Key requirements for autonomous micromobility vehicle simulators

Lennart Luttkus, Lars Mikelsons

Angaben zur Veröffentlichung / Publication details:

Luttkus, Lennart, and Lars Mikelsons. 2023. “Key requirements for autonomous micromobility vehicle simulators.” In 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 1-4 October 2023, Honolulu, Oahu, HI, USA, edited by Adrian Stoica, Ferat Sahin, Robert Kozma, Kyungtae (KT) Han, and Celal Savur, 1506–11. Piscataway, NJ: IEEE.
https://doi.org/10.1109/smc53992.2023.10394587.

Key Requirements for Autonomous Micromobility Vehicle Simulators

Lennart Luttkus[1] and Lars Mikelsons[2]

Abstract— **With the growing demand for autonomous micromobility vehicles, developing robust and effective simulators for them becomes increasingly important. This research paper examines the essential requirements of a simulator for autonomous micromobility vehicles, focusing on aspects such as accurate sensor modeling, realistic pedestrian behavior, customizability, scenario and vehicle library, scalability, and user-friendliness. By analyzing these key features, we provide a comprehensive understanding of the necessary components for an effective simulation environment, aiming to enable researchers, developers, and other stakeholders to design, test, and evaluate autonomous micromobility vehicles in a safe and controlled manner. Addressing these requirements, simulators can significantly contribute to the advancement of autonomous micromobility technology, leading to safer and more efficient urban transportation systems in the future.**

I. INTRODUCTION

Urban environments face growing challenges such as congestion, pollution, and inefficient transportation. Micromobility emerges as a promising and sustainable solution to address these issues. In this research paper, we begin with discussing the motivation for micromobility and provide a clear definition of micromobility vehicles within the context of our study.

Subsequently, we investigate the potential advantages and challenges related to autonomous micromobility. In examining this emerging field, the significance of simulation comes to the forefront. Consequently, we underscore the pivotal role of simulation in the development of autonomous micromobility vehicles, emphasizing its utility for testing and evaluation within a controlled and adaptable setting. Ultimately, we present the need for a sophisticated simulator specifically tailored for autonomous micromobility vehicles.

A. Motivation for Micromobility

The growing urban population has placed increasing strain on traditional transportation systems, leading to issues such as congestion, longer travel times, and environmental concerns [1]. Micromobility has emerged as a viable solution to address these challenges by offering a range of benefits.

Key benefits include encouraging outdoor activity, improving health and well-being, and offering greater efficiency thanks to superior vehicle-to-person weight ratios, reducing energy consumption and increasing cost-effectiveness. Electric micromobility options provide environmental benefits by reducing local emissions, and their smaller size alleviates congestion and parking issues, making cities more navigable.

Micromobility enhances accessibility, bridging the gap between public transit and individual transportation needs, ensuring wider city access. It also contributes to walkable and bike-friendly urban environments, promoting sustainable city planning and enhancing urban livability. Overall, micromobility presents a compelling alternative to traditional transportation methods, fostering healthier, sustainable, and accessible cities for all residents.

B. Definition of Micromobility Vehicles

Micromobility vehicles have become popular, sustainable urban transportation solutions. These lightweight vehicles, typically operating below $25\,\mathrm{km/h}$ and weighing less than $50\,\mathrm{kg}$, are characterized by electric drivetrains, which lower emissions and enhance energy efficiency, contributing to cleaner urban environments.

Their compact size and maneuverability allow for easy navigation in congested areas, fitting on bike lanes, and enhancing user experience. Common micromobility options include electric scooters and shared or electric bicycles. Electric scooters, particularly three-wheeled variants, hold great potential for automation and integration into shared transportation systems, expanding accessibility and convenience.

C. Motivation for Autonomous Micromobility

The advent of autonomous micromobility vehicles has gained much attention due to its potential to revolutionize urban mobility systems. Recently, several studies have investigated the potential benefits and revealed the significant advantages of autonomous micromobility vehicles over traditional micromobility systems.

