Scenario-Based Testing with BeamNG.tech (Hands-On Training)

Chrysanthi Papamichail and David Stark
BeamNG GmbH, Germany
{cpapamichail, dstark}@beamng.gmbh

Alessio Gambi
AIT Austrian Institute of Technology GmbH, Austria
alessio.gambi@ait.ac.at

***Abstract*—Autonomous Driving Systems (ADS) are safety-critical Cyber-Physical Systems that require thorough validation. Currently, scenario-based testing in simulations is the cornerstone of ADS validation, complementing expensive and dangerous natural field operational testing. Scenario-based testing in simulation can systematically assess ADS in diverse, relevant, and critical driving situations. However, it requires sophisticated tools and rich content to let developers quickly generate the intended testing scenarios. This hands-on training illustrates how to use the BeamNG.tech framework for effective scenario-based testing, focusing on manual scenario generation, which is often neglected in research despite its central role in practice. Learning material and additional descriptions are available at: https://github.com/BeamNG/scenario-based-testing-tutorial**

***Index Terms*—Autonomous driving, scenario-based testing, procedural content generation**

I. Introduction

Autonomous Driving Systems (ADS) are safety-critical Cyber-Physical Systems that require thorough validation to achieve the vision of safer, sustainable, and more comfortable transportation. Currently, ADS are tested by coupling expensive and ineffective naturalistic field tests [1] with inexpensive, more controllable, and reliable simulation-based tests. Thanks to photorealistic and physically accurate driving simulations, like BeamNG.tech [2], ADS can be effectively tested in nominal [3] and critical scenarios, such as collisions [4], [5]. The soft-body physics modeling approach of BeamNG.tech has demonstrated efficiency and versatility [6].

Scenario-based testing in simulation can systematically assess autonomous driving systems across diverse and relevant driving situations. However, developers need content (e.g., maps, terrains, vehicles), the ability to generate scenarios quickly, and the possibility to execute simulations in real time.

This hands-on training illustrates how to achieve effective and efficient scenario-based testing using the BeamNG.tech framework [2]. Thanks to its soft-body physics simulator, plenty of maps and vehicle models, testing tools, and various APIs, BeamNG.tech fulfills the major requirements of scenario-based testing [7]. The trainers will guide the participants using the available tools to practice manual scenario generation and enable basic automation.

II. Key Novelty and Overall Objectives

According to Menzel et al. [8], scenario-based testing for ADS identifies four scenario abstraction levels developers should consider during testing: *Functional* scenarios that describe the environment and the behaviors of different traffic participants using natural language. *Abstract* scenarios that refine them and focus on causal relations between traffic participants and expected behaviors. *Logical* scenarios that parameterize abstract scenarios by specifying the relevant input parameters and their acceptable value ranges. *Concrete* scenarios that assign concrete values to the parameters and can be executed as driving simulations.

Unlike tutorials focusing on test automation, this hands-on training allows participants to practice the top-down scenario-based testing process by blending manual work and automation. The trainers will show how to use BeamNG.tech's tools to generate scenarios that include complex roads, advanced sensors, and dynamic traffic. The training also covers aspects of automation, thus enabling search-based and fuzz testing.

This hands-on training aims to stimulate further research in the ADS domain, promote best practices for validating ADS, and reflect BeamNG's commitment to fostering the innovation and development of new technologies for mobility. Additionally, for participants with experience using other driving simulators in ADS testing research, this training is an occasion to assess BeamNG.tech's ease of use, superior performance, better physics simulation, and improved photorealism.

III. Target Audience

BeamNG.tech's straightforward installation and user-friendly interface reduce barriers for those interested in exploring ADS testing methodologies. Consequently, this hands-on training targets a relatively broad audience, including validation and verification experts, students approaching ADS development and scenario-based testing, educators who plan to include scenario-based testing in their classes, and Machine Learning experts who need large collections of labeled data.

IV. Tools Overview

BeamNG.tech is a suite that includes an efficient and accurate soft-body physics simulator (implemented in C and C++), tools for manually editing environments and scenarios (implemented in LUA and JavaScript), extensive content (e.g., props, vehicles), and various APIs to programmatically control the simulator and integrate it with other environments, like Simulink and ROS (implemented in Python).

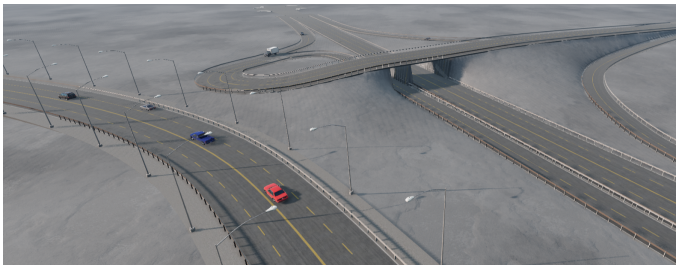

Fig. 1: Sample roads created with the Road Architect tool.

