

System for software testing in real-time simulation environment

Vanja Mijatov, Bojana Ivanović, Branko Milosavljević

Faculty of Technical Sciences, Novi Sad, Serbia

vanja.mijatov@uns.ac.rs, bojana.vanovic@uns.ac.rs, mbranko@uns.ac.rs

Abstract—This paper presents an implementation of the System for Software Testing in Real-Time Simulation Environment. The main goal of this system is to allow real-time simulation of software in an isolated environment according to hardware-in-the-loop concept. Due to numerous challenges in development of software that run on devices that process lots of signals and that are supposed to be integrated in large systems, hardware-in-the-loop testing has emerged as standard technique for their validation by simulating their work as if devices were in real environment. Due to the nature of the testing process, only one simulation can be executed at a time and usually the same environment is reused for different setups or different processes in the development of different components of system. This proposed system allows having clean environment with only necessary dependencies for a specific setup and allows running multiple simulations at the same time. In addition, it provides tools for observing reports of finished test sessions and searching executed tests by different parameters.

Keywords: software testing, real-time simulation, hardware-in-the-loop, HIL, containerization, Docker, Typhoon HIL Control Center, TyphoonTest

I. INTRODUCTION

Developing software that operate on controllers and other devices in the area of power electronics, microgrids and distribution networks is expensive and it has proved challenging to validate how they work before utilizing them in intended use. In order to test the behavior of the software for controller for example while it is being developed as if it was integrated in real system, it is important to simulate its work in real time by sending the inputs to the device and reading its outputs. By testing the device in an environment that consists of both physical and virtual subsystems where virtual subsystems are emulated by a special device (simulator), we can evaluate how tested device operates in different cases in real-time which allows engineers to detect defects in a timely manner. Real-time simulation of a component is a type of simulation where signal values (inputs and outputs) are time-dependent in a same way as in a real component [1].

Hardware-in-the-loop (HIL) is a concept that implies simulating how device operates in real time in the natural environment. HIL simulation requires having a model of simulated process and a device that is being tested where simulation model provides all the process signals in real-time that are converted and supplied to the device [2]. HIL testing has become a standard method for testing control units and control strategies during the development [3]. The device that performs the simulation is called HIL



Figure 1. Illustration of a simple setup for HIL testing process

simulator. HIL simulator is “a setup that emulates a system by immersing faithful physical replicas of some of its subsystems within a closed-loop virtual simulation of the remaining subsystem” as defined in paper [4]. Figure 1 illustrates a simple setup for HIL testing process. HIL simulation provides numerous benefits that simplify the process of development [4]. HIL simulation can also be used for validating models with setups that consists almost only of virtual subsystems where HIL simulator and machine that is used for running simulation are the only hardware parts. HIL testing is commonly used for testing of devices in different fields of industry such as power electronics [5] [6], marine [7] [8], automotive [9][10][11] and aerospace [12][13].

In order to perform HIL testing of a certain device, it is required to create a setup where HIL simulator would be connected to both tested device and a computer that is used for running the simulations. Therefore, only one simulation can be executed at a time. In case of large-scale testing, device is obligated to be tested with different setups and in order to decrease the duration of a test session, test executions should be executed in parallel which is not suitable with traditional process of HIL testing where only one simulation can be executed at a time. In addition, using same environment for testing the same device with different setups or for testing different devices can pollute the environment with unnecessary software that are not required for testing of a specific device which can cause issues in testing process. Finally, after the test session is finished, it is desirable to have a detailed report of the testing process.

II. TECHNOLOGIES USED FOR DESIGNING THE SOLUTION FOR SOFTWARE TESTING IN REAL-TIME SIMULATION ENVIRONMENT

A. Containerization process

Solution to the two identified drawbacks of the traditional HIL testing process in case of large-scale testing is containerization process. Containerization is a lightweight type of virtualization that results in a container. Container contains self-contained application that is ready for deployment with all the dependencies that are required for its operation [14]. Compared to

hypervisor type of virtualization, it provides better performance and less resource consumption [15] [16] which makes it great for utilization and distribution. In order to allow HIL testing, container is obligated to contain all necessary technologies and software that are required for testing process. Docker [17] has emerged in one of the leading containerization solutions and was therefore used in this research. Each time a Docker container is started, it contains only dependencies that are necessary for the testing process and therefore it presents a clean environment which is suited for a specific process, such as software testing for example.

B. HIL testing using Typhoon HIL software stack

Typhoon HIL Control Center [18] is a software toolchain that allows designing, simulating and testing different models in the area of power electronics, automotive, microgrids, distribution networks and other fields. Using this toolchain user can define a model of tested device and simulate its behavior in real-time or artificial realistic scenarios.

TyphoonTest [19] is IDE with Python library that allows writing and executing tests which can verify the design of a piece of software in environment or installed on tested device. Using the library test can be executed and a result of this process is Allure report that is generated using Allure framework [20]. Allure framework collects results of each executed test case and generates report that can be served on a web server as static files and that provides details for each executed test case.

