A Survey on Small-Scale Testbeds for Connected and Automated Vehicles and Robot Swarms

Preprint · February 2024

DOI: 10.13140/RG.2.2.16176.74248/1

12 authors, including:


Armin Mokhtarian
RWTH Aachen University


Patrick Scheffe
RWTH Aachen University


Maximilian Kloock
RWTH Aachen University


Simon Schäfer
RWTH Aachen University

A Survey on Small-Scale Testbeds for Connected and Automated Vehicles and Robot Swarms*

Armin Mokhtarian[1], Patrick Scheffe[1], Maximilian Kloock[1], Simon Schäfer[1], Heeseung Bang[2], Viet-Anh Le[2], Sangeet Ulhas[5], *Student Members, IEEE*,
Johannes Betz[3], Sean Wilson[4], Spring Berman[5], *Members, IEEE*,
Amanda Prorok[6], Bassam Alrifaee[1], *Senior Members, IEEE*.

Fig. 1: Collage showcasing diverse testbeds in the realm of Connected and Automated Vehicles and Robot Swarms.

[1]RWTH Aachen University - Chair for Embedded Software, 52074 Aachen, Germany email: {lastname}@embedded.rwth-aachen.de
[2]Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. email: {heeseung,vietale}@udel.edu
[3]Professorship of Autonomous Vehicle Systems, TUM School of Engineering and Design, Technical University Munich, 85748 Garching, Germany; Munich Institute of Robotics and Machine Intelligence (MIRMI) email: johannes.betz@tum.de
[4]Georgia Tech Research Institute (GTRI), Institute for Robotics and Intelligent Machines (IRIM), Georgia Institute of Technology, Atlanta, GA, 30332 email: sean.wilson@gtri.gatech.edu
[5]School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, USA, 85281 email: {sulhas, spring.berman}@asu.edu.
[6]University of Cambridge, UK: email: asp45@cam.ac.uk.

***Abstract*—Connected and Automated Vehicles (CAVs) and Robot Swarms (RS) have the potential to transform the transportation and manufacturing sectors into safer, more efficient, sustainable systems. However, extensive testing and validation of their algorithms are required. Small-scale testbeds offer a cost-effective and controlled environment for testing algorithms, bridging the gap between full-scale experiments and simulations. This paper provides a structured overview of characteristics of testbeds based on the sense-plan-act paradigm, enabling the classification of existing testbeds. Its aim is to present a comprehensive survey of various testbeds and their capabilities. We investigated 17 testbeds and present our results on the public webpage www.cpm-remote.de/testbeds. Furthermore, this paper examines seven testbeds in detail to demonstrate how the identified characteristics can be used for classification purposes.**

DOI:10.13140/RG.2.2.16176.74248

***Index Terms*—Small-Scale Testbeds, Connected and Automated Vehicles, Robot Swarms, Survey.**

I. Introduction

A. Motivation

Connected and Automated Vehicles (CAVs) and Robot Swarms (RS) can significantly improve the safety, efficiency, and sustainability of the transportation and manufacturing sectors. However, achieving their full potential requires extensive testing and validation of their algorithms. Full-scale experiments are considered the gold standard for testing, as they provide comprehensive results in real-world environments. These experiments can be prohibitively expensive and time-consuming, and they often present challenges in terms of reproducibility. Simulations can provide an alternative to full-scale testing, but the quality of simulation results may not always be reliable due to simplifications in the models. Small-scale testbeds can bridge the gap between full-scale experiments and simulations by providing a cost-effective and controlled environment. These testbeds can simulate real-world scenarios with varying degrees of complexity and realism, allowing researchers to evaluate algorithms under different conditions. In this paper, we will use the term *testbed* to refer to small-scale testbeds and the term *CAV/RS* to denote the domain of connected and automated vehicles and robot swarms.

Advancements in the CAV/RS domain are facilitated by testbeds that focus on particular use cases. For the development and testing of each use case, a testbed must fulfill specific requirements. In this paper, we derive characteristics to describe whether and how a specific requirement is fulfilled. To structure these characteristics, we follow the well-known sense-plan-act paradigm. This paradigm divides a system into three main components: perception, decision-making, and control, each with specific testing needs. The significance of certain characteristics varies depending on the researcher's specific focus. For instance, when considering decision-making, a researcher may find the sensors provided by a testbed irrelevant, as long as the current location of a vehicle is accessible. In this case, instead of *sensors*, a characteristic such as *vehicle count* would serve as a more important factor, as it helps in determining possible scenarios. Conversely, for a researcher concentrating on perception, information about the available sensors is more important, as it describes the tools available for localization, object identification, lane tracking, and distance measurements. Through the identification of domain-specific requirements, we were able to derive 56 characteristics, including *vehicle count* and *sensors*. We use these characteristics to provide a structured overview of current CAV/RS testbeds to assist researchers in selecting the most suitable testbed for their use case. Offering a comprehensive overview also helps researchers in identifying areas of interest and potential gaps in research. Figure 1 displays a collage of some of the testbeds investigated in this paper.

B. Contribution of this Article

This paper:

- presents a survey to cover the topic of small-scale testbeds for CAV/RS.
- provides a comparison of testbeds based on various characteristics, including hardware, software, documentation, accessibility, etc.
- introduces a webpage that allows researchers to access and contribute updates on the testbeds.

In the following sections, we discuss characteristics for classifying testbeds derived from the sense-plan-act paradigm. The characteristics are based solely on objective information and are not intended to rank any of the testbeds. We then apply the characteristics to classify existing testbeds and publish the results on our webpage. We have investigated 17 different testbeds [1]–[17]. However, the number of testbeds is growing and their characteristics are changing, as many of these testbeds are in continuous development. Hence, we discuss the characteristics of only some of the investigated testbeds in this paper and provide a table comparing all investigated testbeds at www.cpm-remote.de/testbeds. Figure 2 shows a screenshot of the webpage that displays the table. This webpage offers a overview and enables continuous updates beyond the publication of this paper. In addition, we invite creators of testbeds to contribute to this table.

