

Utilizing Behavioral Models in Experimental Hardware-in-the-Loop

Chad N. Miller
US Air Force

Michael Boyd
PC Krause & Associates

ABSTRACT

This paper introduces a method for conducting experimental hardware-in-the-loop (xHIL), in which behavioral-level models are coupled with an advanced power emulator (APE) to emulate an electrical load on a power generation system. The emulator is commanded by behavioral-level models running on an advanced real-time simulator that has the capability to leverage Central Processing Units (CPUs) and field programmable gate arrays (FPGA) to meet strict real-time execution requirements. The paper will be broken down into four topics: 1) the development of a solution to target behavioral-level models to an advanced, real-time simulation device, 2) the development of a high-bandwidth, high-power emulation capability, 3) the integration of the real-time simulation device and the APE, and 4) the application of the emulation system (simulator and emulator) in an xHIL experiment. The first topic will be addressed by targeting a behavioral-level model of a brushless dc motor drive with a pulse-width modulated inverter to a real-time simulator. The results of the real-time model will be compared to that of its non-real-time counterpart. For the second topic, hardware descriptions and data will be presented showing the APE's ability to track high-frequency command profiles. The third topic will cover the communication requirements and methods used for integrating the real-time simulator with the APE. Finally, for the last topic, the APE and real-time simulator will be integrated to emulate the brushless motor's dc-link current drawn by the inverter in a hardware experiment. In using the APE and the real-time simulator, it will be shown that it is possible to run detailed electrical models in real-time and emulate detailed electrical characteristics in hardware without requiring the actual hardware component. In following this process, it is possible to reduce risk and accelerate hardware and software development.

CITATION: Miller, C. and Boyd, M., "Utilizing Behavioral Models in Experimental Hardware-in-the-Loop," *SAE Int. J. Aerosp.* 9(1):2016, doi:10.4271/2016-01-2042.

INTRODUCTION

Physics-based modeling of integrated systems exists at four levels of detail: device physical, behavioral, functional, and architectural [1]. This paper will focus on behavioral-level models that aim to capture nonlinearities to better understand power quality, peak and regenerative power, and failure modes. The first topic of this paper is to evaluate a procedure to incorporate behavioral-level models of an electrical power system (EPS) into an xHIL system. Behavioral-level models pose a computational challenge, in which results generally take hours, or days, of actual time for fractional amounts of simulation time. In an attempt to mitigate this issue, there are two common solution implemented. In the first method, simulations are typically focused on a short, predetermined time range, in which the states of the system are initialized to known critical events [2]. This presumes there is prior, detailed knowledge of the system and its interactions and that critical events can easily be discovered and reproduced. In the second method, model fidelity and dynamics are

removed from the model. This inherently increases the execution speed of the model but does this at a cost of model accuracy. Therefore, there is a desire to increase the execution speed of detailed, behavioral-level models thus decreasing the overall simulation time required. By leveraging the speed of advanced computational devices, it is possible to execute behavioral-level models in real-time so they can be used in xHIL experiments. One could also benefit from increased model execution speeds for simulation-only based analysis since the direct outcome of increased execution speeds is a higher throughput of simulation results. For example, an optimization routine, previously too computationally intensive and not practical, may now be feasible using an advanced computational device. Even running a model at real-time, a run time ratio (RTR) of one, a behavioral model execution would be 10-2000 times faster.

Coupling hardware to software introduces more system interactions that can lead to identifying potentially unknown interactions. xHIL is a technique that focuses on the integration of multiple system components in an attempt to detect unknown interactions early in the development cycle. In an xHIL experiment, both the plant and controller are hardware components with emulated and/or simulated subsystems supplying boundary conditions to each other [1]. Historically, the Air Force Research Laboratory, Aerospace Systems Directorate has primarily focused on using dynamotors to emulate a turbine engine [3, 4, 5, 6, 7, 8]. In these scenarios, a model of an engine is executed on a real-time simulator and is emulated, in hardware, by sending shaft speed commands to a dynamotor that tracks the command. Models of mechanical systems, such as this, are simulated with large time constants lending themselves to a real-time simulator using a traditional CPU based architecture. A communication bandwidth of 200Hz between simulator and emulator is sufficient to track the command [3]. Detailed electrical system models are simulated using much smaller time constants, and therefore are more challenging to execute under a traditional architecture that will need to communicate at a much higher bandwidth. The APE will be used to address the requirement for high-bandwidth emulation of electrical system components. In a similar fashion as the drive stand, the APE can be used in an xHIL experiment to provide dynamic and interactive boundary conditions to the EPS. Command signals from a model can be sent to the APE at a sufficiently high-bandwidth such that the signals are accurately reproduced in hardware. An APE can be commanded to emulate a system component or device under test, creating dynamic stimulus to test hardware, providing a more realistic operating scenario with more relevant boundary conditions. As a result of coupling detailed models with highly-dynamic emulation devices, this capability can be used to identify system integration issues prior to hardware build or system integration, or in support of upgrades to fielded systems [8, 9].

