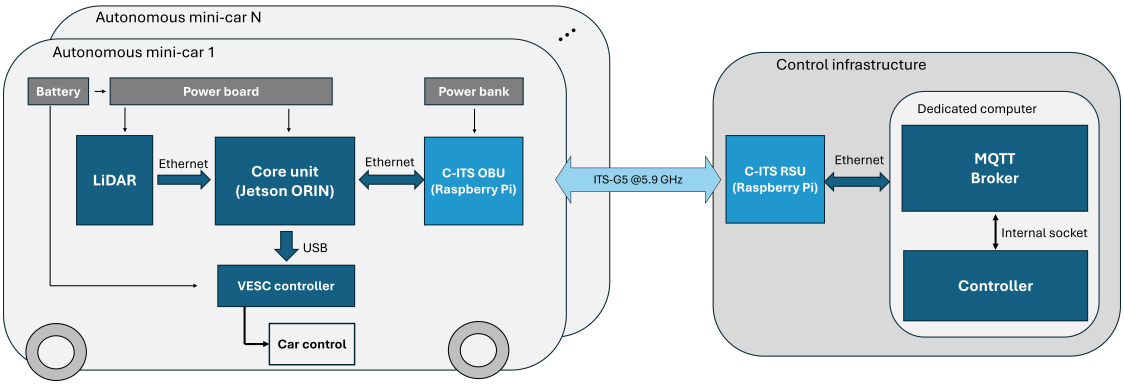


Fig. 1: Testbed architecture.

1:10 scale CAVs operating on a figure-eight track and interacting with a centralized controller. Communication between vehicles and the controller is implemented using message queuing telemetry transport (MQTT) over ITS-G5 in the ITS band at 5.9 GHz. As a representative example of the capabilities of the platform, the system implements the coordination algorithm described in [9], which is scalable and lightweight from a communication perspective.

The objective of this work is to demonstrate the integration of autonomous driving, wireless communication, and centralized scheduling within a single experimental platform. The resulting testbed provides a practical environment for studying cooperative intersection management and validating communication-assisted coordination mechanisms in a controlled setting.

II. Testbed Architecture Overview

The system architecture consists of three main components: the autonomous mini-cars, the wireless communication subsystem, and the control infrastructure. A block scheme is provided in Fig. 1.

A. Autonomous Mini-Cars

The autonomous mini-cars are based on the Roboracer platform [10] (formerly F1TENTH), which is an open-source framework widely used for autonomous driving research. The platform provides modular hardware and open source software components that can be adapted to a variety of experimental scenarios.

From a hardware perspective, each vehicle uses a commercially available 1:10 scale radio-controlled chassis equipped with a custom deck hosting the computing, sensing, and power subsystems. A Vedder Electronic Speed Controller (VESC) controls both the brushless DC motor (speed) and the steering servomotor, providing motor abstraction and sensor feedback through an integrated BMI160 Inertial Measurement Unit (IMU). Environmental perception is primarily provided by a Hokuyo UST-10LX 2D LiDAR sensor.

The central computing unit is an NVIDIA Jetson Orin platform (either the 8-core NX or the 6-core Nano Super), which provides both CPU and GPU resources for running the autonomous driving stack.

The software architecture is built on Robot Operating System 2 (ROS2) [11], leveraging its publish–subscribe communication model to enable a modular and extensible system. The main software components include sensor and VESC drivers, a pure pursuit trajectory-following controller [12], and a custom localization module combining LiDAR and IMU measurements. Future work will integrate an open-source localization component based on Monte Carlo particle filtering [13].

B. Wireless Communication Subsystem

The cooperative intersection management system requires vehicles to exchange information with the control infrastructure. This communication is implemented through short-range vehicular wireless communication operating in the ITS band at 5.9 GHz.

In this work, we employ a compact open-source communication module previously developed to support IEEE 802.11p and ETSI ITS-G5 operation; a detailed description of the hardware architecture and its configuration can be found in [14]. The core of the unit is a Raspberry Pi 5 Model B coupled with a MikroTik R11e-5HnD wireless card based on the Atheros AR9580 chipset. Although originally designed for IEEE 802.11a/n operation, this chipset can be patched to support communication in the ITS band [15].