C. Related Work

A systematic review by Caleffi et al. [18] summarizes existing literature in the field of CAV testbeds. While it identifies trends, such as the recent increase in publications and the predominant focus on software, it does not conduct a comparative analysis. Although there have been previous attempts to survey testbeds, the most recent comparative paper dates back to 2013 [19]. The authors focus on RS and they mainly describe two approaches to classify the testbeds. The first approach is a classification by level of complexity: a distinction is made between (i) 'non-integrated' e.g. multi-robot testbeds, (ii) 'partially integrated' e.g. multi-robot testbeds with sensor networks, and (iii) 'highly integrated' e.g. federated testbeds. In this context, integrated refers to the level of coordination and combination among different components or subsystems within the testbed environment. The second approach classifies according to the following criteria: (i) range of experiments (application-driven, functionality-driven and general-purpose), (ii) architecture flexibility, (iii) target users of the testbed, (iv) proximity between experiments and the final application, and (v) testbeds in real deployments (such as industrial sites, urban environments). Both approaches describe testbeds on a rather abstract level. However, more specific criteria are necessary to accurately characterize recently developed testbeds. Hence, most of the recently published literature includes its own related work section, often featuring a table to compare selected testbeds. However, these tables significantly vary in both covered testbeds and detail. One of the more extensive examples can be found in the work by Samak et al. [20].

To address the redundancy observed in the literature, we aim to unify the fragmented information by providing an

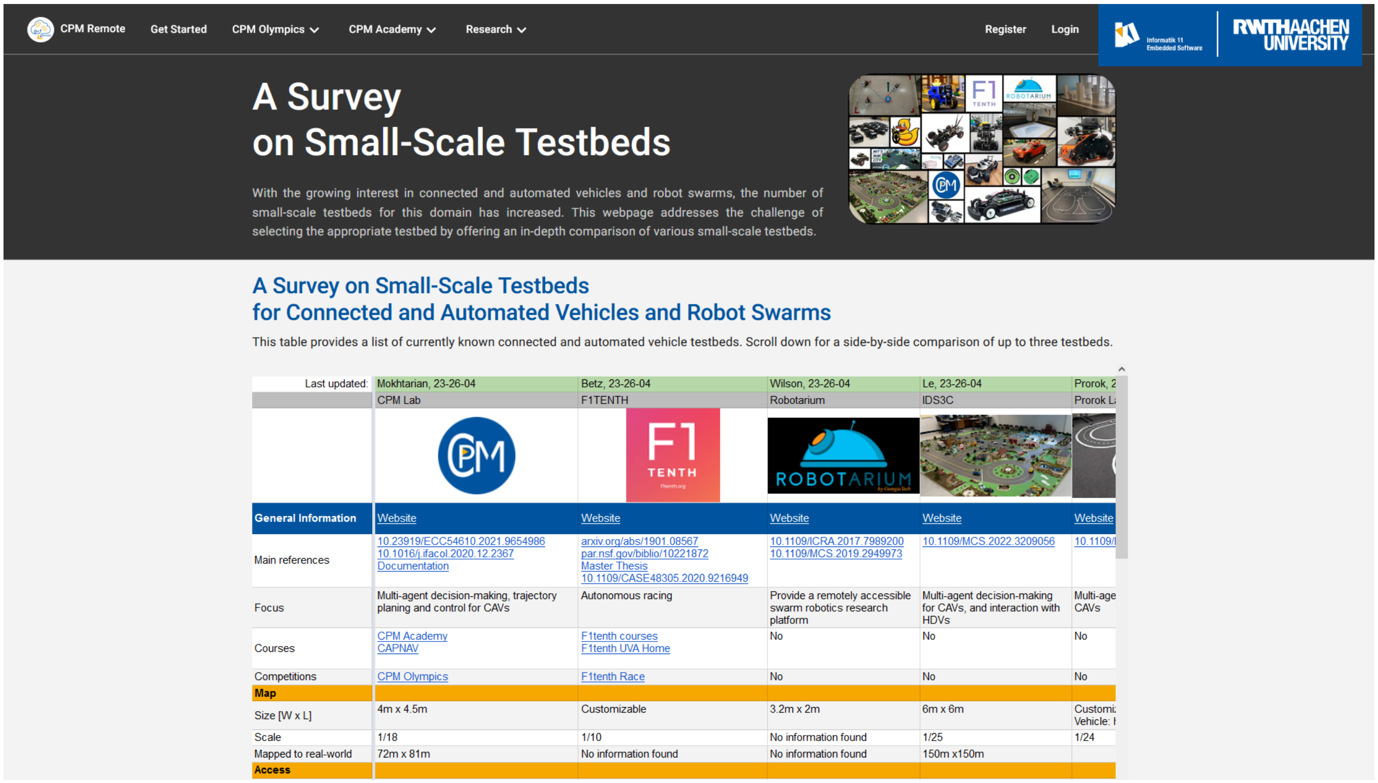

Fig. 2: A screenshot of the publicly accessible webpage [21] that lists all the testbeds investigated in this study.

up-to-date table that contains as many testbeds as possible. To maintain the relevance and accuracy of the information, we are committed to regularly updating it on our dedicated webpage [21]. Our webpage currently contains information on 17 different testbeds. For the purposes of our discussion in this paper, we selected seven testbeds to focus on in more detail: The CPM Lab [1], IDS3C [22], Duckietown [3], F1TENTH [5], the Robotarium [4], The Cambridge Minicar by the Prorok Lab [6], and CHARTOPOLIS [9]. These testbeds were chosen because of their unique characteristics and the insights they can provide into the sense-plan-act paradigm. By analyzing these testbeds, we aim to describe a set of characteristics that can be used to classify other CAV/RS testbeds.

II. Characteristics of Testbeds

In this section, we present and explore a subset of the 56 characteristics that we have identified. The purpose is to offer an overview and discuss some of the key characteristics. First, we will introduce general characteristics that cannot be specifically assigned to the sense, plan, or act domains. Subsequently, we will delve into the characteristics that are unique to each of these domains, providing a more detailed examination of their specific attributes. For an extensive list containing all 56 characteristics, please refer to our webpage.

A. General Characteristics

The primary design objective for most testbeds is to support users in testing algorithms within a scaled-down environment. Despite the diversity in the specific purposes of individual testbeds, commonalities exist among them. This section aims to outline the general characteristics by highlighting these shared features.

1. Focus: The focus characteristic encapsulates a short, high-level description of a testbed. Typically included in the introduction to the testbed, it describes the specific use cases that the testbed is designed to address. This description serves as a foundational overview, providing a quick insight into the primary objectives of the testbed.

