Interfacing Issues in Multi-Domain Simulation Tools

IEEE Power and Energy Society Task Force on Interfacing Techniques for Simulation Tools

M. O. Faruque, V. Dinavahi, M. Steurer, A. Monti, K. Strunz, J. A. Martinez, G. W. Chang, J. Jatskevich, R. Iravani, and A. Davoudi

***Abstract*—This task force paper presents multi-domain tools and their interfacing issues to perform simulations that require expertise in multiple areas of physics, such as electrothermal, electromechanical, electrochemical, and electromagnetic simulations, etc. Features of single multi-domain simulation tools, which are capable of simulating more than one system such as electrical chemical, mechanical, etc., are discussed. In addition, programs that are not capable of performing multi-domain simulation independently but can be interfaced with other programs to perform multi-domain simulation are also covered. Two case studies are reported where multi-domain simulations are performed in offline and realtime environments through the interfacing of more than one simulation tool.**

***Index Terms*—Interfacing, multi-domain, multi-physics, simulation tools.**

I. Introduction

THE USE of computer simulation in industry is rapidly growing. Computer simulations are routinely used to optimize product properties and to reduce product development cost and time to market. In the past, it was considered sufficient to simulate subsystems separately; however, the current trend is to simulate increasingly complex physical systems composed of subsystems from multiple domains, such as mechanical, electric, hydraulic, thermodynamic, and control system components [1]. For example, the design and manufacturing of integrated circuits (ICs) and systems needs reliable simulation tools that also take into account the systems' behavior in different environmental conditions. Predefined standard conditions (e.g., constant temperature) are not always a reasonable assumption since the circuit behavior is influenced greatly by these conditions. Other physical aspects, such as electromagnetic fields and strong and weak coupling between components, are also important as efforts to integrate not only electronics but micromechanical, optical, and other parts on the same chip are underway. As a result, a multi-domain/physics-based simulation is increasingly becoming a necessity. At the same time, the availability of inexpensive work stations with high-performance parallel computational capability has made this type of multi-domain simulation a reality.

In the power and energy systems sector, generation from distributed sources, such as wind power, photovoltaic cells, fuels cells, biomass, and other renewable energy sources and storage are being constantly added to the grid [2]. The increased penetration of energy from these sources into the power grid is changing the system dynamics to a new level where modeling and simulation of renewable technologies using multi-domain simulation tools has become essential due to the use of compact power-electronic devices/converters. The modeling and simulation of power systems containing such new technologies involves multiple disciplines since their outputs are influenced not only by electrical properties but also by other factors, such as ambient temperature, wind velocity, chemical conditions (in case of fuel cells and battery systems), etc. For example, stacked power-electronic switches, such as insulated-gate bipolar transistors (IGBTs) and gate-turnoff thyristors (GTOs), are consistently used in grid interface converters for renewable sources. These converters are modular in nature and several of them are assembled together so that they can be used at higher voltages. Such an assembly of power-electronic devices is called the power-electronic building block (PEBB) [3], which needs careful attention in the design process to remove the generated heat in the block. Moreover, power systems developed for the nonterrestrial system, such as the one used in modern vehicles, aircraft, and electrical ships poses significant challenges (due to the higher requirements for reliability and redundancy, compactness and reduced weight) while removing the heat generated from various components and converters [4]. It is therefore essential to run a thermoelectric cosimulation for designing and commissioning of such a system. Therefore, the study of power system dynamics must include multi-domain physics associated with them so that accurate responses can be obtained.

The aim of this task force paper is to give a brief overview of the challenges associated with using multi-domain simulation tools, and their possible interfacing methods. This paper will be useful to practicing engineers and researchers who are dealing with the modeling and simulation of complex power and energy systems and their components that require multi-domain representation as well as the use of a variety of potentially different simulation tools. Two case studies are presented that illustrate the effectiveness of interfacing these tools.

Manuscript received September 13, 2011; accepted September 26, 2011. Date of publication November 09, 2011; date of current version December 23, 2011. Paper no. TPWRD-00775-2011.

The authors are Contributors to the IEEE Power and Energy Society Task Force on Interfacing Techniques for Simulation Tools.

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), M. O. Faruque, 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.

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.2011.2170861

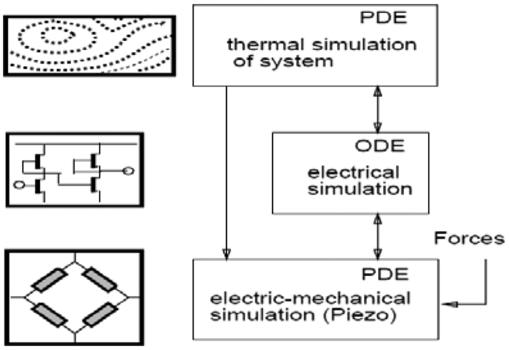
PDE
thermal simulation of system
ODE
electrical simulation
Forces
PDE
electric-mechanical simulation (Piezo)

Fig. 1. Block diagram demonstrating the multi-solver requirement for simulating multi-domain systems [4].

II. Challenges in Multi-Domain Simulation

A. *Multi-Rate and Multi-Solver Tools*

As the dynamic behavior and time response of the subsystems involved in a multi-domain system might be significantly disparate, using a single simulation time step or solver is not very efficient. In some cases, multi-rate, multi-solver approaches may yield the better solution. For example, thermal, mechanical, or chemical systems are not as fast as an electrical system. A multi-rate cosimulation is the optimal way in systems. A complex system can be decoupled into the respective physical domains, modeled with the best available tools for each domain, and then interfaced using multi-rate transition To illustrate the complexity of the problem, let us consider an automotive example consisting of a sensor for acceleration measurement built on a wheel of a car together with an electronic-control system to prevent locking [4]. Normally, piezoresisitive elements are used for the sensors which are temperature sensitive which requires the influence of heat developed between system parts to be modeled. The usual technique used in silicon-based accelerometers is that a piece of metal is attached as weight on a bendable silicon tongue whose deformation is detected by four piezoresistive elements consisting of a Wheatstone bridge. Other electronic components, such as analog-to-digital (A/D) conversion, signal conditioning units, etc. are also attached to the same chip or a nearby circuit. To simulate such a system, it requires the system to be represented into different subsystems as shown in Fig. 1.

The described system can be considered as an integration of the electrical system, thermal system, and electromechanical system. Even though the electrical system can be represented through ordinary differential equations (ODEs), the other two systems require the solution of partial differential equations (PDEs). Similarly, hydropower systems as shown in Fig. 2, consisting of hydraulic, mechanical, and electrical systems need a multi-domain simulation approach to obtain better accuracy and efficiency. In most studies, the three systems are studied separately and the coupled interactions between these systems are either ignored or too simplified, resulting in reduced accuracy of the study [6].

For simulating such multi-domain systems, the widely different time constants pose significant challenges. For example, the time constant of the electrical system is much smaller than that of thermal, electromechanical, or the hydraulic system. The simulation of such a system is challenging due to the interdependency between subsystems and their time-varying behavior with largely different dynamics. Decoupling of such interdependent subsystems would also require intelligent techniques to minimize errors. Moreover, different subsystems require different solvers with obviously different accuracy constraints. Therefore, for efficient simulation of such systems, a multi-rate, multi-solver, and/or multi-domain tool could be the best option.

