High-speed real-time digital emulation for hardware-in-the-loop testing of power electronics: A new paradigm in the field of electronic design automation (EDA) for power electronics systems

Michel Kinsy, Dusan Majstorovic, Pierre Haessig, Jason Poon, Nikola Celanovic, Ivan Celanovic, Srinivas Devadas

High-Speed Real-Time Digital Emulation for Hardware-in-the-Loop Testing of Power Electronics: A New Paradigm in the Field of Electronic Design Automation (EDA) for Power Electronics Systems

Michel A. Kinsy, Dusan Majstorovic*, Pierre Haessig, Jason Poon, Nikola Celanovic*,
Ivan Celanovic, and Srinivas Devadas
{mkinsy@mit.edu}
Massachusetts Institute of Technology, Cambridge, USA.
* Faculty of Technical Sciences, Novi Sad, Serbia.

Abstract

This paper details the design and application of a new ultra-high speed real-time emulation for Hardware-in-the-Loop (HiL) testing and design of high-power power electronics systems. Our real-time hardware emulation for HiL system is based on a custom, heterogeneous, reconfigurable, multicore processor design that emulates power electronics, and includes a circuit compiler that translates graphic system models into processor executable machine code. We present the digital processor architecture, and describe the process of power electronic circuit compilation. This approach yields real-time execution on the order of 1μs simulation time step (including input/output latency) for a broad class of power electronics converters. We present HiL experimental results for three representative systems: a variable speed induction motor drive, a utility grid connected photovoltaic converter system, and a hybrid electric vehicle motor drive.

1 INTRODUCTION

Power electronics is one of the key technologies enabling wider proliferation of renewable energy generation and realization of the smart grid vision. Power electronics could potentially reduce overall electricity consumption by more than 30% [1]. In order for power electronics to reach this point, advancements are needed in the area of *Electronic Design Automation (EDA)* tools, among others. Design automation could significantly reduce development cycles, improve reliability, and accelerate the development of more complex systems. In particular, we believe that an integrated *EDA* tool would immensely benefit the area of automated control testing and rapid design prototyping.

In this paper, we propose a new software/hardware co-design *EDA* platform for comprehensive testing and validation of power electronics controller hardware, firmware, and software performance by means of high-fidelity, real-time emulation of power electronics in *Hardware-in-the-Loop (HiL)* configuration. In HiL testing configuration, the power electronics converter is replaced by an ultra-fast real-time emulator that interacts with the controller via high-speed physical input/output interfaces. From the controller point of view, the emulation is so seamless that there is no distinction between the real-time control of the physical plant and the hardware emulator. The paper is organized in five sections. Section 2 frames the challenges facing the design automation and testing of power electronics. Section 3 describes the proposed framework for real-time hardware emulation. Section 4 compares the fidelity of the proposed real-time emulator with the real power electronics system, and presents two more design examples. Section 5 summarizes the paper.

2 DESIGN REQUIREMENTS FOR HiL POWER ELECTRONICS

2.1 Benefits of Hardware Emulation for Power Electronics

Today, power electronics engineers mostly rely on off-line simulations in the early design stage, and on hardware prototypes–low voltage simulators–in the final stages of the design and testing. However,

off-line simulations are generally slow and only provide a certain level of functional faithfulness, especially when modeling embedded controller and its interactions. Hardware emulation in *HiL* configuration enables more realistic testing while reducing the complexity and cost. The key benefits of the real-time emulation for HiL testing are: (1) accelerated testing and validation; (2) reduced testing time needed in the lab; (3) simulation of all operating points and scenarios that are difficult or impossible to recreate with a real system (4) fault injection capability; (5) real-time access to all signals that are difficult to measure in a real system.

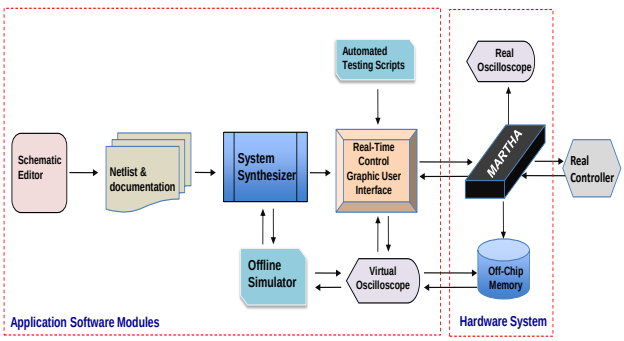
2.2 Hardware Emulation: A Known Design Concept in Design Automation