The platform is capable of exchanging ETSI-compliant messages through the open-source OScar stack [16], [17]. However, this functionality is not used in the cooperative intersection management application presented in this paper, since maneuver coordination messages are still under standardization. Instead, vehicles and controller exchange application-level messages using the MQTT protocol over TCP/IP.

Each mini-car is equipped with one of these units acting as an on-board unit (OBU). An additional unit is deployed at the controller node and operates as a road side unit (RSU), enabling wireless communication between the vehicles and the centralized controller.

C. Control Infrastructure

The control infrastructure runs on a centralized processing node, implemented on a standard laptop. The node hosts two

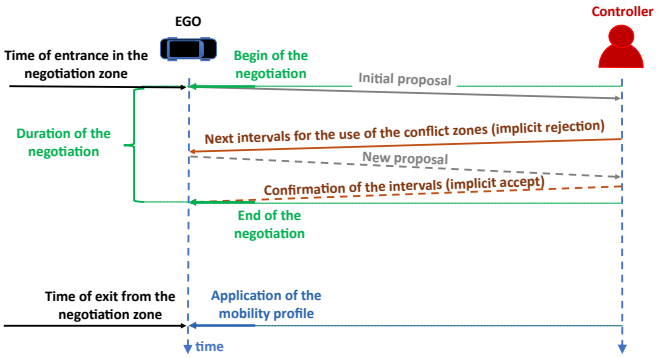


Fig. 2: Temporal sequence of the negotiation procedure between a vehicle and the controller. In this example, the initial proposal is rejected and the second proposal is accepted.

main software components.

- The controller, which executes the scheduling algorithm responsible for coordinating the mini-cars and preventing collisions at the intersection. The algorithm is implemented in Python and its core logic is described in Section III.
- An MQTT broker used to support communication between the vehicles and the controller. The broker manages a set of topics to route messages: one shared topic for messages sent by the vehicles to the controller, and one dedicated topic per vehicle through which the controller sends coordination commands. The messaging service is implemented using the open-source `Eclipse Mosquitto` daemon.

III. Algorithm for the Intersection Management

The cooperative intersection management demonstrated in this work relies on a scheduling algorithm, called seamless mobility of vehicles over intersections (Moveover), and the related communication protocol. A detailed description of the algorithm can be found in [9]; here, we summarise the key aspects of the solution.

The execution of Moveover is distributed between the individual CAVs and the centralized controller. The goal of the algorithm is to determine, for each vehicle, a mobility profile[1] that allows it to traverse the intersection safely and without stopping, except in situations of traffic congestion.

Only one vehicle at a time can negotiate its mobility profile with the controller, and the CAV performing this operation is referred to as the EGO. Once scheduled, the vehicle follows the assigned mobility profile during normal operation.

If abnormal conditions occur, such as communication failures or unexpected traffic situations, the system switches to the backup mode described in Section III-C.

A. Zones

Moveover models the intersection using two distinct areas, called the *negotiation zone* and the *conflict zone*. This abstraction allows the algorithm to be applied to intersections with different geometries.

- Negotiation zone: the area where the EGO and the controller negotiate the mobility profile. One negotiation zone is defined for each incoming lane, and the vehicle must complete the negotiation before leaving this area; otherwise, the backup mode is triggered.
- Conflict zone: a road portion inside the intersection that can be occupied by only one vehicle at a time. The controller prevents simultaneous access to the same conflict zone by enforcing strictly non-overlapping temporal reservations.

Upon entering the negotiation zone, a CAV initiates the negotiation procedure, but it effectively becomes the EGO only when previously entered CAVs have completed the negotiation. Vehicles are therefore scheduled according to a first-in-first-scheduled (FIFS) policy, based on their arrival order.

B. Communication Protocol

The communication protocol consists of a negotiation loop between the EGO and the controller to determine a collision-free mobility profile. The negotiation procedure is illustrated in Fig. 2, which shows an example where the controller requests the EGO to delay its entry into the conflict zones.

1) Proposal by the EGO: Upon entering the negotiation zone, the EGO calculates an initial mobility profile based strictly on its own physical capabilities and constraints (e.g., maximum road speed, safe turning speeds, and comfort-based acceleration/deceleration limits). This profile is then submitted to the controller.

