

Model-Based Design Approaches for Plug-In Hybrid Vehicle Design

Mendes, C.J.^{1*}, M.B. Stevens², E.J. Wilhelm⁴, M.W. Fowler², R.A. Fraser³

¹CrossChasm Technologies, 150 Water St. South, Cambridge, Ontario, N1R 3E2

²Department of Chemical Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1

³Department of Mechanical Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1

⁴Energy Systems Analysis, Paul Scherrer Institute, OVGA/426, Villigen PSI, CH-5232, Switzerland

*Presenting Author

Plug-In hybrid vehicles represent a significant transition for conventional vehicles in both component type and available modes of operation. The resulting increase in degrees of freedom in the architecture and control, further amplifies the need for advanced design tools currently under development in the conventional vehicle design space. The design tools must give accurate insight into architecture selection, component sizing, and control algorithms within a short development timeframe and with limited complexity. The use of model-based design principles, which have the ability to achieve these objectives, will be required for the design and prototyping of plug-in hybrid vehicles.

This paper develops a model-based design process for plug-in hybrid vehicle design. Specifically, the intermediary steps between initial concept and working vehicle prototype are discussed, highlighting on Software-In-The-Loop (SIL), Hardware-In-The-Loop (HIL), and Component-In-The-Loop (CIL) design phases. The associated objectives, challenges, and software tools are discussed for each phase. The process of initially developing virtual prototypes and the gradual transition to physical prototypes is presented. The ability to utilize auto-generated control code to seamlessly connect the simulations to the real-world implementation is highlighted. The major benefits and complications are outlined, as the process is contrasted against the conventional design process. In addition, the existing drawbacks of current simulation environments, and the potential benefits of integrating an advanced vehicle simulation network into the design process is considered.

Key Words: Plug-In Hybrid, Simulink, Model-Based Design, Hardware-In-The-Loop, Software-In-The-Loop, Component-In-The-Loop, Vehicle Simulation

Introduction

There are a number of reasons for the increased research focus on hybrid electric vehicles (HEV's) in recent years: Consumer's attention focusing onto the impact of the transportation system on the environment, the rising price of petroleum-derived products, and sensitivity of the North American economies to geopolitical events in the oil-producing regions of the world [1]. These factors are combining to encourage a fundamental shift in the automotive sector, during which it may become, or perhaps has already become, economically advantageous to produce vehicles which have higher initial costs, but lower relative fueling costs. Thus far, the approach taken by the automotive manufacturers has been to start producing vehicles hybridized with electric subsystems – high voltage batteries and motors.

The design of hybrid electric vehicles focuses around the concepts of peak shaving, or energy supply optimization. These are similar concepts to those used by the power generation industry involved in planning capacity; certain technologies are naturally more efficient (either cost or energy specific) in certain operational regimes, and multiple different technologies may be aggregated together to provide a more optimum solution to a particular energy profile requirement. In the power industry, the focus is cost-efficiency in the context of the market's diurnal and seasonal energy profile: Base demands and peaking demands may be more efficiently served by many disparate technologies, and the handoff between them is driven (in part) by the open market prices. In an internal combustion engine (ICE) powered HEV, the peak demand could be served by the electric system, and the base demand by the engine. The process by which the system is controlled is called the hybrid control strategy (HCS).

Since the concept of hybridization is predicated on the energy profile, the drive cycle the vehicle follows will play a paramount role in the effectiveness of any hybrid vehicle. The components are generally selected in such a way as to optimize the system with respect to fuel consumption, energy efficiency, range, total greenhouse gas production, secondary emissions, and performance. Since the interactions between the vehicle drive cycle, the specific components, and each of the output metrics can be complex and interdependent, simulation tools play an important role in the design of these vehicles. Three simulation tools are widely used in the research area; Powertrain Systems Analysis Toolkit (PSAT), ADvanced VehIcle SimulatOR (Advisor), and CRUISE. All three tools give the vehicle designer the ability to evaluate HEV components, control strategies, and overall vehicle performance. One key note here is that the fundamental modeling approach used in Advisor differs from that taken in PSAT and CRUISE in that Advisor is a so-called backwards facing model, making it less useful for some of the advanced techniques to be discussed in this paper.