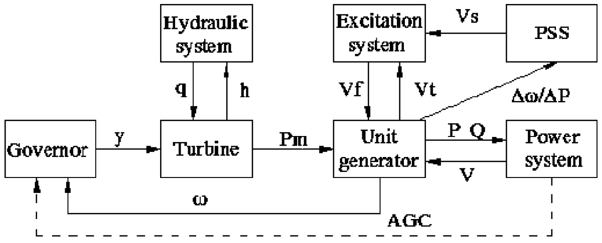

Fig. 2. Coupled multi-domain systems [6].

B. *Issues Related to Hardware, Software, and Interfacing*

To perform multi-domain simulation, there are also challenges related to the type of hardware, software, and interfacing between various hardware components. Generally, two types of simulation platforms are used for multi-domain simulation: *homogeneous* and *heterogeneous*. A homogeneous platform involves one specific type of digital hardware media, such as central-processing unit (CPU), digital signal processor (DSP), or field-programmable gate array (FPGA), etc. On the other hand, the heterogeneous platform involves a combination of more than one type of hardware. Model interfacing on a homogeneous platform requires mostly interfacing through software whereas on a heterogeneous platform, it requires both software and hardware interfacing. Interfacing of data/signals between multi-physics systems is more challenging in heterogeneous simulators than in homogeneous simulators. For example, if Matlab/Simulink models are used for real-time simulation of an electrical system in a PC-based simulator, the modeling of a simple thermal or mechanical system using CPU-based standard programs or custom-developed code or using the supplied toolboxes is quite straightforward. However, if a real-time simulator such as RTDS (using IBM PowerPC processors and DSPs) is used for real-time simulation of an electrical system, and Dymola (on a PC) is used to model a thermal system, real-time interfacing of these two platforms poses significant challenges [7].

Multi-domain simulation can be performed either using a single multi-domain simulation program (known as *independent multi-domain simulation tool*) or by interfacing more than one simulation tool (known as *interfaced multi-domain simulation tools*). Although independent programs, such as Dymola, Simulink, Comsol, and Ansys can perform multi-domain simulations, these tools may not have sufficient simulation capabilities in all areas of physics and engineering design. For example, popular SIMULINK toolboxes can be used for the simulation of control and electrical systems with good accuracy; however, the thermal toolbox available in SIMULINK is suitable only for rudimentary purposes and is insufficient to simulate complex thermal systems consisting of chillers, heat exchangers, and air conditioning units. Similarly, many other multi-domain simulation tools may have good accuracy and a set of built-in library components for particular types of simulations and applications; however, they might lack the capability of modeling and simulating other physical systems at the required level of detail. With the objective of exploiting the best features of each simulation tool, it would therefore, be beneficial to have a complex system modeled with multiple tools in separate domains, such as electrical, mechanical, thermal, chemical, etc. and then interface them so that their respective system dynamics and interactions can be preserved with sufficient accuracy.

Cosimulation or multi-domain simulation require a data exchange between models of various subsystems. One way of exchanging data could be the use of shared memory or the use of a fast communication network using a standard data-communication protocol. For simulators located geographically apart, IP-based protocols may be used although with increased time delay in the output which may affect the accuracy of the simulation, especially for systems with low time constants. However, if the simulators are located at the same physical location, data can be exchanged by establishing a direct link between the simulators through input/output (I/O) terminals or digital hardware. In either case, the challenging issues for performing multi-domain simulation through the interfacing of more than one simulation tool are:

1) decoupling the complex system into multi-physics domain;
2) modeling of multi-physics phenomenon in a common platform in the most efficient way;
3) choosing the right hardware and software to perform the cosimulation;
4) requirement of data compatibility, in case heterogeneous simulation platforms are used;
5) requirement of a high-speed data-communication network in case of a real-time communication requirement
6) bandwidth limitation for I/O interfacing;
7) communication latency.

III. Independent Multi-Domain Simulation Tools

Table I provides a list of independent multi-domain simulation tools with their specific features and areas of application. It is a partial list of the most commonly used multi-domain tools and is by no means comprehensive since this area of research is continually growing.

IV. Interfacing of Multi-Domain Simulation Tools

Although tools mentioned in the previous section are considered independent multi-domain simulation tools, their strength in all areas of physics is not the same. Some are widely used for the simulation in a particular field whereas their usage in the other fields is somewhat limited. For example, DYMOLA is a very good tool for the simulation of thermal, hydraulic, and mechanical systems, but for simulating an electrical system, an Electromagnetic Transients Program (EMTP)-type transient simulator, such as SIMULINK/SIMPOWERSYSTEM or PSCAD/EMTDC may be a better choice for a transient simulation. Therefore, to obtain the most accurate simulation results, interfacing two or more simulation tools might be a good idea. In this section, a few such interfacing examples are discussed.

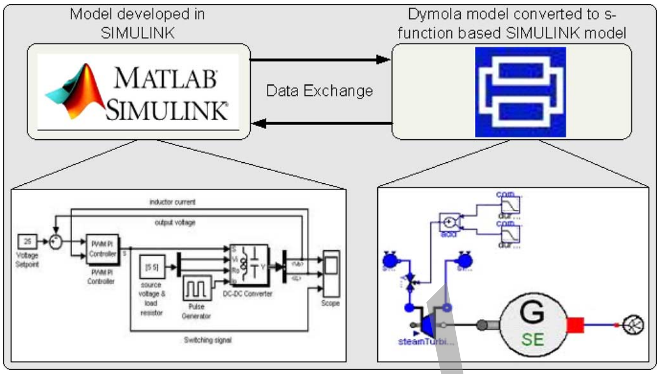

Fig. 3. Generic illustration of Dymola-Simulink interfacing in the Simulink modeling environment [8].

A. DYMOLA and MATLAB/SIMULINK

Models developed in Dymola can be interfaced with SIMULINK as shown in Fig. 3. The DymolaBlock block, with the Dymola logo, representing the Modelica model can be interfaced to other Simulink blocks, and to other DymolaBlocks [8]. The I/O lines of the block are ports which become visible after the compilation of the model. The DymolaBlock is a shield around a S-function MEX block (i.e., the interface to the C code generated by Dymola for the Modelica model). Double-clicking on the DymolaBlock opens a form where the name of the model and optional file names to be interfaced are added. A Simulink model may contain several DymolaBlocks as long as they all use different underlying MEX files. Details about the interfacing procedure are available in [8]. Aside from Simulink, Dymola can also be interfaced with low-level Matlab files (e.g., Dymsim.m) which allows the use of Modelica models from Dymola. Dymola also provides support for real-time simulation (e.g., hardware in the loop) of DymolaBlock on several platforms: dSPACE, Real-Time Workshop, and RT-LAB. A case study involving SIMULINK-Dymola interfacing for real-time simulation has been reported in [9] where a closed-loop real-time simulation of a vehicle with a climate control system model was developed in Modelica and integrated with a controller model developed in the Simulink environment. The integrated model was then simulated in an Opal-RT real-time environment and Modelica's capabilities in creating models for real-time simulations were demonstrated. Similar studies of multi-disciplinary simulations using either real-time or offline simulation tools have also been reported in [10] and [11], where Dymola or Modelica models were interfaced with Simulink models to perform simulations of complex multi-physics systems.