Sanchez et al. proposed to implement autonomous driving technology in bicycle-sharing systems to address challenges in micromobility systems and to create a new on-demand shared mobility platform [2]. The authors demonstrated that a fleet of shared autonomous bicycles would eliminate the need for rebalancing and docking stations, improving the user experience by resolving the difficulty of finding available bikes or docks and reducing walking distance. The study indicated that, with the same demand, the fleet size needed for autonomous shared bicycles is 4.31 times smaller than that of a station-based system and 10.56 times smaller than that of a dockless system. Additionally, the cumulative cost

---

[1] Lennart Luttkus is with the Chair of Mechatronics at the University of Augsburg, 86199 Augsburg, Germany `lennart.luttkus@uni-a.de`

[2] Lars Mikelsons is Head of the Chair of Mechatronics at the University of Augsburg, 86199 Augsburg, Germany `lars.mikelsons@uni-a.de`

of the first five years of operating an autonomous shared bicycle system could be 73% lower than a station-based system and 27% lower than a dockless system.

Kondor et al. [3] evaluated the benefits of autonomous micromobility vehicles using real-world data of shared bicycle usage and public bus use for short trips in Singapore. The study demonstrated that self-repositioning can help achieve up to 10 times higher utilization of vehicles than in current share systems. The authors also presented the ideal fleet size and vehicle utilization required to serve trips currently taken by shared bikes, characterizing the main benefits and challenges for autonomous micromobility vehicles in cities.

Furthermore, Coretti et al. [4] proposed a simulation tool to assess the performance of a fleet of autonomous shared micromobility systems and compare it with existing station-based and dockless schemes. The study revealed that shared autonomous micromobility systems can efficiently combine the benefits of vehicle sharing, electrification, autonomy, and micromobility, creating effective mobility systems. The authors argued that the results of this simulation tool could provide valuable insights to stakeholders such as fleet operators, engineers, city planners, and governments.

In addition, Segway offers the T60 as a first version of a "semi-automatic, teleoperating, shared scooter" which underlines the economic attractiveness of autonomous micromobility vehicles [5].

In conclusion, the motivation for autonomous micromobility vehicles stems from their potential to provide efficient first- and last-mile transportation, address challenges in micromobility systems, reduce costs for operators, and create a more robust micromobility network. The studies discussed above provide strong evidence for the benefits of autonomous micromobility vehicles.

D. Importance of Detailed Micromobility Simulations

The increasing interest in autonomous micromobility vehicles underscores the necessity for comprehensive simulation environments for their development, testing, and validation. Although macroscopic simulations efficiently represent fleet behavior, more detailed simulations are crucial for addressing the unique challenges associated with these vehicles.

Simulation offers a cost-efficient and safe alternative to real-world testing, allowing for the reproduction of identical scenarios and conditions. It enables rapid iteration and optimization of vehicle designs and configurations, while simulating diverse environmental and weather conditions. The scalable nature of simulation facilitates efficient testing of edge cases and potential issues, and aids in identifying areas for improvement in a less risky manner.

Moreover, simulation helps establish confidence in the safety and performance of autonomous micromobility vehicles, accelerating their adoption and deployment. It also allows for the accurate modeling of other road users and pedestrians in complex urban environments. The shareability and replicability of simulations further contribute to the widespread testing and validation of autonomous micromobility vehicle technologies.

II. METHODS

The paper's objective is to present a list of simulators that can be employed for simulating autonomous vehicles. Furthermore, it extracts requirements from existing simulators to outline an ideal simulator specifically designed for autonomous micromobility. Moreover, it offers guidance regarding the most appropriate simulator currently available for autonomous micromobility vehicle simulation.