The automotive industry has been using HiL simulation as a ubiquitous tool for testing the engine control unit (ECU). Connecting an ECU to a real-time simulator of a car enables exhaustive testing of ECU. In computer engineering, hardware emulation, or in-circuit emulation, is used in processor design by using another system (hardware emulator) that imitates the behavior of the system under test. In power electronics, replacing the power converter, grid, and electromechanical components with a real-time digital emulation takes high-power hardware out of the testing environment with minimal loss of fidelity (from the controller point of view). However, high-fidelity testing of power electronics requires a real-time emulation system with: (1) ultra-low latency and high-speed simulation ($\sim$1μs); (2) ability to simulate switched and nonlinear systems; (3) flexibile modeling; (4) ease of use, and variable levels of modeling abstractions. Most HiL simulators, based on off-the-shelf processors, exhibit large latencies on the order of 50μs. Yet, power converters operate at switching frequencies of $\sim$10kHz, thus limiting the fidelity and applicability of HiL platforms based on the standard processors.

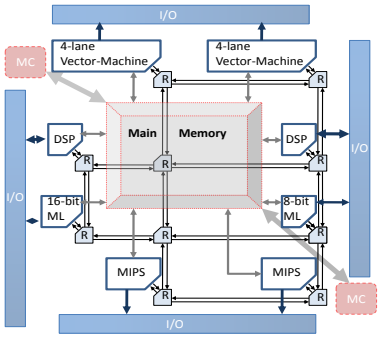
3 HiL SYSTEM ARCHITECTURE

We present a new design automation framework based on a unified software/hardware platform that spans the design space from off-line simulation and real-time hardware emulation, to design prototyping, and testing. The proposed software simulation platform is used for offline tasks, such as power electronics converter modeling, analysis, and optimization. System simulations can automatically be ported to the hardware platform that can serve as an accelerator. In addition, the hardware platform is used for real-time emulation and rapid system prototyping. Our approach to high-speed, low-latency simulation of power electronics is based on a switched hybrid system framework, which is suited for modeling of power electronics, and also lends itself to digital architecture implementation [2]. The real-time hardware emulator system consists of two main components:

- Application software: including a schematic editor, circuit compiler, and emulation controller;
- Custom-designed multicore hardware system: including real-time emulation processors with fast analog/digital input/output interfaces.



(a) Software/Hardware Architecture for Power Electronics Emulation and Rapid Prototyping.



(b) Custom-designed Multicore Architecture for Real-Time Hybrid Applications (MARTHA).

Figure 1: HiL Software/Hardware System Architecture.

3.1 Modeling Approach: Switched Hybrid Automaton

The proposed switched hybrid system approach to power electronics converter modeling relies on piece-wise linear passive elements, piece-wise linear switches, current and voltage sources (both controlled and idependent). The switched hybrid system model is given in the state space form, as:

$$\dot{x}(t) = A_q x(t) + B_q u(t) \tag{1}$$

where $x(t) \in X \subset R^n$ is the continuous state space vector, $A_q \in R^{n \times n}$; $B_q \in R^{n \times m}$; and $u \in R^m$ is the input vector. Any discrete state of the circuit belongs to a finite set $Q = \{q_1..q_n\}$ and further defines the given state space representation. Every discrete state q_i therefore has a unique dynamic behavior associated with it that defines the operation of the circuit. In this framework one can also define a mode, denoted m_q, where $q \in Q$, is the operation of the system defined by given state space $\dot{x}(t) = A_q x(t) + B_q u(t)$ and a given q.

Definition 1 (Switched Hybrid Automaton) A switched hybrid system is a collection $H = (Q, X, f, E, G)$ where: $Q = \{q_1..q_n\}$ is set of discrete states, $X \subseteq R^n$ is the continuous state space; $f : Q \mapsto (X \mapsto R^n)$ assigns to every discrete state continuous vector field on X; $E \subseteq Q \times Q$ is a collection of discrete transitions; $G : E \mapsto 2^X$ assigns each $e = (q, q') \in E$ a guard.