One key to understanding the design of the current fleet of available HEV's, is a phenomena not well captured by any of the existing modeling tools: Battery degradation. Current HEV's mitigate battery degradation primarily by sizing the battery such that the battery may fulfill its function of supplementing high power to the drivetrain, while only shallowly cycling the battery's state of charge[2]. This approach works well in that it enables the battery to last for the life of the vehicle, however it drastically limits the total amount of time the vehicle may run in all-electric mode. This type of battery control is referred to as "charge-sustaining" – all energy must ultimately come from the internal combustion engine.

However, the powertrain efficiency of HEV's may be further increased by turning to a HCS referred to as "charge-depleting". Here, the vehicle must have significantly more *available* on-board electrical energy than a "charge-sustaining" HEV, in order to allow for an extended all-electric range, as well as additional electric power in order to provide a more comprehensive all-electric ability than a charge-sustaining HEV. This vehicle, the plug-in hybrid vehicle (PHEV), brings with it more challenging engineering problems: deep state-of-charge cycling holds more negative implications for battery durability, the

sensitivity to the drive cycle is driven even higher than pure HEV's due to the modal nature of the HCS – the vehicle will switch from a “charge-depleting” to a “charge-sustaining” HCS upon reaching some level of battery depletion, and the battery now requires both high energy density and high power density (instead of simply high power density for HEV applications, and high energy density for EV applications). It should be clear that the HEV designer's dependence on simulation for component sizing and control optimization will only be further heightened upon their transition to PHEVs.

This paper focuses on the problems that HEV and PHEV vehicle integrators will encounter as they design/refine/build these vehicles. It is intended that the role of simulation and the benefits of simulation in this design environment will be clearly demonstrated in the areas that are *unique* to the HEV/PHEV sector. These areas are as follows:

Component selection

A method for evaluating existing components is given using commercially available software. Further, a method to map out and help identify components (existent or not) that would be ideally suited for a given application is given. A real-world example of how design space mapping may yield benefits in preliminary vehicle design feasibility studies is offered.

Hybrid control strategy and component-level control development

The development of effective controls for HEVs is often the most challenging part of HEV development. It is extremely daunting to evaluate and tune different hybrid control strategies in-vehicle, due to the inherent, extended length of such tuning exercises. Tuning these strategies in a virtual environment can speed this process up dramatically. An example of how the simulation of these strategies may be linked in real time to the testing of physical components in order to evaluate critical metrics that would be otherwise difficult to model is given.

The role of simulation in the development of lower level component controllers is important. Attempting to evaluate or tune prototype control loops on real-world component can poses a higher risk than doing so in a simulation environment. An example of how and where lower level control loops can be implemented into a real vehicle is given, and how using this method can improve confidence in obtaining vehicle control optimality. Further, the additional software development needed to support the in-vehicle code is given a framework that may yield additional lead-time benefits. In addition, an example of how the control hardware destined for the vehicle can be integrated into the simulation loop is discussed, and the benefits of such an approach illustrated.

Design Process Streamlining

It will be shown how simulation may play a central role in the entire vehicle development program, starting with component evaluation, and architecture design, up to the process of final bug tracking inside vehicle controllers. This methodology may yield large increases in deployment speed to the vehicle integrator. How simulation networks can play a role in speeding deployment time is discussed.

Example Vehicle Primer

The majority of hardware and vehicle examples used in this paper will focus on the work performed since 2004 out of the University of Waterloo's WATCAR institute. In particular, the mule vehicle is a 2005 Chevrolet Equinox, which was converted to a fuel cell hybrid electric vehicle (FCHEV) as part of the University's entry into 'ChallengeX', an advanced vehicle competition sponsored by the U.S. Department of Energy and GM.

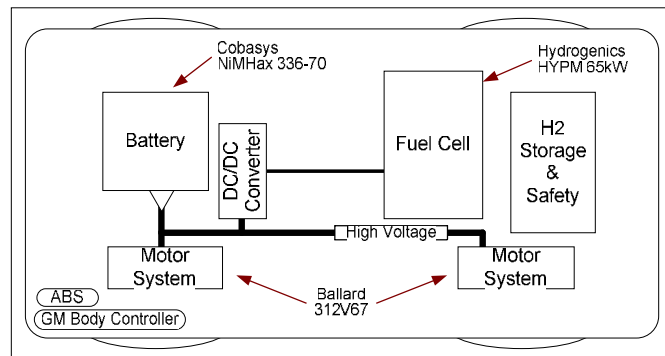


Figure 1: UW's 2005 Fuel Cell Equinox Powertrain Schematic

The stock Equinox's powertrain and driveline were replaced with two electric axle-motors, and a 65kW fuel cell and 60kw NiMH battery pack were used in order to provide electrical power. While this vehicle uses non-conventional hybrid components, the design concepts and methodology used in its design is very similar to that used to design more conventional HEVs or PHEVs. The vehicle schematic can be seen in Figure 1.