A. Survey of Autonomous Cars Simulators

To establish the requirements for an autonomous micromobility vehicle simulator, we examined a variety of 2D and 3D simulators, evaluating their strengths and weaknesses. This assessment, combined with key criteria from Kaur et al.'s study on self-driving car simulators [6], informed our comprehensive list of requirements. The complete list of analyzed simulators is presented in Section III-A, while the derived requirements can be found in Section III-B.

B. Assessing the Suitability of Autonomous Vehicle Simulators for Micromobility Applications

We employ a comparative analysis methodology to assess the suitability of existing simulators for autonomous micromobility vehicles. First, we identify the most popular simulators through a comprehensive review of research literature and online resources. Next, we use the requirements for simulators of autonomous micromobility vehicles and collect information on each simulator's features, capabilities, and limitations from primary and secondary sources. Finally, we compare each simulator against the predefined criteria to assess each aspect. An overall ranking of the simulators is not present because the weights assigned to each of the requirements heavily depend on the research question at hand and the performance is subject to change.

III. RESULTS AND DISCUSSION

The results of our paper are a list of potential simulators in Section III-A, the derived requirements for an optimal simulator in Section III-B and the evaluation of the top three simulators in a comparative analysis in Section III-C.

A. List of Autonomous Vehicle Simulators

In the 2D domain, we examined the following simulators: highway_env [7], CARLO [8], SUMO [9], the MIT Race Simulator [10], and Arena-Rosnav [11] which is based on Flatland [12]. For 3D Simulators, we analyzed the Unreal Engine-based CARLA [13], AirSim [14] and Deepdrive [15]. Based on the Unity game engine [16], we examined the AWSIM [17], the LGSVL Simulator [18] and AutoDRIVE [19]. The Gazebo-based [20] simulators analyzed were AutoCarROS2 [21], OSRF Car Demo [22] and Arena-Rosnav-3D [23]. Additionally, we analyzed VISTA [24], DuckieTown Simulator [25], Webots [26] and TORCS [27]. Despite the extensive list of analyzed simulators, it remains possible that we overlooked some simulators, which might have influenced our study results. We did not consider closed-source simulators like Beam NG, MathWorks Automated

Driving Toolbox, CarMaker, Vector, Hexagon, MORAI Sim Drive, Nvidia Drive and Isaac Sim.

B. Requirements for an Autonomous Micromobility Simulator

The rapid advancement of autonomous micromobility solutions has highlighted the need for a comprehensive simulator to facilitate the development, testing, and validation of these cutting-edge technologies. A robust autonomous micromobility simulator must encompass a range of critical requirements, including realistic vehicle dynamics, accurate sensor modeling, and authentic pedestrian behavior representation. Additionally, it should offer a high degree of customizability, scalability, and determinism, along with a rich library of pre-built scenarios and vehicle models to streamline the development process. By addressing these requirements, an autonomous micromobility simulator can provide an essential tool for researchers, engineers, and developers, enabling them to optimize the safety, efficiency, and usability of emerging micromobility solutions in complex urban environments.

1) Vehicle Dynamics: In autonomous micromobility, accurately modeling vehicle dynamics is essential for ensuring that control algorithms developed in simulation transfer to real-world performance. As micromobility vehicles exhibit unique dynamics compared to larger vehicles, a simulator must capture these characteristics accurately. Simulating various micromobility vehicle concepts from a dynamics perspective aids in optimizing design as well as enhancing safety, efficiency, and usability.

An ideal simulator for realistic vehicle dynamics should include key features such as realistic tire and road interaction models, multi-body dynamics, accurate actuator and powertrain modeling, integrated control system modeling, and vehicle parametrization and customization. These features enable the development and testing of advanced control strategies and allow users to evaluate different vehicle designs and configurations, enhancing the utility of the simulator in developing autonomous micromobility vehicles.

2) Sensor Models: Accurate sensor modeling is vital for developing autonomous micromobility vehicles, which heavily rely on sensors to perceive their environment and make informed decisions. An ideal simulator should accurately model various sensors, such as LiDAR, cameras, radar, ultrasonic, GPS, and IMU, to ensure perception algorithms designed in simulation perform well in real-world scenarios.

