Emerging Cutting-Edge Applied Research and Development in Intelligent Traffic and Transportation Systems, M. Shafik (Ed.)

doi:10.3233/ATDE241196

Multi-Agent Simulation Scheme for Cooperative Connected Automated Mobility Based on ROS and Carla

David Yagüe-Cuevas [a,1], María Paz-Sesmero [a] Pablo Marín-Plaza [a] Stephen Smith [b] Araceli Sanchis [b] and José María-Armingol [b]

[a] *University Carlos III of Madrid, Spain*
[b] *Carnegie Mellon University, Pittsburgh, USA*

Abstract. The current scarcity of viable simulation solutions for cooperative and connected driving research is one of the main reasons to start dedicating efforts to generate robust validation environments. Most current simulators focus on the egocentric perspective of the automated vehicle, with cooperative information being either non-existent or difficult to access. If soon the industry migrates towards connectivity and cooperation paradigms, it will then be necessary to build simulation environments to validate the cooperative behaviors that arise from the interactions between traffic agents. This work focuses on the possibilities offered by Carla simulator integrated with ROS framework for multi-agent simulation and arbitration capabilities between intelligent infrastructure and automated connected vehicles. By combining a wide variety of methodologies, the solution proposed in this work allows generating an effective validation framework for coordinated and autonomous cooperative driving.

Keywords. Cooperative Connected Automated Mobility (CCAM), Intelligent Transportation System (ITS), Multi-Agent Simulation, ROS, Carla

1. Introduction

Intelligent Transport Systems [1], [2] encompass a set of technological solutions to improve the operation and safety of land transport, both on urban and rural roads and railways. Since its inception in the mid-1960s, the industry has been providing valid solutions to the challenges of autonomous mobility itself as advances in the fields of telecommunications and information technology have been made [3]. Advanced Driver Assistance Systems (ADAS) [4], [5] are a practical example of such solutions, which have increased in capabilities and complexity as new techniques and technologies have emerged in industry, research and development. The main goal of this set of solutions is clear, since they not only aim to reduce traffic congestion and improve road safety, but also to optimize the use of existing traffic infrastructure and, in turn, to reduce environmental impact [6], [7] and provide a more efficient and pleasant experience [8]. Major studies investigating the potential socio-economic impact of automated and connected mobility

[1]Corresponding Author: David Yagüe-Cuevas; E-mail: dayaguec@inf.uc3m.es.

present favorable predictions for the future of modern cities [9],[10], where smart infrastructure plays a role as important as automated vehicles capable of connecting and sharing information in real time.

On the one hand, traffic management, focused on traffic control centers and accident detection systems, will be a fundamental pillar for the efficient use of resources and actions in case of emergency. On the other hand, road user service systems will be positively affected with ubiquitous information on the state of the roads, leading to the optimization of resources deployed for the transport of people and goods, which has now become a fundamental pillar around which Intelligent Transport Systems development and research orbits. However, such advances cannot be made effectively without advances in simulation systems [11], since the rapid generation of novel solutions and technologies in this field is increasingly demanding more robust environments for validation and testing. Hence, the development of methodologies and paradigms that allow effective experimentation for the development and testing of state-of-the-art solutions is directly affected by the number of resources available for simulation. The more robust and useful these environments are the more secure, safe and reliable developed solutions will become.

Simulators are generally valuable tools in a wide range of fields due to their multiple benefits. They allow the generation and validation of developed skills in a risk-free environment, avoiding potential damage or hazards, which indirectly reduces potential validation costs. They facilitate access to environments beyond the reach of developers, provide immediate feedback and ground truth on the performance of the algorithms used, replicate complex and/or dangerous situations that are difficult to manage, etc. In short, the industry has been using all kinds of simulators for several decades now providing a robust methodology to test the solutions adopted for one or more specific areas. In particular, the automotive industry has been increasingly demanding realistic simulators capable of generating kinematics, dynamics and in general classical mechanics for close-to-reality situations to validate the implementation of autonomous systems in newly manufactured vehicles. The evaluation of the performance of a new subsystem and the early detection of faults and errors ensure the effectiveness of the developed tools, especially in terms of exploring scenarios and hypothetical situations that can be studied in detail to provide better solutions. In addition, given that the number of possible road situations that a vehicle may encounter are many, simulators present themselves as a valuable tool for the development, training and validation of autonomous vehicle systems.