Component Selection and Hybrid Control Strategy Development

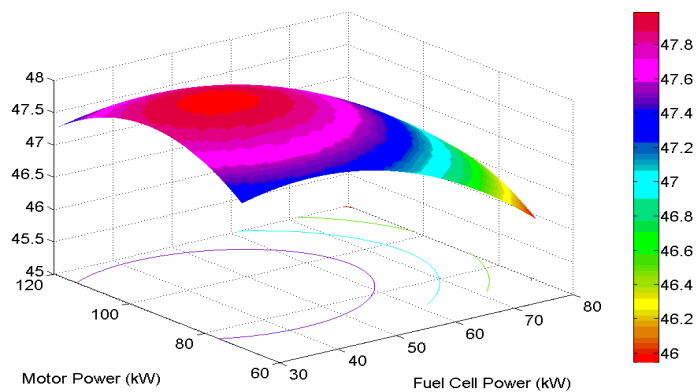
Component Selection and Sizing

There are several simulation software packages available to perform both component selection studies, and HCS development. In general, these packages attempt to simulate a virtual vehicle using quasi-static standardized component models that estimate the power flows, efficiencies, and other operating point of each of the relevant vehicle components. This includes engine models, motor models, battery models, vehicle drag models, mechanical and electrical accessory load models, among others. These different components are aggregated together in order to create a virtual vehicle in the specified hybrid configuration. These vehicles are run through standard drive cycles in order to estimate performance metrics such as fuel economy and acceleration performance, as well as to track the performance of individual components. This method of aggregating standardized component models to build vehicles allows the rapid evaluation of many different types of the same component. For example, to evaluate a multitude of engines

while holding all other component models constant, a change to the specified engine model in the vehicle would be the only modification required. Further, models can typically be scaled (in terms of power, or energy, or both), in order to allow the evaluation of a component that does not exist.

In general, these simulation tools are used to map design spaces, to identify subtle interactions between components, to model individual vehicles for component sizing optimization, and/or to develop hybrid control strategies. Parametric studies may also be run to help optimize component sizing by scaling components automatically, and batch processing multiple trials.

In addition, an experimentation method called the ‘Design of Experiments’ may be used to reduce the total number of simulations that must be performed to map the design space. This method allows the reduction in simulation runs by discounting the impact of high level interactions occurring between sundry components. An implication of this might be as follows: simultaneous downsizing of both an engine and a transmission may yield measurable benefits greater than the simple sum of the individual benefits associated with downsizing each component (this positive interaction between components may be found by the DOE method), however the DOE method would assume negligible interactive benefits if this concept is expanded to three or more components (ie. assuming the net benefit is equal to the sum of the individual benefit in those cases). This assumption will, of course, obscure any higher-level (greater than three in this case) interactions between components, but will allow the designer to drastically reduce the number of cases they must evaluate. The DOE method was used by the University of Waterloo in evaluating all possible components (existent or not) while designing the fuel cell powertrain. One result of that study is given in Figure 2.



**Figure 2: Results from the D.O.E. - UW Equinox
Estimated Vehicle mileage (mpg) as a function of fuel cell
power and motor power**

The DOE method was used by the University of Waterloo in evaluating all possible components (existent or not) while designing the fuel cell powertrain. One result of that study is given in Figure 2.

Hybrid Control Strategy Development

These quasi-static vehicle models may also be used effectively to create, test, and tune hybrid control strategies either concurrently with the component selection process, or as a next step in the design process.