In the aforementioned interfacing technique, Dymola generates S-functions from the Modelica models and then enables the models to be simulated together with the rest of the models in

TABLE I
LIST OF INDEPENDENT MULTI-DOMAIN SIMULATION TOOLS AND THEIR SALIENT FEATURES

Multi-domain Simulation tool	**Field of Application**	**Salient Features**
Dymola	Mechanical, electrical, thermodynamic, hydraulic, pneumatic, thermal, control system and mixed systems comprised of one or more of them [1], [8], [10].	Built upon Modelica language, supports non-causal modeling, uses object oriented methodology, supports standard definitions, and mathematical expressions such as ordinary differential equation systems (ODEs) and differential algebraic equations (DAEs).
SIMULINK	Electrical system, control system, mechanical system, thermal system [13], [14].	High-performance multifunctional multi-domain program, supports linear and nonlinear systems, modeled in continuous time, discrete-time, or a hybrid of the two in both single time-step and multi-rate environments.
SIMPLORER	Electrical, thermal, electromechanical, electromagnetic, and hydraulic [15].	Multi-domain system simulation software program mainly used for the design, modeling, analysis and optimization of high-performance systems, offers multiple modeling techniques including circuits, block diagrams, state machines, equation level, and modeling languages such as VHDL-AMS, SML (Simplorer Modeling Language), and C/C++.
COMSOL	Electrical, electromagnetic field, thermal, fluid flow, structural mechanics [16]	Finite element based software package for various physics and engineering applications, especially coupled phenomena, or multi-physics, offers an extensive interface to MATLAB and its toolboxes for a large variety of programming, preprocessing and post processing possibilities, allows the simulation of coupled systems of partial differential equations (PDEs).
VTB	Electrical, thermal, energy conversion [17]-[22]	A multi-physics multi-formalism tool able to capture expertise from different expertise [17], Supports different type coupling between the model coherently with recent standard in the definition of simulation languages such as VHDL-AMS, also provides co-simulation interfaces with other simulation platforms such as Matlab/Simulink, ACSL, Labview [17]-[22].
ANSYS	Mechanical, structural mechanics, fluid dynamics, electromagnetism [23]	Based on the Finite Element Method and is capable of performing static (stress) analysis, thermal analysis, modal analysis, frequency response analysis, transient simulation and also coupled field analysis.
Mathmodelica	Mechanical, electrical, thermal, control, hydraulics, bio-chemical [24]	A platform for engineering as well as life science modeling and simulation, based on Modelica language, allows custom modeling and simulation.
TRNSYS	Electrical and thermal systems, solar thermal and PV, low energy buildings and HVAC systems [25]	Performs dynamic simulation of the energy systems' behavior, modular nature and allows custom mathematical models.
MagNet	Electromagnetic Field, electro-mechanical [26]	Finite element method tool for simulating static, frequency dependent or time varying electromagnetic fields.
PLECS	Electrical, Electronic and Thermal [27]	Simulates semiconductor devices and their thermal losses, uses ideal models for switching circuits.
JMAG	Electro-mechanical system simulation [28]	Finite element based program used for the modeling of electro-mechanical energy conversion system.
AMESim	Hydraulic, pneumatic, mechanics, electrical, thermal, electromechanical [29]	1-D simulation suite, models and analyzes multi-disciplinary behavior, validated analytical models are used.
MapleSim	Mechatronic, Machine Design, Electronics, Control System Design, Real-Time Simulation and HIL Testing [30]	MapleSim generates model equations, runs simulations, and performs analyses using the symbolic and numeric mathematical engine, includes traditional signal flow components that can be combined with other physical components.
INSEL	Solar Energy, Thermal Collectors and Meteorological [31]	Simulates systems like on-grid PV generators with MPP tracker and inverter, supports with databases for photovoltaic modules, inverters, thermal collectors and meteorological parameters.
VisSim	Electrical, Electronic, Mechanical, Thermal, Control and Communication [32]	Model based dynamic simulation tool for linear, nonlinear, continuous time, discrete time, time varying and hybrid system designs.
CASPOC	Electrical and drives, Control systems, Mechanical system [33]	Circuit level simulation software for modeling power electronics, uses switch models.

the SIMULINK environment. However, it was reported that in the case of some complex physical models, the Simulink solvers are unable to cope reliably with the generated S-function models [12]. In such a case, another tool, Simelica, can be used to convert Simulink models into equivalent Modelica models, and use the Modelica environment to simulate the whole system. The conversion is achieved while retaining the original structure of the Simulink model. The equivalent Modelica models are built from a new library of components, known as the Advanced Blocks library.

B. SIMPLORER With MATLAB/SIMULINK

C/C++ programs, MATLAB/Simulink, MathCAD, and other specialized programs can be directly integrated into Simplorer. This allows users to utilize customized code and existing feedback/control legacy designs. The direct integration of models in their native environment avoids model translation, saves design time, and enables communication and model exchange across departments and between suppliers and users. An example of multi-domain modeling of a hybrid-electric vehicle using Simplorer and Simulink can be found in [34]. Therein, the overall vehicle drive system is divided into multiple subsystems, including electrical, mechanical, and control, which are then interconnected using the distributed heterogeneous links. The partitioning into subsystems is carried out to maximize the use of the domain-specific features of each simulation language (e.g., switching converters in Simplorer, and slow electromechanical components in Simulink).

C. COMSOL With MATLAB/SIMULINK

COMSOL multi-physics also offers an extensive interface to MATLAB/SIMULINK [6] and its toolboxes for a large variety of coupled systems involving partial differential equations (PDEs), 2-D or 3-D, nonlinear and time-dependent systems, and preprocessing and postprocessing of simulation results. Using the export feature in COMSOL, models can be used in Simulink or state-space form to incorporate finite-element-based models from COMSOL. If the time step in a time-dependent COMSOL multi-physics model is very small compared to the time step in the Simulink model, a static model can be exported. This means that the terms of the equations that contain time derivatives are neglected, so that the output of the COMSOL multi-physics subsystem block can be computed from the input variables by solving a stationary PDE problem. For example, if COMSOL multi-physics is used to model electromagnetic phenomena as a part of a mechanical system, the dynamics of the COMSOL multi-physics model can be neglected. By default, COMSOL multi-physics uses general export (i.e., its solvers will be called in every simulation step from Simulink). However, for a linearized model, only a set of matrices is exported and the COMSOL multi-physics solver does not need to be called during the simulation, thereby making the simulation quite fast for linear systems. In Simulink, the COMSOL multi-physics model appears as an S-function.