However, the current scarcity of viable simulation solutions for cooperative and connected driving research is one of the main reasons to start dedicating efforts in this field toward the development of robust validation environments. Most current simulators focus on the egocentric perspective of the automated vehicle, with cooperative information being either non-existent or difficult to access. If soon the industry migrates towards connectivity and cooperation between agents [12], an effect of an inexorable push toward the manufacture of vehicles that are fully computerized and connected, it will then be necessary to build simulation environments to validate the cooperative behaviors that arise from the interactions between traffic agents. However, it is not necessary to start from scratch. Current state-of-the-art software and simulation environments combine valid tools and methodologies than can be built on to develop cooperative behaviors and connected systems. While the telecommunications industry advances in the use of new technologies such as 5G [13] [14] and V2X communication protocols [15], [16], solutions for vehicle and infrastructure simulation such as the one proposed in this work will allow parallel advances in the methodologies for action, arbitration, decision-making, and behavior generation derived from the processing of information from connected traffic agents under the same environment.

Specifically, this work focuses on the possibilities offered by Carla simulator [17], integrated with ROS robotic development framework, for validation and development of algorithms for navigation, localization, automatic control, decision-making and arbitration. Unlike other systems such as Autoware [18] or LGSVL Simulator [19], the solution proposed here focuses on multi-agent simulation and arbitration capabilities between infrastructure and vehicle, i.e. cooperative skills between different traffic agents. Therefore, by combining a wide variety of methodologies and tools, the solution proposed in this work allows generating an effective validation framework for coordinated and autonomous cooperative driving, not only offering the point of view of an automatic vehicle, but also of several vehicles, as well as the possibility of simulating intelligent infrastructure with multiple sensors able to communicate with any connected agent.

2. Software Ecosystem

The ecosystem this work proposes is based on two well-known, yet important frameworks for development and simulation in robotics and automated vehicles.This section summarizes the capabilities of each and establishes the background necessary to understand how the proposed multi- agent simulation is constructed.

2.1. Carla (Car Learning to Act)

Carla [17] is an open-source simulator born in 2017 as a software layer implemented based on the Unreal Engine. It is specifically designed for the validation of autonomous driving algorithms. Carla architecture is based on a Client-Server paradigm. On the one hand, a server runs as a layer of the Unreal Engine and generates the world and all its actors with complete physics and kinematics computations. On the other hand, a Python-based client communicates with the server to extract information from the world allowing interaction with its actors. In addition, Carla allows the simulation of automatically controlled agents, but also offers a wide range assets for the simulation of traffic infrastructure and the generation of customized worlds generated with synthetic or real data. This makes it possible to easily generate scenarios based on real information.

2.2. ROS (Robot Operating System)

ROS [20] was originally developed in 2007 by the Stanford Artificial Intelligence Lab. The project stems from the growing need for a robotic development standard that solves the problem of specificity of robotic software and hardware. ROS is considered a middleware, i.e., a set of tools designed to simplify the creation of robotic applications, offering solutions for hardware abstraction, inter-process communication, package management tools, open-source reuse or agile development. The management of processes in real-time systems, the use of a distributed architecture of independent and collaborative processes, and the communication layer based on the Data Distribution Service, have made it a very attractive proposal for the development of automatic vehicle systems, but also for the implementation of communication systems between cooperative agents.

