Interfacing Issues in Real-Time Digital Simulators

IEEE Task Force on Interfacing Techniques for Simulation Tools

W. Ren, M. Sloderbeck, M. Steurer, V. Dinavahi,
T. Noda, S. Filizadeh, A. R. Chevrefils, M. Matar, R. Iravani, C. Dufour, J. Belanger, M. O. Faruque,
K. Strunz, and J. A. Martinez

***Abstract*—This paper deals with the current state-of-the-art in interfacing issues related to real-time digital simulators employed in the simulation of power systems and power-electronic systems. This paper provides an overview of technical challenges encountered and their solutions as the real-time digital simulators evolved. Hardware-in-the-loop interfacing for controller hardware and power apparatus hardware are also presented.**

***Index Terms*—Field-programmable gate arrays (FPGAs), hardware-in-the-loop simulation, interface algorithms, modeling, power-electronic systems, power system simulation, real-time systems.**

I. Introduction

HARDWARE-IN-THE-LOOP (HIL) simulation is being increasingly used as an important design and development step in the manufacturing process of many industries [1]–[6]. HIL simulation provides a means for the operation of physical hardware, such as power components and control hardware, while interfaced to a computer simulation of the system in which the physical hardware is intended to function. To achieve meaningful results, the computer simulation must proceed in real-time and in a synchronous manner, meaning that each simulation time-step corresponds exactly with the equivalent wall clock time. Furthermore, the simulation time-step must be sufficiently small both to accurately represent the dynamic and transient behavior of the simulated system, and to provide for adequate control and measurement of the physical hardware.

The HIL approach provides several advantages that other analysis and testing methods do not provide [2]–[4]. It allows for power system apparatus to be investigated repeatedly and thoroughly in a true-to-nature test condition even before the actual system is built and commissioned. It also minimizes the cost and risk to examine various extreme conditions and maximizes the likelihood to identify hidden defects in the apparatus before their impact manifests in actual operation, which in turn may have serious consequences. Moreover, the HIL approach inherently incorporates the transient response of the yet-to-be-built system to stimulus by the hardware. Therefore, this method has the potential to reveal the full extent of system interactions to be expected in the final design.

The HIL system is generally composed of three indispensable parts, namely: 1) a piece of hardware under test; 2) a simulated system; and 3) an interface that links the hardware and the simulated system. Depending on the power level of the interface [4]–[6], the HIL simulation can either be categorized as controller HIL (CHIL) simulation or as power HIL (PHIL) simulation. In a CHIL simulation, the hardware under test is a controller, which exchanges signals with the simulated system at a low power level. Examples of CHIL simulation include protection relay testing under simulated fault scenarios, power electronics controllers operating with simulated motor drives, and an electronic engine control unit reacting to an automobile engine simulation. In contrast, the hardware under test in a PHIL simulation involves actual power devices that require significant power flow between the hardware and the simulation system. In these conditions, specially designed power amplifiers and actuators become necessary to establish the interface. Examples of PHIL simulation are propulsion motor testing on a simulation electric ship system, operation of a real motor drive circuit on various simulated motors, and a distributed generator connecting to a simulated utility grid. The difference in the interface structure of typical CHIL and PHIL simulations is illustrated in Fig. 1. To establish and conduct an HIL simulation requires extensive, intricate care. Due to the real-time requirements and the disturbances introduced by the interfaces, issues, such as system stability and simulation accuracy, must be carefully considered.

The paper is organized as follows: Sections II and III discuss the interfacing issues related to CHIL and PHIL simulations, respectively. Conclusions are given in Section IV.

II. Controller Hardware-in-the-Loop Simulation

A. Basic Input and Output Requirements

In addition to the requirements of real-time synchronization and an appropriate time step, accurate inputs and outputs (I/O), both analog and digital, must be provided by the simulator system. This I/O should have a known latency and synchronization to the simulation time-step. In certain cases, it may also be necessary to compensate for the error caused by the

Manuscript received July 07, 2010; accepted August 24, 2010. Date of publication November 11, 2010; date of current version March 25, 2011. Paper no. TPWRD-00518-2010.

Task Force on Interfacing Techniques for Simulation Tools is with the Working Group on Modeling and Analysis of System Transients Using Digital Programs, IEEE Power and Energy Society, T&D Committee. Task Force members: U. Annakkage, G. W. Chang, V. Dinavahi (Task Force Chair), S. Filizadeh, A. M. Gole, R. Iravani, J. Jatskevich, A. J. Keri, J. Mahseredjian, J. A. Martinez, A. Monti, L. Naredo, T. Noda, A. Ramirez, M. Rioual, M. Steurer, K. Strunz

Corresponding author: V. Dinavahi is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: dinavahi@ece.ualberta.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRD.2010.2072792

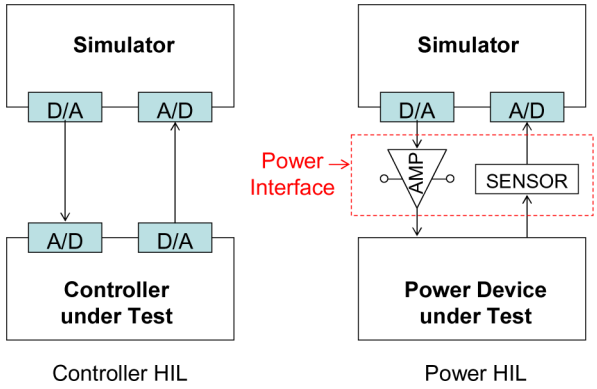

Fig. 1. Controller HIL simulation versus power HIL simulation.

mid-time-step change of an external input to the simulator. To reduce costs, use of low voltage (±10 V) analog-to-digital (A/D) and digital-to-analog (D/A) converters is typical. Digital I/O of 5 V and 3.3 V are also common; however, more emphasis should be made regarding 3.3 V digital I/O, as 5 V becomes less common. An early and continuing use of HIL simulation in the electric power industry has been the testing of commercial relays. In this case, a simulated power system, with simulated events such as faults, is interfaced to a physical relay. The inputs measured by the relay in the simulated environment are identical to what would be seen in the real world, enabling the manufacturer and relay engineer to explore the operation of a physical device without impacting to an actual power system.

Since the relay is usually connected to current transformers (CTs) and/or potential transformers (PTs), which are quite different from the low-voltage D/A provided by the simulator, a challenging interfacing issue arises. One cumbersome solution is to amplify the simulated three-phase signals via external amplifiers before connection to the physical relay. Fortunately, relay manufacturers have begun to provide low-voltage inputs to their hardware that are algorithmically equivalent (with the proper internal scaling) to the CT and PT inputs. However, even in currently available relays, a voltage level of 120 V may still be required for a status signal, and the trip signal may also be higher than the more common computer digital I/O levels of 5 V and 3.3 V. Options for alternate signal voltage levels should then be provided.

If the simulation device accepts pulse-width modulation (PWM) firing pulses from external controllers, the ability to accommodate different firing voltages of multiple manufacturers can be accomplished by substituting resistor packs in an optically isolated circuit.

