A Systematic Literature Review on Characteristics, Challenges, and Limitations of Simulation Environments for Autonomous Driving Systems

Xiao'ao Song[a], Neeraj Kumar[a]

[a]*University of Zurich, Switzerland*

Abstract

Autonomous Driving Systems (ADSs) are systems that are able to collect, analyze, and leverage sensor data from the surrounding environment to control physical actuators at run-time. The interaction of ADSs with the other objects, humans, and other systems, makes the nature and effect of faults in ADSs non-predictable. In this context, industrial and research organizations are increasingly relying on simulation platforms to facilitate the testing of ADSs, to ensure their reliability and safety in such complex environments. However, while the availability of ADSs simulation platforms enable cost-efficient evaluations of ADSs behaviour early on in the development cycle throughout virtual tests, there are no objective quality metrics to discriminate between the different ADSs simulators. The specified boundary values to consider for selecting the right simulators for testing a specific ADS depends from different standards and metric sets. This paper investigate the metrics and factors that in the ADSs sector can be used as quality attributes to characterize ADSs simulation environments for testing purposes. Hence this paper focus on providing an overview of (i) metrics and boundary values discriminating different ADSs simulation environments; (ii) how the metrics match quantitative and qualitative standards; (iii) which quality attributes are addressed by existing ADSs simulation environments; and (iv) what aspect need to be supported to enhance simulation environments for ADSs testing purposes. Our findings from analyzing 1584 papers include a catalog of 11 metric, 35 subcategories of which 25 subcategories define boundary values between different simulators. Most of the identified metrics were useful to discriminate between different ADSs testing and simulation environments. Interestingly, while studying ADSs simulation environments quality metrics, we observed different and complementary aspects where such environments need to be improved or are missing, which is of critical importance for CPSs developers and experts

Keywords: Autonomous systems, Simulation Environments, Quality attributes, Systematic literature review.

1. Introduction

Cyber-physical Systems (CPSs) are systems in which algorithms analyze sensor data collected from the surrounding environment to control physical actuators [1, 2]. Emerging CPS—from medical monitoring systems and devices, industrial robots, and autonomous transportation systems—are expected to play a crucial role in future generations' quality of life and the global economy [3]. These CPSs typically interact with humans as well as other systems in highly dynamic and unpredictable environments. Thus, guaranteeing CPSs' reliability and safety are critical challenges to address, primarily when CPSs operate in critical scenarios, such as driving in traffic or detecting wildfires [4].

Among emerging CPSs application domains, the usage of autonomous systems such as autonomous driving systems (ADSs) is expected to have a profound impact on our society. Human errors cause more than 90% of accidents (e.g., driving while under the influence of alcohol, fatigue, and other distractions) [5]. ADSs have the potential to reduce such errors, which

Email addresses: `xiaoao.song@uzh.ch` (Xiao'ao Song), `neeraj.kumar@uzh.ch` (Neeraj Kumar)

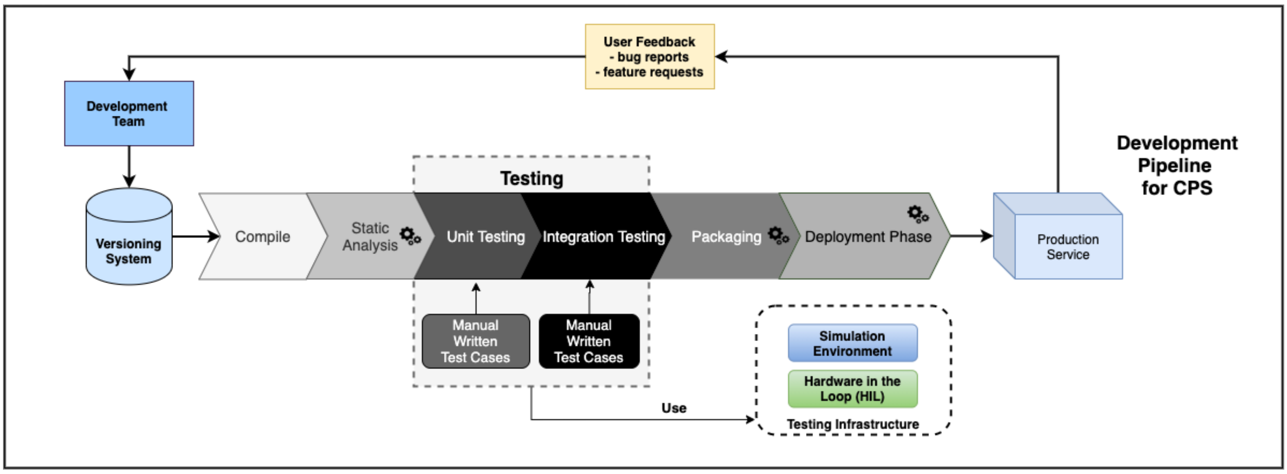

Figure 1: (Idealized) Development pipelines for ADSs, and in general CPSs.

would eliminate most accidents. However, recently reported fatal crashes involving self-driving cars suggest that, at least in the short-term, the usage of ADSs in real-society appear to be too optimistic [1]. The main factor limiting their usage is the lack of tools able to avoid the releasing of ADSs equipped with defective software, which might become erratic and lead to fatal crashes, involving also humans [6].