3. Multi-client Paradigm

The environment described in this work is based on a duality of tools that are integrated within the same ecosystem and allow generation of a framework of interaction between both parts. The entry point for any development in ROS is the ROS node, a program that by itself addresses a specific task and can share information with another ROS nodes through topics and services. On the contrary, the entry point for any software that wants to access the Carla simulation is the Carla Client based on its Python API through which all data from the simulation can be queried and retrieved using the game engine. Since the Carla API is based on modern versions of Python, integration with ROS must be accomplished in Python. Luckily, ROS supports languages such as C++ and Python. In addition, if the only way of communicating simulation information is through a Carla client, the approximation proposed here is as simple as generating a modular architecture in which every component is at the same time a ROS node and a Carla client. Since Carla now supports spawning more than one client and ROS is based on a computing graph of one or more nodes, the combination allows generation of any number of ROS nodes that implement a Carla client, with each node acting independently to produce different information inside specific scopes or ROS namespaces. Hence, each connected agent will function under different namespaces simulating an independent connected automated vehicle that is able to share information through topics and services with other clients in the same network. This network in turn is simulated by the tools provided by ROS communication protocols by standardizing it in messages and sharing them through topics and services. In Figure 1 a general schema of the proposal is shown.

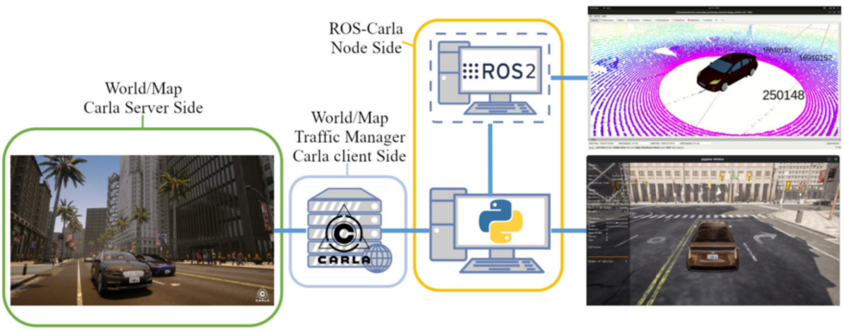


Figure 1. ROS-Carla Interaction Scheme.

In addition, other clients of different nature implement, instead of an ego vehicle control ecosystem, a different component of static intelligent infrastructure able to share information when deployed in important landmarks such as intersections, roundabouts or entry lanes. While in the ego vehicle client side, sensors are attached to the vehicle actor spawned through the Carla client in the World. On the infrastructure side, each sensor is attached to a specific world location and then managed by the ROS-Carla node converting the static sensor readings into ROS standard messages. Given that services and topics can be implemented as a channel of communication between nodes, this information is shared with nearby agents that request cooperation when new transmission channels between actors are created dynamically for this purpose. Additionally, both types of clients i.e., the automated vehicle and the intelligent infrastructure implement several objects of critical importance for the ecosystem. These are (1) the World class, a software abstraction representing the simulation world the client is going to interact with; (2) the Robot Description, a component that generates a virtual representation of the vehicle; (3) the Carla-Rviz Panel, a GUI component to interact with the simulation, and (4) the ROS-Carla node that is responsible for the spawned actor status and the World, and also retrieves information using the Carla API presenting it through topics and services. This last component also spins at the same simulation tick rate provided by the World class.

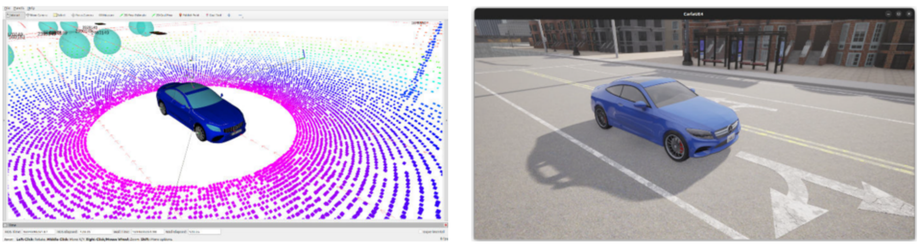
3.1. The World Class

First and foremost, the goal of the world class is to provide a channel of communication between the ROS-Carla node and the Carla server, which can also interact with the agent controlled in the current instance. Through this set up, sensor information, as well as dynamically controlled variables like map layers, simulated weather or traffic lights states can be accessed through the World class and, once prompted, sent to the ROS-Client side that enables information sharing through services and topics. This information can be fed to other ROS nodes for collaboration and cooperation behaviors.