2) Validation by the controller: The controller cross-references the EGO's proposal against already scheduled vehicles across three stages:

- Check for conflicts before the intersection, ensuring a minimum safe following distance behind the vehicle ahead of the EGO in the same lane;
- Check for conflicts within the intersection, verifying that the EGO occupies the conflict zones in time intervals that do not overlap with existing reservations;
- Check for conflicts after the intersection, ensuring that the EGO's exit speed and distance are safely compatible with the vehicle exiting ahead of it.

3) Response from the controller: The controller replies with permissible entry and exit time bounds for each of the conflict zones to be traversed. Acceptance or rejection is implicit in the returned message: if the EGO's proposed mobility profile complies with the specified time windows, the EGO infers that its proposal has been accepted and the controller records the reservation. Otherwise, the EGO interprets the response as a rejection and must recompute its mobility profile using the newly provided bounds as constraints. The new bounds proposed by the controller are always later than those originally proposed by the EGO.

[1]For conciseness, we use the term mobility profile to denote a vehicle's trajectory together with its speed profile.



Fig. 3: Implemented environment for the evaluation of cooperative intersection management.

4) Additional exchanges: Upon rejection, the EGO uses the controller's availability windows to compute and submit a new, delayed mobility profile. This back-and-forth negotiation may iterate multiple times until an agreement is reached. In practice, the number of exchanges is typically small, since each response from the controller can only delay the entry time into the first conflict zone.

C. Backup Mode

The constraint that the controller can only introduce delays guarantees that the negotiation process converges to a deterministic conclusion. However, communication delays may prevent the exchange from completing before the EGO exits the negotiation zone. If the EGO cannot finalize its mobility profile before leaving the negotiation zone, or if the required crossing speed falls below a minimum threshold, the system switches to a backup mode. In this mode cooperative scheduling is suspended and vehicles revert to autonomous driving using conventional right-of-way priorities.

IV. Testbed Scenario and Experimental Setup

To validate the proposed system, we designed a physical testbed that reproduces a cooperative intersection scenario in a manageable environment. The setup consists of multiple autonomous mini-cars moving along a closed track and repeatedly approaching a shared intersection. The objective is to evaluate the practical viability of Moveover by demonstrating how vehicles can coordinate with the controller to cross the intersection without stopping. A summary of the key experimental parameters, including track dimensions and vehicle specifications, is provided in Table I.

A. Track Layout

The demonstration is conducted on a figure-eight track, shown in Fig. 3. The track has a length of approximately 11 m and a width of 8 m, which can be reduced to a minimum of approximately 8.20 m by 4.20 m with lower speed of the mini-cars.

Flexible orange tubes are used to delimit the track and create two loops that intersect at a central crossing point.

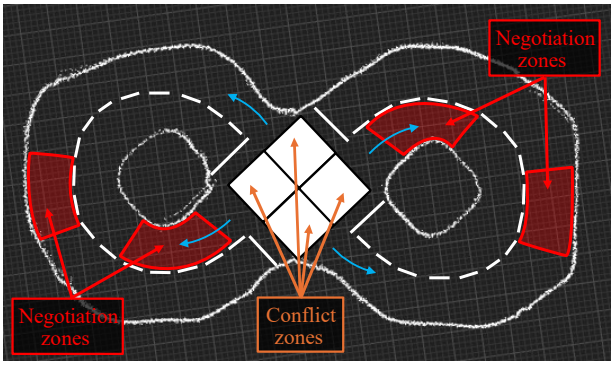
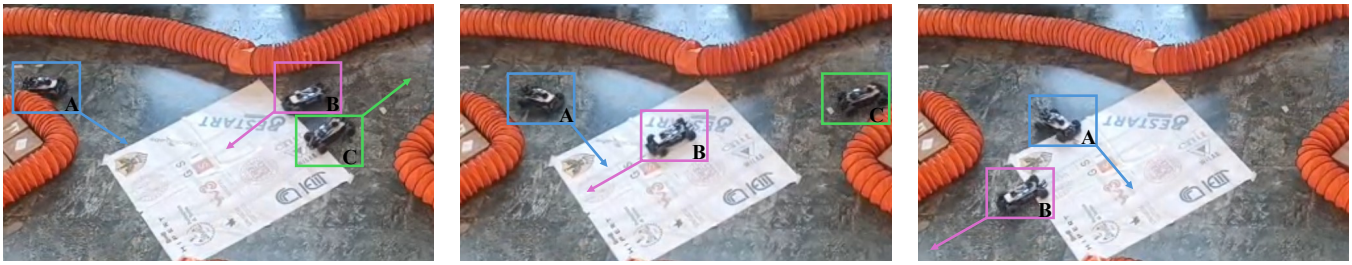