1.1. Problem statement and objective

Current *ADSs development practices have several limitations and drawbacks*, including (i) the limited ability to repeat tests under the same conditions due to ever-changing environmental factors [7]; (ii) the difficulty to test the systems in safety-critical scenarios, with the goal to avoid irreversible damage causes by dreadful outcomes [6, 8, 9]; (iii) not being able to guarantee the system's reliability and safety in its operational design domain due to a lack of testing under a wide range of execution conditions [5]. Consequently, new development practices, environments, and frameworks are needed to address the fundamental development challenges of *observability*, *testability*, and *predictability* of ADSs and support their foreseen wide-spread adoption.

Simulation-based testing of ADSs. ADSs require flexible development and verification strategies to account for Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), and Hardware-in-the-Loop (HiL) paradigms [10]. ADSs are also more difficult and expensive to test and integrate than traditional software systems [11, 4]. Common reasons for this are that the final version of the hardware is often available only late, and integrating hardware components requires a great deal of manual effort. A typical approach to dealing with ADSs safety and testing requirements is to develop *hardware proxies*, such as system simulators and digital twins [12] (see Figure 1). ADSs safety requirements are nowadays evolving under the direction of organizations responsible for software system and development standardization practices and policies. According to these organizations, simulations can facilitate testing of ADSs safety requirements, as they are inexpensive and less dangerous than running the systems in real-life [13, 14]. Consequently, the development and validation of ADSs heavily rely on them.

1.2. Contribution

We argue that simulators need to meet stringent functional and non-functional requirements (e.g., accuracy, efficiency, photo-realism) to be beneficial for developing ADSs. Additionally, they must also support the systematic generation of test scenarios to effectively assess the behavior of ADSs in nominal and critical conditions [15, 16].

This paper makes a case for soft-body simulation as a necessary complement to mainstream rigid-body simulations and remarks on the importance of enabling automation in both test generation and execution also in the context of ADSs However, while the availability of ADSs simulation platforms enable cost-efficient evaluations of ADSs behaviour early on in the development cycle throughout virtual tests, there are no objective quality metrics to discriminate between the different

ADSs simulators. The specified boundary values to consider for selecting the right simulators for testing a specific ADS depends from different standards and metric sets.

This paper investigate the metrics and factors that in the ADSs sector can be used as quality attributes to characterize ADSs simulation environments for testing purposes. In detail, the main contributions of this paper are:

1. An overview of metrics and boundary values discriminating different ADSs simulation environments
2. A discussion on how the identified metrics match quantitative and qualitative standards;
3. A discussion on the quality attributes are addressed by existing ADSs simulation environments;
4. A discussion on the aspects that need to be supported to enhance simulation environments for ADSs testing purposes.

2. Background and Related Work

2.1. Development and Evolution of Cyber-physical Systems

Empirical studies have shown that CPSs such as ADSs are more difficult and expensive to test and integrate than traditional software systems [11, 4]. Common reasons for this are that the final version of the hardware is often not available, and the integration of hardware components requires a high, and error-prone manual effort. Thus, recent studies have investigated the challenges of CPSs development and identified that an effective evolution of CPSs requires more flexible development and verification approaches, integrating Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), and Hardware-in-the-Loop (HiL) paradigms [10].

Giraldo *et al.* [17] conducted a literature review on CPSs research topics, finding that they can be categorized into security, privacy, defense, or domain-specific.

A follow-up study by Törngren and Sellgren [4] discusses how CPSs' engineering deals with the inner complexity of CPSs' design and the challenges that arise from the environments in which CPSs operate. According to Törngren and Sellgren, while semi-automated integration happens through software, there are distinguishing characteristics between software and physical systems that make it hard co-designing hardware and software. Those characteristics entail the usage of different approaches, techniques, abstractions, platforms, faults & failure modes, and development practices [4]. Törngren and Sellgren conclude that CPS development and testing need rapid prototyping, code/test generation, and various testing phases [18] encapsulating model-in-the-loop (MiL), software-in-the-loop (SiL), and hardware-in-the-loop (HiL) activities to effectively identify bugs in CPSs.

A recent work by Garcia *et al.* [19] investigated the bugs affecting two autonomous vehicles (AV) simulation tools. Specifically, they investigated the frequency, root-causes, symptoms, and location (*e.g.,* components) of bugs affecting such systems. To the best of our knowledge, our study represents the first work providing a taxonomy of metrics concerning quality aspects of ADSs simulation environments, which can help developers identifying the root causes of the mistakes made by developers while developing ADSs.

3. Research Design

In this paper, we present a systematic literature review on proposed software simulation environments for testing autonomous vehicles, thus gathering information on the various technological and commercial aspects and supported configurations of these environments. Specifically, we have studied the multi-platform support, documentation, scalability, application domains, configuration limitations, and hardware support of identified industrial and academic simulators. It is important to note that autonomous driving is a very extensive field with wide range of applications (robotics, underwater vehicles, military, aeronautics, space vehicles, etc.). In our study we limited our scope to simulators studied and proposed in the scientific literature body.