Key features for accurate sensor modeling include realistic representation of sensor characteristics, sensor noise and error modeling, sensor occlusion and visibility modeling, and simulation of environmental effects, such as lighting conditions, weather, and road surface properties. Additionally, an ideal simulator should provide output data in formats compatible with industry standards, ensuring seamless integration with existing tools and frameworks. Validation and benchmarking tools should also be available, enabling users to evaluate the performance of perception algorithms under various sensor configurations and conditions. By incorporating these features, a simulator can effectively support the development and testing of robust perception algorithms for autonomous micromobility vehicles.

3) Pedestrian Models: Realistic pedestrian behavior modeling is essential for evaluating the safety and performance of autonomous micromobility vehicles. An ideal simulator should accurately model pedestrian motion, complex interactions, decision-making processes, and pedestrian-vehicle interactions. Additionally, it should consider varying pedestrian behavior based on factors like age, gender, culture, and disability, as well as the influence of environmental factors on pedestrian movement patterns.

Customizability and scenario generation should be included, allowing users to create diverse pedestrian scenarios to test autonomous micromobility vehicle performance under different conditions. The simulator should also provide tools for assessing pedestrian safety, such as collision risk, near-miss incidents, and safety margins.

4) Customizability: Customizability is crucial for simulators of autonomous micromobility vehicles, enabling users to simulate various scenarios and test different algorithms and strategies. An ideal simulator should offer flexible environment and scenario creation, seamless integration with popular third-party software tools, and compatibility with machine learning frameworks. Modularity and extensible software architecture are essential, allowing users to develop and integrate custom plugins, models, and algorithms to address specific research needs.

An intuitive user interface should enable users to manage simulation scenarios, environments, and parameters easily, while import and export functionalities should facilitate integration with external data sources and sharing simulation results. Lastly, a simulator should support the design and adjustment of dynamic traffic scenarios, including varying vehicle types, densities, and pedestrian and cyclist interactions, to assess autonomous micromobility vehicle performance and safety under a wide range of conditions and scenarios.

5) Scenario and Vehicle Library: A scenario and vehicle library is essential in simulators for autonomous micromobility vehicles, as it streamlines the simulation process, reduces implementation time and effort, and fosters collaboration and knowledge sharing. An ideal simulator should provide diverse and realistic scenarios, as well as a comprehensive set of predefined vehicle models with various sensor configurations and control algorithms.

Import and export functions should be available for sharing custom designs and incorporating external resources. User-friendly editors should enable easy creation and modification of scenarios and vehicle models. Finally, scenario and vehicle validation tools should ensure compatibility with the simulator and adherence to specified constraints, contributing to a more robust and reliable simulation environment for the development and evaluation of autonomous micromobility vehicles.

6) Scalability: In autonomous micromobility vehicle development, a scalable simulator is essential for understanding performance in crowded urban environments. A high-performance simulator should handle large numbers of vehicles, pedestrians, and traffic participants without compromising accuracy. Parallel and distributed computing techniques, leveraging multicore processors, GPUs, or cloud resources, should be incorporated to improve performance in large-scale simulations.

An adaptive level-of-detail approach can ensure efficient resource utilization, while efficient data management and storage techniques handle the large volume of generated data. Offering both real-time and faster-than-real-time simulation capabilities allows users to choose the most popular mode for their needs. Robustness and resilience are crucial for maintaining stability and accuracy in complex scenarios or unexpected events. A simulator with these features significantly contributes to the development and evaluation of autonomous micromobility vehicles in various conditions and scenarios.

7) User-Friendliness: User-friendliness is key for autonomous micromobility vehicle simulators, fostering effective use by researchers and developers. This entails clear documentation, engaging tutorials, streamlined setup, customization options, open-source approach, determinism, and result visualization tools.