Using Typhoon HIL software stack, an engineer can develop tests that validate behavior of a component of a real system or the complete system which is represented as a model that is simulated. Before running a test, it can be parametrized and when executed it simulates a model. During the simulation it can feed it with different inputs and it asserts the outputs of the simulation that are usually different signal values that change over time.

III. PROPOSED SOLUTION FOR SOFTWARE TESTING IN REAL-TIME SIMULATION ENVIRONMENT

In order to create an environment for HIL testing process, Typhoon HIL Control Center as software for running simulations and TyphoonTest library (that relies on Pytest testing framework) as technology for running tests with all of their dependencies were containerized using Docker and the result of this process was a Docker image. This required identifying all of the necessary system libraries, packages and other dependencies which are required for their operation. Docker image was later optimized in a way that unnecessary packages were removed and image size was minimalized. This solution has proved suitable for different testing processes because it contains only minimal required packages for running tests and is effortlessly expandable with different software that are required for specific test scenario or setup. In addition, the image is easily distributed using Docker Hub [21]. Based on the image, a container can be started which

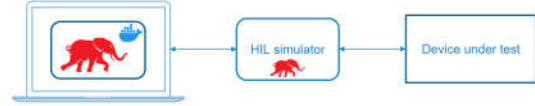


Figure 2. Illustration of a simple setup for HIL testing process

can then be used for installing additional software and after that for testing. The image can also be used as a base image for creating a new Docker image that adjusts existing environment and adds additional software. Multiple containers that are based on this image can be started on the same machine and can communicate with different HIL devices that are parts of different setups which altogether allows running multiple simulations in parallel on the same machine but inside separate containers.

Figure 2 illustrates setup that was used to test the proposed solution. Desktop machine running Docker Engine was used to start a container based on generated Docker image. It was connected to HIL device manufactured by Typhoon HIL Inc. Different sets of tests were executed by using both traditional setup where tests are executed directly on the host machine and illustrated setup and results of the test sessions that were encapsulated as Allure reports were reviewed. By manually reviewing generated Allure results it was concluded that proposed solution works for simple testing processes and that executed test sessions' results in executed tests having expected outcomes.

IV. EVALUATION OF THE RESULTS OF THE PROPOSED SOLUTION

In case of large-scale testing, manual review of generated Allure reports is extremely time-consuming and requires a lot of effort because it requires checking results of each executed test. Therefore, additional subsystem of solution was developed in a form of web application with a modest user interface. It runs in a separate container on premise or in cloud and processes Allure reports and stores the results of each executed test in Elasticsearch platform that was adjusted for indexing and searching test results. It allows searching executed tests that were executed as parts of all test sessions whose reports were processed.

Multiple search criteria can be defined and combined with logical operators “and” and “or”. Each search criteria allows searching by a single parameter. Results of the executed query are listed and it is highlighted where searched terms were identified. Figure 3. illustrates user interface of the described system with an example of a query with multiple criteria. Using the described tool, it is verified that the results of test session that were executed with the proposed solution resulted with identical success as test sessions executed using traditional setup. In addition, test sessions' durations were similar.

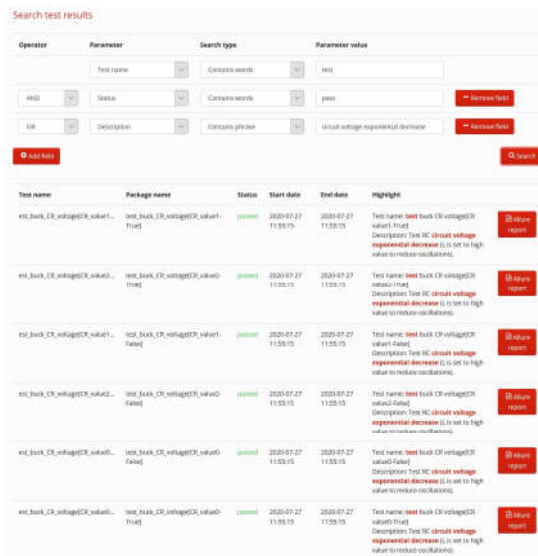


Figure 3. User interface of the application for searching executed tests with an example

Another benefit that the proposed solution provides is that it allows running multiple containers that can execute independent test sessions which use different HIL devices. This way a large set of tests can be divided into multiple subsets of tests that are executed in different containers. In this way, parallelization of a complex test session is achieved and it results in total execution time of a test session to be significantly decreased. Additionally, since each container contains only required software, every time it is started it represents a clean environment which is suitable for the testing process.

Another benefit that the proposed solution provides is allowing software-in-the-loop (SIL) testing. Virtual HIL [23] is a virtual simulator that is a component of Typhoon HIL Control Center which allows simulating model on computer without using HIL device. It can be used for testing models that do not require small simulation steps that HIL provides and that are not too complex which makes the proposed solution suitable for SIL testing.