D. Interfacing Features of ANSYS

ANSYS Workbench can be interfaced to MATLAB/SIMULINK, Mathematica, CASPOC, and other multi-domain simulation tools [35]. This allows the users to run cosimulation in different domains (e.g., magnetic, mechanical, electric circuits, thermal, and control systems simulations). An example of multi-domain simulation for an electromechanical actuator is reported in [36]. The cosimulation process of the actuator is shown in Fig. 1. The electrical circuit and control system is simulated in CASPOC while the magnetic-domain variables (e.g., forces, coenergy, and inductances) are simulated in the ANSYS Workbench. In each time step, CASPOC reads the data from the ANSYS Workbench and accordingly computes the voltage and position value of the actuator for the finite-element (FE) model used in the ANSYS Workbench.

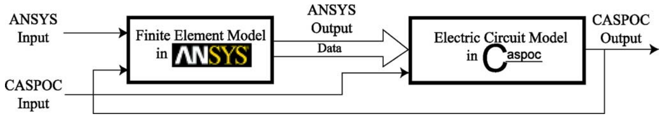
ANSYS Input
Finite Element Model in ANSYS
ANSYS Output
Data
Electric Circuit Model in Caspoc
CASPOC Output
CASPOC Input

Fig. 4. Cosimulation of ANSYS workbench with CASPOC.

The ANSYS Workbench can also be integrated with other simulation tools through the software MOR for ANSYS. Since the finite-element models have high orders, the integration of these models in multi-domain simulation is quite infeasible. Lower order compact models that are good approximations of the original high-order models can be extracted. The software MOR for ANSYS extracts the low-order model from FE ANSYS models. The reduced order models can be directly implemented in MATLAB /SIMULINK, Mathematica, Python, and CASPOC. They can also be used as templates in Saber MAST, VerilogA, and VHDL-AMS [37].

An example of integrating ANSYS Workbench with CASPOC through the software MOR for ANSYS is reported in [38] and shown in Fig. 4. This cosimulation is a detailed electrothermal simulation of an IGBT in a hybrid electric vehicle. The accurate FE thermal model of an IGBT module with three direct copper bondings (DCBs) is generated in the ANSYS Workbench. Each DCB has four IGBTs and six diodes. The software MOR reads the FE model matrices from ANSYS Workbench. The low-order model is extracted by using MOR for ANSYS. The reduced order model is directly implemented in CASPOC for electrothermal simulations.

E. PSIM With ModelSim

The PSIM module, called the ModCoupler Module, can integrate PSIM to ModelSim for supporting VHDL code. The power circuits are simulated in PSIM while the control systems are implemented in ModelSim in VHDL code. The digital simulation of control systems facilitates the implementation of nonlinear control strategies and online optimization. In addition, since FPGA-based digital controllers are described in hardware description languages such as VHDL, the cosimulation between PSIM and ModelSim facilitates the hardware implementation in FPGA [39].

An example of integrating PSIM with ModelSim is reported in [36]. This cosimulation is a detailed case study of the digitally controlled switching power converters. The cosimulation includes the analog simulation of a dc-dc power converter and the digital simulation of the control system. The communication link between PSIM and ModelSim is implemented through the DLL block in PSIM and the FLI block in ModelSim. Both blocks are described in C/C++. The cosimulation process between PSIM and ModelSim is shown in Fig. 5. The DLL block receives the signals generated by the power stage and sends them to ModelSim. The control signals generated by ModelSim are sent to the analog simulator through the FLI block. These

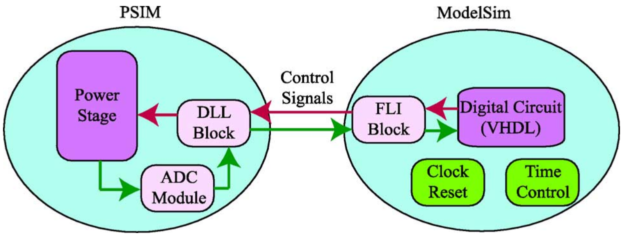

Fig. 5. PSIM and ModelSim cosimulation.

control signals are used as the input signals for the power stage simulation.

F. VTB With MATLAB/SIMULINK

The virtual test bed (VTB) offers different ways to interface with Matlab/Simulink [20]. First, the VTB library is equipped with one library block that is able to perform run-time cosimulation with Simulink. This feature basically allows a linking Simulink diagram to a VTB schematic. A similar library block allows invoking generic Matlab functions for run-time postprocessing of the simulation results. An external toolbox also supports the automatic generation of the VTB native model starting from a Simulink schematic. This feature is based on a customization of the Real Time Workshop (RTW) available from MathWorks. Finally, it is also possible to perform reverse integration (i.e., integrating VTB in a Simulink schematic). VTB is equipped with automatic code generation features and the integration of this code in a standard Simulink S-function is quite straightforward.

A detailed case study involving the integration of three tools to perform advanced analysis of the impact of communication infrastructure on a ship-board power system is discussed in [17]–[22]. The basic idea is to use the best tool for each domain. As a result, VTB is adopted to model the power system with wide penetration of power electronics in the context of MVDC distribution onboard an e-ship; the power converter control is implemented in Simulink; while OpNet is used to model the communication among the converters. In a similar fashion, MATLAB/SIMULINK offers an interfacing facility to other software and together they offer multi-domain simulation capability. For example, The PSpice SLPS and Allegro AMS Simulator SLPS interfaces integrate the proven PSpice technology from Cadence with MATLAB/Simulink. It enables co-simulation of electrical and mechanical systems by supporting the substitution of an actual electronic block, and allows the designer to identify and correct integration issues of electronics within a system.

In addition, the interfacing of different simulation programs can also be achieved by using the distributed heterogeneous simulation links [40]. This approach enables cosimulation between various programs using either fixed or variable communication [41], which can be useful for interfacing multi-domain subsystems with diverse dynamic time constants.

V. Multi-Domain Simulation Case Studies

Two case studies involving multi-domain simulation are reported in this section. The first case study is about performing multi-domain simulation through the interfacing of two offline simulation tools. The second case study involves multi-domain simulation in a real-time environment.

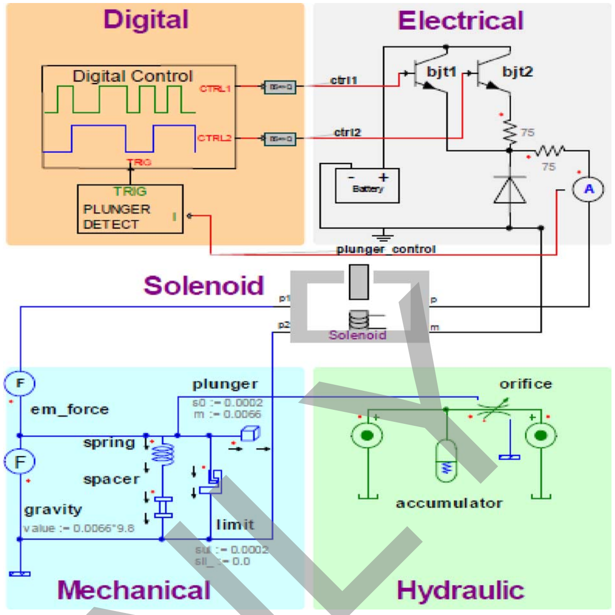

Fig. 6. Linear actuation system divided into multi-domain subsystems.