B. Communication Capability

The engineering community can leverage capabilities beyond the world of analog feedback and control values. As communication speeds and the extent of networking continue to increase, new methods begin to emerge in HIL simulation. For example, a simulation involving phasor measurement units (PMUs) could model them, or alternatively acquire data from a PMU in a test bed or functional power system. Defined protocols, such as the IEEE C37.118 IEEE Standard for Synchrophasors for Power Systems, can allow interoperability over an IP-based Ethernet.

There is still justification for supporting older protocols such as Modbus/TCP and DNP3 (Distributed Network Protocol), but as the community migrates to support IEC 61850, simulation platforms must also migrate. In the context of real-time HIL simulations, asynchronous best-effort network protocols may seem like a contradiction. In fact, external protocols complement the baseline digital and analog I/O of an HIL simulation.

A salient feature of the IEC 61850 interface in the real-time simulator is that it eliminates the need for amplifiers to interface the real-time simulator with protective relays. Thus, significantly reduces the cost and simplifies the connection between the simulator and the relay under test [7].

The main challenges associated with the IEC 61850 interface implementation are to 1) minimize the I/O latency and 2) maintain a low variability for that delay. One approach to achieve these requirements is to implement the IEC 61850 interface using a processing platform dedicated solely for handling IEC 61850 communications [7]. This is the approach adopted in the RTDS® simulators. The RTDS® is the only reported simulator that implements IEC 61850 for real-time simulation [7].

C. Power-Electronic System Simulation and Interface Algorithms

Accurate simulation of power-electronic systems in real-time simulators has been a considerable challenge for a long time. In the context of CHIL simulation, the main issue related to power-electronic system simulation is the synchronization of switching signals with the discrete time-step of the real-time simulator [37], [38]. Due to its discrete-time nature, the digital controller outputs digital signals which are not necessarily in synchronism with the time grid of the simulator. Since the simulator, unlike an actual system, cannot instantaneously respond to the digital output of the control platform and only accounts for it at the end of the encountered simulation time-step, errors are introduced to the simulation results. When a digital controller is interfaced with a real-time simulator that models a power-electronic system, the simulator is required to accurately account for multiple inter-step switching events, which arrive at its input between two calculation cycles. Fig. 2 illustrates the fixed time-step simulation approach with two switching events occur within one time-step. The number of switching events within any given time-step of the simulator depends on three factors: 1) the switching frequency of the power-electronic system (i.e., its digital controller sampling rate); 2) the complexity of the power-electronic system (i.e., the number of switches in the system); and 3) the ratio between the simulation time-step and the switching period [10]. The higher the switching frequency or the larger the system is, the higher the number of interstep switching events will be. Furthermore, the timing of an incoming switching event is not known *a priori* (since it is controlled by processes external to the simulator) and it seldom coincides exactly with the simulator time-steps, thereby creating a switching delay, which produces erroneous simulation results. Under non-real-time conditions, the traditional approach is to use a small integration step-size, however, at the cost of significantly longer simulation times [39]. Interpolation techniques have also been devised and

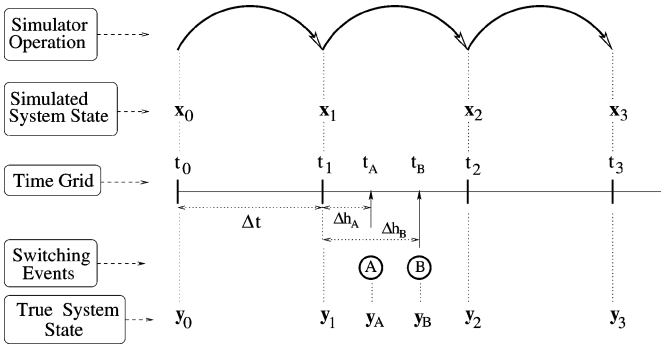

Fig. 2. Inter-step switching events in power-electronic system simulation.

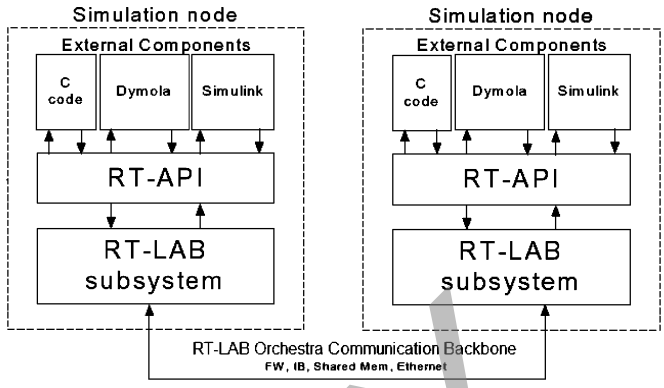

Fig. 3. RT-LAB Orchestra implementation layers.

implemented in non-real-time simulators to compensate for the time delay in switching instants.

Over the years, several switching event compensation algorithms have been proposed and implemented in real-time simulators [37], [38], [12]. The compensation methods included numerical techniques, such as linear interpolation and extrapolation. Generally, these compensation algorithms function based on a two-step procedure: 1) determine the values of the system variables at the instant of switching and 2) resynchronize the simulator outputs with the simulation time grid [9], [13]–[16]. To implement these compensation algorithms in real-time, a switching event capture or time-stamping mechanism is required [40]–[42].

Alternatively, interpolation algorithms based on switching-functions can also be very efficient for transient calculations of power-electronic and motor drives [20]–[23]. The approach enables industrial researchers to design and develop and to test using virtual benches. In [20] and [21], a two-level PMSM drive system was tested using this approach. In [22], a three-level neutral-clamped-point inverter-based induction machine drive was tested using this approach. These tests include special modes like pulse shutdown and rectification. This approach can notably be extended for very complex grid devices (e.g., 48-pulse GTO STATCOM [23]) and used within a real-time simulator such as the one described in [18].

D. Co-Simulation Capability

In some simulation environments, multiple simulation platforms/software may be combined for reasons of computational or software specificity, such as multirate or multidomain simulations, to take advantage of a particular software capability, to incorporate and reuse legacy code due to IP protection issues. It can be advantageous to provide for high-speed signaling between computational components from different manufacturers, using a defined signaling protocol. Co-simulation capability can then be created.

Multiple simulators may then be interfaced with the usual analog and digital I/O capabilities, but the information bandwidth due to the number of available interfaces may be a limiting factor. Co-simulations with relatively slow updates may be accomplished via IP-based protocols over conventional Ethernet, but this is a limited medium [24].

RT-LAB Orchestra [25] is an example of a software application that facilitates integration and interoperability between co-simulation components. Using the Orchestra API, components connect to an RT-LAB framework and exchange data with other co-simulation components, synchronously or asynchronously. Dymola, "C" code and Simulink executable compiled from S-function are the type of external modules currently supported by Orchestra. The interface protocol is public domain so other simulation tools can be attached.

The core of Orchestra architecture is a user-configurable communication layer, sitting on top of the RT-LAB framework, whose role is to provide a transport layer between distributed simulation nodes, and to act as a real-time scheduler for co-simulation. The communication layer consists of a set of shared-memory segments, one per domain. As a result, co-simulation components are co-located within the same simulation node.