Hybrid control strategies are largely responsible for how to manage power flow from the powertrain. The exact responsibilities of the HCS depend on the particular vehicle’s

configuration, the degrees of freedom in the powertrain, and the metrics of interest. The HCS used in the University of Waterloo's vehicle was designed to determine what fraction of the total power needed by the vehicle, the battery should produce. Its goal was to produce an 'optimum' alpha as defined below:

$$\alpha = \frac{\text{BatteryPower}}{\text{TotalElectricalPower Required}}$$

In this case, the definition of optimum was the value of alpha which would:

1. Produce the power required quickly
2. Minimize degradation of the battery
3. Minimize fuel cell on/off cycling
4. Maximize the overall powertrain efficiency
5. Maintain a desired SOC range for the battery

Subject to the constraints:

1. Provide instantaneous power delivery from the battery to allow the fuel cell air delivery system to ramp up/down
2. Respect the maximum and minimum battery power restrictions
3. Respect the maximum and minimum motor voltage restrictions
4. Respect the voltage boost limitations of the DC/DC converter

The method chosen to solve this optimization problem was an in-vehicle search algorithm. A simplified vehicle model was implemented onto the on-board vehicle control hardware, and the optimization metrics were estimated at ten valid alpha values during each control loop iteration. The metrics were scaled and summed, and the alpha producing the largest optimization value was chosen [3]. Tuning such an algorithm in a prototype vehicle is not feasible due to the number of drive cycle iterations required to obtain good results. The rapid simulation of the HEV enabled the parameter tuning of the HCS under a variety of expected drive cycles and outlier conditions. One example of simulation output can be seen in Figure 3 – the transient response of different UW Equinox powertrain components as the power demand of the motors varies, and the HCS commands the Fuel Cell and DC/DC converter in response.

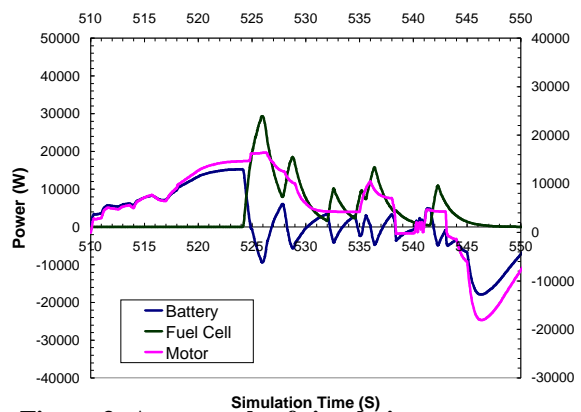


Figure 3: An example of simulation output during HCS simulation

Advanced Modeling for Low Level Controllers

While HCS development may be accomplished using existing tools and simple component models, it is a less defined task to create the lower level controllers that must manifest the HCS's decisions, as well as any other algorithmic goals. This could involve creating control loops to achieve desired power delivery or torque production, or CANbus messaging setup and timing, or component handshake timing.

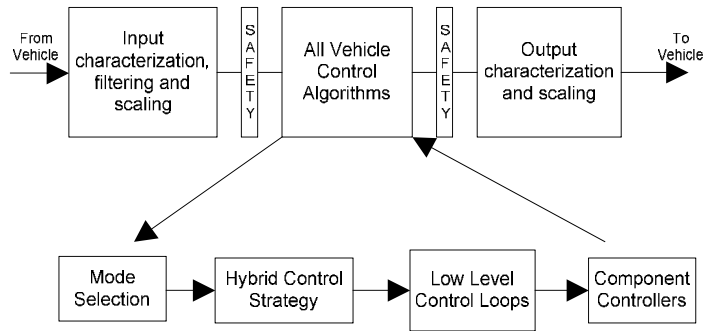


Figure 4: In-vehicle control architecture: Main vehicle control loop

Figure 4 illustrates a typical HEV/PHEV in-vehicle control architecture, showing the hierarchy of the active code while the vehicle is in the run state; once the newest input data is read from the input ports of the controller, safety validations may be run to ensure no unexpected behavior has occurred, the vehicle is put into an active state, and (if in state 'run'), the HCS is computed, followed by the control loops to produce an actionable output, then followed by the low-level component controllers, whom will ensure that the handshaking/communication requirements of the components are being met. Following this process, the data may be again scanned for safety flags, and sent to the scaling algorithms to be prepped for output. These multiple layers of software may also be visualized using the software abstraction diagram in Figure 5, starting from the lowest level (communication interfaces) at the bottom, to the highest level (vehicle modes) at the top.

Note that of all the code in Figure 4 and Figure 5, typically only HCS control development is readily performed in the current software-in-the-loop HEV simulation tool, and the balance of controls development will take place in a separate simulation environment. Obviously, a more ideal technique would allow the development of as much of this code as possible from within the original simulation framework.

