CHEMICAL ENGINEERING TRANSACTIONS
VOL. 114, 2024
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.cetjournal.it
Guest Editors: Petar S. Varbanov, Min Zeng, Yee Van Fan, Xuechao Wang

**ISBN** 979-12-81206-12-0; **ISSN** 2283-9216
DOI: 10.3303/CET24114134

# Embedded System Simulation for Electrical Hardware Test Virtualization

István Szalay, Ferenc P. Speiser, Dénes Fodor*

Széchenyi István University, Department of Power Electronics and Electric Drives, 9026 Győr, Egyetem tér 1., Hungary
fodor.denes@ga.sze.hu

In the automotive industry, the reduction of development costs has a key importance. Also, the duration of the development is a significant aspect. Electrical hardware development is an expensive, time-consuming process with a lot of development stages (e.g., prototypes, electrical tests, mechanical tests, lifecycle tests). Virtualization is an important factor in making electrical component development more sustainable; using this method, less prototype manufacturing is necessary since the simulations make it faster and more effective to find out a large portion of possible faults without building a hardware prototype. The main goal of the paper is to explore the capabilities of seven software solutions that can manage the virtualization of the electrical development toolchain. The simulation environment should manage the simulation of the electrical circuit, including the sensors and loads, the microcontroller, and the execution and debugging of the uploaded program code. The study provides guidance on choosing the proper simulator depending on the simulation focus.

## 1. Introduction

More and more electronic control units are installed in new cars being manufactured today. These usually run some kind of embedded software to ensure proper operation. The development time for these controllers is long and repeated prototyping and testing increase costs (Misbin and George, 2023). There is a growing need to increase the cost-effectiveness of the development and testing phases of embedded software and the hardware that runs it while maintaining reliability (Yang et al., 2023).
In the case of automotive application development, in addition to embedded software development, it is also important to develop the necessary hardware. However, this task involves continuous prototype testing and redesign based on the test results, with many iterations. It is also time-consuming and costly. The purpose of simulating hardware and software together is to verify that the hardware and software work together properly. This task is traditionally performed after the prototype hardware has been built, but if this verification can be performed in a virtual environment, the cost of prototyping and any iterations, and the time required for development, can be reduced. A general-purpose simulation methodology for virtual functional testing of electronic systems is currently not available; the aim of this study is to explore the available software solutions that can handle the virtualisation of the electrical development toolchain.
Functional failures of electronic control units (ECUs) in-vehicle electronic systems can be the consequence of various analogue and digital circuit problems as well as software and communication errors. Currently, in the industry, functional testing for such failures is performed experimentally and is lengthy can be performed only in a late development phase when physical prototypes are available and is time-consuming to correct and retest (Kim et al., 2023). For all these reasons, testing the functional robustness of vehicle electronic systems is costly and time-consuming. The availability of industrially applicable virtual testing platforms that can evaluate the functional robustness of vehicle electronic systems will be valuable to the industry.
The goal is to develop a simulation methodology that is suitable for virtual functional testing of electronic components, taking into account the influencing effect of the software. This can be used in all areas where the following aspects are important: early screening of design errors, shortening the development time, increasing the test coverage and the efficiency of testing, significant reduction of testing costs, and minimising the use of hardware elements (Damle et al, 2022).

Paper Received: 17 May 2024; Revised: 5 August 2024; Accepted: 17 November 2024
Please cite this article as: Szalay I., Speiser F.P., Fodor D., 2024, Embedded System Simulation for Electrical Hardware Test Virtualization, Chemical Engineering Transactions, 114, 799-804 DOI:10.3303/CET24114134

The solution can be used in any electronic system design process that requires functional testing of control units, co-simulation of analogue and digital circuits, or robust design and operation. Unfortunately, there is a lack of information available on the applicable software solutions in the scientific literature.
The simulation of ECUs as embedded systems is a complex task (Amringer and Asemann, 2022). The ECU itself can be decomposed into three very different sections: the analogue electric circuit part, the digital circuit part with the microcontroller, and the software running on it, while the environment of the ECU is a fourth section, which is usually a different physical system, i.e. not just an electric circuit. The four colour-coded sections, their connections, and the main control flow are shown in Figure 1.