Co-simulation components exchange simulation data via the communication layer by calling functions of the Orchestra RT-API, depicted in Fig. 3. Orchestra distinguishes two types of co-simulation components, namely, the RT-LAB framework itself, and external components. Logically, an external component is a process that is not part of the Simulink-based model that the RT-LAB framework instantiates for real-time execution.

Physically, an external component is a cohesive software entity that embeds calls to the RT-API, and that is compiled and linked to form a stand-alone process. A single domain participant acts as a reader or a writer by using the RT-API to send or receive data, respectively.

E. FPGA-Based HIL Simulation

Field-programmable gate arrays (FPGAs) were first introduced in the mid 1980s. Since then, there has been a steady advance in their performance and density and their cost has progressively decreased, enabling them to compete with other digital processors. Most FPGAs are organized as an array of basic logic elements and programmable interconnections between the logic elements, memory, and I/O pins. Due to inherent parallelism, an FPGA provides superior performance compared to a microprocessor or a DSP. In the context of power system simulation, FPGAs are particularly suited for small time-step transient simulation in real time [10], [26], [27], since sequential processors still pose a speed bottleneck for large-scale system simulation [28] with full system representation.

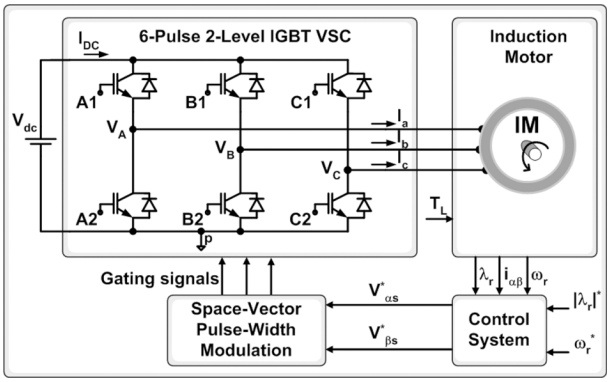

Fig. 4. Induction machine drive system implemented in HIL configuration on an FPGA.

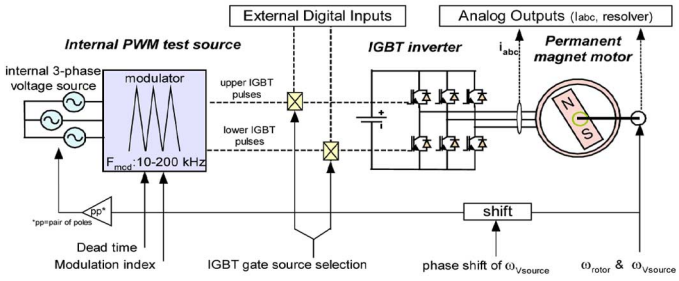

Fig. 5. PMSM drive model implemented on FPGA.

A major advantage of adopting an FPGA as the main computational hardware in a real-time power system simulator is that it enables the design of customized hardware architectures that are tailored to the solution of the mathematical model of the power system under consideration; it also enables the use of submicrosecond simulation time-steps that are not possible with processor-based real-time simulators.

A digital hardware realization of a real-time simulator for a complete induction machine drive (Fig. 4) using a FPGA as the computational engine is presented in [3]. The simulator was developed using Very High Speed Integrated Circuit Hardware Description Language (VHDL), making it flexible and portable. The advantages of this approach include: 1) freedom from reliance on complicated correction algorithms; 2) detailed representation of the device switching characteristics; and 3) the ability to interface the model to a large-scale real-time simulator, such as a PC cluster [18], modeling a larger and more complex host power system in which the power-electronic converter is embedded.

The FPGA simulation of a complete PMSM drive was presented in [29] (Fig. 5). The model was coded using the Xilinx System Generator blockset for Simulink without any VHDL coding. This facilitates the drive design by a non-expert FPGA programmer while retaining code portability by the means of the automatically generated VHDL code.

The drive was implemented to facilitate its testing and hook-up to an external controller. An open-loop internal PWM modulator source with a constant phase difference with respect to the rotor angle is an example of a useful test mode when trying to commission such a system.

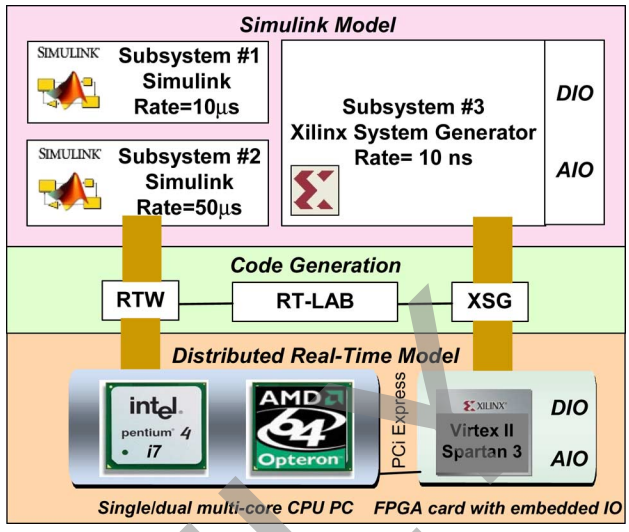

Fig. 6. Workflow structure of the RT-LAB real-time simulator from the model specification to the multi-task real-time execution.

The drive itself is interfaced with a mechanical model running on general-purpose CPUs of an RT-LAB real-time simulator, from Opal-RT Technologies, Inc. This interface with CPU-type computational engines enables the implementation of complex mixed-simulation models using FPGAs only for some parts. The general RT-LAB development and run-time environments used for this model are depicted in Fig. 6.

The simulator can also use finite element analysis results from JMAG to simulate real-time PMSM motors [30].

F. Controller Design Using Real-Time Playback

A real-time playback (RTP) device is capable of generating real-time and synchronized replicas of waveforms produced by non-real-time electromagnetic transient simulators and stored in its memory. A typical application of an RTP device is to test protective relays, where fault waveforms are played back in real time to the protective relay system under study [31].

In a recent adaptation of an RTP, it was used for controller design and development for an advanced vehicular electric motor drive system [32]. The RTP was used extensively for the initial development of the digital controller and for testing and verification of its protective algorithms. The drive system uses a space-vector-modulated (SVM) converter at 10 kHz based on the information provided by a rotor-mounted position sensor.

The motivation for the use of an RTP was to decouple the initial stages of the controller development from the actual motor drive system to minimize the likelihood of damage to the sensitive and expensive drive components and hazard to the vehicle operator. It also allowed safe and convenient testing and tuning of the protective algorithms for rare operating conditions where these algorithms are called.

Several detailed simulation cases of the vehicle were developed in which various operating conditions were simulated. The waveforms generated were then stored in the RTP for subsequent playback into the controller hardware that was suitably interfaced to the RTP. The controller code and its interface to the vehicle were then modified to ensure a proper response is given. The tests conducted include the following stages:

1) General Start-Up Sequence Testing and Debugging: This stage was used for confirming the correct operation of the safety features of the vehicle, such as the throttle position switch, precharge circuit, main relay sequencing, and transmission gear-lever position.

