University of Zurich[UZH]

Department of Informatics

Obere Kirchgasse 2
8400 Winterthur,
Switzerland

Dr. Sebastiano Panichella
Responsible of the Cloud-native development initiative

emails: spanichella@gmail.com , panc@zhaw.ch
https://spanichella.github.io

Binzmühlestrasse 14
CH-8050 Zürich-Oerlikon
Switzerland

Prof. Dr. Davide Scaramuzza
Director of the Robotics and Perception Group

Phone +41 44 635 2409
sdavide@ifi.uzh.ch
http://rpg.ifi.uzh.ch/

Xiao'ao Song
Hagenholzstrasse 78,
8050 Zürich

Matrikel-Nr. 18-747-949
xiaoao.song@uzh.ch

Neeraj Kumar
Friesstrasse 26,
8050 Zürich

Matrikel-Nr. 19-756-303
neeraj.kumar@uzh.ch

November 27, 2020

Master's Independent Study

Title: A Systematic Literature Review on Limitations and Challenges of Simulation Environments for Autonomous Systems

Introduction

Emerging Cyber-physical Systems (CPS)—from robotics, transportation, to medical devices—will play a crucial role in the quality of life of European citizens and the future of the European economy [5]. In this context, autonomous systems development and adoption– e.g., Drones [17] and Self-driving cars [9]– is rising in recent years. Autonomous driving has the potential to eliminate accidents caused by human errors [11]. Similarly, the potential usage scenarios of drones are wide, including the fields of agriculture, aerial photography, delivery, or surveillance. Drones can directly reach places using specific routes [17]. Hence, both drones and autonomous driving systems represent emerging CPS, which can operate in dynamic environments, also involving interactions with humans. The reliability and safety of software systems are of high relevance and it is of even more importance for CPS. One reason is that the environments and working scenario of CPS are often critical or unpredictable (e.g., the scene of a fire in case of drones). In general, the increasing automation fostered by CPS emerging systems, gives rise to many challenges, at the crux of which lies the hardware and software symbiosis [15].

Autonomous Vehicles can find themselves into a wide range of difficult situations, and they do not have a human to instruct them in fine detail all the time [2]. They must be trained to react quickly and sensibly to erratic and inconsistent behaviour from human drivers, pedestrians, or cyclists. When testing CPS systems, we must therefore care deeply about the dangerous, less frequent scenarios in addition to typical, relatively safe scenarios. Reinforcement learning [13], Imitation Learning [3] and transfer learning [10] are the most recent deep learning approaches to solve autonomous driving problem. However, these techniques have very high sample complexity - the amount of training data needed to learn useful behaviours is often prohibitively high. Gathering such data by physical tests can be expensive, difficult and even dangerous as autonomous vehicles are often unsafe and expensive to operate during the training phase. In contrast, a high-fidelity simulator can augment and improve training of algorithms and allow for testing safely and efficiently. Insights gained from simulation could provide important training data and information on algorithmic inaccuracies before actual vehicle testing.

Testing automation is particularly important in CPS systems, as they are going to be extremely complex systems and will need an extensive amount of testing. However, when testing CPS, most developers rely on human written test cases, assessing the CPS behaviour in the real environments, which has drastically higher costs than in traditional systems. Hence, current CPS testing practises have several drawbacks including (1) difficulty to reproduce faulty scenarios as CPS environments are non-deterministic [12]; 2) difficult to standardize benchmark to ensure that CPS behaviour is safe [11]; 3) some scenarios are too dangerous to be recreated in the real world (e.g., a child running onto the road ahead of the autonomous vehicle). Therefore, we need new development/testing strategies and regulations to support the wide-spread usage of such systems.