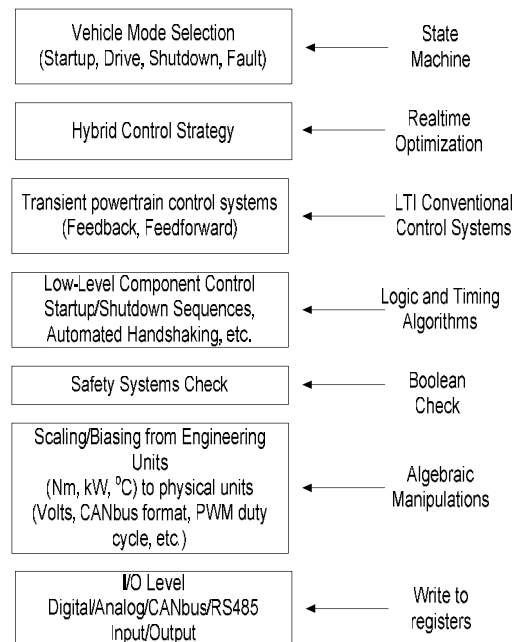


Figure 5: In-vehicle control architecture: Software abstraction layers

The conventional approach to low level control loops development is to simulate in a Matlab/Simulink environment, using custom

vehicle component models to perform preliminary tuning. There, an evaluation of worst-case scenarios may be performed to ensure the system remains stable and, if necessary, catches and gracefully corrects exceptions.

One example of this type of control loop tuning involves the torque control strategy on the UW Equinox. One possible implementation is shown in the code schematic of Figure 6. In this two stage control scheme, the presence of the fuel cell is ignored, and motor torque is controlled through a conventional proportional scheme (stage

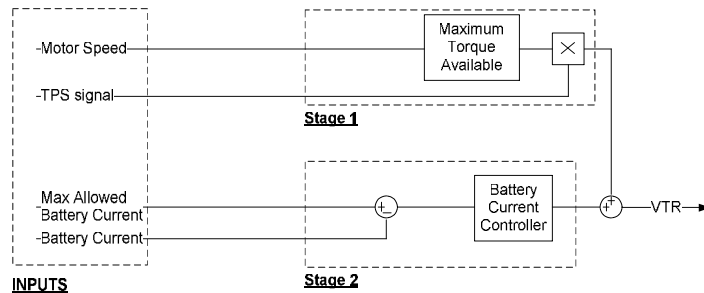


Figure 6: Possible torque control strategy for the UW Equinox (TPS signal = throttle position sensor signal, VTR = vehicle torque request)

1), and a battery current limiter (stage 2). This scheme allows the driver to nominally request as much torque as desired, subject to only the motor limits, until the battery current gets too close to the maximum limits allowed by the batter controller. At this point, the “battery current controller” latches on and will actively mitigate the torque request in order to maintain the battery current to acceptable levels. Once there is sufficient throttle release from the driver, the “battery current controller” de-latches, and torque control is returned to stage 1. A simulation result from an acceleration run is shown in Figure 7. Note the tight control of the system once the battery current control latches.

It is also desirable to develop component models to represent not only the physical mechanisms of the device, but each of the component controllers as well. One of the main objectives in tuning closed loop controllers is to evaluate the impact of signal propagation delay on the control loop. While this could (and should) be done in Simulink during the development of the closed loop controllers, one further step would be to integrate *component* controller behavior (modes, failures, timeouts, handshaking, limits, communication) directly into the vehicle modeling platform. This would allow the evaluation of the balance of the software abstraction layer (see Figure 5), with the exception of the I/O level, inside the same software platform as the preliminary HEV simulation tool and would allow control validation traditionally requiring hardware-in-the-loop systems, or mule vehicles, to be performed completely in software. Obviously, achieving worthwhile results by doing so would even more heavily