In the context of our exploration, we leverage this characteristic to explain the use cases investigated, the methodologies employed, and the insights garnered from the experiments conducted within the testbed. This enables an understanding of the testbed's overarching purpose. Furthermore, testbeds can concentrate on specific problems within an area of focus. For example, within the focus on multi-vehicle coordination, specific attention could be given to traffic management, communication, and human-autonomy interactions.

2. Software: The diversity in software languages, architectures, frameworks, and computation models across different testbeds reflects the versatility of the testbeds. Different testbeds may utilize various software components depending on factors such as the targeted application domain, the availability of libraries and tools, and the preferences of the researchers involved. For instance, some testbeds may use Python, a widely adopted language known for its simplicity and extensive scientific computing libraries. Others may leverage languages such as C++ for performance optimization or MATLAB for domain-specific functionalities.

Another dimension of the software characteristic is the selection of a software architecture. The architecture determines how the software components are organized and communicate with each other. Examples include service-oriented architectures, client-server architectures, peer-to-peer architectures and event-driven architectures. The choice of software architecture impacts factors like scalability, fault-tolerance, modularity, and interoperability within the testbed.

Frameworks also contribute to the software characteristic of testbeds. Frameworks provide reusable software components, libraries, and tools that simplify the implementation of software. Testbeds can leverage existing frameworks tailored for specific purposes, for example simulation frameworks, Internet of Things (IoT) frameworks, or machine learning frameworks.

Computation models represent another vital aspect of the software characteristic. Some testbeds employ deterministic computation models, such as Finite State Machines for software design, which involve predictable and reproducible outcomes based on defined input conditions. Deterministic models are particularly suitable for experiments that require precise and reliable results. However, probabilistic computation models introduce randomness and uncertainty into the experiment outcomes, allowing researchers to explore scenarios where stochastic factors play a significant role. Hybrid models combine both deterministic and probabilistic approaches, providing a balance between predictability and stochasticity in the experimental outcomes.

3. Documentation: The documentation characteristic of a testbed refers to the availability and quality of accompanying documentation that describes the testbed's features, functionalities, usage guidelines, and other relevant information. It plays a critical role in the selection of a testbed, as it significantly impacts the user's ability to understand, utilize, and effectively conduct experiments within the given environment.

4. Accessibility: The accessibility of a testbed refers to the ease with which interested parties can access and utilize the facility for conducting experiments. This section explores the different strategies employed to make testbeds accessible.

1) Open-Source Blueprints:
   Some testbeds provide comprehensive and detailed open-source blueprints to enable interested individuals or organizations to replicate their experimental setup. These testbeds document their design, construction processes, component specifications, and operational procedures in detail. By sharing such blueprints, these testbeds empower users to replicate the testbed, facilitating the advancement of scientific knowledge through experimental testing.
2) Ready-to-Use Setup:
   Testbeds may offer a ready-to-use setup for sale that can be readily deployed by users without the need for extensive technical expertise or complex construction. These testbeds provide a complete package of the required hardware and software, allowing users to quickly set up the experiment and focus on their specific research objectives. Such testbeds streamline the accessibility process, particularly for researchers who may not have the necessary resources to build their own testbed from scratch.
3) Walk-In Access:
   Certain testbeds prioritize accessibility by allowing researchers to physically visit the testbed and utilize the experimental facilities on-site. These testbeds may require users to schedule their visits in advance to ensure availability. By offering walk-in access, testbeds encourage direct engagement and hands-on experimentation, fostering collaboration and enabling users to have a more interactive experience with the testbed infrastructure.
4) Remote Access:
   Another approach to enhancing accessibility is through remote-access capabilities. These testbeds enable users to access and control the experimental setup remotely, without the need to be physically present at the testbed. Remote access may involve using a web-based interface or specialized software that allows users to interact with the testbed, adjust parameters, and collect data from any location. This characteristic proves particularly advantageous for researchers who are geographically distant from the testbed or have limited mobility. Furthermore, this is the most cost-efficient way to access a testbed.

5. Scenario: A scenario refers to a specific set of conditions, events, and interactions that are conducted within the testbed. Scenarios often involve the deployment of different maps and the presence of diverse actors (e.g., pedestrians, obstacles, etc.), each contributing to the complexity and realism of the experimental setup. The choice of scenarios enables researchers to address specific research questions. Therefore, the selection of a scenario directly correlates with the researchers' focus.

B. Sense-driven Characteristics

Within the sense domain of the sense-plan-act paradigm, the primary focus is on perceiving the environment and gathering relevant information using sensors. The derived characteristics to describe the sense domain are described in the following.

1. Sensors: Sensors form the foundation of perceiving the environment in a testbed. Different sensors, such as cameras, LiDAR, radar, and ultrasonic sensors, provide distinct information about the surrounding objects and their properties. The availability and quality of sensors in a testbed greatly affect the richness and accuracy of the perceptual data collected. Researchers can evaluate the sensor suite and its capabilities to determine if the testbed adequately supports their sensing needs.

2. Positioning System: Many applications benefit from having accurate and reliable positioning information. A testbed could include an accurate positioning system such as an indoor positioning system, which is capable of precisely tracking the position and orientation of vehicles.

3. Accuracy: The testbed's sensors supply perceptual information. The accuracy of this information is an aspect to consider. It relates to how closely the sensed data represents the actual state of the environment. Researchers should consider the accuracy of individual sensors and the overall perception system to assess whether the testbed meets their specific accuracy requirements.

4. Traffic Management Infrastructure: This assessment includes features such as road networks, traffic signs, traffic lights, and other components that simulate real-world traffic infrastructure. A testbed with a comprehensive traffic management infrastructure enables researchers to study and evaluate the performance of their algorithms and systems in realistic traffic conditions.

5. Surroundings: The environment surrounding a testbed contributes to its suitability for experiments. The characteristics of the surroundings, such as urban, suburban, or rural settings, can influence the complexity of the testbed. For instance, an urban testbed might offer challenges related to high-density traffic, while a suburban testbed could have different obstacles and road conditions.

The final webpage [21] showcasing the comparison table presents the derived characteristics of the sense domain in more detail. We expand several characteristics further to provide in-depth information. For instance, we distinguish between on-vehicle and global sensors, offering a more detailed understanding of the sensor suite employed within the testbed.

C. Plan-driven Characteristics

The plan domain plays a crucial role in enabling CAV/RS to generate appropriate actions based on the information gathered from the environment. This section focuses on deriving characteristics related to the plan domain that can be used to evaluate and compare testbeds.