The state space representation of hybrid automaton modes, as defined in Equation 1, can be discretized. We use the exact discretization method via state-transition matrix. The discretized state space system matrices, for a given mode are given as:

$$\dot{x}(k+1) = A_{d(q)} x(k) + B_{d(q)} u(k) \tag{2}$$

$$y(k) = C_{d(q)} x(k) + D_{d(q)} u(k) \tag{3}$$

$$g(k) = C_{gd(q)} x(k) + D_{gd(q)} u(k) \tag{4}$$

where k is the discrete time, y is the output vector, g is the guard vector (it is treated as the output vector). In the discretized form the set of matrices $\{A_{d(q)}, B_{d(q)}, C_{d(q)}, D_{d(q)}, C_{gd(q)}, D_{gd(q)}\}$ defines the dynamic behavior of the system.

3.2 Application Software

Application software is the interface for specifying the system that is emulated in hardware. After the system is specified in the schematic editor, the circuit compiler analyzes the circuit, and extracts the full switched hybrid system representation, followed by compilation and linking of the model into the processor executable binary file. The simulation controller loads the binary file into the processor and provides control of the simulation via start/stop commands, etc. and configures the “in-circuit” probing.

Application Software Implementation

The software environment comprises three layers: (1) the power electronics modeling layer, (2) formulation and analysis layer, and (3) the synthesis and simulation layer. In the software domain, we focus on abstraction, structure, and modularity. Figure 1(a) shows the software modules and the data flows. The first layer consists of the schematic editor and the circuit netlist abstractor. The *Schematic Editor* library components include piece-wise linear circuit elements (e.g. resistors, inductors, capacitors), machine models, semiconductor devices (e.g. diodes and IGBTs), independent and controlled sources, contactors and predefined power electronics switching blocks(e.g., a three-phase two-level voltage source inverter etc.) and control blocks. Given a power electronics circuit, the *Netlist Extractor* module generates a list of components, values, and connectivity. The hybrid model formulation is done through the *System Synthesizer* module, which generates the set of state space matrices, modes, transition equations, guards, etc. The synthesizer output files can be used by the offline simulator, or by the real-time controller to be mapped onto the the hardware. The *Offline Simulator* module uses hybrid model formulation (both preset and actively user controlled), and outputs time-domain signals via the *Virtual*

Oscilloscope module. The *Real-Time Control Graphic User Interface* module allows user to run the model in real-time on the hardware, and to control the emulation manually or via script.

3.3 Multicore Hardware Architecture

We develop a domain-specific digital processor architecture that consists of two sets of computational elements: high-speed, low-latency, dynamic system state space solver (discretized with fixed time-step), vector cores, and reduced instruction set computing (RISC) cores that control simulation and input/output access. The architecture guarantees combined real-time emulation latency and execution times of 1μs.

Hardware Implementation

Figure 1(b) shows the macro-view of the architecture. Processing cores are connected via an irregular, two dimensional mesh network of bufferless reconfigurable routers. The number of any class of processing cores (e.g., vector-machines, DSPs, RISC cores) is expandable, and the interconnect network is scalable. The architecture uses two techniques to achieve parallelism: namely, Single Program-Multiple Data (SPMD), where the two autonomous vector processors can simultaneously execute the same program on different data sets, and Multiple Program-Multiple Data (MPMD) style approach, where the different processors and micro-processors simultaneously operate on independent programs. The architecture comprises: two four-lane chained vector-machine cores for fast and parallel matrix computation [3, 4], two RISC cores, two digital signal processors (DSP), a 16-bit programmable microprocessor, an 8-bit programmable microprocessor, a bank style main memory, with no cache for deterministic data access [5], and two memory controllers for off-chip communication.

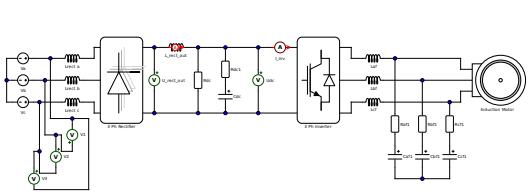
The performance of the proposed hardware architecture is verified on a Xilinx ML506 FPGA development board, with a custom-designed analog/digital I/O board for the controller under test and user outputs. Our current design roughly uses 90% of the slices and BRAMs on the FPGA, and runs at $200MHz$.