2) A/D Converter Calibration: To confirm correct internal numerical scaling of the software based on simulated transducer output from the RTP.

3) Digital Filtering: Digital filtering techniques were tested by using noisy signals generated by the RTP, to ensure proper operation of the vehicle when it is actually driven on the road.

4) Custom-Developed Math Libraries: The controller used several analog signals as inputs. For example, the analog position sensor algorithm that fed the SVM firing pulse generation was confirmed at this stage. This stage also led to the discovery of a delay introduced by the time needed for the execution of some mathematical functions within the controller code. At times, the delay caused interference to the position-sensor signal and the update-sequence for the SVM firing pulse algorithm. Alternate ways for coding these mathematical functions were then undertaken to expedite their execution and resolve the problem.

5) Firing Pulse Generation Algorithm: The RTP was used for the testing and debugging of the SVM firing pulse generator. Since a variable-frequency SVM algorithm based on the position sensor information was used, coordination of its operation with the position sensor required extensive use of the RTP.

The RTP was also useful for testing and debugging the general flow of the code and its exception-handling procedures, such as buffer overflows and incorrect scaling. The RTP analog waveforms were generated to represent overcurrents pulses, dc bus overvoltages, disconnected sensors, etc. to confirm correct operation of the shutdown and protection algorithms.

G. Future Signal Capabilities

Modern FPGA and DSP chips are now offering multiple LVDS (Low Voltage Differential Signaling) signals at greater than 1 Gb/s per channel. High-speed serial signaling may also be provided with methods such as IEEE 1394b. The mere use of such a system, however, does not create interoperability. An open protocol describing the data structure and transfer is needed.

III. Power Hardware-in-the-Loop Simulation

If the hardware under test in an HIL simulation absorbs and/or regenerates considerable power, specially designed power amplification and conversion apparatuses (e.g., actuator, voltage source converter) must be deployed to interconnect the simulator and the hardware. This is the feature where power HIL (PHIL) simulation differs from CHIL simulation where low-power DA and AD converters can fulfill the requirements of the interface.

As a promising technique for system analysis and prototyping, PHIL simulation has been applied [33] to a variety of electrical applications. In [34], a real-time simulator was developed to drive an industrial amplifier to reproduce different types of power-quality disturbances (e.g., spikes, impulses, interruptions). Equipment, such as personal computers and induction motors, was tested under these transient disturbances. A universal platform for motor drive testing was developed in [35]. In this paper, a real-time simulator was used to model and simulate motors with different characteristics. The performance of any motor-drive under test would then be investigated thoroughly by connecting it to the simulator as it was driving a motor. In [36], the PHIL simulation was applied to study the dynamics of alternative power sources in a dc zonal electrical distribution system. The models of a synchronous generator and a diode rectifier were emulated in a real-time simulator. A dc/dc current converter was applied to couple the simulated system and a hardware load. The simulation model representation and the small-signal stability analysis were discussed in this paper. In [37], a PHIL system with considerable complexity and power rating was studied. It consisted of a notional destroyer-class all-electric ship power system and a commercial propulsion motor. The system's dynamic response during the crashback maneuver was investigated.

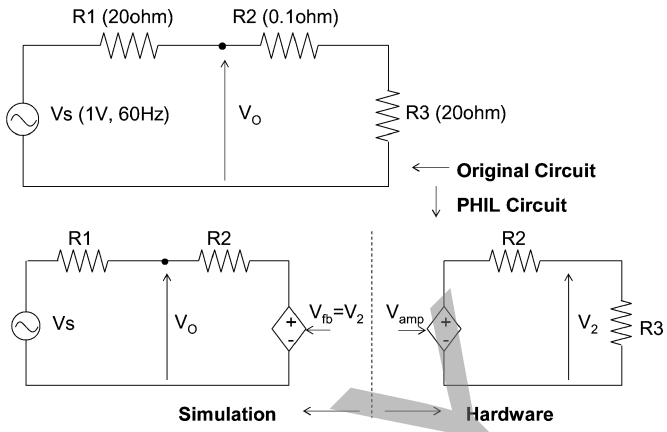

Fig. 7. PHIL example illustrating the simulation accuracy issue.

Ideally, the interface between the hardware under test and the simulated system should have unity gain with infinite bandwidth and zero time delay in order to ensure proper correspondence between the HIL system and the original system. However, such an ideal interface is neither achievable nor affordable in practice. As a result, all HIL simulations contain errors caused by the imperfection of the interface. When the errors are too excessive, the fidelity and, hence, the validity of the simulation are compromised, rendering the result less meaningful or even misleading. This problem is most severe in electrically coupled PHIL simulations because of the challenges with existing technology to build high-power amplification apparatus with satisfactory precision. At times, even high-precision interface amplifiers could result in unacceptably large simulation errors. One such example is given in Fig. 7. If a simple voltage divider circuit, as shown at the top of the figure, is going to be studied with PHIL simulation, one approach for its implementation is the one shown at the bottom of the figure. At the simulation side, the voltage $\mathrm{V_O}$ before $\mathrm{R_2}$ is simulated and amplified by the interface amplifier $\mathrm{V_{amp}}$. As the feedback, the voltage $\mathrm{V_2}$ across $\mathrm{R_3}$ at the hardware side is measured and sent into the simulator where a controllable voltage source represents $\mathrm{R_3}$.

With an ideal interface (i.e., $\mathrm{V_{amp}} = \mathrm{V_O}$ and $\mathrm{V_{fb}} = \mathrm{V_2}$), the PHIL circuit will behave exactly the same as the original

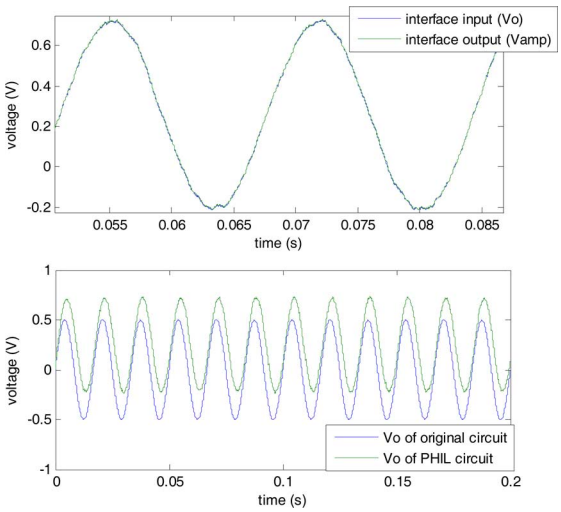

Fig. 8. Simulated result shows large error compared to that of the original circuit.

circuit does. In reality, however, this ideality does not exist. If we assume that a 5 μs time delay and a 0.5% standard deviation of white noise are present in the amplified voltage $\mathrm{V_{amp}}$, although this is still high precision, the resulting PHIL simulation voltage $\mathrm{V_O}$ contains a large dc offset and phase shift compared to the output of the original circuit (Fig. 8).