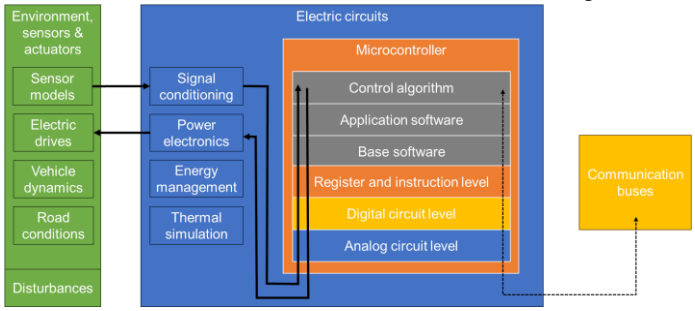


*Figure 1: The ECU and its environment*

The simulation of the four sections requires different simulation tools, shown in Figure 2, with the same colour coding. The simulation of the ECU's environment can be done with traditional system simulators, such as Simulink, which usually follow a causal modelling approach. The suitable tools for the simulation of the analogue circuit section are the circuit simulators, such as SPICE-based programs, that follow an acausal modelling approach. The microcontroller and the software require an instruction set simulator, and the simulation of the communication buses may require an additional simulator. Traditionally, the four fields are somewhat isolated. Most simulator software suites only aim for one or sometimes two sections.

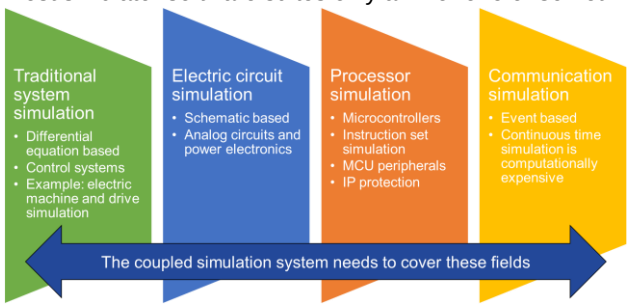


*Figure 2: Coupled simulation of embedded systems*

The literature on complete embedded system simulators is limited, and there is a growing need for such tools in the automotive industry to increase development efficiency. The main goal of the paper is to explore the available software solutions that can manage the virtualisation of the electrical development toolchain. The Materials and Methods section presents the envisioned completely virtual testing process for embedded systems. In the Results and Discussion section, the simulator tools were evaluated and compared based on which sections of embedded system simulation they covered. The study provides guidance in choosing the proper simulator software depending on which system layer is in focus.

## 2. Methods and materials

### 2.1 The envisioned simulation process

The envisioned virtual testing process is shown in Figure 3. The circuit simulators solve the numerical simulation and signal logging steps. However, there is a gap between the schematic and the circuit model, and the inference of the functional state from the simulated signal values requires extensive research. Both of these two steps will depend on the specific design under test (DUT) and the available knowledge about its expected operation. The final objective is to develop a generalised process for streamlining these two steps and for the extraction of the additional products of the process, such as reusable subcircuits and state inference methods.