A. Case Study1- Multi-Domain Simulation Using SIMPLORER and Maxwell [43]

In this case study, VHDL-AMS modeling language is used for the behavioral modeling of a linear actuation system that contains subsystems and components belonging to different physical domains, such as the solenoid (electromechanical), the valve (hydraulic), and the driver circuit (electrical). Linear actuation systems typically include electrical, mechanical, and hydraulic components as well as control systems as shown in Fig. 6. The system employs the mechanical motion of a solenoid to control hydraulic flow. A digital subsystem controls the excitation of the solenoid's electromagnet by switching a pair of transistors on the electrical side. The force from the solenoid displaces a plunger in the mechanical domain. This translational motion of the plunger serves to block and unblock the flow of a fluid in the hydraulic domain. The solenoid plays the role of an interfacing component between the electrical and mechanical domains of the model. It is a multi-domain component in itself, comprising the electrical driver circuit and the mechanical plunger behavior.

1) Modeling of Electrical Subsystem: A simple dynamic battery model is used to power two bipolar junction transistors. The transistors have been modeled at the system level and operate as ON-OFF switches based on their input control signals. The transistors, along with the associated resistor, behave as a voltage divider and provide the voltage control for the solenoid's electromagnet. A free-wheeling diode is provided between the solenoid's electrical terminals to ensure a gradual decay of the solenoid's coil current when the plunger opens to unblock the fluid flow. An ammeter is used to measure and feedback the solenoid's coil current to the digital control subsystem.

2) Modeling of the Digital Control Subsystem: The digital subsystem consists of two parts: 1) a pure digital model called "digital control" developed in VHDL and 2) a mixed signal VHDL-AMS model called "plunger detect" that identifies the closing of the plunger. The control signals for the two transistors are called ctrl1 and ctrl2, and the control stimuli are modeled using behavioral PROCESS constructs in VHDL. Special interface models called *omnicasters* are used to connect the digital control signals to an analog input of the transistor switches. *Omnicaster* models are used to interface ports of different data types in analog and digital domains. In this example, the *omnicasters* function as 1 bit digital-to-analog (D/A) converters that transform the digital BIT data to an analog quantity based on a threshold value.

3) Modeling of the Mechanical Subsystem: The plunger is modeled in the mechanical domain as a mass model with a single conservative node (position is the across quantity and force is the through quantity). A preloaded spring is represented by a spacer-spring combination and the maximum displacement of the plunger is limited with a limit stop model. When the solenoid is energized initially, the plunger closes the gap beginning with an initial position of 0.0002 m to a final position of 0 m as specified in the limit stop model. A force source is used to model the force of gravity on the plunger with a force value set to 9.8 m/s2 times the plunger mass in kilograms. The force at different points in the mechanical subsystem may be measured using force sensor models.

4) Modeling of Hydraulic Subsystem: A simplified hydraulic system is modeled using an accumulator, a hydro-mechanical variable orifice model, and a pair of pressure sources that provide differential pressure to enable hydraulic flow. The plunger displacement in the mechanical domain is used by the variable orifice model to either block or unblock the hydraulic flow due to the differential pressure.

5) Modeling of Solenoid Subsystem: The modeling of the solenoid is the most challenging due to the variation of inductance with the position of the plunger and the coil current. This cannot be represented by a constant lumped value of the inductance. Due to the dependency of the inductance on a variety of geometric and material parameters, a better prediction of the real value of the inductance can be obtained using a finite-element analysis (FEA)-based tool. However, this is not feasible due to the large number of equations generated by the FEA method. Therefore, model extraction technology was used to refine the model behavior. The FEA solver was used to generate a lookup table that included the nonlinear effects of the solenoid by performing a parametric sweep. This lookup table was then used to accurately simulate the solenoid behavior in a multi-domain system. Details about the modeling of the solenoid are available in [43].

6) Multi-Domain Simulation Results: The system dynamics due to the opening and closing of the plunger are observed by carrying out a transient simulation for 80 ms. Both control signals are reset for the first 10 μs of simulation time and then ctrl1 is set to 1. This causes electromagnetic excitation of the solenoid and as soon as a trigger from the plunger model is received indicating that the plunger has closed at 10.5 ms, ctrl2 was set to 1 and ctrl1 was reset to 0 (see Fig. 7). This ensures that the minimum current required to keep the plunger closed is supplied. The initial closing of the plunger causes a back electromotive force (emf) that is reflected as a transient dip in the solenoid coil current (Fig. 8). The force exerted on the plunger displaces it from its initial position of 0.0002 m to 0 m and the plunger is held in this position until about 50 ms after its closing. The hydraulic flow through the variable orifice is blocked during the period that the plunger is closed and is continued again when the plunger opens. After the second transistor is switched off, the plunger opens and moves back to its original position of 0.0002 m due to the force of the spring, and a transient increase in the solenoid coil current was noticed due to the back emf.

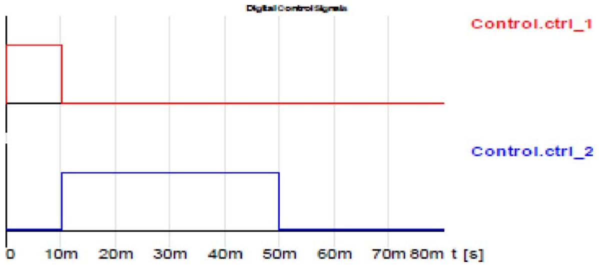

Fig. 7. Control signals from the digital circuit.

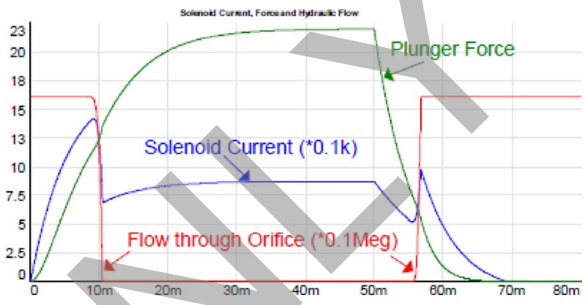

Fig. 8. Solenoid current, plunger force, and hydraulic flow with the change of control signals.

VHDL-AMS is a powerful modeling language and is well suited for multi-domain mixed-signal simulation. The solenoid model based on the lookup table was simulated using SIMPLORER Modeling Language (SML). Simulation results shown in this case study were generated through the use of SIMPLORER and Maxwell products from Ansoft Corporartion.

B. Case Study 2 – Interfacing RTDS and OPAL-RT Simulators for Thermoelectric Cosimulation [7], [44]

This is a unique case of multi-domain cosimulation where two RTDSrs located approximately 3500 km apart were interfaced to perform thermal-electric cosimulation of an electric ship. The electrical model was simulated on the RTDS simulator at the Center for Advanced Power Systems (CAPS), Florida State University (FSU), Tallahassee, FL, while the thermal model was simulated on an OPAL-RT simulator located at the University of Alberta (U of A), Edmonton, AB, Canada. The two simulators exchange data in an asynchronous mode on the Internet utilizing the TCP/IP and UDP protocols. Due to the involvement of completely different types of hardware and software environments in addition to the geographic separation of the simulators, this is one of the challenging co\simulation cases reported in the literature.