The example above brings up two questions. First, how can one predict the accuracy of a PHIL simulation when the reference system (either the accurately modeled system or the all-hardware setup) is unavailable? Second, if the performance of an interface amplifier is impossible to be further improved, is there any possibility to improve the simulation accuracy? In the following text, answers will be given to each of these questions.

A. PHIL Accuracy Evaluation

Although there have been a number of successful PHIL applications, only limited research has been performed on the simulation accuracy issue and it remains an open research problem that needs better solutions.

In [38], an idea of using "transparency performance index" to evaluate the fidelity of a PHIL simulation was discussed. This approach compares the difference between the actual and the equivalent subsystems' impedance seen from the other side of the interface. A smaller difference indicates higher transparency of the interface. However, this approach addresses only the precision of the interface and overlooks the closed-loop property of a PHIL system. One can easily find counterexamples (e.g., the system in Fig. 4) in which high transparency can result in large simulation errors.

Two concepts of "performance mismatch (PMM)" and "probability of PMM" were described in [39] to evaluate the simulation performance. But this method requires a lot of approximations and assumptions. Even for the simplest case, it could involve enormous computation, which compromises the applicability of the method.

One effective method for PHIL accuracy evaluation is proposed in [40] and [41]. This method categorized the source of

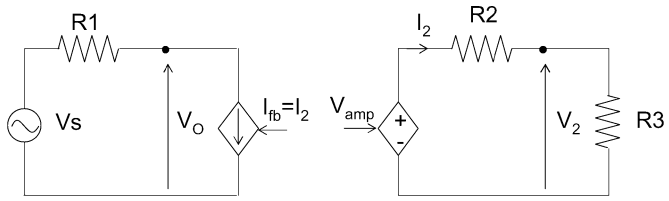

Fig. 9. Different implementation of the previous PHIL system.

the simulation error as from two different types of interface perturbations. For each type, a quantifiable error function is defined in the frequency domain. Solving the error functions revealed that the accuracy of a PHIL simulation is determined by its open-loop transfer function and the non-ideality of the interface amplifier. Although the proposed method is based on linear system analysis, it is shown to be applicable to a wider range of PHIL simulations. For systems whose detailed transfer function is difficult to derive due to complexity or nonlinearity, the method finds out the upper bound of the simulation error as long as the complex or nonlinear parts can be ranged by equivalent gains. However, this upper bound could be overly conservative in some cases when the system is highly nonlinear and the equivalent gains range too widely.

Another method proposed in [42] provides a way to quantify the error bounding of a PHIL simulation in the time domain. This method is based on two basic assumptions. First, the simulated system and the PHIL interface (i.e., actuator, sensor) are linear time-invariant and can be expressed in state space equations. Second, the hardware under test, though nonlinear, is smooth and has an elliptical bounded output. By expressing the simulation error (between the real system and the PHIL system) also in state space equations, its ellipsoidal bound can be derived from the Minkowski sum [41] of the hardware output bound and the system state bound. This method has the same limitation as the one in [40]. When the hardware under test has high nonlinearities and the ellipsoidal bound has to be very large, the resulting error bound will be too conservative to be meaningful.

B. Interface Algorithms

The interface of a PHIL simulation can be implemented in more than one way and each different way could result in different simulation accuracy. To show this, an example is given in Fig. 9. The same PHIL system in Fig. 4 is implemented differently. Instead of feeding back the voltage $\mathrm{V_2}$, current $\mathrm{I_2}$ injected in the hardware is measured and sent to the simulator where a controllable current source represents $\mathrm{R_2}$ and $\mathrm{R_3}$. With no change to the precision of the voltage amplifier (i.e., still 5 μs time delay and 0.5% white noise), this new setup results in much higher simulation accuracy as illustrated in Fig. 10 by its good match between the simulated and the original voltage $\mathrm{V_0}$. The ways of the interface implementation are referred to as the interface algorithm.

Early in 1995, this interfacing issue had been discussed [43] in expanding analog HVDC simulators capability by interfacing with a real-time simulator, from RTDS Technologies Inc. The performances of three different interface algorithms were compared and an optimal one was suggested for the application. In [44], a summary of several existing interface algorithms was

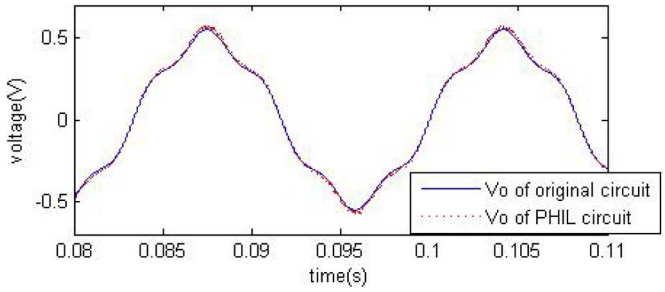

Fig. 10. More accurate simulation result from the revised PHIL simulation.

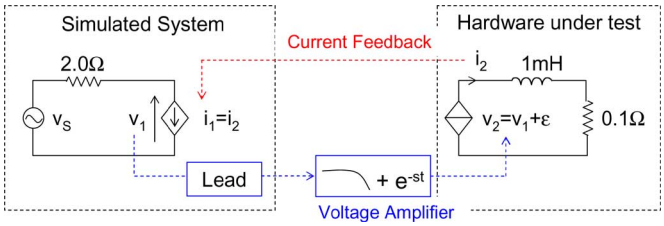

Fig. 11. Illustrative interface compensation application.

given and a new algorithm based on first-order approximation of the hardware-under-test was proposed. However, except for the experimental results, neither work provided an analytical explanation to the different behaviors of the interface algorithms. It is not clear which interface algorithm will result in higher simulation accuracy and under what conditions, and what performance index should be considered in designing a better interface algorithm in the future. To address this problem, a theoretical analysis was performed in [41] and [45]. The relationship between the simulation accuracy and the interface algorithm is for the first time established on a systematical base. It was revealed that different interface algorithms influence the PHIL simulation by altering the system's open-loop transfer function. The system's open-loop transfer function, especially its magnitude, has a direct relationship to the system stability (small gain theorem) and to the simulation accuracy (the damping factor of a simulation error in the close loop). Therefore, it can be utilized as the performance index for interface algorithms.

C. Interface Compensation

Since a PHIL simulation inherits the flexibility of the software simulation, various function blocks can be easily implemented in the simulation to preprocess a signal before it is sent to the interface. This leads to the possibility to alter the overall performance of a PHIL interface by inserting certain compensators. For example, assume the transfer function of an interface is $G(s)$. Adding a compensator with a transfer function of $G^{-1}(s)$ will idealize the interface to unity gain. Although the inversion of $G(s)$ is not always rational (e.g., the inversion of the time delay) in reality, implementing certain compensation can still be effective somehow.

