

# Electric Drive Vehicle Development and Evaluation Using System Simulation

A. Rousseau\*, S. Halbach\*, L. Michaels\*, N. Shidore\*, Na. Kim\*, N. Kim\*. D. Karbowski\*  
M. Kropinski\*\*

\*Argonne National Laboratory, Argonne, IL 60439  
USA (Tel: 630.252.7261; e-mail: arouseau@anl.gov).

\*\*General Motors, 3300 General Motors Road, MI 48380 USA (e-mail:  
michael.kropinski@gm.com)

---

**Abstract:** To reduce development time and introduce technologies faster to the market, many companies have been moving to Model-based System Engineering (MBSE). In MBSE, the development process centers around a multi-physics model of the complete system being developed, from requirements to design, implementation and test. Engineers can avoid a generation of system design processes based on hand coding, and use graphical models to design, analyze, and implement the software that determines system performance and behavior. This paper describes the process implemented in Autonomie, a Plug-and-Play Software Environment, to design and evaluate electric drive powertrain and component technologies in a multi-physics environment. We will discuss best practices and provide examples of the different steps of the V-diagram including model-in-the-loop, software-in-the-loop and component-in-the-loop simulation.

**Keywords:** Simulation, System Engineering, Control, Electric Drive Vehicles, Model Based System Engineering.

---

## 1. INTRODUCTION

Building hardware is expensive, time consuming, and limiting with respect to comprehending and accounting for variation in a system design. Traditional design processes in the automotive industry often delay control system design until late in the product development process, in some cases requiring several costly hardware iterations. To reduce costs and improve time to market, OEMs are turning towards Model-based System Engineering by placing greater emphasis on modeling and simulation early in the development process. MBSE is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities, beginning in the conceptual design phase and continuing throughout development and later life cycle phases [1]. To fully realize the benefits of math-based design, flexible and reusable multi-physics models must be created.

But greater reliance on modeling and simulation does come at some cost. Even if institutional inertia can be overcome, new processes must be put in place to facilitate communication between the model creators and consumers and to handle an increase in the number of files, which can become quite significant and overwhelming.

Model management introduces a set of requirements that are quite similar to software management requirements. There is a need to maintain version control for the models, as well as to implement a system-level model assembly capability, comparable to the need to interface, compile, and link the code modules that comprise a complete software build.

Project management requirements also exist, to ensure that all the files and data needed to create a complete simulation are organized, maintained, and provided as a complete package.

The Autonomie software [2, 3] provides an integrated environment and set of processes for managing, interconnecting, and integrating dynamic models of vehicle components and subsystems, to build and execute multi-physics system simulations. These simulations are then used for many different types of analyses, from trade-off studies of alternative propulsion system architectures to detailed control system design. Autonomie's Plug-and-Play architecture facilitates the use of component and subsystem models of selectable fidelity, permitting users to focus in detail on the area of interest they want to investigate.

## 2. MODEL BASED SYSTEM ENGINEERING

Model-Based System Engineering [4] provides an efficient methodology that includes four key elements in the development process: modeling a plant (from first principles or system identification), synthesizing and analyzing a controller for the plant, simulating the plant and controller together, and programming/deploying the controller. MBSE supports all these multiple phases and provides a common framework for communication throughout the entire design process.

The MBSE paradigm is significantly different from traditional methodology. Rather than using complex workflows and extensive software code, designers can formulate advanced functional characteristics by using

continuous-time and discrete-time computational building blocks. These models and associated simulation support tools can provide rapid prototyping, virtual functional verification, software testing, and hardware/software validation. In MBSE, models are at the core of the development process, from requirements development, through design, implementation, and testing. The control algorithm model is an executable specification that is continually refined and elaborated throughout the development process. MBSE allows one to improve efficiency by:

- Using a common design environment across project teams
- Linking designs directly to requirements
- Integrating testing with design
- Investigating the effects of variation on system performance and robustness
- Refining algorithms through multi-physics simulation
- Automatically generating embedded code
- Developing and reusing test suites
- Automatically generating documentation
- Reusing designs to deploy systems across multiple processors and hardware targets

Different steps are supported by a variety of approaches from Model-in-the-Loop (MIL), to Software-in-the-Loop (SIL), Hardware-in-the-Loop (HIL), Rapid-Control-Prototyping (RCP), or Component-in-the-Loop (CIL). Each process is used to address different stages of the development. Automotive companies have widely implemented these processes from code generation [5], to RCP [6] and HIL [7].