To conduct our investigation, we have done a literature review of characteristics, challenges, and limitations of Simulation Environments for Autonomous Driving Systems including identification, analysis, interpretation and comparison of the results. The systematic literature review was conducted by performing five phases:

- *Preparation*: Research questions are formulated and corresponding keywords (or strings) are selected for the search.
- *Data collection and Dataset cleaning*: We collected research papers from multiple sources, by performing a a systematic keyword search among academic sources, including DBLP[1], CSBIB[2] and Semantic Scholar[3]. We removed duplicates, i.e., potential cases in which same papers were selected from various sources. Moreover, paper are classified as relevant or not relevant by using inclusion and exclusion criteria.
- *Papers Selection*: Relevant papers were selected by using inclusion and exclusion criteria explained in the table.
- *Papers Analysis & Categorization*: Relevant papers are qualitatively analyzed and categorized in topics, based on the designed research questions.

The overall research and analysis process pipeline shown in Figure 2 is explained in more details in subsequent sections.

3.1. Research objectives and research questions

In the context of this paper we focused on addressing the following research questions.

- **RQ$_1$**: *What quality and quantity attributes characterize different ADS simulation enviroments?*
- **RQ$_2$**: *What types and domains characterize ADS simulation environments?*
- **RQ$_3$**: *Which boundary values exist among quality and quantity attributes of ADS simulation enviroments?*

3.2. Data collection procedures

We have followed a multistage process to collect relevant research papers, as illustrated in the data collection pipeline in Figure 3. The data collection was conducted via keyword-based web search on academic repositories such as DBLP, CSBIB and Semantic Scholar. We considered such sources, since they comprehensively contain updated information on journal and conference papers from various disciplines relevant to our investigation (e.g., computer science and robotic simulation). Thus, a keyword-search was applied on titles and abstracts of articles of these sources, in order to eliminate duplicates and irrelevant results. The subsequent sections provide details on the search procedure.

3.3. Preparation

In the study preparation phase, we focus on the definition of the research questions, the choice of relevant literature sources, the development of search queries strings and criteria to exclude of include papers, according to our study objectives.

3.4. Query string construction and automated search

We defined the search strings to manually search for publications. The goal of these search string was to get a set of publications dealing with metrics in Autonomous Driving Software Simulators.We used the following approaches to refine the search:

- *Inclusion of reference publications* : We collected a set of reference publications before conducting the data collection. One useful criteria for checking the appropriateness of a search is to find if the result set contains the expected reference publications. If a particular search string results in a set which does not contain some expected publications, we revised the search string.

[1] https://dblp.org/
[2] https://liinwww.ira.uka.de/bibliography/
[3] https://www.semanticscholar.org/

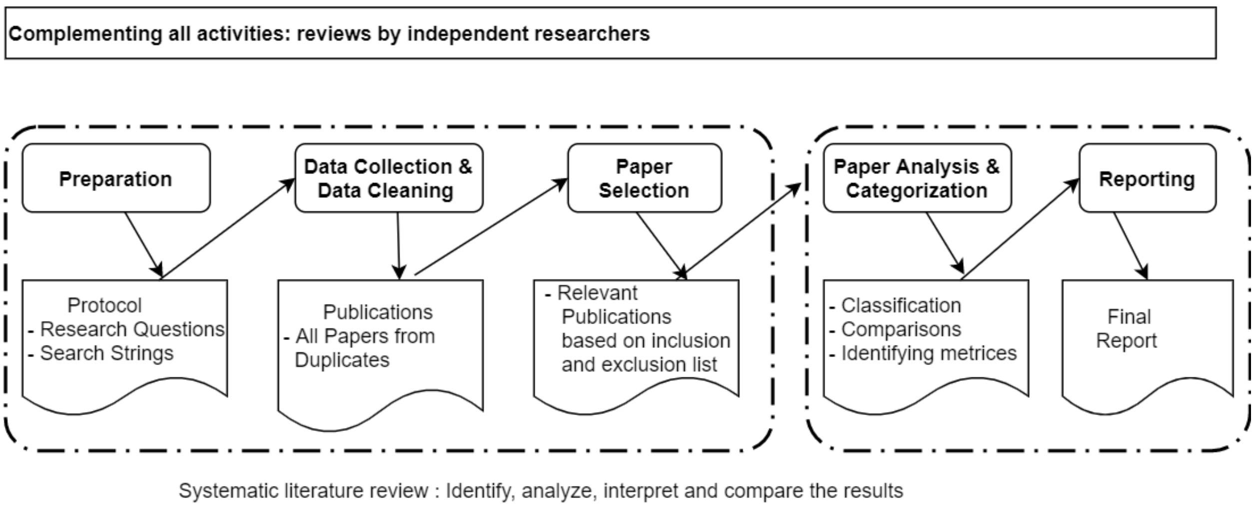

Figure 2: Overview of the multi staged systematic literature review method to identify, analyze, interpret and compare the results

- *Checking the Result Set*: Visually inspected the search results to ensure sufficient precision; for instance, if searching a particular domain, how many hits are outside the domain of interest?
- Timelines Constraint: We discarded during the keyword-search papers published before 2015, thus, considering papers from 2015 and 2020 (included).

Table 1 shows the concluded set of keywords used to generate the initial data set. The search yielded 1584 papers in total, across 8 domains. Furthermore, a thorough analysis of abstracts, titles and journals was carried out in order to eliminate duplicates and results that were found in more than one database, leading to the elimination of 580 repetitions, some of which were even found 3 times, reflecting the thoroughness of the search, delivering a total amount of 1004 unique publications.