4 EXPERIMENTAL RESULTS AND APPLICATION EXAMPLES

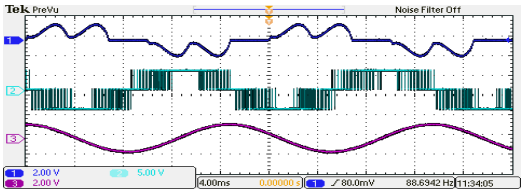
To demonstrate the fidelity and the versatility of our full electronic systems design and automation approach, and our integrated software/hardware system solution, we present Hardware-in-the-Loop simulations of three representative systems: namely, a variable speed induction motor drive, a utility grid connected photovoltaic converter system, and a hybrid electric vehicle motor drive.

4.1 Variable Speed Induction Motor Drive

The variable speed drive consists of a three-phase voltage source, a three-phase diode bridge rectifier, a DC-link, a three-phase two-level voltage-source inverter, an output filter and an induction machine.

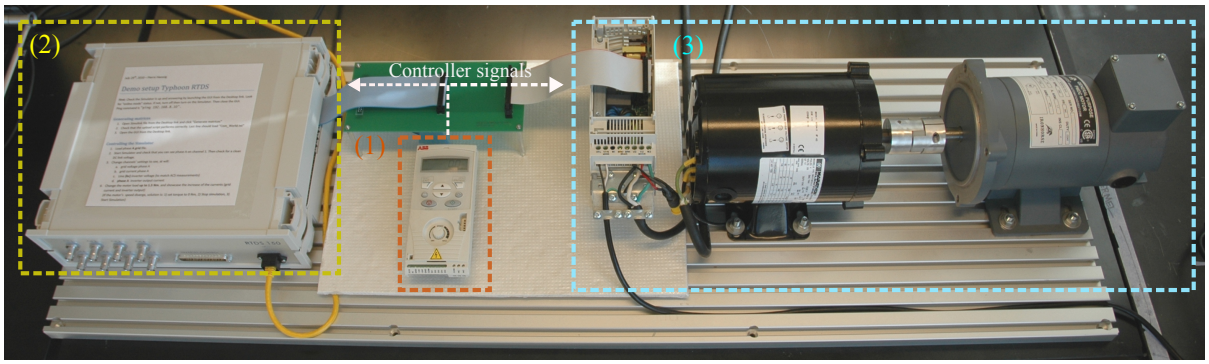


(a) Schematic diagram of the motor drive model of a three-phase rectifier, two-level voltage source inverter, filter and induction machine.



(b) Experimental measurements of the real-time emulation system rectifier input current, line to line inverter voltage, and stator current (channels 1, 2, and 3).

Figure 2: Variable Speed Induction Motor Drive Application Experimental Results.

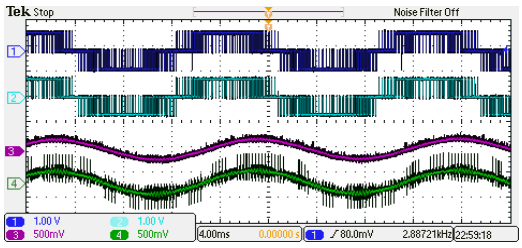


(2) HIL platform simulating variable speed drive connected to three-phase

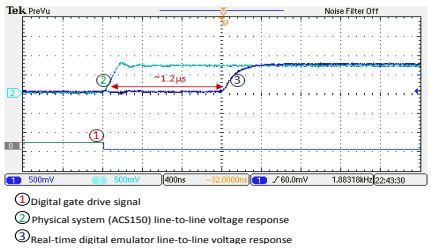
(1) ACS-150 variable speed drive controller. Device-under-test (DUT)

(3) Variable speed drive converter connected to three-phase induction machine.

Figure 3: Motor-drive experimental setup for parallel physical system and real-time digital emulator testing.



(a) Experimental measurements of the line-to-line inverter voltage and the motor phase current of the physical system are shown on channels 2 and 4, and the matching real-time hardware emulator waveforms are on channels 1 and 3.



(b) Three-phase inverter line-to-line voltage response on the physical system, and real time emulation system to a gate drive signal.

Figure 4: High-Fidelity Emulation of Variable Speed Induction Motor Drive Application.

To illustrate the fidelity of our real-time digital emulation platform, we run in parallel an industrial grade open-loop variable speed drive, ABB ACS-150, and our real-time hardware emulator, both driven from the same controller. Figure 2 shows the system comprising three-phase diode rectifier and three-phase two-level IGBT inverter. Figure 3 shows the experimental setup, where the same controller is used for both the real system and the real-time emulated system. Measured waveforms on both the real system and the emulator are shown in Figure 4(a) with almost one-to-one matching. Emulator response latency is measured to be less than(1.3μs), as shown in Figure 4(b).