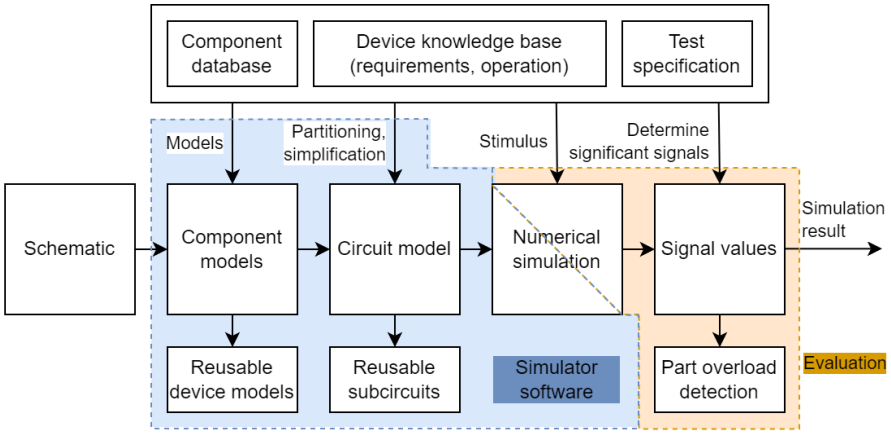


*Figure 3: The envisioned simulation process*

### 2.2 Modelling approach

In causal models, the outputs of the modelled system are explicitly expressed in terms of the inputs, i.e. the information flow has a defined direction, and the model equations must be formulated as ordinary differential equations (ODE). In acausal modelling, the modelled system is, directly or indirectly, expressed as a system of undirected differential-algebraic equations (DAE) in implicit form (Schweiger et al., 2020). Solving DAEs is, in general, more difficult than solving explicit ODEs, where the differential variable is a valid system state candidate; however, the DAE formulation means that the modeller can focus more on what to model rather than on how to model it to, for example, facilitate numerical simulation. Acausal modelling is also known as declarative, mathematical, physical, or equation-based modelling in the literature. Simulink is causal, Simscape and the SPICE-based circuit simulators are acausal.

### 2.3 Microcontroller modelling

Simulating an embedded system, including the analogue circuit and the whole microcontroller unit (MCU), using just SPICE models would be very slow (Gauthier et al., 2000). The MCU simulation tools break up the circuit in parts and simulate the analogue part with SPICE and the digital part with other means.
People identify the various blocks in the circuit and simulate each independently with the most appropriate tool. Typically, for an analogue block with a few inputs from the microcontroller, you would use a spice tool and simulate inputs coming from the MCU using simple voltage sources for which you assign the appropriate waveform (using pulse at predefined times, for example). As an alternative solution, it is possible to simulate the high-level MCU code on a PC. In this case, the MCU peripherals used by the high-level functions have to be emulated. This can be time-consuming for a complex app with lots of peripherals, but it gives a lot of freedom during tests. In the end, it led to the development of a custom simulator.

## 3. Results and Discussion

The following summary gives some insight into the advantages and disadvantages of the investigated software solutions in embedded system simulation. The list of the tools for evaluation has been narrowed down based on the intended use, support availability, and if the tool is an integrated simulator environment.

### 3.1 MATLAB and Simulink

The Mathworks MATLAB-Simulink environment is a widely used modelling and simulation tool that provides a powerful computational engine with a large number of numerical algorithms, such as integration solvers, a scripting environment, and a versatile graphical user interface to create, run, and evaluate simulations.
Simulink is a general-purpose simulator; it is not limited to certain types of physical systems. It supports the simulation of both continuous and discrete time models, and analogue and digital models can be built with it. Thanks to its integration with MATLAB features available through its GUI, it provides very good automation capabilities. In general, it is a reliable and extensible tool frequently used in the automotive industry. Although it has no built-in MCU simulation capabilities. It follows the causal modelling approach. Therefore, an equation-based model that can be manually work-intensive is required to develop and is usually detached from the actual physical structure of the simulated system. The causal modelling approach does not scale well for electrical circuits that contain a high number of repeating components. For example, a resistor can have very different

roles in a circuit based on its position, and in the equation-based approach the engineer has to develop the equations for every case separately.

**Simscape**

Simscape is the acausal modelling subsystem of the Mathworks MATLAB/Simulink environment designed for physical systems modelling. Simscape provides a modelling language and a toolchain to compile the models into Simscape blocks that can be used in Simulink models. Mathworks also provides prebuilt electrical circuit element models as parts of various separately available toolboxes and block libraries.
As a consequence of the causal modelling approach, the visual representation of the model is close to the actual physical structure of the modelled system. Simscape models scale well for repeating components, which is an important advantage in electrical circuit simulation. For example, the engineer can reuse the same resistor model regardless of its position and role in the circuit. It has no built-in MCU simulation capabilities. The built-in model libraries cover a limited set of physical systems and components, although these can be connected to each other and to normal Simulink models. In electrical circuit modelling, it is an important disadvantage that Simscape offers only a limited level of SPICE compatibility.