1. Testbed Architecture: The testbed architecture refers to the overall design and structure of the experimental setup. It encompasses the physical layout, the presence of specific infrastructure elements, and the integration of hardware and software components. When evaluating a testbed for the plan domain, researchers should consider whether the architecture supports the execution and evaluation of planning algorithms and strategies effectively. For reproducible experiments, a deterministic timing of the testbed components is required, i.e., all computing and sensing devices have to be synchronized. The architecture may guarantee deterministic and reproducible experiments.

2. Distributed Computations: In many real-world scenarios, CAV/RS operate in distributed environments, where the processes are spread out across multiple nodes. Assessing the capability of a testbed to handle distributed computation is essential, as it determines whether the testbed can efficiently manage coordination and communication between vehicles during the planning phase.

3. Computation Power: The computation power available within a testbed significantly impacts the complexity and scalability of planning algorithms that can be deployed. Higher computation power allows for more advanced planning techniques, such as optimization-based algorithms. Researchers should consider characteristics such as the processing capabilities of the testbed's hardware infrastructure or availability of parallel computing resources.

4. Computation Schemes: The computation scheme refers to the methodology or algorithms employed for planning within the testbed. Distributed planning algorithms follow different computation schemes, e.g., sequential, parallel, and hybrid computations. Hybrid computations involve a combination of sequential and parallel processing methods to optimize planning within the testbed.

5. Human-robot Interaction: In many real-world scenarios, CAV/RS coexist and interact with humans in shared environments. Evaluating the testbed's characteristics related to human-robot interaction is essential for assessing its suitability for scenarios involving mixed traffic. Characteristics such as the availability of realistic human models, capabilities for simulating human behaviors, or an interface for humans to interact with the testbed are important considerations.

6. Different Kinds of Vehicles: Assessing the capability of a testbed to handle heterogeneous vehicles is vital when studying CAV/RS. Different types of vehicles may have distinct sensing and acting capabilities, leading to diverse planning requirements. The testbed should support the integration of various vehicle models, allowing researchers to evaluate planning algorithms for scenarios involving vehicles with different capabilities, such as cars or trucks.

7. Vehicle Count: The number of vehicles in a testbed is an important characteristic, particularly in scenarios involving transportation or traffic management. Evaluating the vehicle count characteristic involves understanding the testbed's capacity to handle multiple vehicles concurrently, the scalability of planning algorithms with increasing vehicle count, and the impact on the planning system's performance as the number of vehicles increases.

D. Act-driven Characteristics

This subsection aggregates information related to translating a plan into action, which is known as trajectory following. In a hierarchical architecture such as the sense-plan-act paradigm, planning is a separate layer from trajectory following. A vehicle model links the two layers by representing constraints on the vehicle dynamics. The planning layer is responsible for collision avoidance, which requires knowledge of the vehicle dimensions.

1. Dynamics: Modeling the system dynamics is a fundamental step in classical control design. When planning respects the constraints on the vehicle dynamics, more accurate trajectory following is possible. If the vehicle model significantly deviates from the actual vehicle dynamics, a theoretically collision-free trajectory could lead to a collision when executed. Hence, it is crucial to test the trajectory planning model through experiments with real vehicles. Depending on the steering geometry, we differentiate between differential-drive and Ackermann-steering vehicles. Models for differential-drive vehicles include the point-mass model with double-integrator dynamics and the unicycle model, which separates control inputs for translational and rotational velocity [13], [23]. Models for Ackermann-steering vehicles include the point-mass model with double-integrator dynamics, the kinematic bicycle model, the kinetic bicycle model, and more complicated models. An overview of vehicle models is given in [24]. Depending on the model choice, different vehicle parameters must be known. Generally speaking, the limits on the vehicle's speed and acceleration affect the constraints on its longitudinal dynamics, and the limits on the vehicle's

Fig. 3: Cyber-Physical Mobility Lab at RWTH Aachen University [1].

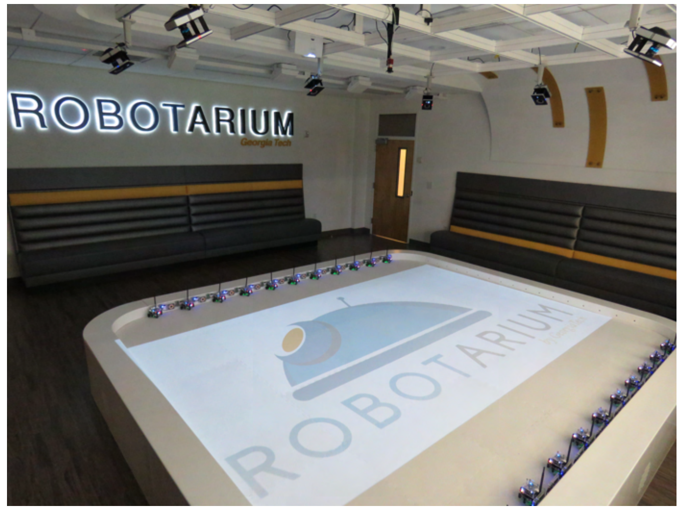

Fig. 5: Robotarium Testbed at Georgia Institute of Technology [4].

Fig. 4: IDS Scaled Smart City at Cornell University [22].

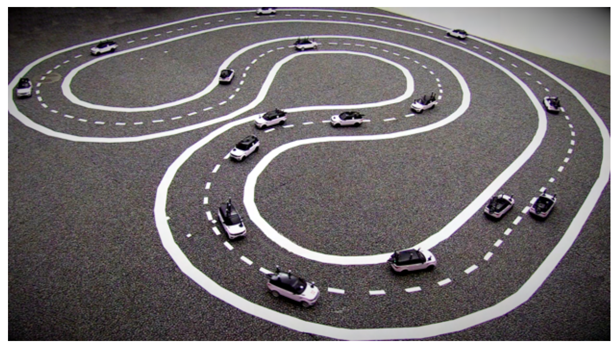

Fig. 6: Cambridge Minicars at the Prorok Lab at Cambridge University [6].

steering angle and steering rate affect the constraints on its lateral dynamics.

2. Geometry: The models for vehicle dynamics given above describe the motion of a vehicle's center of gravity. The occupied area of a vehicle is distributed around this center of gravity. In order to ensure collision-free plans, we need to guarantee that no occupied areas intersect. The vehicle's geometry is needed to compute its occupied area.