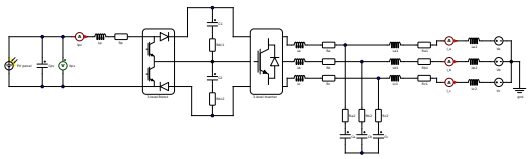
4.2 Utility Grid Connected Photovoltaic Converter System

To illustrate, that beyond the real-time high-fidelity, our EDA environment also provides support for flexible modeling, design, prototyping, and HiL testing for a wide range of applications, we present another example system: a grid connected photovoltaic converter system.

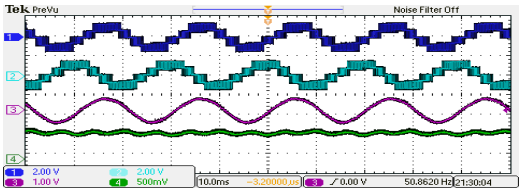
Figure 5(a) shows a two-stage utility grid connected photovoltaic converter model, that consists of a photovoltaic panel module, boost converter, a three-level NPC inverter, an input filter, and a grid. Figure 5(b) shows voltages V_{ab} and V_{bc} on channels 1 and 2, respectively.

4.3 Hybrid Electric Vehicle Motor Drive

In recent years, we have witnessed a significant increase in the development efforts focused on hybrid and electric vehicle drivetrains. Being a safety critical application, these system require ever increas-



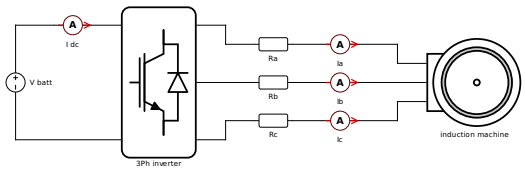
(a) Schematic diagram of the PV panel, boost converter, three-level voltage source inverter, filter and utility grid.



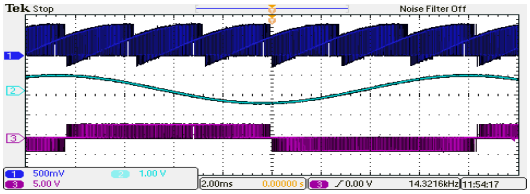
(b) Experimental measurements of the real time emulator variables, line to line inverter voltages, output current, and upper DC-link voltage.

Figure 5: Utility Grid Connected Photovoltaic Converter System Application.

ing attention to testing and verification of power electronics designs. Standard vehicle HiL simulators can provide good fidelity for mechanical systems while lacking speed and latency to simulate power electronics subsystems. In this example, we present HiL simulation, based on the proposed platform, of a typical electric vehicle drivetrain with an induction machine, two-level voltage source inverter, and battery as shown in 6. High-fidelity HiL simulation enables engineers to observe all the fine details, e.g.: DC-link and battery current, induction machine currents, even common mode voltages.



(a) Schematic diagram of the hybrid electric vehicle motor drive with battery model, two-level voltage source inverter, and induction machine.



(b) Experimental measurements of the real time emulation system waveforms, DC link current, rotor current, and inverter output line-to-line voltage.

Figure 6: Hybrid Electric Vehicle Motor Drive System Application.

Figure 6 shows a simplified voltage-source inverter model of the system, with switching frequency of 4 kHz.

5 SUMMARY AND CONCLUSIONS

This work proposes an industrial grade real-time emulation platform based on a novel digital processor design and a software tool chain that makes it flexible, easy to use, and scalable. The proposed platform brings a true emulator performance to the control design and test engineers desks. It enables a high-fidelity (with 1μs latency and simulation time-step), safe, and fully realistic testing and validation of the most detailed aspects of power electronics systems.

References

[1] Fed C. Lee and et. all. Power electronics system integration–a cpes perspective. *14th IEEE Power Electronics Society Newsletter*, 20, 2008.

[2] Matthew Senesky, Gabriel Eirea, and T. Koo. Hybrid modelling and control of power electronics. 2623:450–465, 2003.

[3] Krste Asanovic. Vector microprocessors. 1998. AAI9901978.

[4] D. Bhandarkar and R. Brunner. Vax vector architecture. pages 204 –215, May 1990.

[5] Bryce Cogswell and Bryce Cogsweil. A cacheless architecture for real time systems. 1990.