Well-structured documentation and tutorials, along with responsive support, simplify user understanding of the simulator. Ease in installation and setup, alongside customizable simulation parameters, enhance user experience and enable tailored simulations to individual requirements.

An open-source approach encourages collaboration and continuous improvement. Determinism, providing consistent results for the same input, simplifies debugging, testing, validation, and performance comparison. Visualization tools are crucial for result analysis and communication to stakeholders, contributing to successful development, testing, and deployment of autonomous micromobility vehicles.

C. Evaluation of Potential Simulators for Autonomous Micromobility Vehicles

In accordance with the requirements delineated in Section III-B, we conducted a thorough evaluation of each simulator discussed in Section III-A. In this section, we present the findings for the top three open-source candidates, i.e., CARLA, AWSIM and Arena-Rosnav-3D. It is important to note that we focused solely on open-source simulators for the final assessment, as their reproducibility and shareability are particularly advantageous for academic purposes. Figure 1 shows the final evaluation results in a radar diagram. Each axis represents one of the aforementioned requirements, and the simulators were graded from 0 to 10.

1) CARLA: CARLA is an open-source simulator for autonomous vehicles based on the Unreal Engine [13]. Launched six years ago, it is supported by Intel, Toyota, and NVIDIA, making it one of the most actively developed open-source simulators for autonomous vehicles.

The vehicle dynamics in CARLA are based on the NVIDIA PhysX engine. However, the documentation for PhysX and its integration in CARLA has significant gaps. While car-like vehicles have their own class, accurately modeling micromobility vehicles with specific dynamics is not well-supported. Overall, the vehicle dynamics in the Unreal Engine receive a rating of 6/10.

The sensor models in CARLA are comprehensive, including sensor occlusion, noise, and lighting conditions. However, validation of sensor models is not explicitly supported, resulting in a score of 9/10. The pedestrian models offer detailed customization options, such as various gaits, sizes, movement sequences, and appearances. The behavior is modeled using Recast & Detour, but there is limited literature discussing the accuracy of this software. With global path planning and local collision avoidance available, the pedestrian models receive a score of 7/10.

The Unreal Engine and CARLA, both being open-source, provide significant customizability. However, their build tool chain is intricate and may exhibit fragility. A ROS bridge facilitates the interface between the Python API and both ROS 1 and 2. While tools for importing new vehicles, maps, and assets are accessible, most adaptations come with their distinct challenges, leading to a customizability rating of 7/10.

The scenario and vehicle library, featuring approximately ten different towns, is extensive. However, not all scenarios are directly applicable to micromobility vehicles and may require adaptation to increase sidewalk detail levels. Overall, the available scenarios score 8/10.

CARLA's complexity, level-of-detail, and computational intensity limit its scalability. Although, the server-client architecture helps balance loads, a recent update enables scenarios with varying levels of detail, and dockerization supports cloud computing; not everything works seamlessly out of the box. Consequently, CARLA's scalability score is 5/10.

CARLA offers user-friendly features, as evidenced by its comprehensive documentation, a wealth of instructional tutorials, an active development community, and ongoing enhancements. However, there are a few points of concern. Firstly, it does not assure determinism across all simulation aspects. Secondly, it has approximately 800 unresolved GitHub issues. A particularly limiting factor is its official support being restricted to Ubuntu 18.04, a version which has reached its end-of-life status. Given these considerations, we rate CARLA's user-friendliness as 4 out of 10.

AirSim, developed by Microsoft, is another simulator based on the Unreal Engine, but its development has been discontinued, leading to its exclusion from further consideration in this study. However, Microsoft has announced plans to develop a successor to AirSim. Deepdrive 2.0 is also developed based on the Unreal Engine, but did not receive an update within the last three years.