It should be noted that interface compensation differs from the interface algorithm. Interface compensation does not alter the PHIL system topology but instead inserts a function block in the path of an interfacing signal in order to compensate for the time delay, the injected noise, or the magnitude attenuation in

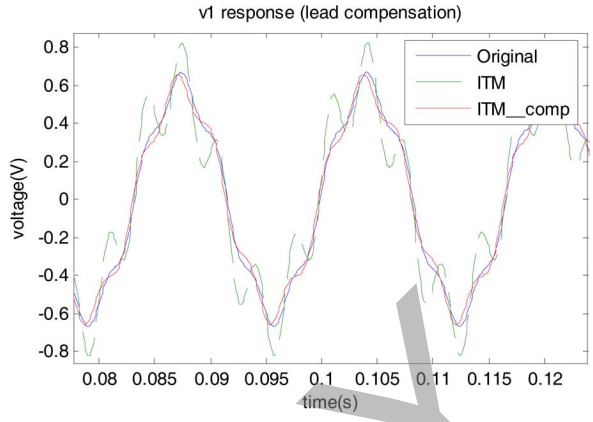

Fig. 12. Simulation result (v_1) before and after interface compensation.

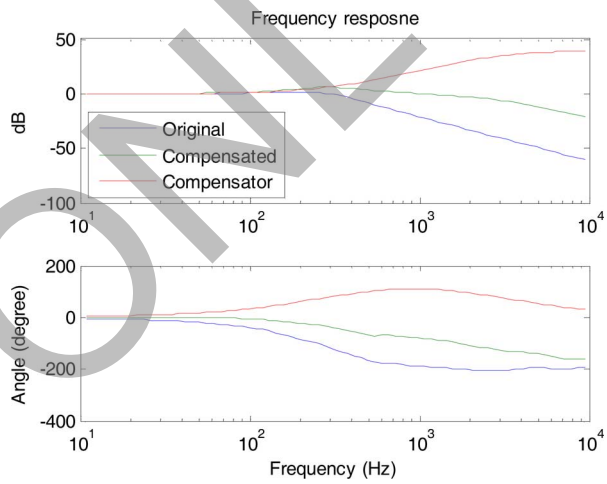

Fig. 13. Overall interface frequency response before and after compensation.

the interface. To understand how interface compensation works, consider the PHIL system in Fig. 11. In this setup, the simulated voltage v_1 is reproduced by an amplifier, and the load current i_2 is fed back in the simulator. From experiment, it is found that the characteristics of the voltage amplifier can be expressed as a pure time delay plus a low-pass filter. To decrease the simulation error caused by this imperfection, a lead compensator is inserted before the voltage amplification.

Assume the transfer function of the voltage amplifier is

$$G_{\text{int}} = \frac{1}{1 + s/1800 + (s/1800)^2} e^{-s*300e-6}. \quad (1)$$

By adding the lead compensator in (2), the simulation result is significantly improved as illustrated in Fig. 12. A comparison between the overall interface frequency responses before and after the compensation (Fig. 13) shows how the compensated interface approaches ideal unity gain

$$G_{\text{comp}} = \left(\frac{1 + s/1800}{1 + s/18000} \right)^2. \quad (2)$$

Some other examples of interface compensation include extrapolation prediction to compensate for time delays and the band-pass filter to block interface noises. In [46], the large

power factor error caused by time delay in a PHIL simulation was reduced by phase shifting the feedback current.

D. Interface Protection

In extreme conditions, a simulation error could be so large to drive the simulation into instability and even damage the testing hardware. For this reason, it is necessary to incorporate suitable protection schemes in the PHIL interface to prevent these disasters from occurring in the first place. An interface protection scheme can be carried on at two layers. The hardware layer is implemented in the interface amplifier to provide the most basic protection. It usually defines the hard limits of the signal magnitude and is constant for different PHIL applications. The software layer is implemented in the simulation to give the most flexible protection. It works with the same mechanism as interface compensation to preprocess the input and output interfacing signals and prevent unacceptable deviation. It can be easily tailored to a specific PHIL application while costing minimum effort.

E. PHIL Experience at CRIEPI

At the Central Research Institute of the Electric Power Industry (CRIEPI) in Japan, a high-voltage analog simulator has been used for power system studies. The operating voltage of the simulator is 3.3 kV, which allows to accurately represent mechanical dynamics of rotating machines, that is, carefully designed real 3.3-kV rotating machines are used for representing generators and rotating loads. However, studies of wide-area system operation have become more important these days, and it has been recognized that the number of generators equipped in the simulator is not sufficient to this end. Thus, a digital simulator has been connected to the existing analog simulator so that a wide-area power system can be modeled in the digital simulator and the important study zone is modeled in the existing analog simulator.

1) Developed Interface Technique: Bergeron's equivalent circuit of a transmission line is often used for connecting analog and digital simulators. The propagation time delay represented by Bergeron's equivalent circuit is utilized to decouple the circuit equations of the digital part from the analog circuit [43]. The circuit topology used in [43] is shown in Fig. 14. Since an analog device which realizes a current source is usually slower in response compared with a voltage source device, the analog side of Bergeron's equivalent circuit is transformed into a Thevenin equivalent consisting of a voltage source and a series resistor (Norton-Thevenin transformation).

However, if the circuit topology shown in Fig. 14 is used for the high-voltage simulator of CRIEPI, the following problems arise. When a fault occurs in the analog simulator, the voltage source V_{s2} has to generate a very high voltage and, thus, the voltage source must have a fairly wide dynamic range. At the same time, since the power consumed by the series resistor, R_1 is very large, special consideration must be given for the cooling facility of the resistor. Considering the aforementioned problems, the circuit topology shown in Fig. 15 has been developed by CRIEPI and Mitsubishi Electric [47]. The reactor L is connected in parallel to R, and with this reactor, the voltage source V_{s2} does not have to generate a high voltage. Also, the power

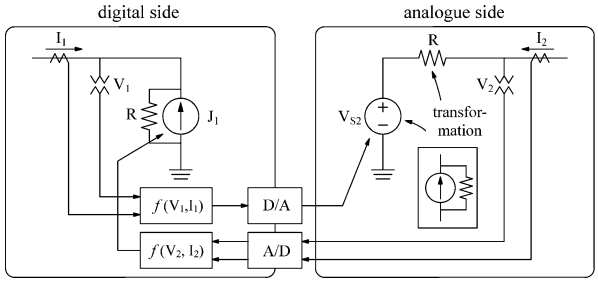

Fig. 14. Conventional circuit topology for the interface (Bergeron's equivalent circuit).

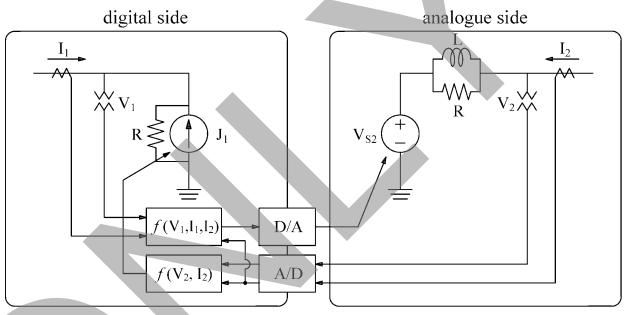

Fig. 15. Developed circuit topology for the interface.

that has to be consumed by R is now greatly reduced (from 50 kW to 1 kW in the case of CRIEPI's simulator). The effects due to the addition of L are compensated by V_{s2}.