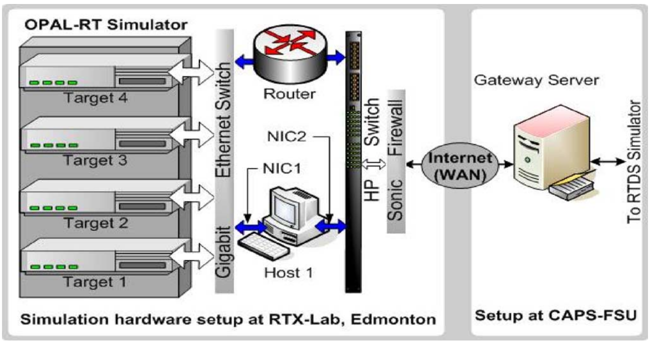

Fig. 9. Hardware setup at the RTX-Lab (University of Alberta) for geographically distributed cosimulation.

1) RTDS Simulation Setup at CAPS-FSU: As shown in Fig. 6, digital I/O ports on the RTDS are used to transfer data between the RTDS and two microcontrollers using a time-division-multiplexing (TDM) scheme. The microcontrollers are Rabbit RCM3200, with Ethernet, TCP/IP, and Modbus TCP support, and custom C software was developed at CAPS-FSU to support the TDM transfer and data buffering. One microcontroller unit is dedicated for input and one for output.

The simulation cases are constructed, downloaded to the simulator, and monitored and controlled using RSCAD software. The software enables construction of cases using a graphical schematic editor, provided with libraries containing models for typical power system components, such as machines, transmission lines, and power-electronic components. In addition, the software provides the user with the capability to develop new components.

2) Simulation Setup at RTX-LAB (University of Alberta): For geographically distributed simulation, a communication link was required for exchanging data between the two simulators over such a long distance. Among the available cost-effective options, the publicly available Internet was used as the backbone of the communication link between the two simulators. The high-speed Internet backbones used for interconnecting research networks in the U.S. and Canada (Florida Lambda Rail, National Lambda Rail, and CANET), provide high-throughput, low-latency communication.

Fig. 9 illustrates the hardware setup at the RTX-Lab where *Target* node 1 was used as the main computation engine and *Host* 1 was used to prepare the model using SIMULINK and RT-LAB. The *Host* is connected to two network switches using two network interface cards (NIC1 and NIC2) installed on it.NIC1 communicates with the *Target* and NIC2 communicates with the external world Internet (WAN) through the HP switch and the Sonic Firewall. The main requirement of NIC2 link is to acquire the RT-LAB license from the license manager running on a separate computer. A router was added and configured to link the *Target*s to the WAN so that simulation results can be sent directly to any remote computer such as the gateway server located at the CAPS-FSU, Florida.

In addition to the operating systems (Windows and Linux) required for *Hosts* and *Targets*, the *Host* computer also needs MATLAB/Simulink and RT-LAB software which were basically used to create, compile, and load the thermal model into the *Target*. With the help of the Real-Time Workshop available in Simulink, RT-LAB generates C/C++ code of the model and then creates executable files which are loaded into the *Target* for execution. A client communicator program, developed in C++, was embedded with the Simulink model which allows the *Target* to send data over the Internet to a predefined remote server with a fixed IP address and to receive data from the remote server.

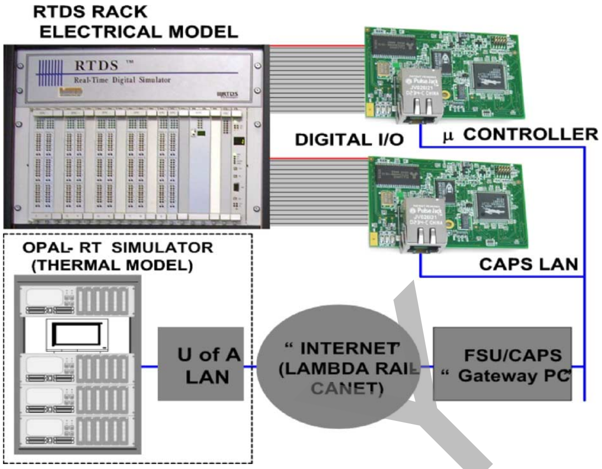

Fig. 10. Hardware setup at CAPS-FSU for geographically distributed cosimulation.

3) Server-Client Communication: A *Server* at CAPS was dedicated for TCP-based data transfer, using custom networking software developed in Microsoft Visual C++. This software connects to both Modbus TCP slaves, and waits for a connection from the remote simulation on the OPAL-RT system (*Client*). After the remote connection was established, the gateway waits for a transfer of a data set from the OPAL-RT system, and then replies to the remote system with a data set from the RTDS. The gateway program must also communicate with the two Modbus TCP microcontrollers, sending data to one, and receiving from the other. In turn, the microcontrollers communicate with the RTDS via the TDM digital I/O scheme. The process was repeated, with the OPAL-RT system initiating each two-way data transfer. The current data set was a set of 32 floating- point values in each direction.

4) Modeling of E-Ship Electrical System: The architecture of the electrical system of the e-ship is depicted in Fig. 11 which shows a power generation system, load center, and a ship propulsion system. A 4.16 kV medium-voltage (MV) ring bus is powered by two 36 MW main gas turbine-driven synchronous generators (MTG 1 and MTG 2) and two 4 MW auxiliary gas turbine generators (ATG 1 and ATG 2). The MV subsystem supplies power to two 36.5 MW propulsion motors via variable speed drives. The system was modeled using RSCAD and run on the real-time-simulator RTDS at CAPS. Details about the modeling of the notional electrical system can be found in [44].

5) Modeling of E-Ship Thermal System: In general, the thermal-management system of an all-electric ship consists of cooling fluids, such as chilled water, chilled air, or seawater-based heat exchangers and heat sinks, all controlled by a system-level process control. However, for demonstration purposes, metallic heat sinks are used which remove the heat

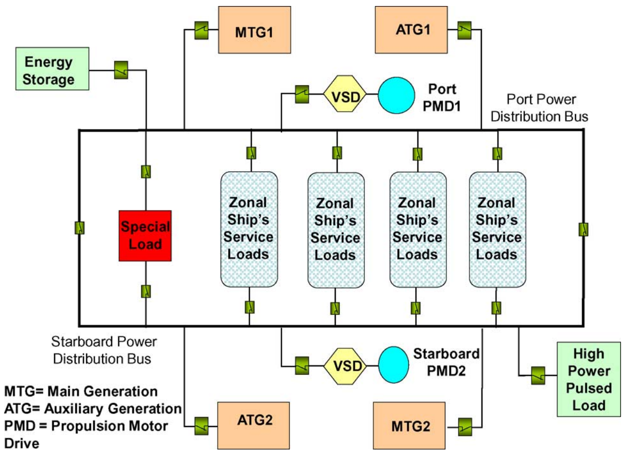
Energy Storage
MTG1
ATG1
VSD
Port PMD1
Port Power Distribution Bus
Special Load
Zonal Ship's Service Loads
Zonal Ship's Service Loads
Zonal Ship's Service Loads
Zonal Ship's Service Loads
Starboard Power Distribution Bus
VSD
Starboard PMD2
High Power Pulsed Load
MTG= Main Generation
ATG= Auxiliary Generation
PMD = Propulsion Motor Drive
ATG2
MTG2

Fig. 11. Notional architecture of the electric power system of an e-ship.