From the World, the ROS-Carla client has the ability to gather the simulator ground truth and convert it to ROS standard messages. This can be very useful for the evaluation of deep learning perception models used for classification and detection, or even to assess the performance of tracking algorithms. At each simulation tick, the client retrieves vehicles, pedestrians, traffic lights, map landmarks and signs near the ego vehicle location through the World functions. This information is then stored into ROS messages that are published for use between other clients inside the same ecosystem.

3.2. The Robot Description

ROS uses the Unified Robot Description Format [21] (URDF) to define the robot's physical characteristics, including its kinematic and dynamic properties. Through the usage of links connected to joints, the system can define not only a visual representation of the robot in the virtual world but also the point of view of each component, and the transformation tree for each measurement [22]. From a single reference frame, sensor readings become directly usable by different algorithms ensuring consistent interpretation of distances and positions within the chosen coordinate system, which can be seamlessly combined for accurate localization, navigation, control, etc. In Figure 2 a vehicle description in Rviz and Carla, the ROS visualization tool, can be seen to understand these concepts.



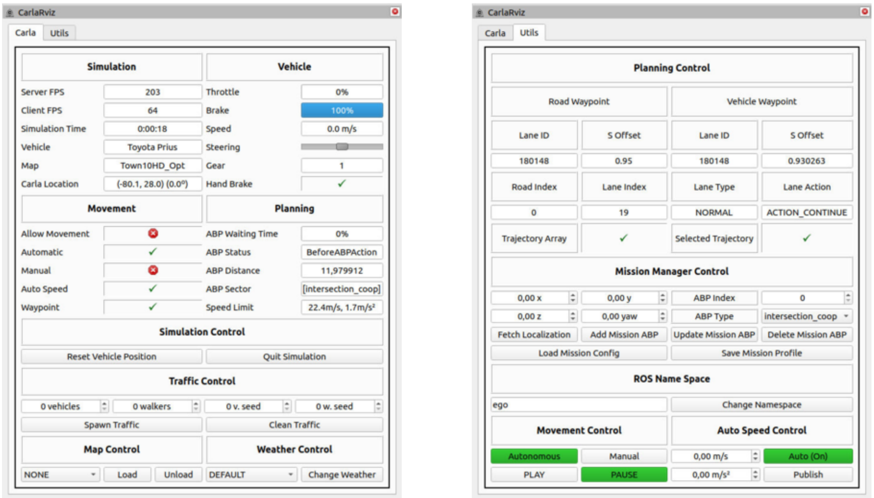
(a) TF Tree in ROS Rviz. (b) Original Carla Model.

Figure 2. TF Tree and Carla Vehicle Model.

The Ego Vehicle description is based on a URDF implemented with Xacro of an Ackermann geometry vehicle that retrieves from simulation the physical characteristics of the vehicle requested to be simulated, and generates a robot description based on this information. Then, through the sensor YAML configuration file processed by the ROS-Carla Client (see section 3.4), sensors are attached to the actor vehicle and assigned to a link in the transformation frame if it exists to produce readings from that specific frame. For the infrastructure side, since it is a logical part of the world, the transformation tree is constructed with several sensors attached to the world global frame that is the origin of the world in Carla.

3.3. The Carla-Rviz Panel

This component is a GUI panel based on Qt, a tool to develop user interfaces, developed to show important information about the simulation and ROS components during execution. It allows interaction with the World, Traffic Manager and also with the ROS components on the client Side through several services specific for the client. In Figure 3 the panel design is shown. As can be seen there are multiple fields, buttons and interactive items that allow adjustment other component configurations within the same ecosystem.



(a) Panel General Tab. (b) Panel Utilities Tab.

Figure 3. ROS-Carla Rviz Panel.

The panel is attached to Rviz for management purposes. It also provides buttons to interact with Carla Server through ROS-Carla node allowing to dynamically adjust several parameters to customize the current simulation.