Fig. 4: Screenshot of the map with highlighted negotiation and conflict zones.

TABLE I: Characteristics of the experimental setup.

Parameter	Value
Track	
Length	11 m
Width	8 m
Minimum length	8.20 m
Minimum width	4.20 m
Conflict zone side	0.9 m
Negotiation zone length	2 m
Mini-cars	
Length	0.5 m
Width	0.3 m
Maximum speed	1 m/s

The intersection area is subdivided into four square regions corresponding to the conflict zones, each with a side length of 0.9 m. Square prints are used to visualize the conflict zones, although they are not used by the mini-cars or the controller. Similarly, tape markings on the floor indicate the lanes and the negotiation zones, which are defined for each lane approaching the intersection. Each negotiation zone has a length of 2 m.



(a) A and B approach the intersection (b) B is scheduled to traverse first (c) A crosses the intersection as B leaves

Fig. 5: Sequence of images from the cooperative intersection demonstration. Multiple mini-cars approach the intersection and traverse it without stopping under the coordination of the controller. The full video is available at [18].

B. Demonstration Workflow

During the demonstration, the mini-cars continuously circulate along the figure-eight track and repeatedly encounter the intersection. Each vehicle follows a predefined trajectory while the onboard autonomous driving system ensures accurate path tracking. In our experiments, four mini-cars were used, each measuring 0.5 m in length and 0.3 m in width, operating at a maximum speed of 1 m/s. To generate diverse yet repeatable interaction scenarios, two vehicles always proceed straight through the intersection, one always turns right, and one always turns left.

Accurate path tracking requires continuous self-localization. For this purpose, each mini-car is equipped with a map of the track generated prior to the experiments by manually driving a vehicle along the track while recording LiDAR data. The resulting point cloud is refined offline and uploaded to all vehicles. An example of the resulting map for the scenario in Fig. 3 is shown in Fig. 4. The same map is also used to design the reference trajectories followed by the mini-cars, using proprietary computer-aided design (CAD) software.

Once the map and reference trajectories are loaded, the mini-cars can navigate the track autonomously. When entering a negotiation zone, a mini-car transmits an initial proposal to the controller, initiating the negotiation procedure described in Section III-B.

In the current implementation, if a vehicle triggers the backup mode, centralized scheduling is suspended and all mini-cars stop. During our experiments, this situation occurred only rarely during long test runs, typically due to battery depletion in one of the vehicles. Alternative implementations of the backup mode are left for future work.

Using these specific settings, the system sustained autonomous, collision-free operation for more than one hour, halting only due to onboard battery exhaustion. The four 1:10 CAVs operated simultaneously with full localization, perception, and control loops, coordinated by the central infrastructure. The system maintained stable connectivity, consistent timing in command execution, and precise synchronization between the vehicles and the intersection controller. These experiments confirm that the integrated platform achieves a high level of reliability and is suitable for extended experimental studies.

Fig. 5 shows a sequence of instances from the demonstration, where two mini-cars approach the intersection and traverse it without stopping under the coordination of the controller. A video of the demonstration is available at https://www.youtube.com/watch?v=Mt51gEt45E0.

V. Conclusion

This work demonstrated, through a scaled experimental testbed, the feasibility of cooperative maneuvers among connected and autonomous vehicles at non-stop non signalized intersections. The demonstration integrates perception, control, and communication components within a unified experimental platform. The results confirm the practical applicability of the intersection coordination algorithm used in the system and highlight the suitability of the proposed platform as a research and validation environment for cooperative intersection management. More broadly, the testbed provides a flexible framework for experimental studies in the field of cooperative, connected, and automated mobility.