ID	Search string	Papers Found
1	Robotic Simulations	241
2	eHealth Simulations	324
3	aviation Simulations	280
4	Carla Autonomous Driving	278
5	Gazebo Autonomous Driving	56
6	Cyber-physical systems simulation	393
7	AirSim	8
8	Manually Specified	4

Table 1: Search string and Papers Found

3.5. Inclusion and exclusion criteria, search execution, data collection and extraction, and evaluation

To select the papers for the study, we defined the inclusion and exclusion criteria shown in Table 2. Paper selection process is illustrated in Figure 4. All papers were evaluated based on above mentioned inclusion and exclusion criteria. The evaluation of the result set was independently performed by two researchers using the inclusion and exclusion criteria. Each one of 1004 papers were evaluated for inclusion or exclusion in the study,votes of both reviewers were recorded, and 56 papers were accepted for further analysis. There were 13 cases of conflicts between the researcher's evaluation, requiring further discussion with a bigger team to decide the inclusion of these papers in the study.

A data table was extracted from these 56 papers containing information regarding general details(year,Venue,journal) of the paper.

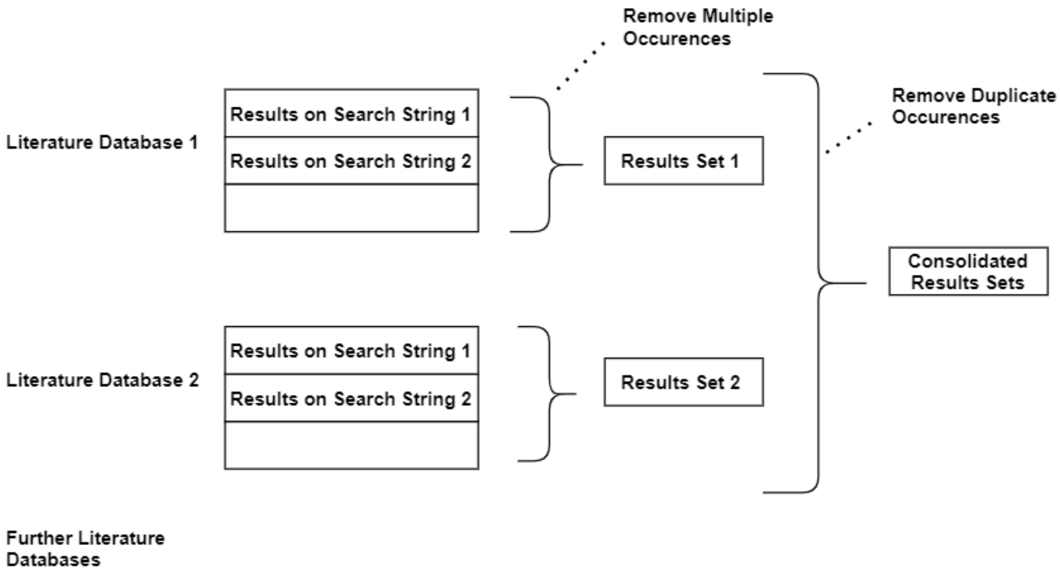

Figure 3: Multistage procedure of step wise integration and cleaning of literature databases.

Nu	Exclusion and Inclusion Criteria
IC1	The Paper discusses the Autonomous Driving Simulation Software
IC1	The Paper talks about important algorithmic implementations in Autonomous Driving Simulation Software
EC1	The Paper is not in English.
EC2	The paper publication date is outside the considered timelines.
EC3	The Paper is not in domain of Autonomous Driving Testing Simulation? E.g., refer to simulators, but not for ADSs
EC4	The Paper is not available for download
EC5	There are multiple occurrences of the paper in the test set.

Table 2: Exclusion and Inclusion Criteria

3.6. Identification of Relevant Papers

We selected 31 papers out of these 56 papers as final datasets for our study by going through the complete content of these papers and marking the paper as relevant or not relevant by using the criteria defined in table 3.

Output Label	Relevant and Non Relevant Criteria
Relevant	The discussed software simulator is popularly used in academics or industry.
Relevant	The paper describes the effectiveness, challenges, application potentials of software simulation in autonomous driving.
Relevant	The Paper discusses current state-of-the-art on deep learning,computer vision, reinforcement learning algorithms in autonomous driving and provides an insight into the strengths and limitations of deep learning and AI approaches and design choices.
Non-Relevant	The paper relates to the topic in its related work only
Non-Relevant	The discussed application use case is outside the scope of autonomous driving simulation.

Table 3: Criteria for identifying paper as relevant or not relevant

3.7. Analysis procedures

We extracted a data table from the reviewed papers containing information regarding supported configuration, features, ap-