2) AWSIM: The AWSIM Simulator, developed by Tier4, employs the Unity engine and primarily supports Autoware simulation [17]. Previously, Autoware utilized the LGSVL Simulator, but its development was discontinued. AWSIM, first released in 2022, lacks the extensive development his-

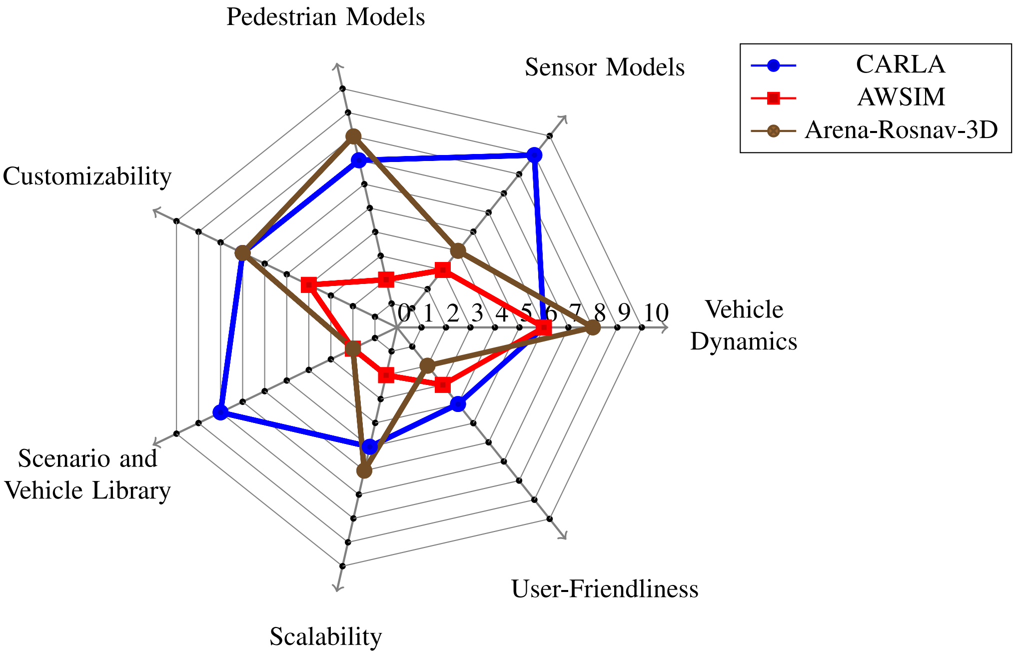

Fig. 1. Evaluation of CARLA, AWSIM and Arena-Rosnav-3D for the Simulation of Autonomous Micromobility Vehicles

tory of CARLA. Its sparse documentation contributes to the uncertainty in our assessment.

Vehicle dynamics in AWSIM are based on Unity, which uses NVIDIA's PhysX for 3D simulation, resulting in a similar score to CARLA at 6/10. Sensor models are more limited, offering only LiDAR, camera, IMU, and GNSS sensors, without error or noise simulation, thus receiving a score of 3/10. The pedestrian model is available, but lacks documented behavior, leading to a score of 2/10.

Though Unity itself is not open-source, it is well-documented. Combined with AWSIM's open-source code, customizability is reasonable. A native ROS 2 interface is available, but further customization options are limited, yielding a score of 4/10. The scenario and vehicle library is considerably limited compared to CARLA, with only one scenario presented in the documentation. The scenario library scores 2/10, as existing Unity assets can still be imported.

Unity and Unreal Engine typically exhibit similar computational performance, but AWSIM has not yet implemented any scaling features. Consequently, scalability is rated as 2/10. As AWSIM is a recent development, documentation is limited, the community is smaller, and few tutorials are available, resulting in a user-friendliness score of 3/10.

3) Arena-Rosnav-3D: Arena-Rosnav-3D is a 3D navigation platform constructed on the Gazebo 3D ROS simulator and integrated with a modified Pedsim Simulator [23]. This platform offers realistic dynamic 3D scenarios for evaluating and benchmarking ROS navigation approaches on various robot platforms. The repository encompasses local planners for dynamic obstacle avoidance, task generators, multiple detailed scenario-worlds, and robot models. Additionally, it features automated creation of random 3D worlds with static and dynamic obstacles, realistic behavior patterns for obstacles, and intermediate planner classes. The platform seamlessly converts randomly generated ROS maps to Gazebo worlds and supports a "random world" task mode that loads a new Gazebo world with each task reset.