Acknowledgment

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - RESTART), project MoVeOver, and national research centre on mobility (CN00000023 - MOST).

References

[1] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti, “Revisiting street intersections using slot-based systems,” *PloS one*, vol. 11, no. 3, p. e0149607, 2016.

[2] Z. Yao, Y. Zhao, H. Jiang, and Y. Jiang, “A review of cooperative intersection: From design to management,” *IEEE Transactions on Intelligent Vehicles*, vol. 11, no. 1, pp. 197–219, 2026.

[3] Z. Wang, K. Han, and P. Tiwari, “Digital twin-assisted cooperative driving at non-signalized intersections,” *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 198–209, 2021.

[4] Y. Zhang and C. G. Cassandras, “Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees,” *Automatica*, vol. 109, p. 108563, 2019.

[5] R. Hult, M. Zanon, S. Gros, and P. Falcone, “Energy-optimal coordination of autonomous vehicles at intersections,” in *2018 European control conference (ECC)*. IEEE, 2018, pp. 602–607.

[6] M. D. Vaio, P. Falcone, R. Hult, A. Petrillo, A. Salvi, and S. Santini, "Design and experimental validation of a distributed interaction protocol for connected autonomous vehicles at a road intersection," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9451–9465, 2019.

[7] A. Katriniok, B. Rosarius, and P. Mhnen, "Fully distributed model predictive control of connected automated vehicles in intersections: Theory and vehicle experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 288–18 300, 2022.

[8] A. Mokhtarian, J. Xu, P. Scheffe, M. Kloock, S. Schfer, H. Bang, V.-A. Le, S. Ulhas, J. Betz, S. Wilson, S. Berman, L. Paull, A. Prorok, and B. Alrifaee, "A survey on small-scale testbeds for connected and automated vehicles and robot swarms: A guide for creating a new testbed [survey]," *IEEE Robotics & Automation Magazine*, 2025.

[9] L. Farina, L. M. Amorosa, M. Rapelli, B. M. Masini, C. Casetti, and A. Bazzi, "V2N-based algorithm and communication protocol for autonomous non-stop intersections," 2026. [Online]. Available: https://arxiv.org/abs/2603.05165

[10] M. O'Kelly, V. Sukhil, H. Abbas, J. Harkins, C. Kao, Y. V. Pant, R. Mangharam, D. Agarwal, M. Behl, P. Burgio, and M. Bertogna, "F1/10: An open-source autonomous cyber-physical platform," 2019.

[11] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, 2022.

[12] R. S. Wallace, A. Stentz, C. E. Thorpe, H. P. Moravec, W. Whittaker, and T. Kanade, "First results in robot road-following," in *IJCAI 1985*.

[13] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, *Particle Filters for Mobile Robot Localization*. Springer New York, 2001, pp. 401–428.

[14] L. Farina, M. Piccoli, S. Iandolo, A. Solida, C. A. Grazia, F. Raviglione, C. Casetti, and A. Bazzi, "Low cost c-its stations using raspberry pi and the open source software oscar," in *IEEE VTC2025-Spring*.

[15] S. Iandolo, C. A. Grazia, M. Klapez, and M. Casoni, "Odu: Open dsrc unit toward effective dissemination of v2x applications," in *IEEE VTC2025-Spring*.

[16] M. Rapelli, F. Raviglione, and C. Casetti, "OScar: An ETSI -Compliant C- ITS Stack for Field- Testing with Embedded Hardware Devices," in *MedComNet 2024*.

[17] L. Farina, F. Gavioli, S. Iandolo, F. Moretti, G. Perrone, M. Piccoli, F. Raviglione, M. Rapelli, A. Solida, P. Burgio, C. A. Grazia, and A. Bazzi, "Open-source based and etsi compliant cooperative, connected, and automated mini-cars," 2026. [Online]. Available: https://arxiv.org/abs/2603.06343

[18] "Demonstration of cooperative autonomous vehicles crossing an intersection," 2026, accessed: March 12, 2026. [Online]. Available: https://www.youtube.com/watch?v=Mt51gEt45E0