2) BTB as a Voltage Source: In order to connect the analog side of the interface circuit to the 3.3-kV analog simulator, the voltage source V_{s2} has to generate voltages of an order of a few kilovolts. To this end, we cannot use an analog amplifier. Thus, back-to-back (BTB) inverters, whose primary side is connected to a commercial power line and the other side to the analog simulator, are used. The total capacity of the BTB is 400 kVA and its carrier frequency is 6 kHz so that the BTB can represent an external power system up to 1 kHz. Since 2003, this analog-digital simulator, called a hybrid simulator, has been in operation at CRIEPI.

3) Simulation Case Study: The four-machine system shown in Fig. 16 was simulated by the hybrid simulator for validation purposes. The generators G1 and G2 are represented in the analog part, and G3, G4 and the infinite bus are represented by the digital part. As a simulation scenario, a 3LG-O (3 line-to-ground fault, and then cleared) at the bus of G1 was simulated. The result is shown in Fig. 17, where the result obtained when all generators are represented in the analog part is superimposed. It is confirmed from this comparison that the developed analog-digital interface gives correct results.

IV. Conclusion

Real-time digital simulators (RTDSs) have been used in the hardware-in-the-loop configuration for the design and development of new equipment for more than two decades now. This paper provided an overview of particular interfacing challenges encountered and their solutions as these simulators evolved from their analog predecessors. The interfacing challenges are divided into two groups according to the real-time simulator

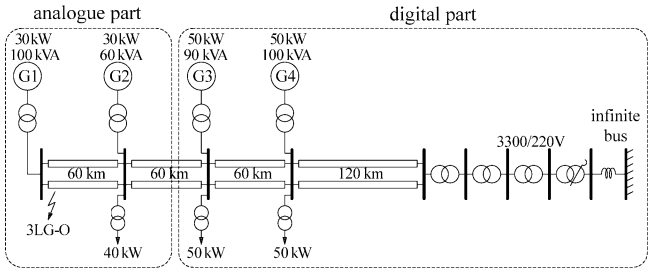

Fig. 16. Four-machine system for validation.

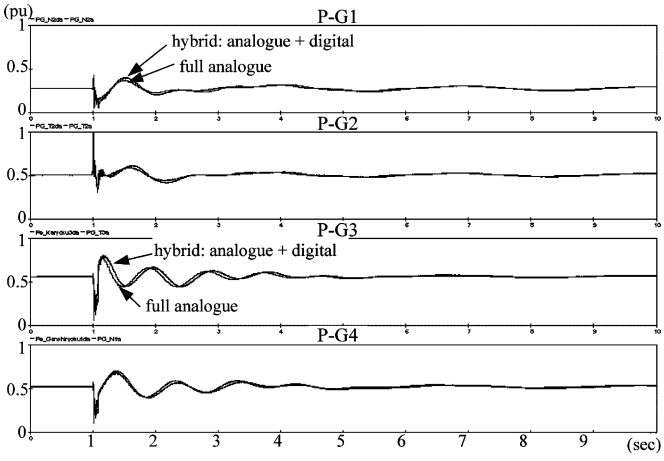

Fig. 17. Simulation result with CRIEPI's hybrid simulator.

application: 1) controller hardware-in-the-loop (CHIL) and 2) power hardware-in-the-loop (PHIL). Case studies are provided and the reader is referred to relevant literature. The area of CHIL is quite mature where several feasible solutions have been proposed especially with FPGA technology which is currently being used in the industry. While the area of PHIL has also made significant progress, research is still needed for cost-effective interfacing solutions.

REFERENCES

[1] H. Li, M. Steurer, S. Woodruff, L. Shi, and D. Zhang, "Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1144–1151, Jun. 2006.

[2] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 2, pp. 919–931, Apr. 2007.

[3] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives," *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1235–1246, Apr. 2007.

[4] A. Bouscayrol, "Different types of hardware-in-the-loop simulation for electric drives," in *Proc. IEEE ISIE*, Cambridge, U.K., Jun. 2008, pp. 2146–2151.

[5] A. Bouscayrol, X. Guillaud, R. Teodorescu, P. Delarue, and W. Lhomme, "Energetic macroscopic representation and inversion-based control illustrated on a wind-energy-conversion system using hardware-in-the-loop simulation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4826–4835, Dec. 2009.

[6] M. O. Faruque and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1146–1158, Apr. 2010.

[7] M. Desjardine, P. Forsyth, and R. Mackiewicz, "Real time simulation testing using IEC 61850," in *Proc. Int. Conf. Power Systems Transients*, Lyon, France, Jun. 4–7, 2007.

[8] V. Dinavahi, M. R. Iravani, and R. Bonert, "Real-time digital simulation of power electronic apparatus interfaced with digital controllers," *IEEE Trans. Power Del.*, vol. 16, no. 4, pp. 775–781, Oct. 2001.

[9] M. O. Faruque, V. Dinavahi, and W. Xu, "Algorithms for accounting of multiple switching events in the digital simulation of power electronic apparatus," *IEEE Trans. Power Del.*, vol. 20, no. 2, pt. 1, pp. 1157–1167, Apr. 2005.

[10] M. Matar and R. Iravani, "FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 852–860, Apr. 2010.

[11] M. O. Faruque, Y. Zhang, and V. Dinavahi, "Detailed modeling of CIGRE HVDC benchmark system using PSCAD/EMTDC and PSB/SIMULINK," *IEEE Trans. Power Del.*, vol. 21, no. 1, pp. 378–387, Jan. 2006.

[12] C. Dufour, J. Bélanger, and S. Abourida, "Accurate simulation of a 6-pulse inverter with real time event compensation in ARTEMIS," *Math. Comput. Simul.*, vol. 63, no. 3–5, pp. 161–172, Nov. 2003.

[13] K. Strunz, "Flexible numerical integration for efficient representation of switching in real time electromagnetic transients simulation," *IEEE Trans. Power Del.*, vol. 19, no. 3, pp. 1276–1283, Jul. 2004.

[14] K. Strunz, L. Linares, J. R. Marti, O. Huet, and X. Lombard, "Efficient and accurate representation of asynchronous network structure changing phenomena in digital real time simulators," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 586–592, May 2000.

[15] L. Linares and J. R. Marti, "A resynchronization algorithm for topological changes in real-time fast transients simulation," presented at the Power Systems Computation Conf., Seville, Spain, 2002.

[16] C. Dufour and J. Belanger, "Discrete time compensation of switching events for accurate real-time simulation of power systems," in *Proc. IEEE 27th Annu. Conf. Industrial Electronics Soc.*, 2001, vol. 2, pp. 1533–1538.

[17] V. Dinavahi, M. R. Iravani, and R. Bonert, "Design of a real-time digital simulator for the D-STATCOM system," *IEEE Trans. Ind. Electron.*, vol. 51, no. 5, pp. 1001–1008, Oct. 2004.

[18] L. F. Pak, M. O. Faruque, X. Nie, and V. Dinavahi, "A versatile cluster-based real-time digital simulator for power engineering research," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 455–465, May 2006.