3.4. The ROS-Carla Node

The ROS-Carla node generates a software bridge with the Traffic Manager and World classes to effectively simulate the agent and its connections to the server. The ROS-Carla node is the most important module in this ecosystem, since it is the module through which the connected automated agents are spawned with required sensors attached through several queries to the World class. All information used for the parameterization of the simulation comes from YAML files, processed by this ROS-Carla node and redistributed to the other software classes. Sensor information is provided in configurations where one or more sensors for each available type can be instantiated to be spawned in simulation, given the transform and frame to which the Client will attach depending on the case, e.g., the world for infrastructure and the actor for ego vehicle simulations. Sensor type can be any available in Carla: GNNS receiver, IMU, RGB Camera, Semantic Camera, Radar, LiDAR and Semantic LiDAR. In Figure 4 the general idea of these interactions can be seen, where each ROS-Carla client represents an agent able to extract and share information about its current state in the simulation.

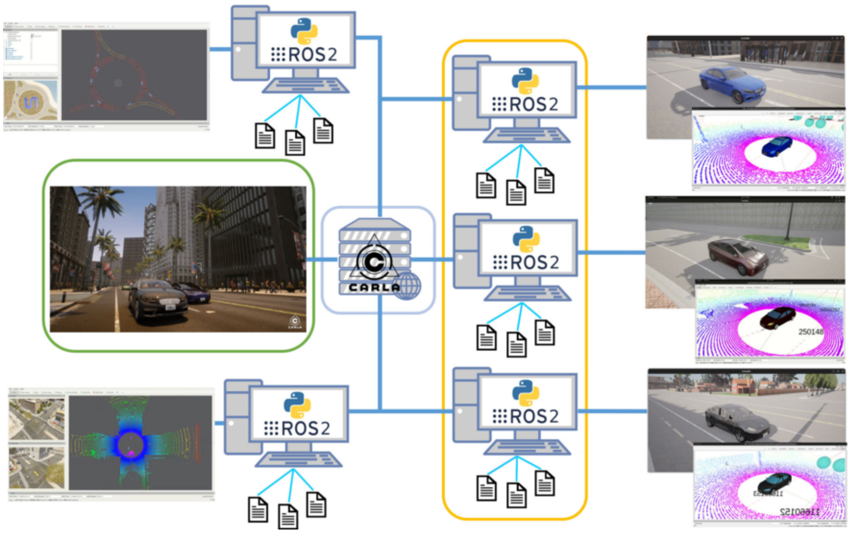


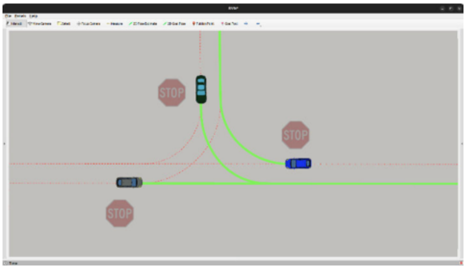
Figure 4. Multi-Client Architecture.

Additionally, though ROS parameters this information can be completed with things such as the spawning location for the vehicle or the main frame of the infrastructure system, the type of ego vehicle to simulate, selected from the list available in Carla assets, the specific world to generate, from custom to already implemented worlds in Carla, the map layers to load, the current weather of the simulation, etc. Each time a new agent needs to be spawned, the ROS-Carla node is launched, it reads the parameters in the YAML file and the ROS parameters given, and generates the actor required. This actor can be a vehicle to be controlled or an infrastructure item such as an intersection with multiple cameras and LiDAR sensors. The launch file mentioned uses the launch system provided by ROS and contains a namespace to distinguish between different ROS-Carla clients. Hence, with this set up, multiple clients can be spawned in the world generating multiple agents, governed independently by the control software of each, and then an intelligent infrastructure agent able to arbitrate using the information of ROS services and topics shared by the agents. This allows simulation for the arbitration of intersections or the collaborative generation of trajectories.

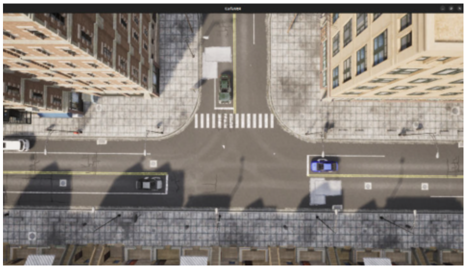
4. Experimentation

Several use cases were designed to test the effective simulation of multiple agents within the same Carla world. This experiments were ran in a high-end machine based on Ubuntu 22.04, Carla 0.9.15 in asynchronous mode, and ROS2 Humble. In asynchronous mode Carla server is in full control of the simulation and provides more realistic calculations. The machine specs are Intel Xeon w5-2455X CPU, 256GB DDR5 RAM Memory, and GeForce RTX 4090 GPU. Although, multiple machines can be used to act as Carla-ROS clients, for these experiments all were run in the same computer.