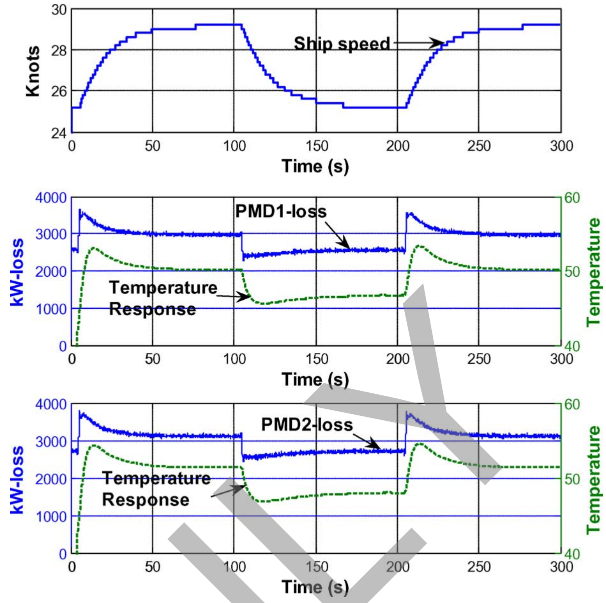

Fig. 12. Transient response of power losses and temperature changes for propulsion motor drives when the ship speed has changed.

through natural air cooling or forced-air cooling methods. A simple thermal model of a heat sink was implemented in the OPAL-RT system in the SIMULINK environment.

6) Results of Cosimulation: To study the temperature response of the thermoelectric cosimulation, the ship speed was changed several times at regular intervals in the RTDS model. The change in speed changed the power losses in various parts of the electric system. These power losses were sent to the OPAL-RT simulator through the established communication link. The thermal model in the OPAL-RT simulator received that data as input and calculated the temperature of each heat sink. All of the temperature data were then sent to the RTDS model where they were used for generating control signals to take further actions. The power-loss signals that were supplied by the RTDS model were from Propulsion Motor Drive 1 (PMD1), Propulsion Motor Drive 2 (PMD2), Rectifier (PREC), DC-DC Converter 1 (PCONV1), DC-DC Converter 2 (PCONV2), Main Turbine Generator 2 (PMTG2), and Auxiliary Turbine generator 1 (PATG1).

Fig. 12 shows the simulation results of an illustrative maneuver where the speed of the ship was consecutively changed between 25 nautical mi/h (knots) and 29 knots at an interval of 100 s. It is found that with the increase of ship speed, the power loss in the propulsion motor drives increases. The maximum power loss from PMD1 reaches a value of 3.6 MW (approximately 10% of the installed power) which leads to a corresponding maximum temperature of 53 °C. Similarly, the minimum power loss was found to be 2.3 MW, which brought down the temperature of the heat sink to 46 °C. Similar results were observed for PMD2 as well. During each change of ship speed, a small transient overshoot in power loss was observed which stabilized within a few seconds. Similar transient overshoots were observed in the results of the propulsion motors and the turbine-generator systems. Further results on this case study are available in [3].

This case study demonstrates the multi-domain cosimulation using distributed simulation resources which is suitable if the two modeled systems have widely different time constants. Otherwise, latency due to the geographical distance and communication delay may make the simulation unstable. However, if the two simulators are located in the same place and data exchange is performed through a faster network, such as direct digital connection between the two simulators, simulations of systems with smaller time constants would also be feasible.

VI. Conclusions

Multi-domain modeling and simulation are being used in the industry to tackle increasingly complex problems. This task force paper gives a brief overview of the issues and challenges involved in multi-domain simulation tools which have the common capability of simulating electrical networks along with others. It also discusses how two or more multi-domain simulation tools can be interfaced to perform special simulation tasks. Case studies illustrate such applications where at least two or more tools are interfaced to perform multi-domain simulation. Special care is needed in designing cosimulation through the interfacing of multi-domain simulation tools so as to avoid numerical inaccuracies and instabilities due to interfacing latency and data exchange.

References

[1] V. Engelson, H. Larson, and P. Fritzson, "A design, simulation and visualization environment for object-oriented mechanical and multi-domain models in modelica," presented at the 3rd Int. Conf. Inf. Visual., London, U.K., Jul. 1999.

[2] J. M. Carrasco, L. G. Franquelo, J. T. Bialasiewicz, E. Galvan, R. C. P. Guisado, M. A. M. Prats, J. I. Leon, and N. Moreno-Alfonso, "Power-electronic systems for the grid integration of renewable energy sources-A survey," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1002–1016, Jun. 2006.

[3] N. G. Hingorani, "Power electronics building block concepts," presented at the IEEE Power Eng. Soc. Gen. Meeting, Toronto, ON, Canada, Jul. 13–17, 2003.

[4] T. W. Webb, T. M. Kiehne, and S. T. Haag, "System-level thermal management of pulsed loads on an all-electric ship," *IEEE Trans. Magn.*, vol. 43, no. 1, pt. 2, pp. 469–473, Jan. 2007.

[5] B. Klassen, "A method for tightly coupled thermal-electrical simulation," *Microelectron. J.*, no. 28, pp. 239–245, 1997.

[6] L. Li and J. D. Yang, “Advanced simulation of hydroelectric transient process with Comsol/Simulink,” presented at the 25th IAHR Symp. Hydraulic Mach. Syst., Timisoara, Romania, Sep. 20–24, 2010.

[7] M. O. Faruque, M. Sloderbeck, M. Steurer, and V. Dinavahi, “Thermo-electric co-simulation on geographically distributed real-time simulators,” presented at the Power Energy Soc. Gen. Meeting, Calgary, AB, Canada, Jul. 2009.

[8] *“Dymola User Manual,”* Dynasim AB, Lund, Sweden, 2004.

[9] A. Pitchaikani, K. S. Jebakumar, S. Venkataraman, and S. A. Sundaresan, “Real-time drive cycle simulation of automotive climate control system,” presented at the 7th Modelica Conf., Como, Italy, Sep. 20–22, 2009.

[10] D. Winkler and C. Gühmann, “Synchronising a Modelica real-time simulation model with a highly dynamic engine test-bench system,” presented at the Modelica Assoc. Conf., Vienna, Austria, Sep. 4–6, 2006.

[11] H. Elmqvist, S. E. Mattsson, H. Olsson, J. Andreasson, M. Otter, C. Schweiger, and D. Brück, “Real-time simulation of detailed automotive models,” presented at the 3rd Int. Modelica Conf., Linköping, Sweden, Nov. 3–4, 2003.