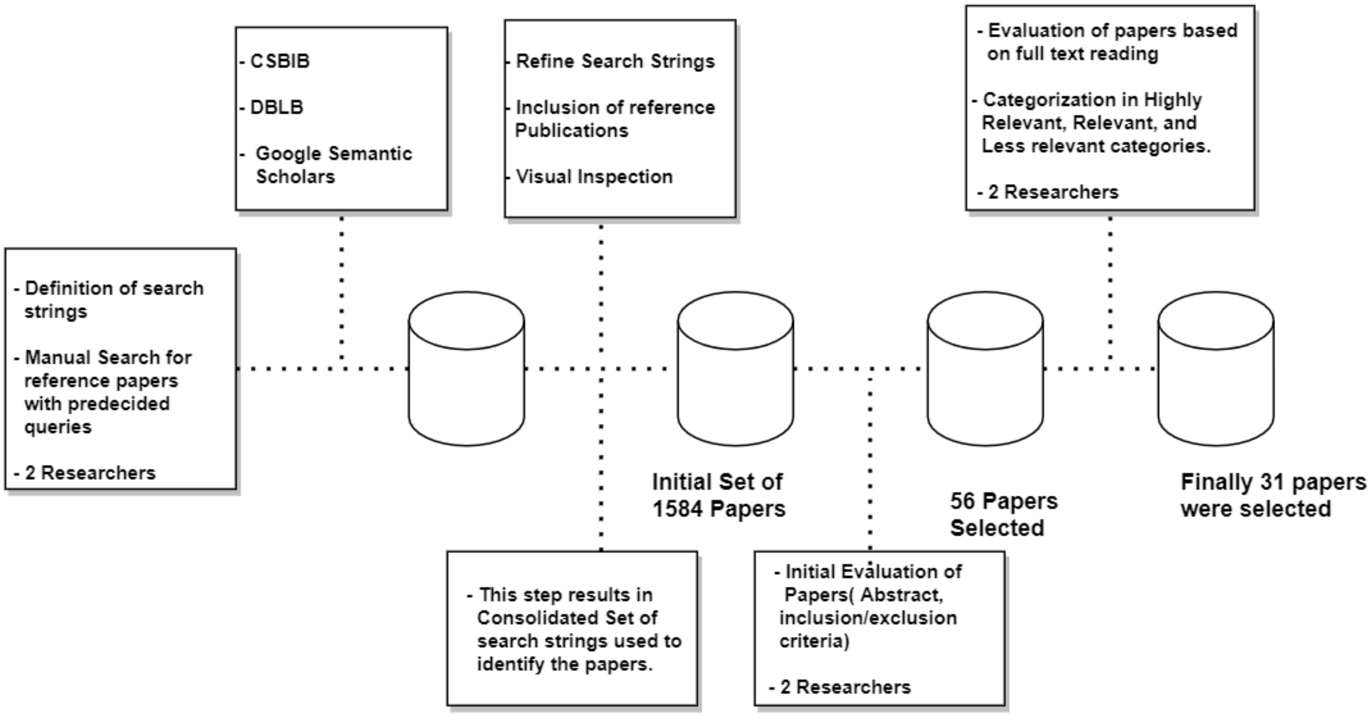

Figure 4: Data collection method implemented in the literature review

plication use cases, source code availability and personalization support. On the basis of this extracted data, we create a feature map of considered tools and use this feature map to compare the reviewed tools for different applications.

3.8. Analysis of publications over time

Using this method, we can get a good insight about the origin and development of a research field over time. As shown in Figure 5, we can observe a continuous increase in the topic of interest. Even though our data corresponds to limited time span(2015 to 2020), we can see the number of published papers are continuously increasing through this period.

3.9. Venue analysis

A count analysis was conducted to analyse the distribution of papers among Journals, Conference Paper or Online repository. As shown in Figure 6, highest number of papers belongs to conference papers, followed by Journal and Online Repositories.

3.10. Citation analysis

Citation analysis was conducted as it is a well-established procedure and one of the most effective tools to analyze research performance and the the impact of published articles in the field. Please refer to Table 4.

4. Study Results

4.1. Overview of the result set

Please refer to the Tables 5, 6, 7, 8 and 9 to get a summary of main features of simulation platforms used in Autonomous Driving. In section 3.1, we posed three research questions and following are the corresponding findings:

RQ1: The first research question was concerned with the categorization of attributes for different ADS simulation environments reported in literature. Our systematic review of 31 papers yielded 11 different metrics with 35 subcategories.

RQ2: The second research question was concerned with the types and domains that characterize ADS simulation environ-

Cites	Authors	Title	Year	Source
1495	Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, Laura Bieker	Recent Development and Applications of SUMO - Simulation of Urban MObility	2012	International Journal On Advances in Systems and Measurements
976	Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun	CARLA: An Open Urban Driving Simulator	2017	Conference on Robot Learning(CoRL)
549	Shital Shah, Debadeepta Dey, Chris Lovett, Ashish Kapoor	AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles	2017	Field and Service Robotics conference(FSR)
96	Jesse A. Grimes, Jonathan W. Hurst	THE DESIGN OF ATRIAS 1.0 A UNIQUE MONOPOD, HOPPING ROBOT	2012	Climbing and Walking Robots and the Support Technologies for Mobile Machines
91	Iker Zamora, Nestor Gonzalez Lopez, Victor Mayoral Vilches, Alejandro Hernandez Cordero	Extending the OpenAI Gym for robotics: a toolkit for reinforcement learning using ROS and Gazebo	2016	arXiv preprint arXiv:1608.05742
88	Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, Gigel Macesanu	A Survey of Deep Learning Techniques for Autonomous Driving	2019	Journal of Field Robotics
62	Francisca Rosique, Pedro Javier Navarro Lorente, Carlos Fernandez, Antonio Padilla	A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research	2019	Sensors 19(Perception Sensors for Road Applications):648

Table 4: Top 7 citations

ments. In our study, we identified 2 metrics with 5 subcategories to characterize different simulation environments.