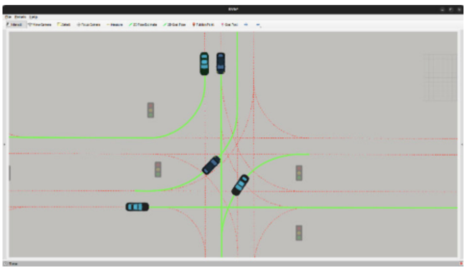
In Figure 5 a multi-agent simulation with several cooperative automated vehicles can be seen. In Figures 5a and 5b three vehicles meet at a T junction with several stop signs marked in the High Definition Map, while in Figure 5c and 5d, the vehicles can be seen driving in a typical four-way intersection. Respective cooperative systems running in each vehicle allow interaction with an arbitration module deployed in the infrastructure side whose aim is to allocate resources on demand. The vehicles can also interact with each other since each controller is running under the same ecosystem. Red arrows are HD Map lanes while green paths are road plan followed by each vehicle planning system.



(a) 3 Vehicles Rviz



(b) 3 Vehicles Carla Server

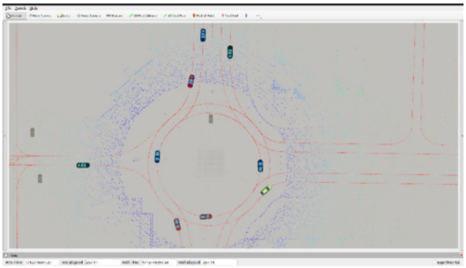


(c) 5 Vehicles Rviz

(d) 5 Vehicles Carla Server

Figure 5. Simulation of Multiple Agents.

In Figure 6 a similar simulation environment is presented but, instead an intersection, a roundabout managed by an arbitration system with a LiDAR deployed on its center can be seen. Multiple vehicle systems were spawn together with the infrastructure agent. These scenarios are built as a test-bed for cooperation design and collaboration between vehicles and infrastructure arbitration systems such as in previous works [23], [24], and [25]. All simulation runs were able to maintain a rate $\approx$ 60 FPS in the machine used, but in other machines, simulation time could be constrained by the requirements needed.



(a) Vehicles and Infrastructure Rviz

(b) Vehicles and Infrastructure Carla Server

Figure 6. Multiple Vehicle Agents in Roundabout System.

5. Conclusions

Multi-agent simulation is an important feature of rigorous research and development in the field of connected and automated vehicles. This paper proposes a comprehensive scheme based on well-known state-of-the-art open-source frameworks available to any researcher and developer to continuously work on new methods for arbitration and cooperation. The benefits of this environment are multiple, such as real scale scenarios, realistic physics empowered by the underlying Unreal Engine, and robust robotics development based on ROS solutions. The scheme proposed not only serves as a testbed for Cooperative Connected Automated Mobility but also for Intelligent arbitration system development and validation of Detection/Tracking deep learning models. In this work, several vehicles' models were simulated successfully in a common traffic scenario in which an intelligent intersection was deployed, given the possibility to specify and test cooperative behavior between them. It was proven how a combination of these tools are sufficient to provide realistic simulation of multiple agents that can work concurrently within the same environment.

However, the problem of resources required to run multi-agent realistic simulations is a matter to consider. Most automated driving simulators are still memory expensive tools whose requirements tend to increase in proportion to the number of assets simulated. With the proposed solution, when simulating more than 10 agents, a drop in performance can be easily registered leading to unexpected results of the simulation. In that regard, more efforts must be made for the development of efficient mechanisms in current automated driving simulation to improve resources usage, especially when it comes to generating dense sensor information like LiDARs.

Acknowledgements

Grant PID2021-124335OB-C21, PID2022-140554OB-C32 and PDC2022-133684-C31 funded by MCIN/AEI/10.13039/5011000 11033.