[19] L. F. Pak and V. Dinavahi, "Real-time simulation of a wind energy system based on the doubly-fed induction generator," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1301–1309, Aug. 2009.

[20] M. Harakawa, H. Yamasaki, T. Nagano, S. Abourida, C. Dufour, and J. Bélanger, "Real-time simulation of a complete PMSM drive at 10 μs time step," in *Proc. Int. Power Electronics Conf.*, Niigata, Japan, 2005, pp. 1–4.

[21] M. Harakawa, C. Dufour, S. Nishimura, and T. Nagano, "Real-time simulation of a PMSM Drive in faulty modes with validation against an actual drive system," presented at the 13th Eur. Conf. Power Electronics and Applications, Barcelona, Spain, Sep. 8–10, 2009.

[22] C. Bordas, C. Dufour, and O. Rudloff, "A 3-level neutral-clamped inverter model with natural switching mode support for the real-time simulation of variable speed drives," presented at the 8th Int. Symp. Advanced Electromechanical Motion Systems, Lille, France, Jul. 1–3, 2009.

[23] C. Dufour and J. Bélanger, "Real-time simulation of a 48-pulse GTO STATCOM compensated power system on a dual-Xeon PC using RT-LAB," presented at the 6th Int. Conf. Power Systems Transients, Montréal, QC, Canada, Jun. 19–23, 2005.

[24] M. O. Faruque, M. Sloderbeck, M. Steurer, and V. Dinavahi, "Thermoelectric cosimulation on geographically distributed real-time simulators," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2009, pp. 1–7.

[25] RT-LAB User Guide. ver. 8.1, , OPAL-RT Technologies Inc., 2007.

[26] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892–902, Apr. 2009.

[27] M. Matar and R. Iravani, "FPGA implementation of a modified two-layer network equivalent for real-time simulation of electromagnetic transients," presented at the IPST, Kyoto, Japan, Jun. 3–6, 2009.

[28] X. Nie, Y. Chen, and V. Dinavahi, "Real-time transient simulation based on a robust two-layer network equivalent," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1771–1781, Nov. 2007.

[29] C. Dufour, J. Bélanger, and V. Lapointe, "FPGA-based ultra-low latency HIL fault testing of a permanent magnet motor drive using RT-LAB-XSG," *Int. Trans. Soc. Modelling Sim.*, vol. 84, no. 2/3, pp. 161–172, Feb./Mar. 2008.

[30] C. Dufour, H. Blanchette, and J. Bélanger, "Very-high speed control of an FPGA-based finite-element-analysis permanent magnet synchronous virtual motor drive system," presented at the 34th Conf. IEEE Industrial Electronics Society, Orlando, FL, Nov. 10–13, 2008.
[31] M. S. Sachdev, T. S. Sidhu, and P. G. McLaren, "Issues and opportunities for testing numerical relays," in *Proc. IEEE Power Eng. Soc. Summer Meeting*, Jul. 2000, vol. 2, pp. 1185–1190.
[32] A. R. Chevrefils and S. Filizadeh, "Transient simulation of an ac synchronous permanent magnet motor for an all-electric all-terrain vehicle," presented at the Vehicle Propulsion and Power Conf., Arlington, TX, 2007.
[33] L.-F. Pak, V. Dinavahi, G. Chang, M. Steurer, and P. Ribeiro, "Real-time digital time-varying harmonics modeling and simulation techniques," *IEEE Trans. Power Del.*, vol. 22, no. 2, pp. 1218–1227, Apr. 2007.
[34] E. Acha, O. Anaya-Lara, J. Parle, and M. Madrigal, "Real-time simulator for power quality disturbance applications," in *Proc. 9th Int. Conf.*, Oct. 2000, vol. 3, pp. 1–4.
[35] H. J. Slater, D. J. Atkinson, and A. G. Jack, "Real-time emulation for power equipment development. II. The virtual machine," *Proc. Inst. Elect. Eng., Elect. Power Appl.*, vol. 145, no. 3, pp. 153–158, May 1998.
[36] W. Zhu, S. Pekarek, J. Jatskevich, O. Wasynczuk, and D. Delisle, "A model-in-the-loop interface to emulate source dynamics in a zonal DC distribution system," *IEEE Trans. Power Electron.*, vol. 20, no. 2, pp. 438–455, Mar. 2005.
[37] W. Ren, M. Steurer, S. Woodruff, and M. Andrus, "Demonstrating the power hardware-in-the-loop through simulation of a notional destroyer-class all-electric ship system during crashback," presented at the Advanced Naval Propulsion Symp., Arlington, VA, Oct. 30–31, 2006.
[38] M. Bacic, "On hardware-in-the-loop simulation," presented at the 44th IEEE Conf. Decision and Control, and the Eur. Control Conf., Seville, Spain, Dec. 12–15, 2005.
[39] S. Ayasun, S. Vallieu, R. Fishl, and T. Chmielewski, "Electric machinery diagnostic/testing system and power hardware-in-the-loop studies," presented at the 4th IEEE Int. Symp. Diagnostics Electric Machines, Power Electronics Drives, Atlanta, GA, Aug. 24–26, 2003.
[40] W. Ren and M. Steurer, "Accuracy evaluation in power hardware-in-the-loop (PHIL) simulation," presented at the Summer Computer Simulation Conf., San Diego, CA, Jul. 15–18, 2007.
[41] W. Ren, "Accuracy evaluation of power hardware-in-the-loop simulation," Ph.D. dissertation, Elect. Comput. Eng. Dept., Florida State University, Tallahassee, FL, 2007.
[42] M. S. MacDiarmid and M. Bacic, "Quantifying the accuracy of hardware-in-the-loop simulations," presented at the American Control Conf., New York, 2007.
[43] R. Kuffel, R. P. Wierckx, H. Duchen, M. Lagerkvist, X. Wang, P. Forsyth, and P. Holmberg, "Expanding an analogue HVDC simulator's modeling capability using a real-time digital simulator (RTDS)," in *Proc. 1st Int. Conf. Digital Power System Simulators*, College Station, TX, Apr. 1995, pp. 199–204.
[44] X. Wu, "Methods for partitioning the system and performance evaluation in power hardware-in-the-loop simulations," Ph.D. dissertation, Elect. Comput. Eng. Dept., Univ. South Carolina, Columbia, 2005.
[45] W. Ren, M. Steurer, and T. L. Baldwin, "Improve the stability and accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms," presented at the Industrial and Commercial Power Systems Technical Conf., Edmonton, AB, Canada, May 6–10, 2007.
[46] W. Ren, M. Steurer, and S. Woodruff, "Applying controller and power hardware-in-the-loop simulation in designing and prototyping apparatuses for future all electric ship," presented at the IEEE Electric Ship Technologies Symp., Arlington, VA, May 21–23, 2007.
[47] M. Takasaki, K. Yamashita, N. Gibo, K. Takenaka, K. Kuroda, Y. Kono, and S. Tamai, "Development of interfacing technique for connecting digital and analogue simulators," presented at the Inst. Elect. Eng. Annu. Meeting Rec., Japan, 2005.