RQ3: The third research question was concerned with the boundary values defined for the different literature review metrics. In total, boundary values were defined for 23 out of 35 subcategories.

5. Threats to Validity

5.1. Limitations and Strengths

This section describes the limitation and strengths of the study that should be taken into account while interpreting the existing results. Firstly, our search is limited to papers published in English and relevant literature published in other languages was excluded.Secondly we took into consideration papers from last five years and only three databases were used to help searching articles. Some of the published work may have been be ignored, as they are not included in the searched databases. Main Strength of this study comes from the fact that it used an extensive search strategy to locate papers and rigorously screened papers through well-defined inclusion/exclusion criteria. Second, the quality of the included papers was assessed in a standardized and reproducible way.

6. Conclusion

In this paper, we presented findings from a systematic literature review in which we analyzed 31 papers for metrics used in automotive driving testing simulation. We categorized the simulator tools across 11 metrics and 35 subcategories and devel-

Simulator	License	GPU	RAM	Disk Memory	Supported OS
AirSim	GPL/Open Source	4GB minimum	8GB	50-80GB (with UE4)	Windows, Linux, Mac
Carla	GPL/Open Source	4GB minimum	8GB	50-80GB (with UE4)	Windows, Linux
BeamNG	Restricted: Graphical engine(torque 3D) is open source; Physics engine is not	Minimum: Radeon HD 7750 / Nvidia GeForce GTX 550 Ti	8GB	15GB	Windows
Gazebo	GPL/Open Source	A dedicated GPU is recommended	3GB	30GB for Windows; 500 MB for Ubuntu	Windows, Linux, Mac
UdaCity	GPL/Open Source	N/A	N/A	N/A	Windows, Linux, Mac
DeepDrive	GPL/Open Source	CUDA capable GPU	8GB	10GB	Linux
Sumo	GPL/Open Source	N/A	N/A	N/A	Windows, Linux, Mac
USASSim	GPL/Not Open Source	N/A	N/A	512MB	Windows, Linux, Mac
Webots	GPL/Open Source	An NVIDIA or AMD OpenGL (minimum version 3.3) capable graphics adapter with at least 512 MB of RAM is required	N/A	2GB	Windows, Linux, Mac
Flightmare	GPL/Open Source	N/A	N/A	30GB for Windows; 500 MB for Ubuntu	Windows, Linux, Mac
Matlab AV Driving Toolbox	Restricted	Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended.	4GB	28GB	Windows, Linux, Mac
Baidu Apollo	GPL/Open Source	NVIDIA GPU	8GB (16GB for Apollo 3.5 and above)	N/A	Linux(Ubuntu)

Table 5: Summary of main features of simulation platforms used in Autonomous Driving - Part A

Simulator	Vertical Scaling	Horizontal Scaling	Has Dockerized version	Ground Vehicles	UAV	Support MultiAgents	Data Recording
AirSim	No	No	Yes	Car; user needs to specify the vehicle type while creating multiple agents.	Quadrotor, 3 vehicle types are supported: arducoptersolo, simpleflight, arducopter	Yes	Yes
Carla	No	No	Yes	Car; custom car models can be added	N/A	Yes	Yes
BeamNG	N/A	N/A	No	Car, support customized Vehicles	Quadrotor	Yes	Yes
Gazebo	Yes	Yes	Yes	Car	Quadrotor	Yes	Yes
UdaCity	No	No	Yes	Car	N/A	No	Yes
DeepDrive	No	No	Yes	Car	N/A	No	Yes
Sumo	Yes	Yes	Yes	Car	N/A	Yes	Yes
USASSim	Yes	Yes	No	Car	N/A	Yes	Yes
Webots	Yes	Yes	No	Different Robot Shapes	N/A	Yes	Yes
Flightmare	Yes	N/A	No	N/A	Quadrotor	Yes	Yes
Matlab AV Driving Toolbox	N/A	N/A	No	Car	N/A	Yes	Yes
Baidu Apollo	N/A	Yes	Yes	Support customized vehicles	N/A	Yes	Yes

Table 6: Summary of main features of simulation platforms used in Autonomous Driving - Part B

Simulator	Hardware in Loop	API support	Programmable simulations	3D Rendering Engine	ROS Support	Gravity Support	Weather Conditions	Pedestrian Simulation
AirSim	Yes(PX4)	Python, C++	Yes	Unreal	Yes	Yes	8 conditions	No
Carla	No	Python, C++	Yes	Unreal	Yes	Yes	14 conditions	Yes
BeamNG	No	Python	Yes	Torque 3D	No	Yes	No(future feature)	No(future feature)
Gazebo	Yes	C++	Yes	Ogre3D(OpenGL)	Yes	Yes	No	No
UdaCity	No	Python, C++	Yes	Unity	Yes	No	No	No
DeepDrive	NA	Python	Yes	Unreal	Yes	No	No	No
Sumo	No	Python	No	No rendering engine	No	No	No	Yes
USASSim	No	Real Engine Scripts	Yes	Unreal	No	No	No	No
Webots	N/A	C, C++, Java, Python,MATLAB	Yes	OpenGL	Yes	No	No	No
Flightmare	No	Python	Yes	Unity	Yes	N/A	N/A	NO
Matlab AV Driving Toolbox	Yes	Matlab, C, C++	Yes	Unreal Engine	Yes	No	No	No
Baidu Apollo	Yes	Python	Yes	Unity	Yes	No	3 conditions	Yes