### 3. PLUG-AND-PLAY ARCHITECTURE TO SUPPORT MBSE

While most companies currently use MBSE, there is currently no single tool that allows them to share and reuse models from one phase to the next. Autonomie [3] is a software package designed in collaboration with General Motors (GM) to support the ideal use of modeling and simulation for math-based automotive control system design. While Autonomie does not meet all the needs of the development process, it provides a framework to link and integrate multiple tools (i.e. requirements, plant modeling, optimization, database management...) into an integrated workflow. Autonomie supports the assembly and use of models from design to simulation to analysis with complete plug-and-play capabilities. Models in the standard format create building blocks, which are assembled at runtime into a simulation model of a vehicle, system, subsystem, or component. All parts of the Autonomie graphical user interface (GUI) are designed to be flexible to support architectures, systems, components, and processes not yet envisioned. This allows the software to be molded to individual uses, so it can grow as requirements and technical knowledge expand [2]. This flexibility also allows for integration of legacy code, including models, controller code, processes, and post-processing equations.

### 3.1 Configuration Definition

The relative position of systems and their interconnections are defined by a set of configuration files. For example, a conventional vehicle can be represented by multiple configurations. One configuration may have a torque converter and a gearbox. Another may have those two systems grouped together under a single transmission system. Both configurations are valid for a conventional vehicle, and the selection between the two of them depends on the application. Figure 1 shows an example of a completed configuration in Simulink for generic component systems.

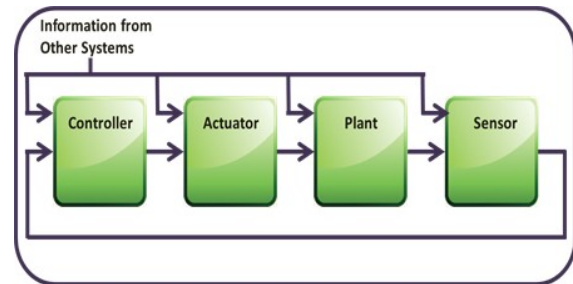


Figure 1: Configuration Example for Component (MIL)

### 3.2 Selection through the GUI

The GUI manages the system definition complexity and makes it possible for novice users, as well as experts, to quickly find the files they need, to stitch them together into a cohesive simulation, to execute standard work processes, and to perform database management functions without becoming overwhelmed. The first step in defining a model is to select its configuration. As the user selects various systems in the GUI, several configuration options may be available in the GUI and are selected through Drag-and-Drop. Once the configurations are selected, experts select their models and initialization files to define a system.

### 3.2 Building the Entire Model Automatically

Autonomie uses a unique approach that allows users to build a complete system model on the basis of configurations and systems selected from the GUI. This gives users the flexibility of saving and versioning models independently without having to manually connect each model's inputs and outputs. Users select the correct files in a user interface, and the automatic building algorithm uses metadata associated with the models to create the correct connections. This GUI also uses the metadata to facilitate the other necessary functions, such as compatibility checks and file selection.

Figure 2 shows how each selected system from the GUI is built and connected to represent the final model. It also allows component experts (e.g., engine experts) to focus on the areas they are familiar with, while being able to evaluate their component both as a standalone system as well as within a vehicle context. Systems can be exported and transferred that way, or the systems can be shared through the database. Autonomie includes as many pre-built systems as possible to give users a "jump start" into creating vehicles from scratch.