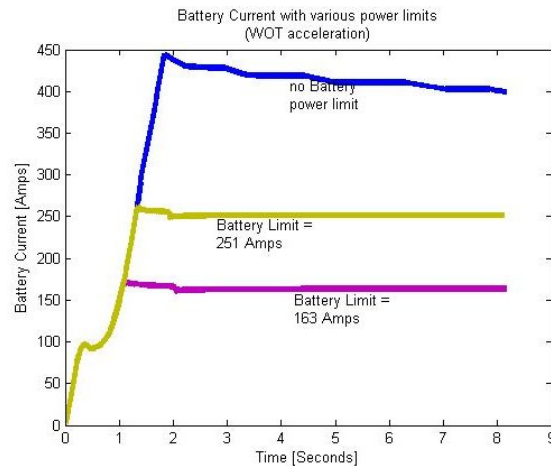


Figure 7: The impact of differing battery power limits on a 0-100kph acceleration run using the two-stage torque control strategy depicted in Figure 6.

depends on the quality and fidelity of the component models, however as will be shown in the “*Hardware Development*” section, upstream integration of this sort within the Simulink environment can yield great benefits in light of the recent advances in auto-generated embedded code, and vehicle-ready controller hardware. To the author’s best knowledge, performing vehicle controller design so early in the design process is typically not done. In general, these lower level vehicle controls are developed in HIL/CIL, or on the vehicle prototype itself.

Hardware development using the Hardware-in-the-Loop concept

So far, HEV/PHEV simulation tools have been discussed that can help select hybrid vehicle components and can develop and tune a preliminary HCS. A development process for the lower-level vehicle controller code has been given, and the suggestion made that a standardized integration of component *controllers* into a simulation tool can yield code that is near vehicle-ready. The next logical question is: How should the code be taken into the real-world?

One of the benefits of working in the Simulink environment is the presence of the Real-Time Workshop (RTW). RTW is a software tool developed by Mathworks that allows the compilation of native simulink code. Taking advantage of this, several companies have marketed in-vehicle controllers that no longer need to be programmed directly by a low-level language such as C. Two of these devices are the MicroAutoBox from dSPACE Inc, and the Motohawk development system from Mototron. These devices are designed to be paired directly with Matlab, allowing single-click programming from the Simulink workspace, which in the process described above, is ideally, our simulation environment.

These systems can enable fast and accurate deployment of control code, and have the added bonus of reducing the documentation requirements of the project due to the increased readability of Simulink code compared to C-code. However in making this leap, reliability of the code must be considered; the computing platform has changed, a different, usually less powerful, processor is being used, vastly less memory is available, there is a completely different operating system, the I/O is no long simulated but is instead real I/O and real external serial ports, the controller code is running in real-time, and further, there are no easy methods to pause the code and inspect variables, as there are in the development environment.

However, the existence of a good simulation tool provides the ability to more easily check the presence of bugs in the deployment process by using a method called Hardware-in-the-Loop (HIL). HIL is a way to implement testing of an actual controller before implementing it into a real-world application. This is accomplished by running the simulation on a computer that:

- 1) Is capable of simulating the vehicle and its components in real-time
- 2) Has the ability to communicate via the outside world using analog/digital I/O and a serial data bus (or CANbus for vehicle applications)

The computer is interfaced with the controller via each device's respective I/O, and the vehicle/controller simulation is started. The controller effectively behaves as if it is controlling a real vehicle and its in-situ behavior can be observed within the laboratory without the safety risk of trying untested code on a prototype vehicle. This step is crucial in determining if the controller is computing values accurately and timing-critical sequences are being performed correctly. In addition, common

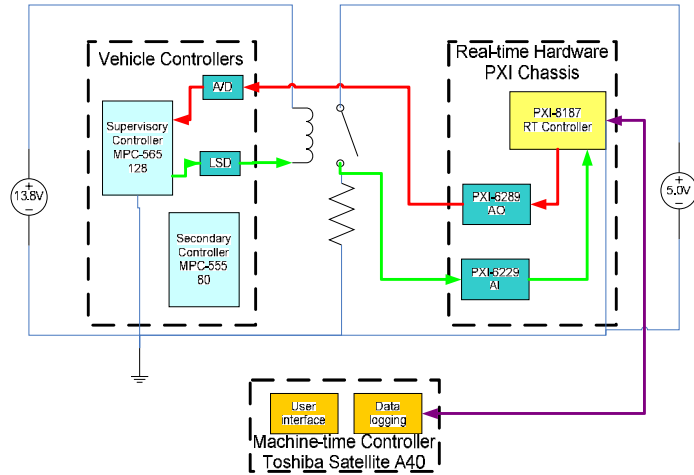


Figure 8: Hardware in the Loop simulation system for the UW Equinox

bugs such as typecast errors, and correct access to board I/O can be detected easily. The HIL system established for the UW Equinox is shown in Figure 8.