III. Overview of Testbeds

Through this paper, our aim is to provide researchers with a understanding of the current landscape of CAV/RS testbeds. By examining the sense-plan-act paradigm characteristics, we can gain deeper insights into the capabilities, strengths, and potential areas of advancement of each testbed. Discussing all 17 investigated testbeds would exceed the scope of this paper. Therefore, we have selected seven representative testbeds to provide an overview and to present how the characteristics can be used to describe a testbed. The testbeds are: The Cyber-Physical Mobility (CPM) Lab [1], IDS3C [22], Duckietown [3], F1TENTH [5], the Robotarium [4], the Cambridge Minicar [6] and CHARTOPOLIS [9]. Figures 3 to 6 show images of four of these testbeds.

A. General Information

This section discusses the specific application of the general characteristics derived in the previous section, i.e., focus, software, documentation, accessibility, and scenario, to characterize the seven testbeds.

1. Focus: The CPM Lab, the Cambridge Minicar, CHARTOPOLIS, and IDS3C are primarily designed for multi-vehicle planning and control of CAVs. The CPM Lab, CHARTOPOLIS, and IDS3C can also be used to investigate interactions between CAVs and human-driven vehicles (HDVs). For example, two driving modes are being developed for CHARTOPOLIS, as they were for its smaller predecessor described in [8]: one in which a vehicle navigates autonomously, and one in which it is remotely driven by a human participant. In contrast, testbeds such as Duckietown and F1TENTH primarily focus on perception. Duckietown implements a miniature-city-scale environment for exploring camera-based localization, while F1TENTH operates on a larger scale and employs a LiDAR sensor.

While the aforementioned testbeds are all situated in the domain of CAVs, the Robotarium operates within the domain of multi-robot coordination (RS). This testbed enables the control of a large number of robots and facilitates the study

Fig. 7: The Go-CHART at Arizona State University [8].

Fig. 8: An exemplar F1TENTH vehicle built at the University of Pennsylvania [5].

of emergent behaviors and distributed algorithms in swarm robotics, a field focused on the coordinated behavior and interaction of multiple autonomous robots operating as a collective system.

Although most platforms are primarily utilized for research purposes, some are also integrated into educational settings and employed for competitions. In particular, F1TENTH has been utilized in university courses focused on autonomous vehicles [25] and Duckietown offers an online open course on AI and robotics [26]. Additionally, the CPM Academy allows users to learn and address specific problems in intelligent transportation systems [27], [28]. In terms of competitions, F1TENTH organizes autonomous racing competitions that challenge participants to avoid crashes and minimize lap times [29]. Duckietown hosts the AI Driving Olympics, a semiannual competition that focuses on machine learning and artificial intelligence [30]. The CPM Lab introduced the annual CPM Olympics [31], which enables users to remotely access challenging real-world CAV scenarios deployed on the testbed.

2. Software: Some of the testbeds are built upon common open-source software architectures. For instance, the

Fig. 9: Duckietown at Massachusetts Institute of Technology [3].

CPM Lab utilizes the Data Distribution Service (DDS) and IDS3C and F1TENTH employ the Robot Operating System (ROS). Duckietown has developed a containerized architecture, in which the system is divided into smaller, isolated units called containers that are each responsible for a specific part of the main functionalities. The Robotarium follows a comparable approach, where the architecture consists of three main groups: simulation-based components, testbed interface components, and coordinating server applications. Similarly, CHARTOPOLIS uses a modified Donkeycar [32] library or the ROS library based on the application requirements. Concerning the supported programming languages, the testbeds utilize some widely used options, such as C/C++, Python, and MATLAB.

3. Documentation: The majority of the testbeds provide users with comprehensive manuals and access to source code. CPM Lab and F1TENTH offer complete open-source code that facilitates the reconstruction and comprehension of the testbed's architecture. Conversely, the Robotarium and the Cambridge Minicar offer code that is specifically designed for simulating simple examples. Furthermore, CPM Lab and F1TENTH furnish a construction manual detailing the process of building vehicles and environments.

4. Accessibility: Each testbed offers varying degrees of accessibility. The CPM Lab [33] and the Robotarium [23] enable users to remotely interact with the testbeds through the internet. The CPM Lab has seamlessly integrated its entire interface, including a simulation environment, within a web application. Conversely, the Robotarium necessitates a local installation for development purposes, yet its interface can still be accessed through a web application.

To facilitate the utilization of some testbeds, simulators have been developed for the CPM Lab, IDS3C, Duckietown, Robotarium, and F1TENTH. The simulators provide users with the opportunity to test their algorithms without the inherent risks associated with hardware experimentation. The simulators have varying levels of complexity. For example, the CPM Lab includes digital twin representations of the testbed, which are virtual replicas that mimic the behavior and characteristics of the physical system they represent. Moreover, there is a substantial variation in the technologies employed, contributing to the diverse complexity of these simulators. IDS3C has introduced a Unity-based simulator known as IDS3D City [34], which allows users to rapidly iterate their control algorithms and experiments before deploying them to IDS3C. The Robotarium provides a simulator compatible with both MATLAB and Python interfaces. CHARTOPOLIS employs a map in the CARLA [35] simulation environment to mimic the testbed in simulation. Additionally, Duckietown and F1TENTH have developed simulators built upon the OpenAI Gym framework [36], [37]. Notably, F1TENTH expands its simulation capabilities by providing a simulator integrated within the ROS Gazebo environment [5]. These simulators empower users to implement and evaluate their algorithms within meticulously controlled virtual environments.

An alternative approach to enhancing testbed accessibility has been adopted by Duckietown and F1TENTH, aiming to make the setup more affordable and easily replicable. In the case of Duckietown, users have the option to purchase ready-to-use setups, reducing the barriers to entry and facilitating broader adoption.

5. Scenario: IDS3C primarily focuses on urban driving scenarios such as intersections, roundabouts, merging roadways, and corridors [38], [39]. It is also equipped with driver emulation stations (remote vehicle operation), which enables the exploration and study of human driving behaviors and their interaction with CAVs. The Cambridge Minicar testbed specializes in providing a multi-lane freeway scenario. In contrast to these, F1TENTH, Duckietown, and the Robotarium offer users the ability to customize maps and scenarios according to their preferences.

The CPM Lab offers 1:18 scaled representations of highways and intersections and also allows users to customize the map to suit their needs. Additionally, the CPM Lab provides real-world scenarios for a competition and supports another benchmark framework called CommonRoad [24]. IDS3C and Duckietown enable users to validate traffic management algorithms, as they are equipped with stop and yield signs to simulate realistic traffic conditions.