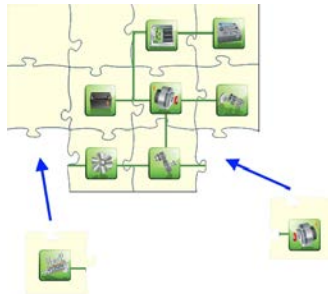


Figure 2: A Powertrain Is Assembled by System Experts on the basis of Selected Configuration

The unique architecture of Autonomie allows users to perform multi-physics system simulation through the different steps of the V-diagram using a single tool. The following sections provide examples of the tool's capabilities.

#### 4. MODEL IN THE LOOP EXAMPLE

Model-in-the-loop (MIL) can be used to address a very wide range of questions. As mentioned previously, pure simulation is usually used toward the beginning of the process, to allow engineers to study the performance of the system and design the control algorithm(s). Due to the short simulation time, a very large number of simulations can be quickly performed. This section provides an overview of some of the main applications that can be addressed using Autonomie for MIL.

##### 4.1 Vehicle Model Validation

Before performing any analysis, it is important to validate the component and sub-system models [8, 9] to provide confidence in the results. Autonomie allows users to import and analyze component and vehicle test data within the same environment as the models, thus facilitating the analysis and validation. Most of the vehicle test data used to validate the models comes from Argonne's Advanced Powertrain Research Facility (APRF). For hybrid and plug-in hybrid electric vehicles, one of the critical steps of the analysis consists of understanding the vehicle energy management. Indeed, it is important to understand the different states of the system (i.e. engine ON/OFF, shifting, regenerative braking...) and the selection criteria from a control point of view. For example, the engine turn ON conditions of the Prius HEV 2010 are shown in Figure 3 (i.e. the engine will be turned ON above a wheel torque demand of 400Nm at a wheel speed of 20 rad/s).

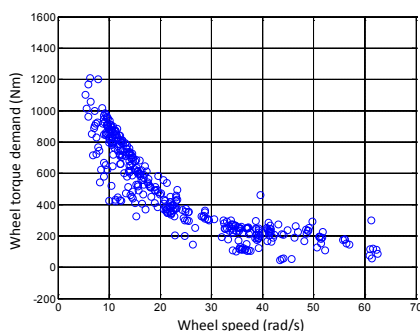


Figure 3. Prius HEV Engine Turn ON Conditions Used for Vehicle Level Control

Once the states are selected, the next step is to define the component operating points under different conditions. Particular focus should be paid to thermal effects as shown in Figure 4.

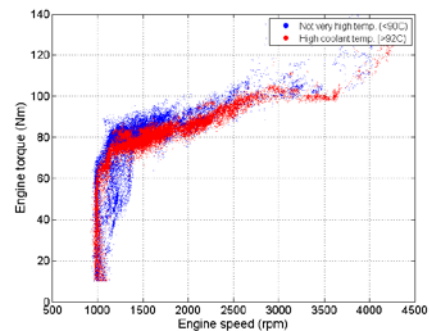


Figure 4 Prius HEV Engine Operating Conditions for Different Temperatures

The complete vehicle model can then be validated ensuring that each individual component behaves properly compared to test data. Figure 5 shows an example for the 2010 Toyota Prius HEV model comparing fuel consumption, battery state-of-charge and component temperature. Outstanding differences are attributed to test driver uncertainties which influence engine ON events and consequently battery SOC.

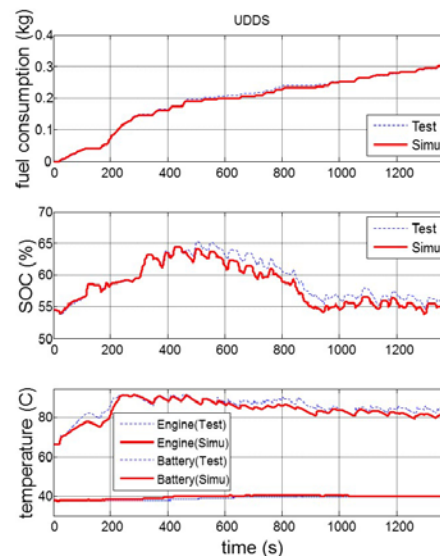


Figure 5. Prius HEV Model Validation with Test Data

##### 4.2 Impact of Powertrain Comparison

Electric drive vehicles offer increased flexibility from a powertrain configuration point of view with hundreds of possible combinations. As a result, it is critical for any multi-physics system simulation tool to handle a wide number of powertrain configurations to understand the advantages and drawbacks of different options under various driving and thermal conditions [10]. Autonomie currently includes close to 200 different powertrain configurations.