Vehicle dynamics are simulated in Gazebo, which supports four different physics engines: ODE, Simbody, Bullet, and DART. The broad array of physics simulators suggests potential improvements in the accuracy of Gazebo's physics simulation. Consequently, Arena-Rosnav-3D receives a score of 8/10 for vehicle dynamics.

Sensor models currently include only LiDAR and localization, but additional models could be implemented through Gazebo. However, the current state warrants a rating of 4/10.

Pedestrians are modeled using the Pedsim Framework, which employs the Social Force model, a commonly used approach for pedestrian simulation. Thus, the pedestrian models receive a score of 8/10.

Customizability is satisfactory. Arena-Rosnav-3D, Gazebo, and several extensions used in the simulator are open-source. However, everything is designed to utilize ROS 1 exclusively, not ROS 2, resulting in a customizability score of 7/10.

The scenario and vehicle library feature nine different maps, but all maps are indoor environments. No city environments are implemented, suggesting significant development is needed for micromobility vehicles in a representative environment. The score is evaluated to be 2/10.

Scalability is decent since execution in Docker is supported, and Gazebo is not as computationally intensive as the Unreal Engine, earning a score of 6/10.

User-friendliness is suboptimal due to the smaller community, unclear development goals, fewer tutorials, and gaps in documentation. User-friendliness is rated 2 out of 10.

D. Interpretation and Discussion

The comprehensive list of simulators presented in Section III-A illustrates the complexity involved in selecting the most suitable simulator tailored to a specific research question. Notably, none of the reviewed simulators have been explicitly designed for the development of autonomous micromobility vehicles.

The optimal simulator requirements, from our perspective, are outlined in Section III-B. Addressing these requirements can expedite the evaluation and adaptation of existing simulators for the simulation of autonomous micromobility vehicles. However, potential biases may exist in these requirements, and they should be applied cautiously.

Section III-C assesses three potential open-source simulator candidates for autonomous micromobility vehicles. Each candidate offers distinct advantages and limitations. The implications of selecting a particular simulator must be evaluated individually, as no universal answer can be provided for all research scenarios.

IV. CONCLUSION

This paper highlights the necessity for a more advanced simulator tailored to autonomous micromobility vehicles. It presents an array of compelling simulators for autonomous vehicles and delineates the requirements for simulating autonomous micromobility. Subsequently, three promising candidates are evaluated based on their ability to fulfill these requirements. A crucial conclusion drawn from this study is the need for enhancing existing simulators to optimally simulate autonomous micromobility vehicles.

The benefits of this work include the provision of a well-defined list of requirements that future research can address and improve upon in specific categories to develop more suitable simulators. Additionally, the paper presents a list of potential candidates for researchers to consider. However, potential shortcomings include biases in the requirements, the possibility of overlooking a simulator in our review, or a lack of knowledge regarding the full potential of a given simulator.

Future research areas involve the improvement of existing simulators and an examination of their adaptability to the specified requirements. Moreover, further investigation is needed to identify any unmentioned requirements and to assess the significance of each requirement.

REFERENCES

[1] Q. Wang, S.-d. Wu, Y.-e. Zeng, and B.-w. Wu, “Exploring the relationship between urbanization, energy consumption, and CO2 emissions in different provinces of China,” *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1563–1579, Feb. 2016.

[2] N. C. Sanchez, L. A. Pastor, and K. Larson, “Autonomous Bicycles: A New Approach To Bicycle-Sharing Systems,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2020, pp. 1–6.

[3] D. Kondor, X. Zhang, M. Meghjani, P. Santi, J. Zhao, and C. Ratti, “Estimating the Potential for Shared Autonomous Scooters,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4651–4662, May 2022.