Table 7: Summary of main features of simulation platforms used in Autonomous Driving - Part C

Simulator	Static Objects Support	Dynamic Objects Support	Maximum Supported Dynamic Objects	Frequency	Daytime Support	Used in Academia	Used in Industry
AirSim	Yes	Yes	More than 20 drones	1000Hz; 50Hz in Python API communication	Yes	Yes	Yes
Carla	Yes	Yes	200	30fps in rendering mode, higher in non rendering mode	Controlled through weather settings	Yes	Yes
BeamNG	Yes	Yes	6-8	No Info	Yes	No	Yes
Gazebo	Yes	Yes	255	1000Hz	No	Yes	Yes
UdaCity	Yes	No	No Info	No Info	No	Yes	No
DeepDrive	Yes	Yes	No Info	No Info	No	No Info	No Info
Sumo	Yes	Yes	No Info	10Hz	No	Yes	Yes
USASSim	Yes	Yes	No Info	No Info	No	Yes	No
Webots	Yes	Yes	No Info	25 Hz(default 10Hz)	No	Yes	Yes
Flightmare	Yes	Yes	More than 150 agents	Physics Simulation 200, 000Hz	No	Yes	Yes
Matlab AV Driving Toolbox	Yes	Yes	No Info	No Info	No	Yes	Yes
Baidu Apollo	Yes	Yes	30	100Hz(default 15Hz)	Yes	Yes	Yes

Table 8: Summary of main features of simulation platforms used in Autonomous Driving - Part D

Simulator	Data-driven (visual or others)	Soft/Rigid Body	Usage in Autonomous systems	Human or natural inspired robotics	Use in Domains other than AV simulation
AirSim	Yes	Rigid	Autonmous Driving Research, Synthetic Data Generation	No	Precision Agriculture, pathogen surveillance, weather monitoring
Carla	Yes	Rigid	Autonmous Driving Research, Synthetic Data Generation	No	a) Smart City Applications(Pedetrian, Traffic light Simulations). b) V2X(Vehicle to Everything) Simulation
BeamNG	Yes	Soft	Autonmous Driving Research, Synthetic Data Generation	No	Gaming, Soft Body Simulation Applications
Gazebo	Yes	Rigid	Robotics Simulation	Yes	Mobile Robotics Simulation for terrestrial, aerial and space robotics applications
UdaCity	No	Rigid	Autonomous Driving Research	No	Teaching
DeepDrive	Yes	Rigid	Autonomous Driving Simulation	No	No Info
Sumo	Yes	No Info	Traffic Simulation	No	No Info
USASSim	Yes	Rigid	Mobile Robotics Simulation	Yes	Mobile Robotics Simulation for terrestrial, underwater, aerial and space robotics applications.
Webots	Yes	No Info	Mobile Robotics Simulation	Yes	Mobile Robotics Simulation
Flightmare	Yes	Rigid	Quadator simulation	No	No Info
Matlab AV Driving Toolbox	Yes	Rigid	Autonmous Driving	No	No Info
Baidu Apollo	Yes	Rigid	Autonmous Driving Research, Synthetic Data Generation	No	a) Smart City Applications(Pedetrian, Traffic light Simulations). b) V2X(Vehicle to Everything) Simulation

Table 9: Summary of main features of simulation platforms used in Autonomous Driving - Part E

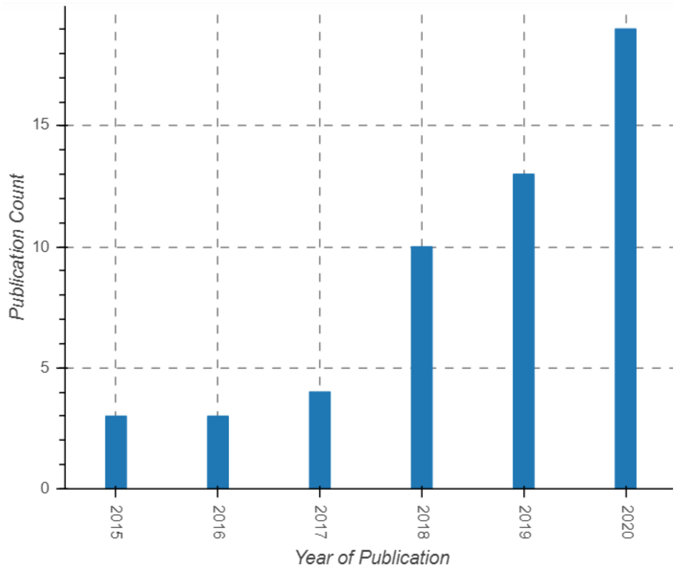

Figure 5: Analysis of Publications over time

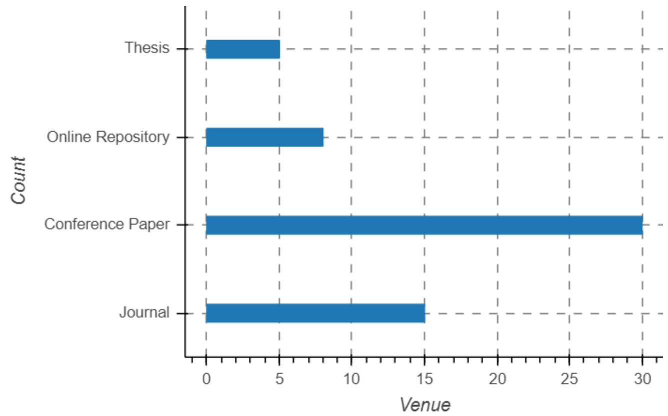

Figure 6: Analysis of Publication Venues

oped a catalog of 11 metrics including a description/definition for these metrics along with indication for tool support. The findings from our study provide an extensive data basis to be used for the selection of a simulator tool for different applications scenarios. Although the most widespread simulators used in the field of research are robotics simulators considering that autonomous vehicles are a branch of robotics, however, not all robotic simulators are prepared to provide the necessary realism required in these cases.