Figure 6 shows a comparison between the GM Volt ec powertrain configuration and a series hybrid configuration for a PHEV application. Such analysis provides researchers with insight on component sizing, performance and energy

management. In the example shown below, the sizing results showed that the GM Voltec powertrain requires less component power to meet the vehicle technical specifications than does a series hybrid as a result of the many component efficiencies between the engine and the wheel. In addition, simulations were performed to characterize the impact on component operating conditions and fuel consumption during urban and highway driving. Using the series mode in the GM Voltec implies that a relatively larger motor is required to address the vehicle's power requirement. In the GM Voltec power-split mode, the motor is used to assist the engine; in the plug-in series hybrid, the motor works as a generator. It was determined that the GM Voltec powertrain consumed up to 5% less energy during all driving condition modes than did a pure series hybrid configuration on the UDDS cycle.

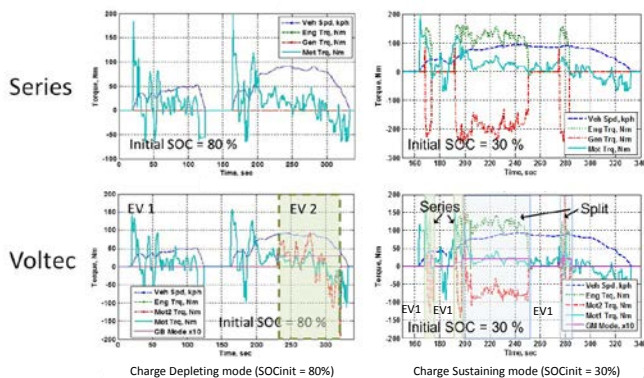


Figure 6. GM Voltec PHEV and Series PHEV Comparison on the UDDS driving Cycle

#### 4.4 Impact of Vehicle Energy Management through GIS

Another critical benefit of system simulation is the ability to quickly and efficiently develop and evaluate the impact of advanced vehicle level energy management algorithms. In the case of plant models that link to expert tools (i.e. AMESim, GTPower...), Autonomie supports integration of third party tools for specific use cases. Since optimization based control methods for Plug-in Hybrid Electric Vehicles require the knowledge of an entire driving cycle and an elevation profile to obtain the optimal performance over fixed driving route, linking with additional tools such as HERE for traffic information is quickly becoming of utmost importance [12, 13]. Figure 7 shows an example of route based control including a geographical information system (GIS).

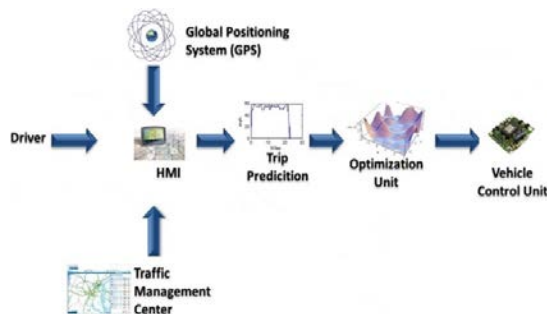


Figure 7: Route Based Control Implementation using Geographical Information System

#### 5. SOFTWARE IN THE LOOP EXAMPLE – PRODUCTION CONTROL DEVELOPMENT (GM)

Once high level analysis is performed during the MIL phase, higher fidelity models and controls can be used to develop and test production controls. During that phase, it is critical that the tool be able to handle a wide range of modeling languages and model complexity. Autonomie is currently being used by GM in several critical projects to accelerate the implementation of the next generation of advanced technologies for engine, transmission, and hybrid applications (including the next generation Chevrolet Volt and Chevrolet Malibu with eAssist). The methodology used for virtual ECU software development is that of system simulation, which calls for the entire system to be modeled, including plants, sensor/actuators, controller hardware, and algorithm/application software.