Besides testing, solution can be utilized for simulating digital twins of different power electronics systems which presents the final benefit of proposed solution. A model that presents a digital twin of a certain device or group of devices can be simulated and by using specific libraries it is allowed to interact with the simulation.

V. CONCLUSION

This paper describes a system for software testing in real-time simulation environment which allows HIL testing of different devices and their components. It is achieved by containerizing stack of different technologies that allow HIL testing. Described solution allows testing in clean environment without unnecessary software. Additionally, it allows running multiple parallel test sessions that execute multiple HIL simulations on different HIL simulators. Another benefit of the solution is that it allows SIL testing and simulation of digital twins of different power electronics systems.

Moreover, this paper describes a tool for searching results of executed tests that belong to the reports of executed test sessions. This tool simplifies the process of reviewing the results of test sessions and comparison of them to the results of HIL testing sessions that were executed using traditional setup. Using this tool, it is concluded that test sessions which were executed using the described solution resulted with identical success as test sessions which were executed using traditional setup and that they had similar execution durations.

REFERENCES

- [1] Isermann, Rolf, Jochen Schaffnit, and Stefan Sinsel. "Hardware-in-the-loop simulation for the design and testing of engine-control systems." *Control Engineering Practice* 7.5 (1999): 643-653.
- [2] Grega, Wojciech. "Hardware-in-the-loop simulation and its application in control education." FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings (IEEE Cat. No. 99CH37011. Vol. 2. IEEE, 1999.
- [3] Kelemenová, Tatiana, et al. "Model based design and HIL simulations." *American Journal of Mechanical Engineering* 1.7 (2013): 276-281.
- [4] Fathy, Hosam K., et al. "Review of hardware-in-the-loop simulation and its prospects in the automotive area." *Modeling and simulation for military applications*. Vol. 6228. International Society for Optics and Photonics, 2006.
- [5] Ivanović, Zoran R., et al. "HIL evaluation of power flow control strategies for energy storage connected to smart grid under unbalanced conditions." *IEEE Transactions on Power Electronics* 27.11 (2012): 4699-4710.
- [6] Graf, Christian, et al. "Real-time HIL-simulation of power electronics." *2008 34th Annual Conference of IEEE Industrial Electronics*. IEEE, 2008.
- [7] Skjetne, Roger, and Olav Egeland. "Hardware-in-the-loop testing of marine control system." (2006): 239-258.
- [8] Zhou, Jiadong, Guangyao Ouyang, and Minghe Wang. "Hardware-in-the-Loop testing of electronically-controlled common-rail systems for marine diesel engine." *2010 International Conference on Intelligent Computation Technology and Automation*. Vol. 2. IEEE, 2010.
- [9] Krejci, P. "Testing of automotive park assistant control unit by hil simulation." *Mechatronics 2013*. Springer, Cham, 2014. 487-493.
- [10] Fathy, Hosam K., et al. "Review of hardware-in-the-loop simulation and its prospects in the automotive area." *Modeling and simulation for military applications*. Vol. 6228. SPIE, 2006.
- [11] Schulte, Thomas, Axel Kiffe, and Frank Puschmann. "Hil simulation of power electronics and electric drives for automotive applications." *Electronics* 16.2 (2012): 130-135.
- [12] Fang, Juan, et al. "FMI-Based Multi-Domain Simulation for an Aero-Engine Control System." *Aerospace* 8.7 (2021): 180.
- [13] Peng, Qi, Ying-Qing Guo, and Hao Sun. "Modeling and fault diagnosis of aero-engine lubricating Oil system." *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018.
- [14] Pahl, Claus. "Containerization and the paas cloud." *IEEE Cloud Computing* 2.3 (2015): 24-31.
- [15] Scheepers, Mathijs Jeroen. "Virtualization and containerization of application infrastructure: A comparison." 21st twente student conference on IT. Vol. 21. 2014.
- [16] Felter, Wes, et al. "An updated performance comparison of virtual machines and linux containers." 2015 IEEE international symposium on performance analysis of systems and software (ISPASS). IEEE, 2015.
- [17] Rad, Babak Bashari, Harrison John Bhatti, and Mohammad Ahmadi. "An introduction to docker and analysis of its performance." *International Journal of Computer Science and Network Security (IJCSNS)* 17.3 (2017): 228.
- [18] <https://www.typhoon-hil.com/products/hil-software/>, accessed May 2022.
- [19] <https://www.typhoon-hil.com/products/typhoon-test-ide/>, accessed May 2022.

[20] <https://docs.gameta.io/allure/>, accessed May 2022.

[21] <https://hub.docker.com/>, accessed May 2022.

[22] <https://docs.pytest.org>, accessed May 2022.

[23] <https://www.typhoon-hil.com/products/virtual-hil-device/>,
accessed May 2022.