B. *Sense-based Information*

Localization can be accomplished through onboard vehicle sensors or sensors embedded in the environment. A notable characteristic of most of the investigated testbeds is that they combine both methods.

1. Sensors: The Duckietown testbed provides an example of vehicle-centric localization. The vehicles in Duckietown rely on an array of onboard sensors, including a Hall effect sensor (odometer), a front-facing camera, a time-of-flight sensor, and an Inertial Measurement Unit (IMU). These sensors collect data on the vehicle's movement, the surrounding environment, and the vehicle's position relative to that environment, allowing each vehicle to estimate its position and independently navigate the testbed.

The CHARTOPOLIS testbed presents a similar approach to Duckietown, featuring an onboard camera for self-localization. CHARTOPOLIS is also equipped with overhead cameras to perform vision-based tracking of agents. An OptiTrack camera system has been installed to enhance tracking capabilities. However, it is currently not fully integrated, leaving room for future upgrades to the setup. In contrast, the F1TENTH testbed employs a different sensing modality, using a LiDAR sensor mounted on the vehicle. LiDAR enables the vehicles to construct a detailed 3D map of their surroundings and precisely determine their location within this map.

However, not all of the testbeds rely primarily on vehicle-centric localization. Other testbeds, such as the CPM Lab and IDS3C, demonstrate a globally coordinated approach. While the vehicles in these testbeds are equipped with sensors such as an IMU and an odometer, these only play a supplementary role in localization. A global positioning system provides the primary means of positioning, with further details covered in the subsequent section.

2. Positioning System: CPM Lab is an example case of using a global positioning system for localization. It employs a ceiling-mounted Basler camera that detects LEDs mounted on each vehicle. Each vehicle is equipped with four LEDs: three indicate the vehicle's pose, while the fourth blinks at a specific frequency to identify the vehicle itself. This LED-based system is the primary method for localizing the vehicles within the CPM Lab environment.

Similar approaches are utilized by the IDS3C, Robotarium, and Cambridge Minicar testbeds. These testbeds use high-precision camera systems to track vehicle poses: a Vicon camera system in IDS3C and the Robotarium and an OptiTrack camera system for the Cambridge Minicar. These cameras capture the vehicles' locations at high refresh rates, facilitating accurate real-time localization.

In addition to the primary global positioning system, CPM Lab integrates a secondary system based on a sensitive surface layer with pressure sensors. This innovative approach provides a second method for tracking vehicles' positions by detecting the pressure distribution on the surface layer. The ability to obtain position data from multiple sources could enhance the accuracy and reliability of localization via sensor data fusion.

3. Accurancy: Both IDS3C and the Robotarium, which utilize the Vicon system, report localization errors below 1 mm. The Cambridge Minicar testbed, which uses the OptiTrack system, reports an even higher degree of accuracy, with localization errors below 0.2 mm. These values indicate that global positioning systems based on high-precision cameras can achieve extremely high localization accuracy, providing a reliable foundation for complex navigation tasks.

The camera-based global positioning system in the CPM Lab, while not as precise as the Vicon or OptiTrack systems, still provides an acceptable level of accuracy for a significantly lower cost. With the reported localization error under 3 cm, the CPM Lab's system can adequately support a broad range of autonomous navigation tasks.

Conversely, for testbeds that primarily use vehicle-centric localization, such as Duckietown and F1TENTH, the localization accuracy can vary significantly based on the quality of the sensor data, the sophistication of the user's algorithm, and the algorithm's ability to handle uncertainties and noise.

4. Traffic Management: Duckietown and CHARTOPOLIS provide the incorporation of traffic management systems, which include traffic signs and traffic lights. These systems regulate and manage vehicle movement within the testbed, thereby creating dynamic and interactive environments. The inclusion of traffic management systems allows these testbeds to simulate real-world urban traffic conditions, providing a platform for testing and developing algorithms for traffic rule compliance, intersection management, and multi-vehicle coordination.

F1TENTH, the CPM Lab, and IDS3C take this concept further by introducing moving obstacles such as pedestrians into the environment. These obstacles present the vehicles with more unpredictable driving scenarios, necessitating more complex planning and decision-making capabilities. The inclusion of moving obstacles thus presents a platform for researching and testing dynamic obstacle avoidance algorithms.

5. Surroundings: The IDS3C, Duckietown, and CHARTOPOLIS testbeds present structured environments that mimic real-world settings. The surroundings in these testbeds consist of scenery (e.g., trees, grass) and buildings, adding layers of complexity and realism to the navigational challenges. These additions allow the testbeds to simulate various urban scenarios, thereby enabling comprehensive testing of navigation algorithms under diverse and complex conditions.

C. Plan-based Information

1. Testbed Architecture: The CPM Lab exhibits a multi-layered architecture. The high-level layer is situated on the computational units, playing a vital role in decision-making. The mid-level and low-level layers are deployed on the vehicles, and they handle tasks such as control of the vehicles' actuators. A unique aspect of this architecture is the usage of middleware which ensures deterministic and reproducible experiments. Communication is enabled through a Data Distribution Service (DDS) that operates via WLAN.

Several other testbeds adopt multi-layered architectures, including IDS3C, CHARTOPOLIS, and the Robotarium. Each of these platforms relies on WLAN-based communication among the testbed's components, enhancing synchronicity and ensuring efficient task execution.

In contrast, Duckietown employs a unique communication strategy among its vehicles. Rather than using conventional wireless communication, it utilizes on-board LEDs that are detected by other vehicles. This method presents an alternative avenue for exploring vehicle communication strategies, demonstrating the adaptability and breadth of potential designs for testbed architecture.

2. Distributed Computations: Most testbeds adopt a distributed approach to computational tasks. This methodology provides the advantage of computational load balancing, error resilience, and localized decision-making capabilities, enabling a higher degree of autonomy for individual vehicles.

The CPM Lab's distributed approach is realized by incorporating an Intel NUC for each vehicle; notably, the NUCs are not on-board the vehicles, but rather serve as external computational units.. This design effectively delegates computational tasks to each vehicle's dedicated unit, thereby facilitating parallel processing and autonomous decision-making. Similarly, Duckietown, F1TENTH, and CHARTOPOLIS use a distributed computation approach, where each vehicle relies on its own computational power. This allows for greater autonomy and real-time decision-making on a per-vehicle basis, enhancing the system's capacity to respond to dynamic changes in the environment.