Although plant models may be available, or can be readily developed, the control algorithm models are not all available, and the only representation of the algorithm functionality may be contained in existing software. In these cases, a full system simulation can be performed only when a complete software build has been created and loaded into the target embedded controller. Then the control algorithm software can be tested in a hardware-in-the-loop (HIL) system, where the actual controller and its application software is connected to a real-time computer system that runs a simulation of the plant model(s) representing the physical system with which the controller will eventually be used. This is only possible when controller hardware and associated software are available, and is quite difficult and costly when multiple controllers are involved.

System simulation has a number of challenging requirements, including:

- The need to support flexible modelling,
- The ability to interconnect many complex models involving hundreds of input and output signals,
- Representation of all control system functionality, when many control function models are not available
- Model sharing facility, where models are made available to a large user community

The challenges listed above were solved with a combination of the Autonomie software developed at Argonne National Laboratory and a unique software-in-the-loop (SIL) capability created at GM Powertrain Engineering [14]. Since the adoption of Autonomie, the Electric Vehicle Controls group at GM has been using this methodology for virtual controls development. GM has created a distributed work environment that leverages the plug-and-play capabilities of Autonomie such that plant models, sensor/actuator models, control algorithm models, and compiled production software components of the system that are developed by different people throughout the organization may be seamlessly integrated into a system simulation. This facility enables the use of models having varying degrees of fidelity that are implemented in a number of commercial simulation tools.



Figure 8 shows the controller model structure and includes:

- A block for the real-time operating system (RTOS) that schedules the execution of the tasks in the software and algorithm models
- A Controller Area Network (CAN) communication block to supply serial data signals to the controller software
- An algorithm model (AM) block that contains the AMs under development
- An hardware I/O (HWIO) block that contains behavioral models of the HWIO functions in the controller

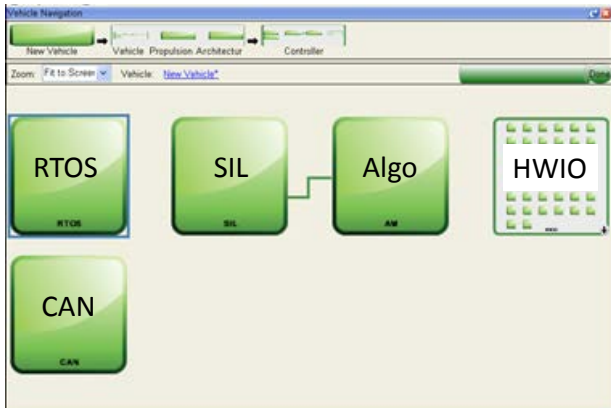


Figure 8: Configuration Selected for Software-in-the-Loop

A formal process has been established to allow users to archive and share models and complete system simulations. Plant models are reused for HIL as well as system simulation, minimizing rework and redundant development. Migration from one controller software release to the next is easily accommodated by the automated features of the Software-in-the-Loop process and the Autonomie interconnect capability. The open architecture of the system simulation facility has permitted integration of GM internal tools, such as the Global Automated Test Tool (GLATT) and the Test Automation Suite (TAS). Figure 9 illustrates an example of the automated insertion of a control algorithm model into the simulation.

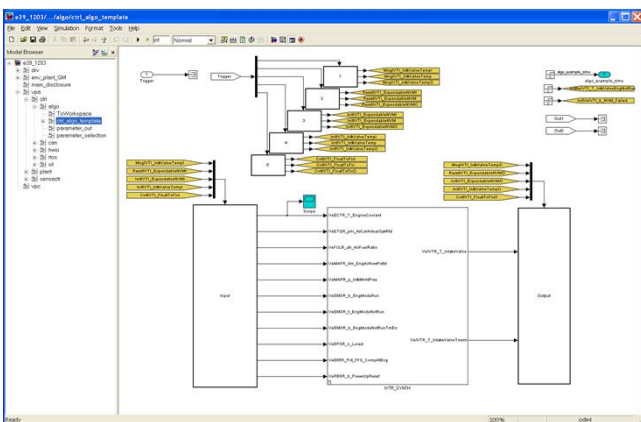


Figure 9. An Algorithm Model Inserted Into a Baseline System Simulation

6. COMPONENT IN THE LOOP EXAMPLE

Figure 10 shows a block diagram representation of BIL.