References

[1] J. C. Miles, "Intelligent transport systems: overview and structure (history, applications, and architectures)," *Encyclopedia of Automotive Engineering*, pp. 1–16, 2014.
[2] S. Telang, A. Chel, A. Nemade, and G. Kaushik, "Intelligent transport system for a smart city," *Security and privacy applications for smart city development*, pp. 171–187, 2021.

[3] S. Sharma and S. K. Awasthi, "Introduction to intelligent transportation system: overview, classification based on physical architecture, and challenges," *International Journal of Sensor Networks*, vol. 38, no. 4, pp. 215–240, 2022.

[4] M. M. Antony and R. Whenish, *Advanced Driver Assistance Systems (ADAS)*. Cham: Springer International Publishing, 2021, pp. 165–181.

[5] S. Ayyasamy, "A comprehensive review on advanced driver assistance system," *Journal of Soft Computing Paradigm*, vol. 4, no. 2, pp. 69–81, 2022.

[6] P. Szymanski *et al.*, "The future of road transport in europe. environmental implications of automated, connected and low-carbon mobility," *Combustion Engines*, vol. 60, 2021.

[7] Z. Yao *et al.*, "Fuel consumption and transportation emissions evaluation of mixed traffic flow with connected automated vehicles and human-driven vehicles on expressway," *Energy*, vol. 230, 2021.

[8] A. Nikitas, N. Thomopoulos, and D. Milakis, "The environmental and resource dimensions of automated transport: a nexus for enabling vehicle automation to support sustainable urban mobility," *Annual Review of Environment and Resources*, vol. 46, pp. 167–192, 2021.

[9] H. U. Ahmed, Y. Huang, P. Lu, and R. Bridgelall, "Technology developments and impacts of connected and autonomous vehicles: An overview," *Smart Cities*, vol. 5, no. 1, pp. 382–404, 2022.

[10] Z. Lv and W. Shang, "Impacts of intelligent transportation systems on energy conservation and emission reduction of transport systems: A comprehensive review," *Green Technologies and Sustainability*, vol. 1, no. 1, p. 100002, 2023.

[11] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," in *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, 2021, pp. 62–70.

[12] V. S. R. Kosuru and A. K. Venkitaraman, "Advancements and challenges in achieving fully autonomous self-driving vehicles," *World Journal of Advanced Research and Reviews*, vol. 18, pp. 161–167, 2023.

[13] A. Gohar and G. Nencioni, "The role of 5g technologies in a smart city: The case for intelligent transportation system," *Sustainability*, vol. 13, no. 9, p. 5188, May 2021.

[14] M. J. Shehab, I. Kassem, A. A. Kutty, M. Kucukvar, N. Onat, and T. Khattab, "5g networks towards smart and sustainable cities: A review of recent developments, applications and future perspectives," *IEEE Access*, vol. 10, pp. 2987–3006, 2021.

[15] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 3, pp. 2027–2057, 2021.

[16] S. Gyawali, S. Xu, Y. Qian, and R. Q. Hu, "Challenges and solutions for cellular based v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 222–255, 2020.

[17] 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*, 2017, pp. 1–16.

[18] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[19] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[20] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[21] D. Tola and P. Corke, "Understanding urdf: A survey based on user experience," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–7.

[22] T. Foote, "tf: The transform library," in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2013, pp. 1–6.

[23] D. Yagüe-Cuevas, P. Marín-Plaza, M. P. Sesmero, and A. Sanchis, "Towards a robust traffic scene representation in cooperative connected automated mobility." in *Proceedings of the 9th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2023, pp. 265–272.

[24] D. Yagüe-Cuevas, P. Marín-Plaza, M. Paz-Sesmero, and A. Sanchis, "Modeling the traffic scene in intelligent transport systems for cooperative connected automated mobility," in *Communications in Computer and Information Science*. Springer International Publishing, 2024, p. to appear.

[25] D. Yagüe-Cuevas, M. Paz-Sesmero, P. Marín-Plaza, and A. Sanchis, "Behavioral based decision-making scheme for cooperative connected automated mobility," in *Proceedings of the 10th International Conference on Connected Smart Cities*. MCCSIS, 2024, p. to appear.