More advanced troubleshooting procedures may be invoked later in the development cycle; programming bugs that only show-up in-vehicle, or perhaps intermittently, can be debugged by comprehensively datalogging the CANbus and later injecting the CAN messages of interest onto the HIL CAN network to observe the results. This allows the repeatable observation of controller behavior while processing data that came directly from the vehicle. During control debugging of the UW Equinox control strategy, this technique was applied. The fuel cell power module fault messages were broadcast consistently incorrectly. This data was captured, and an attempt was made to recreate the undesired behavior on the Waterloo HIL test system. Figure 9 is a screenshot showing appropriate controller response to broadcast failure messages. This test resulted in the rapid identification of the fault in the fuel cell subsystem controller.

ActiveFaults	(None)
FaultCommand	-
OccurredFaults	FCPM_Stack_UnderVoltage
SuspectedFaults	(None)

Figure 9: Injected CAN error message and desired response

Improving Component Models using Component-in-the-Loop Simulation

Critical to the usefulness of HEV/PHEV simulation is the quality of the component models. The component models must be able to model those metrics that are most relevant to the HEV/PHEV design process. Two characteristics which play a central role in the performance and cost of hybrid vehicles are the durability of the battery pack, and the voltage characteristics of the battery pack. The durability of the pack, and the rate of degradation will play a key role in any HCS development project, as the way the battery is used (depth of discharge, loading, temperature) will largely determine the expected life of the battery. The voltage characteristics (which is dependant on the battery state-of-charge, temperature, and instantaneous power production) of the pack is critical to

understanding the acceleration performance capability of the vehicle as, in many cases, the power of the electrical system is capped by the lower voltage limit of the motor inverter. Using simulation to inform the design process about these vehicle performance metrics involves good, in-situ battery knowledge.

One method to generate component models, or to validate existing modeling techniques, is to use a technique called “Component-in-the-Loop”. This involves placing a physical HEV component under the control of a real-time HEV/PHEV simulation, and forcing the component through the operating conditions it would face if in fact that simulated vehicle was a real vehicle. This approach has been taken at the University of Waterloo to investigate the interplay between the HCS and battery degradation [4]. If the results of these trials can be extrapolated into a model that can be brought back into a purely simulated environment, it may allow HEV designers to tune the HCS such that degradation of the battery occurs at the same rate as the rest of the vehicle. This is beneficial as the battery will not need to be replaced during the vehicle’s life, and the fuel economy, which to some extent depends on the allowable depth of discharge, may increase.

The benefits of simulation, and where do we go from here

HEV/PHEV design is a complicated task. The engineered components involved are complex and are still being characterized. Vehicle performance is heavily dependant on the drive cycle the vehicle undergoes, which varies from driver to driver. The process of integrating many varied components together brings control and interaction issues that require troubleshooting. A single, multi-functional simulation tool can have a dramatic impact on the development time of initial vehicle component selection and controller development. This tool should be capable of performing lower-fidelity, rapid computation, HEV/PHEV simulations in order to optimize component selection and perform preliminary evaluation of HCS theories. It should have the ability to compute high-fidelity, transient component models that include controllers/control inputs. It should have a standardized component description, and mechanisms allowing, and encouraging, component manufacturers to add their components to the body of knowledge, and it should support integration into advanced testing procedures such as HIL and CIL. It should also attempt to include the two most constraining HEV/PHEV technical issues, battery voltage modeling, and battery degradation modeling. The development of a tool that can meet all the above criteria, would be a boon to vehicle integrators, and would act to advance the industry as a whole.

References:

-
- 1 Cavallo, Alfred J. Hubbert’s Petroleum Production Model: An Evaluation and Implications for World Oil Production Forecasts, Natural Resources Research, Vol 13. No 4, December 2004
 - 2 Pesaran, Ahmad and Markel, Tony, Battery Requirements and Cost-Benefit Analysis for Plug-in Hybrid Vehicles. Presentation: The 24th International Battery Seminar and Exhibit, NREL/PR-540-42082
 - 3 M.B. Stevens, et al 2006, Fuel Cell Hybrid Control Strategy Development. SAE 2006-01-0214
 - 4 M.B. Stevens, et al 2007, Modeling battery degradation in plug-in hybrid vehicles. PHEV2007, Manitoba