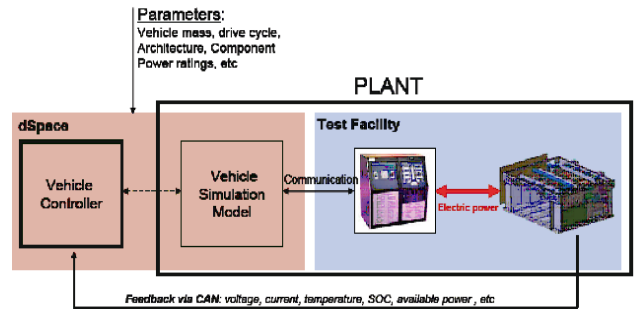


Figure 10: Battery in the Loop Block Diagram

Battery hardware is subjected to power demands from a virtual vehicle in Autonomie [15]. A high-voltage DC power supply (the ABC-170 CE) sinks and sources power from the battery to emulate battery utilization on standard dynamometer cycles or real-world driving. Real-time feedback from the battery (SOC, voltage, temperature, current limits) is used by the virtual vehicle energy management controller as a part of its strategy. To control hardware or receive feedback, specific logic needs to be introduced. This provides a convenient point for the user to implement a testing plan or to enforce checks to ensure safe operation of the hardware. Figure 11 describes the generic configuration setup used for hardware/software interactions. Note that the configuration selected for CXIL application is radically different than for both MIL and SIL.

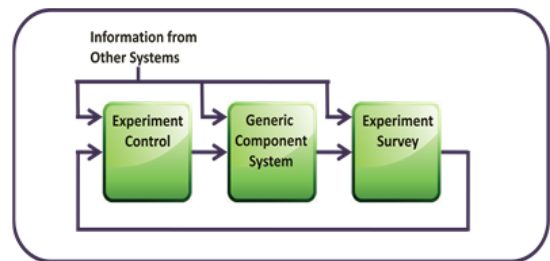


Figure 11: Configuration for Component-in-the-Loop

The Autonomie simulation of a midsize pre-transmission parallel vehicle with a 41-A·h, 10-kWh JCS battery in electric operation was compared to the same vehicle with an actual physical battery (same specification) by using BIL. The aim of the experiment was to validate the battery simulation model. Proper validation is only possible if the battery model can be seamlessly replaced by an actual battery, with the rest of the vehicle model remaining untouched. Autonomie provides the flexibility for such kinds of experiments. After the simulation, Autonomie was configured for CIL, as described in the section above. Figure 12 compares the Autonomie simulation model voltage (red) with the actual battery voltage (blue).