References

[1] R. Baheti, H. Gill, Cyber-physical systems, The impact of control technology 12 (1) (2011) 161–166.

[2] Academies of Sciences, A 21st Century Cyber-Physical Systems Education, National Academies Press, 2017.

[3] H. Chen, Applications of cyber-physical system: A literature review, Journal of Industrial Integration and Management 02 (03) (2017) 1750012.

[4] M. Törngren, U. Sellgren, Complexity Challenges in Development of Cyber-Physical Systems, Springer International Publishing, Cham, 2018, pp. 478–503.

[5] N. Kalra, S. Paddock, Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?, Transportation Research Part A: Policy and Practice 94 (2016) 182–193. doi:10.1016/j.tra.2016.09.010.

[6] T. Guardian, Self-driving uber kills arizona woman in first fatal crash involving pedestrian (2018).
URL https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe

[7] J. Kim, S. Chon, J. Park, Suggestion of testing method for industrial level cyber-physical system in complex environment, in: 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, 2019. doi:10.1109/icstw.2019.00043.
URL https://doi.org/10.1109/icstw.2019.00043

[8] The-Washington-Post, Uber's radar detected elaine herzberg nearly 6 seconds before she was fatally struck, but "the system design did not include a consideration for jaywalking pedestrians" so it didn't react as if she were a person. (2019).
URL https://mobile.twitter.com/faizsays/status/1191885955088519168

[9] F. Ingrand, Recent trends in formal validation and verification of autonomous robots software, in: 3rd IEEE International Conference on Robotic Computing, IRC 2019, Naples, Italy, February 25-27, 2019, 2019, pp. 321–328.

[10] S. Abbaspour Asadollah, R. Inam, H. Hansson, A survey on testing for cyber physical system, in: K. El-Fakih, G. Barlas, N. Yevtushenko (Eds.), Testing Software and Systems, Springer Intern. Publishing, Cham, 2015, pp. 194–207.

[11] P. Helle, W. Schamai, C. Strobel, Testing of autonomous systems - challenges and current state-of-the-art, INCOSE International Symposium (2016) 571–584.

[12] D. S. Johnson, H. Koehneman, D. LaFortune, D. Leffingwell, S. Magill, S. Mayner, A. Ofer, R. Stroud, A. Wallgren, R. Yeman, INDUSTRIAL DEVOPS - Applying DevOps and Continuous Delivery to Significant Cyber-Physical Systems, National Academies Press, 2017.
URL https://itrevolution.com/book/

industrial-devops/

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, V. Koltun, CARLA: an open urban driving simulator, in: 1st Annual Conference on Robot Learning, CoRL 2017, Vol. 78 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 1–16.
URL http://proceedings.mlr.press/v78/dosovitskiy17a.html

[14] E. Bondi, D. Dey, A. Kapoor, J. Piavis, S. Shah, F. Fang, B. Dilkina, R. Hannaford, A. Iyer, L. Joppa, M. Tambe, AirSim-w: A simulation environment for wildlife conservation with uavs, in: E. W. Zegura (Ed.), Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, COMPASS, ACM, 2018, pp. 40:1–40:12. doi:10.1145/3209811.3209880.
URL https://doi.org/10.1145/3209811.3209880

[15] A. Gambi, M. Müller, G. Fraser, Automatically testing self-driving cars with search-based procedural content generation, in: D. Zhang, A. Møller (Eds.), Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2019, Beijing, China, July 15-19, 2019, ACM, 2019, pp. 318–328. doi:10.1145/3293882.3330566.
URL https://doi.org/10.1145/3293882.3330566

[16] A. Gambi, T. Huynh, G. Fraser, Generating effective test cases for self-driving cars from police reports, in: M. Dumas, D. Pfahl, S. Apel, A. Russo (Eds.), Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2019, Tallinn, Estonia, August 26-30, 2019, ACM, 2019, pp. 257–267. doi:10.1145/3338906.3338942.
URL https://doi.org/10.1145/3338906.3338942

[17] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, M. Kantarcioglu, Security and privacy in cyber-physical systems: A survey of surveys, IEEE Design & Test 34 (4) (2017) 7–17.

[18] H. Shokry, M. Hinchey, Model-based verification of embedded software, Computer 42 (4) (2009) 53–59. doi:10.1109/MC.2009.125.

[19] J. Garcia, Y. Feng, J. Shen, S. Almanee, Y. Xia, Q. A. Chen, A comprehensive study of autonomous vehicle bugs, in: ICSE '20: 42nd International Conference on Software Engineering, Seoul, South Korea, 27 June - 19 July, 2020, ACM, 2020, pp. 385–396.