In contrast, testbeds such as IDS3C that do not utilize a distributed computation approach may centralize their computations, which can offer a different set of advantages such as global system coherence and simplified data management.

3. Computation Power: In the CPM Lab, the Intel NUC computation units are located off-board the vehicles, since they are too large to be integrated onto the vehicles themselves. In contrast, F1TENTH operates at a larger scale, which allows it to carry more substantial hardware components. Specifically, it employs a Jetson TX2, an advanced computation unit that provides significant computational power, integrated directly onto the vehicle. The choice of hardware components for these testbeds reflects the careful balance between the scale of the vehicles, the computational requirements of the tasks, and the physical constraints imposed by the hardware.

4. Computation Schemes: The CPM Lab provides support for a range of computation schemes, allowing parallel, sequential, and hybrid computations. This flexibility accommodates various task requirements and optimizes system performance by enabling the simultaneous processing of tasks or sequential execution as per the task dependencies.

Duckietown also accommodates a variety of computation schemes through its API, supporting both parallel and sequential computations. Moreover, it facilitates both onboard and offboard computation, further broadening the scope of its computation strategies.

An edge computing station in CHARTOPOLIS is used to extend the testbed's computational capabilities beyond the computation onboard the vehicles when needed for tasks such as platooning and Vehicle-to-Infrastructure (V2I) communication.

F1TENTH primarily focuses on onboard computation, potentially constraining its computation schemes to those suited for real-time, vehicle-based processing.

5. Human-robot Interaction: The CPM Lab, CHARTOPOLIS, IDS3C, and the Cambridge Minicar testbed offer capabilities to incorporate humans in the loop. Human interaction with these platforms is implemented in various ways, each with its own set of limitations. The specific methodologies and constraints depend on factors such as the overall system design, the complexity of tasks, and the degree of human involvement required.

6. Different Kinds of Vehicles: The CPM Lab, Duckietown, Robotarium, F1TENTH, and CHARTOPOLIS testbeds utilize different vehicle or robot models. IDS3C includes drones in the testbed environment, thereby introducing an aerial dimension.

7. Vehicle Count: The CPM Lab, IDS3C, Robotarium, and Cambridge Minicar testbeds each offer approximately 20 vehicles for experimentation. This sizable fleet enables the exploration of complex interactions and emergent behaviors in multi-vehicle scenarios. In contrast, the vehicle count in Duckietown and F1TENTH varies based on the specific sceanrio. The flexible count in these testbeds allows for a customizable experimental setup, offering scalability based on the demands of the study being conducted.

D. Act-based Information

The vehicles in the testbeds surveyed in this article are either differentially-driven, such as the Duckiebots (Duckietown) and the GRITSBots (Robotarium), or have an Ackermann steering geometry, such as the µCars (CPM Lab) and the F1TENTH vehicles. For vehicles with Ackermann-steering geometry, there are many common models which differ in accuracy and complexity. In this overview, we consider the parameters for the point-mass model and the kinematic single-track model, which is based on geometric parameters of a vehicle. It captures the nonlinearity of the vehicle's motion while being simple to parameterize. The parameters for the point-mass model, the nonholonomic differential-drive model, and the kinematic single-track model of the presented testbeds are displayed on the webpage [21].

IV. Conclusions

This survey provides a detailed overview of small-scale CAV/RS testbeds, with the aim of helping researchers in these fields to select or build the most suitable testbed for their experiments and to identify potential research focus areas. We structured the survey according to characteristics derived from potential use cases and research topics within the sense-plan-act paradigm. Through an extensive investigation of 17 testbeds, we have evaluated 56 characteristics and have made the results of this analysis available on our webpage [21]. We invited the testbed creators to assist in the initial process of gathering information and updating the content of this webpage. This collaborative approach ensures that the survey maintains its relevance and remains up to date with the latest developments. The ongoing maintenance will allow researchers to access the most recent information.

In addition, this paper can serve as a guide for those interested in creating a new testbed. The characteristics and overview of the testbeds presented in this survey can help identify potential gaps and areas for improvement. One ongoing challenge that we identified with small-scale testbeds is the enhancement of their ability to accurately map to real-world conditions, ensuring that experiments conducted are as realistic and applicable as possible. Overall, this paper provides a resource for researchers and developers in the fields of connected and automated vehicles and robot swarms, enabling them to make informed decisions when selecting or replicating a testbed and supporting the advancement of testbed technologies by identifying research gaps.

References

[1] M. Kloock, P. Scheffe, J. Maczijewski, A. Kampmann, A. Mokhtarian, S. Kowalewski, and B. Alrifaee, "Cyber-Physical Mobility Lab: An Open-Source Platform for Networked and Autonomous Vehicles," in *2021 European Control Conference (ECC)*, Jun. 2021, pp. 1937–1944.

[2] A. Stager, L. Bhan, A. Malikopoulos, and L. Zhao, "A Scaled Smart City for Experimental Validation of Connected and Automated Vehicles," *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 130–135, Jan. 2018.

[3] 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)*. IEEE, May 2017, pp. 1497–1504.

[4] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The Robotarium: Globally Impactful Opportunities, Challenges, and Lessons Learned in Remote-Access, Distributed Control of Multirobot Systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, Feb. 2020.

[5] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1TENTH: An Open-source Evaluation Environment for Continuous Control and Reinforcement Learning," *Proceedings of Machine Learning Research*, vol. 123, pp. 77–89, Apr. 2020.

[6] N. Hyldmar, Y. He, and A. Prorok, "A Fleet of Miniature Cars for Experiments in Cooperative Driving," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019, pp. 3238–3244.

[7] A. Carron, S. Bodmer, L. Vogel, R. Zurbrgg, D. Helm, R. Rickenbach, S. Muntwiler, J. Sieber, and M. N. Zeilinger, "Chronos and CRS: Design of a miniature car-like robot and a software framework for single and multi-agent robotics and control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 1371–1378.

[8] S. Kannapiran and S. Berman, "Go-CHART: A miniature remotely accessible self-driving car robot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2020, pp. 2265–2272.

[9] S. S. Ulhas, A. Ravichander, K. A. Johnson, T. P. Pavlic, L. Gharavi, and S. Berman, "CHARTOPOLIS: A Small-Scale Labor-art-ory for Research and Reflection on Autonomous Vehicles, Human-Robot Interaction, and Sociotechnical Imaginaries," *Workshop on Miniature Robot Platforms for Full Scale Autonomous Vehicle Research, IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[10] S. Wilson, R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. Bertozzi, and S. Berman, "Pheeno, A Versatile Swarm Robotic Research and Education Platform," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 884–891, Jul. 2016.