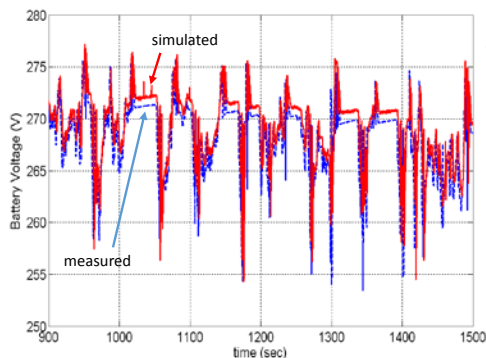


Figure 12: Battery Voltage Comparison

## 7. CONCLUSIONS

To reduce costs, the automotive industry has been embracing Model-based System Engineering for modeling, simulation, testing, and analysis. This paper proposes an ideal modeling process, wherein experts produce libraries of high-quality models in varying levels of fidelity for use throughout an organization and across the automotive industry. These models connect seamlessly for maximum reusability and flexibility, making collaboration quick and easy. The models developed by these experts can be used from the beginning to the end of the design process, from high-level configuration sorting studies, to code testing with production software (such as SIL, HIL, or RCP), and, finally, to solving production problems. Each system (e.g., plant or control) can be either represented by a set of equations or by the actual hardware. To quickly evaluate new technologies in different environments, a generic process was developed to replace any part of the system through the graphical interface. This generic process allows companies to increase their productivity and bring technologies to the market faster. The tools and methods described in this paper have been adopted by a large number of companies. For example, the software-in-the-loop process has been adopted by the General Motors Electric Vehicle Controls group to enable virtual ECU software development via system simulation. Control algorithm development in the context of a complete system simulation provides many advantages that result in high-quality robust control systems. Engineers are able to account for interactions among the many functions, subsystems, and controllers that comprise modern automotive controls, and to study the effects of variation on the system's robustness.

The main features of the software that empower the algorithm developer are:

- Automatically build a complete system simulation from the constituent subsystem models
- Provide a development environment that supports MIL, SIL, HIL, and virtual rapid prototyping
- Establish a set of tools and processes for enterprise-wide collaboration
- Support customization of architectures, models, processes, and analysis
- Allow simulation at any level – single component -> subsystem -> Powertrain -> complete vehicle
- Manage models, data, processes, results, and controller code from research to production

## ACKNOWLEDGEMENTS

This work was supported by DOE's Vehicle Technology Office under the direction of Lee Slezak and David Anderson. The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

## REFERENCES

1. INCOSE SE Vision 2020 (INCOSE-TP-2004-004-02), Sept 2007
2. Argonne National Laboratory, *Autonomie Documentation*, 2013.
3. Argonne National Laboratory, Autonomie, [www.autonomie.net](http://www.autonomie.net), 2013
4. Paul Smith, The MathWorks, "Best Practices for Establishing a Model-Based Design Culture," SAE 2007-01-0777, SAE World Congress.
5. General Motors, "Automatic Code Generation Process," MathWorks Automotive Conference, June 2005.
6. Oguz Dagci, General Motors, "Real Time Interface Blockset Development in Matlab/Simulink for On-Target Rapid Prototyping," SAE 2006-01-0169.
7. Sergey Semenov, Ford Motor Company, "Automation of Hardware-in-the-Loop and In-the-Vehicle Testing and Validation for Hybrid Electric Vehicles at Ford," SAE 2006-01-1448.
8. N. Kim, M. Duoba, N. Kim, and A. Rousseau, "Validating Volt PHEV Model with Dynamometer Test Data Using Autonomie," *SAE Int. J. Passeng. Cars - Mech. Syst.* 6(2):985-992, 2013,
9. N. Kim, A. Rousseau, E. Rask, "Vehicle-level control analysis of 2010 Toyota Prius based on test data", Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering Volume 226 Issue 11, November 2012
10. N. Kim, J. Kwon, A. Rousseau, "Comparison of Powertrain Configuration Options for Plug-in HEVs from a Fuel Economy Perspective", SAE 2012-01-1027, SAE World Congress, Detroit
11. B. Walton, A. Rousseau, "Fuel Efficiency Benefit for Electrified Vehicles from Advanced Spark-ignition Engine Technologies", EVS27, Nov 2013, Barcelona
12. N. Kim and A. Rousseau, "Sufficient conditions of optimal control based on Pontryagin's minimum principle for use in hybrid electric vehicles," IMechE Part D: J. Automobile Engineering, vol. 226, no. 9, Sept. 2012, pp. 1160-1170.
13. D Lee, Suk Won Cha, A Rousseau, N Kim, D Karbowski, "Optimal Control Strategy for PHEVs using Prediction of Future Driving Schedule", EVS26, May 2012, Los Angeles
14. L. Michaels, et al. (Argonne National Laboratory), M. Kropinski, et al. (General Motors), "Model-Based Systems Engineering and Control System Development via Virtual Hardware-in-the-Loop Simulation, SAE 2010-01-2325, SAE Convergence Conference, October, 2010
15. N. Shidore, T. Bohn, "Evaluation of cold temperature performance of the JCS-VL41M PHEV battery using Battery HIL", SAE 2008-01-1333, April 2008, Detroit