Software engineering has evolved from developing projects using a mere compiler, to a much more complex and heterogeneous process, often involving several stacks, languages, and frameworks. A new approach called DevOps [7] promises to allow software organizations to release complex software early and with a high frequency. DevOps is set of practices towards software delivery where the key focus is on speed of delivery, continuous testing in production like environment, continuous feedback, ability to react to change more quickly, teams working to accomplish a global goal instead of an individual task. This allows entirely new concepts of development and acceptance testing. Today, Continuous Delivery is state-of-the-art in many application domains and enables large enterprises to provide new custom releases to a broad base of customers at the push of a button, to receive instant customer feedback [16]. While current DevOps principles apply to code integration, deployment, delivery and maintenance in the software industry, we envision the application of the very same principles at the model level for the development of CPS.

We argue that an alternative, effective and efficient approach to train and validate CPS systems, could be the integration of software simulations [1] into DevOps pipeline. A properly implemented and employed Continuous Integration(CI) System shortens the lead time from coding to deployed products, and increases the overall quality of the code and the system being shipped. Testing soon and testing quickly is logistically simple for IT applications where any standard computer or cloud computing instance can be used for testing. However, for complex and heterogeneous CPS systems, it can be a real issue to perform continuous integration and automated immediate testing. CPS systems are going to become part of every aspect of our daily lives, and their numbers are expected to reach billions. Software and firmware used in these systems must be continuously maintained and updated for security reasons and improvements on its functionality. Over the air (OAU) updates of these devices will cause another issue in CT/CD of CPS. OAU basically consists in deploying new code onto running systems, potentially billions, which might lead to unexpected and disastrous consequences if not properly done. This might be a motivation to test more and test better (and different scenarios) during DevOps.

Using high speed virtual platforms and advanced simulation allows for true automated and continuous integration even for CPS system developers. Specifically, we believe that testing CPS in virtual/simulation environments in DevOps pipelines(as illustrated in Fig.1) could help in addressing most of the emerging challenges of contemporary CPS systems. Ideally, we could learn policies that encode complex behaviors entirely in simulation and successfully run those policies on physical robots with minimal additional training. However, discrepancies between software simulators and the real world make transferring behaviors from simulation challenging. The real world has unmodeled physical effects like nonrigidity, gear

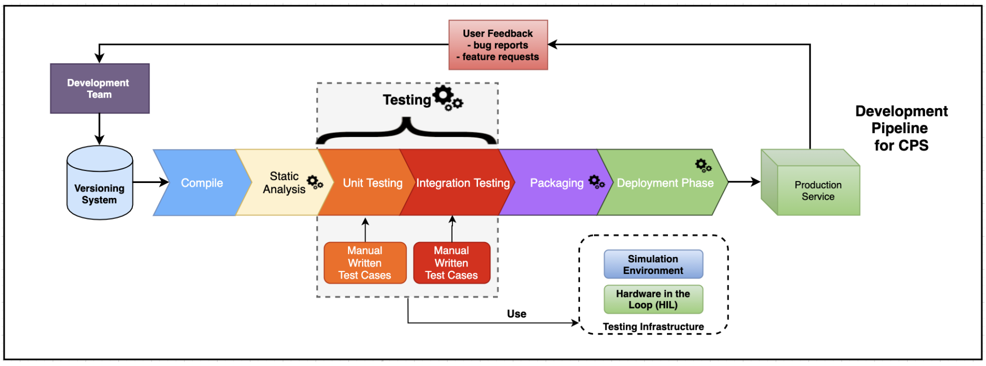

Figure 1: (Idealized) Development pipelines for Cyber-physical Systems.

backlash, wear-and-tear, and fluid dynamics that are not captured by current simulators. For example, for robots that aim to use computer vision in outdoor environments, it may be important to model real-world complex objects such as trees, roads, lakes, electric poles and houses along with rendering that includes finer details such as soft shadows, specular reflections, diffused inter-reflections and so on. These differences, known collectively as the reality gap, form the barrier to using simulated data on real robots. However, in this study, we will be limiting the scope to software simulations.