[11] N. Buckman, A. Hansen, S. Karaman, and D. Rus, "Evaluating Autonomous Urban Perception and Planning in a 1/10th Scale MiniCity," *Sensors*, vol. 22, no. 18, p. 6793, Sep. 2022.

[12] T. Tiedemann, L. Schwalb, M. Kasten, R. Grotkasten, and S. Pareigis, "Miniature Autonomy as Means to Find New Approaches in Reliable Autonomous Driving AI Method Design," *Frontiers in Neurorobotics*, vol. 16, p. 846355, Jul. 2022.

[13] A. Schwab, L.-M. Reichelt, P. Welz, and J. Lunze, "Experimental Evaluation of an Adaptive Cruise Control and Cooperative Merging Concept," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*. Montreal, QC, Canada: IEEE, Aug. 2020, pp. 318–325.

[14] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[15] S. Graham, G. Baliga, and P. Kumar, "Abstractions, Architecture, Mechanisms, and a Middleware for Networked Control," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1490–1503, Jul. 2009.

[16] Crenshaw, Tanya L. and Beyer, Steven, "UPBOT: a testbed for cyber-physical systems," in *CSET'10: Proceedings of the 3rd international conference on Cyber security experimentation and test*. USENIX Association, 2010, pp. 1–8.

[17] J. Dong, Q. Xu, J. Wang, C. Yang, M. Cai, C. Chen, Y. Liu, J. Wang, and K. Li, "Mixed Cloud Control Testbed: Validating Vehicle-Road-Cloud Integration via Mixed Digital Twin," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2723–2736, Apr. 2023.

[18] F. Caleffi, L. d. S. Rodrigues, J. d. S. Stamboroski, B. V. Rorig, M. M. C. d. Santos, V. Zuchetto, and . B. Raguzzoni, "A systematic review

of hardware technologies for small-scale self-driving cars," *Ciência e Natura*, vol. 45, no. esp. 1, pp. 84 071–84 071, Oct. 2023.
[19] A. Jimnez-Gonzlez, J. R. Martinez-de Dios, and A. Ollero, "Testbeds for ubiquitous robotics: A survey," *Robotics and Autonomous Systems Magazine*, vol. 61, no. 12, pp. 1487–1501, Dec. 2013.
[20] T. Samak, C. Samak, S. Kandhasamy, V. Krovi, and M. Xie, "AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education," *Multidisciplinary Digital Publishing Institute*, vol. 12, no. 3, p. 77, Jun. 2023.
[21] Armin Mokhtarian, Patrick Scheffe, Maximilian Kloock, Simon Schfer, Heeseung Bang, Viet-Anh Le, Sangeet Ulhas, Johannes Betz, Sean Wilson, Amanda Prorok, Spring Berman, Liam Paull, Bassam Alrifaee, "A Survey on Small-scale Testbeds," www.cpm-remote.de/testbeds, 2023.
[22] B. Chalaki, L. E. Beaver, A. I. Mahbub, H. Bang, and A. A. Malikopoulos, "A Research and Educational Robotic Testbed for Real-Time Control of Emerging Mobility Systems: From Theory to Scaled Experiments [Applications of Control]," *IEEE Control Systems Magazine*, vol. 42, no. 6, pp. 20–34, Dec. 2022.
[23] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1699–1706.
[24] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, Jun. 2017, pp. 719–726.
[25] J. Betz, H. Zheng, Z. Zang, F. Sauerbeck, K. Walas, V. Dimitrov, M. Behl, R. Zheng, J. Biswas, V. Krovi *et al.*, "Teaching autonomous systems hands-on: Leveraging modular small-scale hardware in the robotics classroom," *arXiv preprint arXiv:2209.11181*, 2022.
[26] Duckietown Foundation, "Duckietown MOOC," https://www.duckietown.org/mooc, 2023.
[27] A. Mokhtarian, L. Hegerath, and B. Alrifaee, "CPM Academy: A Remote Platform for Teaching Current Topics in Connected and Automated Vehicles," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, Dec. 2023, pp. 8894–8900, iSSN: 2576-2370.
[28] A. Mokhtarian, P. Scheffe, S. Kowalewski, and B. Alrifaee, "Remote Teaching with the Cyber-Physical Mobility Lab," *IFAC-PapersOnLine*, vol. 55, no. 17, pp. 386–391, 2022.
[29] F1TENTH, "Racing Competition," https://f1tenth.org/race.html, 2023.
[30] Duckietown Foundation, "AI Driving Olympics," https://www.duckietown.org/research/ai-driving-olympics, 2023.
[31] A. Mokhtarian, S. Schäfer, and B. Alrifaee, "CPM Olympics: Development of scenarios for benchmarking in networked and autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 9–15.
[32] "Donkeycar: a python self driving library," https://github.com/autorope/donkeycar, 2017, GitHub.
[33] A. Mokhtarian and B. Alrifaee, "CPM Remote: A Remote Access to the CPM Lab," in *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, May 2022, pp. 1124–1129.
[34] R. M. Zayas, L. E. Beaver, B. Chalaki, H. Bang, and A. A. Malikopoulos, "A digital smart city for emerging mobility systems," in *2022 2nd Annual International Conference on Digital Twins and Parallel Intelligence (DTPI)*. IEEE, 2022, pp. 1–6.
[35] 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.
[36] M. Chevalier-Boisvert, F. Golemo, Y. Cao, B. Mehta, and L. Paull, "Duckietown environments for OpenAI Gym," https://github.com/duckietown/gym-duckietown, 2018, GitHub.
[37] "The F1TENTH Gym," https://github.com/f1tenth/f1tenth_gym, GitHub.
[38] B. Chalaki, L. E. Beaver, and A. A. Malikopoulos, "Experimental validation of a real-time optimal controller for coordination of CAVs in a multi-lane roundabout," in *2020 31st IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 504–509.
[39] L. E. Beaver, B. Chalaki, A. M. Mahbub, L. Zhao, R. Zayas, and A. A. Malikopoulos, "Demonstration of a Time-Efficient Mobility System Using a Scaled Smart City," *Vehicle System Dynamics*, vol. 58, no. 5, pp. 787–804, 2020.