REAL-TIME BEHAVIORAL MODEL SIMULATION

A model of a well-documented and understood brushless DC motor (BLDC) was chosen to assess the performance of a real-time simulator. The BLDC schematic shown in Figure 1 is of a three-phase, Y-connected, permanent magnetic synchronous machine. The particular circuit model is described in detail in [10]. The final evaluation of the solution would use an implementation of a sine-triangle modulated BLDC motor drive, which is described in detail in [11].

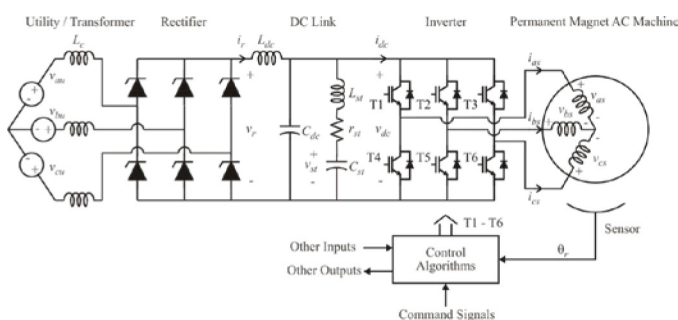


Figure 1. Brushless dc motor model circuit schematic

This particular system was chosen for several reasons. First, it is representative of power electronic based loads that the APE is primarily designed to emulate. Second, it contains switching level dynamics that cannot be simulated on a standard CPU-based real-time target. The model has an average RTR of 19.6 on a high-power workstation computer. In order to meet real-time requirements, a more advanced computational architecture would need to be utilized. Lastly, the resources exist to provide the hardware power and signal requirements for the various system components. In this case, the hardware/software interface will be established at the dc-link. The dc voltage is supplied physically in hardware, with the inverter, motor, and associated controls simulated on the real-time simulator and i_{dc} emulated by the APE as a load on the dc supply, as shown in Figure 2.

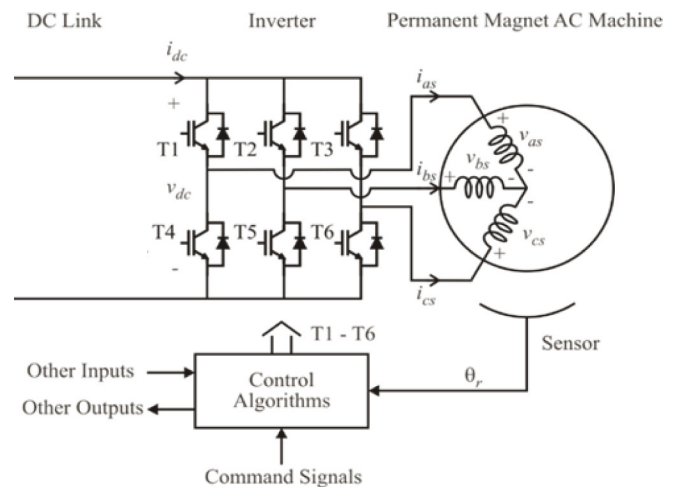


Figure 2. APE emulated dc load.

Using the BLDC model as a challenge problem, multiple real-time simulation solutions were investigated in an attempt to find a system that was able to not only meet the real-time simulation requirements, but also be effectively applied to an xHIL experiment. A proposed Opal-RT solution was determined to have the least risk as both the hardware and the software capability existed to fulfill both requirements.

The solution proposed involves combining two different computational architectures into a single real-time simulator. The first, more commonly used architecture, utilizes a commercial off-the-shelf CPU that is used to execute larger time step models. These typically execute at a rate of ten or more microseconds with common examples being basic control algorithms, averaged value circuit models, and basic physical models (e.g. ideal AC power generation and transformers) [13]. The second computational architecture uses a FPGA to execute behavioral-level models at time steps faster than a model's switching frequency allowing for accurate convergence. Opal-RT developed a specialized solver, called electric Hardware Solver (eHS), which is capable of solving complex behavioral-level models at time steps in the range of 100ns to 1μs [13]. eHS is based on a linearized, discrete equivalent circuit method, in which an ideal switch is represented by a constant conductance in parallel with a current source [14]. This method represents a closed switch as a small inductance and an open switch as a small