### 3.2 Proteus

The Proteus Design Suite is a software tool used primarily for electronic design automation. The software is used mainly for electronic circuit design. It supports creating schematics, performing simulations, and also designing printed circuit boards (PCB). It has tools for microcontroller simulation, and it enables co-simulation along with analogue and digital electronics connected to it. It can be used in a broad spectrum of project prototyping in areas such as motor control, temperature control and user interface design. Proteus has quite a large MCU model library with more than 1,000 models from various vendors and architectures. An important feature of Proteus is the ability to run real-time and interactive circuit simulations.
Unfortunately, Proteus has some modelling limitations. Changing the supply voltage of an MCU is not possible because of the modelling principles of Proteus, which makes it impossible to simulate certain microcontroller functionalities, such as supply voltage drop interrupts and brown-out detection. The clock system of the MCU models is usually significantly simplified. The clock frequency cannot be changed during simulation, and there is no possibility of an external clock source simulation. The communication peripherals are simplified, or the related registers are not modelled. Besides the MCU model limitations, Proteus' modelling approach limits the power line models in general. Their voltage levels have to be defined as constant values.

### 3.3 Designsoft TINA

TINA, originally called Toolkit for Interactive Network Analysis, is a SPICE-based electronics design and training software. It enables analogue, digital, and mixed circuit simulations, as well as PCB design. It allows simulation, design, and real-time testing of hardware description languages (HDL). The schematic editor is easy to use, although the user interface feels a little bit outdated. TINA has a large MCU model library with more than 1,000 models from various vendors and architectures.
The circuit simulation in TINA is partially interactive, and certain controls and indicators are functional. Stress and power dissipation analyses are easily available on the schematic editor. Power rails are not treated as special elements; their voltage value can change. Although a large MCU library is available, most models are limited in some way. MCU supply circuits are partially modelled; for example, interrupt handling, the programmable voltage detector, and brown-out detection can be missing, and clock signal outputs may not be modelled. The configuration of the clock system is not straightforward, which, depending on the clock register configuration, can cause the software to run too fast or too slow compared to the simulation time.
External MCU source code projects can be configured, but re-running the simulations with modified source code requires manual work.

### 3.4 SIMetrix

It is a mixed-signal circuit simulator designed for ease and speed of use. It provides a closely coupled direct matrix analogue and event-driven digital simulator. The core simulator algorithms are based on SPICE, and in the Author's experience, SIMetrix is able to simulate various SPICE models quite reliably. To build the netlist expected by the simulator, it provides a fully integrated hierarchical schematic editor, which is also able to plot the simulation results and perform various post-processing tasks. Cross-probing of voltage, current and device power from the schematic is also possible. SIMetrix provides a full-featured scripting language allowing customised waveform analysis and automated simulation. An important additional feature of SIMetrix is the support of the simulation of Verilog models. The simulator supports Verilog-A for analogue models and mixed-signal simulation using Verilog-HDL.

### 3.5 VLAB

VLAB is an ECU/MCU/SOC simulation software environment that supports safety-critical virtualization-based development processes, including the test, verification, and validation processes. It allows users to define, architect, implement, test, validate and optimise embedded systems without the need for physical hardware. It runs the unmodified software and offers a rapid, automated path to continuous incremental build, integration, test, measurement, and analysis in all phases of development, from functional and test specifications to final release. It has high-performance simulation capabilities. The program execution is distributed and not limited to a single thread. VLAB can connect debug clients to each core for multi-core, multi-process scenarios to trace the execution of the code. It is possible to increase coverage and exercise error handling by injecting errors. Python scripting and PySPICE support are available, and VLAB is able to interact with MATLAB/Simulink, Vector Canoe/VectorCast/SIL Kit solutions. There is a wide range of supported MCUs (ARM, Bosch, Imagination, Infineon, NXP, Renesas, ST, TI).