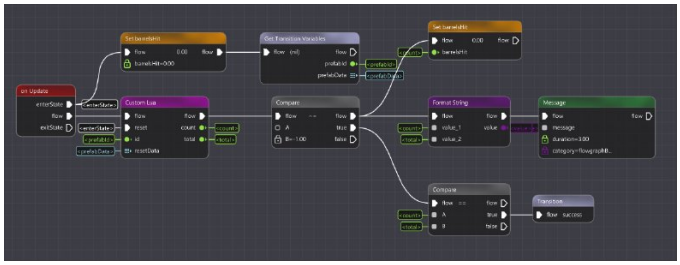

Fig. 2: Sample Flowgraph to control a driving scenario.

During this hands-on training, the participants will mainly practice three tools: the Road Architect, the FlowGraph, and the Python APIs. Moreover, the participants will be introduced to sensor configuration editors that enable placing sensors on vehicles and infrastructure. Thus enabling the creation of scenarios involving ADS and smart infrastructures and the collection of a rich dataset from the scenario executions.

Interested readers can refer to the official documentation[1] for more details and other tools offered by BeamNG.tech.

A. The Road Architect

The Road Architect can generate complex road networks, such as roundabouts, bridges, and tunnels (see Figure 1). This tool can also be used to sculpt terrain, create new roads, import existing roads and junctions, and edit them. Its intuitive graphical interface allows developers to connect various road segments, ensuring drivability while forming road networks. The Road Architect also enables developers to add and alter details, such as road materials and lane markings, and include objects such as lamp posts, crash barriers, traffic lights, and signs. Finally, this tool can import and export road networks defined in OpenDRIVE format, uniformly adhering to industrial standards.

B. FlowGraph Scenario Scripting

During scenario execution, BeamNG.tech enables controlling dynamic elements (e.g., traffic) using an intuitive graphical scripting language called FlowGraph (see Figure 2). This scenario scripting language, coupled with an editor that supports drag-and-drop, allows those without coding experience to generate and control scenarios in BeamNG.tech.

C. Python API

BeamNG.tech enables automation via BeamNGpy [9], a Python library implementing a set of APIs to generate and control scenarios. Thanks to BeamNGpy, concrete scenarios can be instantiated, executed, and monitored. Likewise, sensors' and vehicles' data can be easily collected. BeamNGpy transforms BeamNG.tech scenarios into client-server applications, enabling multiple clients to interact concurrently with the running simulations. BeamNGpy also integrates BeamNG.tech to MATLAB [10], ROS1 [11], and ROS2 [12], thus allowing developers to implement co-simulations and integrate other robotics systems into their scenarios.

[1]https://documentation.beamng.com/beamng_tech/

V. Conclusions

After completing this hands-on training, the participants will have a better understanding of BeamNG.tech's capability and will be able to build on top of it effective and efficient scenario-based testing.

References

[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[2] BeamNG GmbH, "BeamNG.tech." [Online]. Available: https://www.beamng.tech/

[3] S. Klikovits, A. Gambi, D. Dhungana, and R. Rabiser, "Leveraging software product lines for testing autonomous vehicles," in *Proceedings of the 18th International Working Conference on Variability Modelling of Software-Intensive Systems, VaMoS 2024, Bern, Switzerland, February 7-9, 2024*, T. Kehrer, M. Huchard, L. Teixeira, and C. Birchler, Eds. ACM, 2024, pp. 56–60. [Online]. Available: https://doi.org/10.1145/3634713.3634720

[4] A. Gambi, V. Nguyen, J. Ahmed, and G. Fraser, "Generating critical driving scenarios from accident sketches," in *IEEE International Conference On Artificial Intelligence Testing, AITest 2022, Newark, CA, USA, August 15-18, 2022*. IEEE, 2022, pp. 95–102. [Online]. Available: https://doi.org/10.1109/AITest55621.2022.00022

[5] A. Gambi, T. Huynh, and G. Fraser, "Generating effective test cases for self-driving cars from police reports," in *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*, M. Dumas, D. Pfahl, S. Apel, and A. Russo, Eds. ACM, 2019, pp. 257–267. [Online]. Available: https://doi.org/10.1145/3338906.3338942

[6] B. Arya, J. Yao, J. L. Davy, and M. Fard, "A novel and fast crash simulation method: Revolutionising racetrack safety barrier analysis," *Results in Engineering*, vol. 25, p. 103870, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123024021133

[7] S. Tang, Z. Zhang, J. Zhou, L. Lei, Y. Zhou, and Y. Xue, "Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2024, Sacramento, CA, USA, 27 October 2024 - 1 November 2024*. ACM, 2024, accepted for pulbication.

[8] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*. IEEE, 2018, pp. 1821–1827. [Online]. Available: https://doi.org/10.1109/IVS.2018.8500406

[9] BeamNG GmbH. Beamngpy documentation. [Online]. Available: https://beamngpy.readthedocs.io/en/latest/readme.html

[10] MATLAB, *MATLAB version 24.1.0.2578822 (R2024a)*. Natick, Massachusetts: The MathWorks Inc., 2024.

[11] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

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