For this study, we are going to investigate on Flightmare, AirSim, Carla, BeamNG and other simulators for the simulation of drones and cars. For the evaluation and assessment of the simulators, we will use the following selection criteria:

- **Ease usage in real environments:** difficulties of learning and using simulation environment, resources cost, time required to setup scenarios and environments.
- **Open Software v.s. Commercial Software**: open source tools will be prioritized to commercial software.
- **Simulators Capabilities:** reproducible (deterministic execution), number of available features, software reliability and scalability.
- **Development Activities:** only actively developed software is considered.

The goals of this master's independent study

The independent study consists in a systematic literature review (SLR) on relevant CPS simulators such as `Flightmare`, `AirSim`, `Carla`, `BeamNG.research` and etc. Hence, the **research question** that guide this study is:

- *RQ: What are the advantages, limit, and challenges of using contemporary simulators?*

The outcome of this independent study will lay down further research on the role of advanced co-simulation strategies in the context of testing and quality assessment of Drones and other CPS systems (and in general autonomous systems).

In the next phrase (beyond the scope of this current study), we will investigate on identifying which simulator technology(rigid/soft body) is suitable for different test scenarios related to Obstacle avoidance, Lane following, accidents simulations, etc.

Task description

The main **tasks** of this study are:

1. Read reference papers to get familiar with the topics.
2. Conduct a comprehensive systematic literature review (SLR) study on features, limits and challenges of contemporary simulators for CPS:
   (a) Initial list of simulators considered: Flightmare, AirSim, Carla, Gazebo and BeamNG.
   (b) Other simulators will be selected using various selection criteria, as previously described: ease of usage, actively developed simulators, used in research, etc.
   (c) Methodology: after identifying the simulators we will focus on (i) make a list of the features of the various simulators and compare them according to such criteria; (ii) understand what simulators can or can't technically do; (ii) empirical comparisons between simulators in terms of ease of use, and functionality. Results of this analysis consist in a report on features, limits and challenges of contemporary simulators, with tables making direct comparison of investigated simulators.
3. Conduct a survey summarizing and analysing the results gathered in above tasks.

Deliverables

During the independent study, most commonly used software simulators in academia and industry, are examined to understand the current state of the research in the field of autonomous vehicles and a survey report is created. At the end of the study, a survey will be published containing a comparison of selected simulators.

Advisors:

Senior Computer Science Researcher:
Dr. Sebastiano Panichella

Professor:
Dr. Davide Scaramuzza

Signatures:

Xiao'ao Song	Dr. Sebastiano Panichella
Neeraj Kumar	Prof Dr. Davide Scaramuzza

References

[1] A. Afzal, D. S. Katz, C. L. Goues, and C. S. Timperley. A study on the challenges of using robotics simulators for testing, 2020.

[2] Alexander, Hawkins, and Rae. Situation coverage - a coverage criterion for testing autonomous robots. 2015.

[3] J. Bagnell. An invitation to imitation. Technical report, CMU ROBOTICS INST, 2015.

[4] A. Bernstein. So what is a (diploma) thesis? a few thoughts for first-timers. Technical report, Dynamic and Distribution Information System Group, University of Zurich, 2005.

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

[6] A. Dosovitskiy, G. Ros, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. 2017.

[7] F. Erich and C. Amrit. A qualitative study of devops usage in practice. 2017.

[8] A. Gambi, M. Mueller, and G. Fraser. Automatically testing self-driving cars with search-based procedural content generation. 2019.

[9] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu. A survey of deep learning techniques for autonomous driving. 2020.

[10] W. K, T. Khoshgoftaar, and D. Wang. A survey of transfer learning. journal of big data. *IEEE Software*, 2016.

[11] N. Kalra and 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*, 94:182–193, 12 2016.

[12] J. Kim, S. Chon, and 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, Apr. 2019.

[13] J. Kober, A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey.

[14] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. 2017.

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

[16] S. Vöst and S. Wagner. Towards continuous integration and continuous delivery in the automotive industry. 2016.

[17] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling. Vision meets drones: Past, present and future, 01 2020.