### 3.6 Renode

Renode is an open-source software development framework that accelerates IoT and embedded systems development by making it possible to simulate physical hardware systems, including the CPU, peripherals, sensors, environment, and wired or wireless medium between nodes. It is an instruction set simulator with full determinism of execution and shared virtual time. The simulator runs the unmodified software. It can be configured using a human-readable, modular, and extensible platform description format (MCU modelling based on the datasheet), and rich model abstractions with additional functionality are available. Transparent and robust debugging, tracing, and analysis can be managed by the software even in multi-node setups (Interoperability: Jenkins, GDB, Wireshark). It is very well suited to be a part of an automated test scenario. Renode is integrated with the Robot Framework testing suite and provides user-friendly scripts for running tests. It includes an integration layer for HDL simulators. That allows connecting an HDL peripheral with interrupts and external interfaces. Thanks to the integration using SystemVerilog DPI, Renode can co-simulate with virtually any HDL simulator that supports Direct Programming Interface. Supported platforms include Microchip, NXP, RISC-V, ST, TI, ZYNQ.

### 3.7 Qorvo QSPICE

Qorvo QSPICE is an analogue and mixed-signal circuit simulator. It offers significantly better SPICE basics (higher simulation speed, functionality, accuracy, and reliability) than older simulators like PSPICE or LTSPICE, supports digital logic without performance penalties, and provides the speed and accuracy required for reliable power-based simulation with an emphasis on power integrity and noise analysis. QSPICE has native support for cutting-edge semiconductor types, such as the SiC FETs produced by Qorvo. From the embedded system simulation point of view, a very important feature of QSPICE is the support for behavioural modelling using Verilog and C++, which positions QSPICE as a true mixed-mode simulator, bridging the gap between digital and analogue designs. The QSPICE installer also includes the necessary Verilog and C++ compilers. These interfaces provide various ways to extend the SPICE-based electric circuit models with models of different physical systems, such as electric drives through Verilog, digital circuits models, and even custom MCU and software models through C++ extensibility.

### 3.8 Comparison

The previously presented simulator software suites have been compared based on how well they fit into the simulation of certain levels of the ECU and its environment. The result of the evaluation is shown in Figure 4. For every software suite and layer, a score of 2 for good suitability, 1 for acceptable suitability, and 0 for non-suitability was assigned. Based on these, a combined evaluation score was computed that indicates the coverage achieved by each software suite.

Based on the scores, it can be concluded that no single software suite is suitable for the simulation of all four layers of an ECU-environment model. Proteus and TINA provide good SPICE-based simulators, but the MCU models are functionally limited in many ways, and there is no easy way to extend them, develop custom MCU models, or implement non-circuit models, such as the moving components of electric drives.

VLAB and Renode provide good MCU simulation by running unmodified software but with no built-in capabilities for electric circuits or other physical system simulations. The MATLAB/Simulink/Simscape ecosystem is a very good general-purpose simulation environment, but it is not SPICE-compatible, and there is no built-in MCU simulation capability. SIMetrix and QSPICE are quite reliable SPICE-based circuit simulators, and both provide analogue and digital extensibility through Verilog, helping to cover the environment modelling layer, but there is no built-in MCU simulation capability.