[12] M. Dempsey, “Automatic translation of Simulink models into Modelica using Simelica and the Advanced Blocks library,” presented at the 3rd Int. Modelica Conf., Linköping, Sweden, Nov. 3–4, 2003.

[13] 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.

[14] *“Matlab/Simulink User Guide,”* MathWorks, Natick, MA, 2004.

[15] *“Simulation System SIMPLORER 6.0 User Manual,”* Ansoft Corporation, Pittsburgh, PA, 2002.

[16] *“COMSOL Installation and Operation Guide,”* COMSOL AB, Burlington, MA, 2008.

[17] A. Monti, E. Santi, R. Dougal, and M. Riva, “Rapid prototyping of digital controls for power electronics,” *IEEE Trans. Power Electron.*, vol. 18, no. 3, pp. 915–923, May 2003.

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

[19] R. Dougal, T. Lovett, A. Monti, and E. Santi, “A multi-language environment for interactive simulation and development ff controls for power electronics,” in *Proc. IEEE Power Eng. Soc. Conf.*, 2001, pp. 1725–1729.

[20] A. Monti, L. Cristaldi, A. Ferrero, F. Ponci, W. McKay, and R. Dougal, “A virtual environment for remote testing of complex systems,” *IEEE Trans. Instrum. Meas*, vol. 53, no. 5, pp. 123–133, Oct. 2003.

[21] A. Monti, M. Colciago, P. Conti, M. Maglio, and R. A. Dougal, “Integrated simulation of communication, protection, and power in MVDC systems,” in *Proc. Elect. Ship Technol. Symp.*, 2009, pp. 353–359.

[22] A. Monti, M. Colciago, P. Conti, M. Maglio, and R. Dougal, “A co-simulation approach for analysing the impact of the communication infrastructure in power system control,” presented at the Int. Simulation Multiconf., Istanbul, Turkey, 2009.

[23] C. Yongjuan, L. Qiang, and Y. Li, “A software for design and analysis of PMSM based on ANSYS,” presented at the 1st Int. Conf. Inf. Sci. Eng., Vancouver, BC, Canada, 2009.

[24] *“Mathmodelica User Guide,”* Mathcore Engineering AB, Linkoping, Sweden, 2009..

[25] N. Vatcharasathien, J. Khedari, J. Hirunlabh, and M. Daguenet, “A computer tool for designing solar-thermoelectric power generation system,” in *Proc. 22nd Int. Conf. Thermoelect.*, 2003, pp. 623–626.

[26] *“Magnet v7User Guide,”* Infolytica Corporation, Montreal, QC, Canada, 2010.

[27] J. H. Alimeling and W. P. Hammer, “PLECS-piece-wise linear electrical circuit simulation for Simulink,” in *Proc. IEEE Int. Conf. Power Electron. Drive Syst.*, 1999, vol. 1, pp. 355–360.

[28] S. Abourida, C. Dufour, J. Belanger, T. Yamada, and T. Arasawa, “Hardware-in-the-loop simulation of finite-element based motor drives with RT-LAB and JMAG,” in *Proc. IEEE Int. Symp. Ind. Electron.*, 2006, vol. 3, pp. 2462–2466.

[29] D.-X. Zhang, X.-H. Zeng, P.-Y. Wang, and Q.-N. Wang, “Co-simulation with AMESim and MATLAB for differential dynamic coupling of hybrid electric vehicle,” in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 761–765.

[30] *“MapleSim User Guide,”* Waterloo Maple, Inc., Waterloo, ON, Canada, 2009.

[31] U. Eicker, D. Pietruschka, J. Schumacher, JdC Fernandes, T. Feldman, and E. Bollin, “Improving the energy yield of PV power plants through Internet based simulation, monitoring and visualization,” presented at the 20th Eur. Photovoltaic Solar Energy Conf., Barcelona, Spain, Jun. 6–10, 2005.

[32] J. T. Bialasiewicz, E. Muljadi, R. G. Nix, and S. Drouilhet, “Renewable energy power system modular simulator.” RPM-SIM User’s Guide National Renewable Energy Laboratory, Rep. NREL/TP-500-29721, 2001.

[33] P. Bauer, P. Korondi, and P. J. van Duijsen, “Integrated control and circuit simulation for a motion control system,” presented at the EPE, Toulouse, France, Sep. 2–4, 2003.

[34] N. Wu, C. E. Lucas, C. Rands, I. E. Simpson, D. C. Aliprantis, and M. A. Masrur, “Distributed heterogeneous simulation of a hybrid-electric vehicle drive system using the simplorer software product,” presented at the SAE Power Systems Conf., New Orleans, LA, Nov. 7–9, 2006.

[35] ANSYS Workbench Platform. ANSYS, Inc., 2011. [Online]. Available: http://www.ansys.com/Products/Workflow+Technology/ANSYS+Workbench+Platform

[36] J. Sitar and P. Bauer, “Multidisciplinary analysis of electromechanical actuator based on FEM-circuit simulator coupling,” in *Proc. Power Electron. Mot. Control Conf.*, 2006, pp. 1256–1261.

[37] E. Rudnyi, From ANSYS to system level simulation: MOR for ANSYS. 2008. [Online]. Available: http://www.modelreduction.com/doc/papers/rudnyi08infoplaner.pdf

[38] A. Dehbi, W. Wondrak, E. B. Rudnyi, U. Killat, and P. van Duijsen, “Efficient electrothermal simulation of power electronics for hybrid electric vehicle,” in *Proc. Int. Conf. Thermal, Mechanical, and Multi-Physics Simulation and Experiments in Micro-Electronics and Micro-Systems, EuroSimE*, 2008, pp. 1–7.

[39] ModCoupler Module POWERSIM, Inc., 2011. [Online]. Available: http://www.powersimtech.com/index.php?name=modcoupler

[40] C. E. Lucas, E. A. Walters, J. Jatskevich, O. Wasynczuk, and P. T. Lamm, “A distributed heterogeneous simulation of a representative aircraft power system,” presented at the SAE Power Systems Conf., Coral Springs, FL, Oct. 29–31, 2002.

[41] E. A. Walters, M. A. Hasan, C. E. Lucas, O. Wasynczuk, and P. T. Lamm, “Variable communication rates in a distributed simulation,” presented at the 3rd Int. Energy Convers. Eng. Conf., San Francisco, CA, Aug. 15–18, 2005.

[42] T. Juhasz and U. Schmucker, “CAD to SIM: CAD model conversion for Dymola-based mechatronic simulation,” in *Proc. 10th Int. Conf. Comput. Model. Simul.*, 2008, pp. 289–294.

[43] D. Devaranjan, S. Stanton, and B. Knorr, “Multi-domain modeling and simulation of a linear actuation system,” in *Proc. IEEE Int. Workshop Behav. Model. Simul.*, 2003, pp. 76–81.

[44] M. O. Faruque, M. Sloderbeck, M. Steurer, and V. Dinavahi, “Geographically distributed thermo-electric co-simulation of all-electric ship,” in *Proc. ESTS* , Apr. 2009, pp. 36–43.