[4] N. Coretti Sanchez, I. Martinez, L. Alonso Pastor, and K. Larson, “On the simulation of shared autonomous micro-mobility,” *Communications in Transportation Research*, vol. 2, p. 100065, Dec. 2022.

[5] “Kickscooter T60 — Segway AI-Driven Shared E-Scooter.”

[6] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, *A Survey on Simulators for Testing Self-Driving Cars*, Jan. 2021.

[7] E. Leurent, “An environment for autonomous driving decision-making,” 2018.

[8] Z. Cao, E. Biyik, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh, “Reinforcement learning based control of imitative policies for near-accident driving,” in *Proceedings of Robotics: Science and Systems (RSS)*, Jul. 2020.

[9] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, “Microscopic traffic simulation using SUMO,” in *The 21st IEEE International Conference on Intelligent Transportation Systems*, ser. IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2018.

[10] “MIT Racecar,” https://mit-racecar.github.io/.

[11] L. Kästner, T. Buiyan, L. Jiao, T. A. Le, X. Zhao, Z. Shen, and J. Lambrecht, “Arena-rosnav: Towards deployment of deep-reinforcement-learning-based obstacle avoidance into conventional autonomous navigation systems,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6456–6463.

[12] S. Mohanty, E. Nygren, F. Laurent, M. Schneider, C. Scheller, N. Bhattacharya, J. Watson, A. Egli, C. Eichenberger, C. Baumberger, G. Vienken, I. Sturm, G. Sartoretti, and G. Spigler, “Flatland-RL : Multi-Agent Reinforcement Learning on Trains,” Dec. 2020.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” *arXiv:1711.03938 [cs]*, Nov. 2017.

[14] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles,” *arXiv:1705.05065 [cs]*, Jul. 2017.

[15] “Deepdrive,” https://deepdrive.io/.

[16] J. K. Haas, “A history of the unity game engine,” 2014.

[17] “Quick Start Demo - AWSIM document,” https://tier4.github.io/AWSIM/GettingStarted/QuickStartDemo/.

[18] 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)*, Sep. 2020, pp. 1–6.

[19] T. V. Samak and C. V. Samak, “AutoDRIVE Simulator – Technical Report,” Nov. 2022.

[20] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sep. 2004, pp. 2149–2154 vol.3.

[21] W. H and R. Thomas, “AutoCarROS2,” Apr. 2023.

[22] “Demo of Prius in ROS/GAZEBO,” Open Robotics, Apr. 2023.

[23] L. Kästner, T. Buiyan, L. Jiao, T. A. Le, X. Zhao, Z. Shen, and J. Lambrecht, “Arena-Rosnav: Towards Deployment of Deep-Reinforcement-Learning-Based Obstacle Avoidance into Conventional Autonomous Navigation Systems,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 6456–6463.

[24] A. Amini, T.-H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus, “VISTA 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[25] L. Paull, J. Tani, H. Ahn, J. Alonso-Mora, L. Carlone, M. Cap, Y. F. Chen, C. Choi, J. Dusek, Y. Fang, D. Hoehener, S.-Y. Liu, M. Novitzky, I. F. Okuyama, J. Pazis, G. Rosman, V. Varricchio, H.-C. Wang, D. Yershov, H. Zhao, M. Benjamin, C. Carr, M. Zuber, S. Karaman, E. Frazzoli, D. Del Vecchio, D. Rus, J. How, J. Leonard, and A. Censi, “Duckietown: An open, inexpensive and flexible platform for autonomy education and research,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1497–1504.

[26] O. Michel, “WebotsTM: Professional Mobile Robot Simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, Mar. 2004.

[27] E. Espié, C. Guionneau, B. Wymann, C. Dimitrakakis, R. Coulom, and A. Sumner, “TORCS, The Open Racing Car Simulator,” 2005.