| | | | LabCenter Proteus | | DesignSoft TINA | | VLAB | | Renode | | MATLAB/Simulink & Simscape | | SIMetrix | | Qorvo QSPICE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SW stack | Control algorithms | | Missing interrupt handling. Debugging is limited by the poor built-in code editor. | 2 | Missing interrupt handling. Debugging is limited by the poor built-in code editor | 2 | VLAB VDMs run the unmodified ECU software | 2 | Renode runs the unmodified software | 2 | Simulink is very good in this field | 2 | Verilog support | 2 | Verilog and C++ support | 2 |
| | Application SW | | | 1 | | 1 | | 2 | | 2 | Not suitable | 1 | Not suitable | 0 | Not suitable | 0 |
| | Base SW | | | 1 | | 1 | | 2 | | 2 | | 0 | | 0 | | 0 |
| MCU | Register & instuction level | | Missing peripherals and supply models. Closed models. | 1 | Missing peripherals and supply models. Closed models. | 1 | VLAB VDMs | 2 | Renode is an instruction set simulator | 2 | | 0 | | 0 | | 0 |
| | MCU digital circuit level | | | 0 | | 0 | | 0 | | 1 | | 0 | | 1 | | 1 |
| Electric circuit | MCU analog circuit level | | PROSPICE-based simulator. Unusual modelling limitations | 1 | XSPICE-based simulator | 1 | Maybe PySpice or maybe MATLAB and Simulink | 1 | Maybe Verilog with Verilator | 1 | Simscape is not SPICE | 1 | Reliable SPICE-based simulator | 2 | Reliable SPICE-based simulator | 2 |
| | Signal electronics | Power electronics | | 1 | | 2 | | 1 | | 0 | | 1 | | 2 | | 2 |
| Environment, sensors & actuators | | | Extension requires SDK | 0 | Extension requires SDK | 0 | Maybe MATLAB and Simulink or Python | 1 | | 0 | Simulink is very good in this field | 2 | Verilog support | 2 | Verilog and C++ support | 2 |
| Combined evaluation | | | 35% | | 42% | | 63% | | 50% | | 50% | | 65% | | 65% | |

*Figure 4: Standalone tool comparison and evaluation*

## 4. Conclusion

This study investigated which software suites can be used to perform simulations of embedded systems. The main aspects of the study were software support, the feasibility of microcontroller simulation, the simulation of electric circuits connected to a microcontroller and the possibility of implementing environment, sensor and actuator models. The main conclusion is that none of the evaluated software suites provide complete coverage. The study provides guidance in choosing the proper simulator depending on which layer has the project focus. Generally speaking, if microcontroller simulation is the important aspect, VLAB and Renode have an advantage. If the control algorithm development is in the focus, then Simulink/Simscape is a good alternative, and if circuit simulation is in the focus, then SIMetrix or QSPICE is the best choice. The application of these software tools helps to achieve sustainable development in the automotive industry by shortening the development time of vehicle ECUs and lowering the number of prototype iterations. If it is necessary to cover more than one section of embedded system simulation, then certain combinations of the evaluated software suites are possible to achieve through co-simulation. Further research is required in this direction.

### Acknowledgements

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002).

### References

Amringer N., Asemann P., 2022, Simulation of Virtual ECUs in the context of ECU Consolidation. Proceedings of the 9th AutoTest Technical Conference, Test of Hardware and Software in Automotive Development. <https://www.researchgate.net/publication/364694688_Simulation_of_Virtual_ECUs_in_the_context_of_ECU_Consolidation>, accessed 25.10.2024.

Damle A., Nayak O., Gole A., Sinkar A., 2022, Control Systems Testing with a Flexible Co-simulation Interface to PSCAD/EMTDC. 2022 22nd National Power Systems Conference (NPSC), New Delhi, India, 166-171.

Kim H., Lee H., Cho J., 2023, A FMI-based Approach for CAN Bus Simulation and Simulink Model Integration in Vehicle Simulation Environment. 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 279-284, DOI: 10.1109/ICTC58733.2023.10393858.

Misbin R., George A., 2023, QEMU-Based Emulation-in-the-Loop for the Simulation of Small Satellite Flight Software. 2023 IEEE Aerospace Conference, 1–8, DOI: 10.1109/AERO55745.2023.10115569.

Schweiger G., Nilsson H., Schoeggl J., Birk W., Posch A., 2020, Modeling and simulation of large-scale systems: A systematic comparison of modeling paradigms. Applied Mathematics and Computation, 365, 124713, DOI: 10.1016/j.amc.2019.124713.

Yang A., Kim J.Y., Seol W.H., Han W.J., Kim H.R., Cho J., 2023, A Method for Reducing Simulation Timing Deviation in QEMU-Based Virtual ECU, 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 265-270, DOI: 10.1109/ICTC58733.2023.10393554.