ABSTRACT

This presentation focuses on several examples of partnerships between tool suppliers and embedded software developers in which state-of-the-art tools are used to optimize a variety of electric and hybrid vehicle architectures. Projects with Automotive OEMs, Tier One Suppliers as well as with academic institutions will be described. Due to the growing complexity in multiple electronic control units (“ECUs”) inter-communicating over numerous network bus systems, combined with the challenge of controlling and maintaining charges for electric motors, vehicle development would be impossible without use of increasingly sophisticated tools.

Hybrid drive trains are much more complex than conventional ones, they have at least one degree of freedom more. To achieve the development goals (e.g. optimum drive performance and minimum CO2 emissions), an optimal configuration of combustion engine, battery and electric machine(s) can only be defined with the help of computer simulation methods and tools.

Even if the major specs (speed/torque characteristics, performance, high voltage system) of all components are defined and fixed, the operating strategy software (overall hybrid drive train coordination) still significantly influences these goals. The hybrid ECU with the vehicle's operating strategy is often developed, implemented and tested by means of desktop simulation, rapid control prototyping, automatic code generation as well as hard-ware-in-the-loop (HIL) simulation.

The e-mobility technology is developing very rapidly. So cooperative partnerships between OEMs, component and controller suppliers and test equipment and tools companies can be observed world-wide. For example: Lithium-ion battery suppliers are working in cooperation with controller developers and tools companies on battery simulation and controller testing. Additionally, several trends can be observed in the HEV development area:

- HIL simulation replaces track tests to handle the complexity and reduce costs
- Optimization tools are applied on mechanical test benches to automatically find a good compromise between driveability, performance and emissions.

This paper presents application examples where these partnerships and new trends lead to successful products in the automotive industry.

One special summary example of the use of embedded software development tools is the EcoCar Challenge, an industry-supported project organized by the Department of Energy in conjunction with General Motors, and other sponsors, to develop expertise in the automotive development process, especially relative to producing “Greener” vehicles. In this project, competing Universities are supplied with a base vehicle and software development tools, such as rapid prototyping systems and HIL test systems. Numerous powertrain architectures are developed and optimized during the course of the competition.

1). BACKGROUND

The flurry of development activity in the hybrid vehicle powertrain area over the recent several years is continuing to draw lot of excitement and press. The electrification of today's vehicles is approaching a level not foretold by many
pundits even a few years ago. Today, essentially every vehicle OEM has announced new product developments with hybrid, or even fully electric powertrains.

In addition to the media and consumer excitement and concerns, this activity has placed an emphasis on new developments both in electric motors, battery systems, alternative energy storage devices (e.g. ultracapacitors, hydraulics and flywheels), power inverters, and especially in control systems. Not only does the newly electrified powertrain bring requirements for its own control, but the newly developed energy storage systems also energize many auxiliary components, such as steering, brakes, temperature controls, etc. In some cases, especially as the re-charging infrastructure becomes more developed, the ability of the vehicle control system to interact with the grid will place even more demands on embedded controls.

In contrast to developing new versions of internal combustion powertrains, the new hybrid movement is creating opportunities for control systems to be developed from scratch. In the case of IC engines, most control software has been developed over the past 30 years in incremental steps as new emission controls were added, or new features were needed. In those cases embedded software was commonly a challenge of integrating legacy code with newly developed features. Developing control algorithms from scratch has pros and cons. The pro is that Model Based Development (MBD) techniques can be used without the challenge of integrating with legacy code developed before MBD was used. The main con is that there is often no proven baseline code which can be relied on as robust and reliable.

Another area of complexity is the variety of hybrid architectures. In the IC engine era, powertrain architectural decisions were basically between FWD, RWD, or AWD for wheel power, then Diesel or Gasoline for fuel source, then, a variety of cylinder configurations and fuel distribution technologies. The hybrid architecture decisions do not eliminate any of these previous degrees of freedom, but instead add a whole host of additional significant variations: battery technology alternatives (NiMH, LiIon, PbAcid), mild hybrid, two mode hybrid, parallel hybrid, PHEV, range-extender HEV, full electric vehicle, and in the future possibly fuel cell power.

In addition to the complexity added by new hybrid powertrain components, OEMs are finding a whole host of new suppliers to collaborate with, for example battery suppliers such as A123, SAFT, Compact Power, Energy Conversion Devices, EnerDel, etc. Other players include such companies as Azure Dynamics, TESLA, Fiskar, TH!nk, etc.

Considering all of the above, a primary issue to the success of new hybrid vehicles is the importance of system design and performance. Complexity and economy concerns have continuously driven a stronger emphasis on the vehicle as a system even before the hybrid movement. Emphasis has changed from optimizing the sub-systems of the vehicle assembly i.e. primarily the powertrain, the chassis, and the body, to optimizing the architecture and the overall system. The successful modern hybrid vehicle is analogous to a successful sports team. Individual stars cannot guarantee continuous success. Only an emphasis on a balanced set of players and optimized teamwork produces lasting success. From an embedded controls perspective overall hybrid vehicle analyzed as a system (including energy management) is key to success.

2). CHALLENGES IN HEV ARCHITECTURE AND CONTROL DEVELOPMENT

There are several goals and challenges that influence the development of modern vehicles: Good performance and driveability, maximum safety and good driving comfort often conflict with minimized fuel consumption, emissions and carbon footprint. As mentioned in the previous chapter, the development of HEVs is a system approach where optimized components and corresponding component control systems (for example combustion engine, battery, electric motor(s), transmission and braking systems) have to be designed and optimized in combination with the overall (hybrid) vehicle management system.

Vehicle Management System

This hybrid vehicle management system contains the overall operating strategy and guides the different subsystems (Figure 2.1). It decides, for example, when the combustion engine has to be started and stopped and in which operating point (load vs. speed) the engine should do its job. Additionally it has to make sure that the battery state-of-charge (SOC) is always kept in the specified operating range so that on one hand, the electric-only mileage is still significant, and on the other hand, that there is always some potential for recharging it, for example in case of regenerative braking. These intelligent systems are usually adaptive and adjust their operating strategy according to the traveled route (history) or the track planned by a navigation system (future).

Development Processes

The goal of this paper is the presentation of the tool chain and the process steps in the development of the EV or HEV controller networks. The basis for this development is the well-established V-cycle that is one of the standard approaches in automotive software development. Figure 2.2 shows the major phases of the V-cycle in general. Chapter 3 will then introduce the different process steps in detail based on case studies from the area of EV and HEV development.
3. CASE STUDIES FOR HEV/EV DEVELOPMENT PROJECTS

In this chapter, different case studies are presented that illustrate the different development steps used in design and test of HEV electronic control units. These case studies can be seen as examples that explain the process steps in general and the issues and challenges in the development of HEV electronics in particular. At least one case study is described for each of the process steps shown in figure 2.2.

3.1. CONTROL DESIGN AND SYSTEM DEFINITION

The component design and the system definition is the starting point for dimensioning of the drivetrain. There are several tools available on the market, which support the development of the hybrid vehicle system configuration and simulation, based on libraries of the individual components. Some of the tools use the effect-to-cause method [1], which is a kind of feed-forward simulation: Based on a driving cycle as an excitation, the operating points of the individual components are calculated and from there the overall fuel consumption and battery depletion/charge can be calculated. This method is not very useful for controller design and optimization, but can be used for the fast comparison of different hybrid drive train structures and configurations.

The more common approach is the cause-effect method, which is the basis for nearly all hybrid vehicle simulation tools. Here a driver-model follows a driving cycle (task: speed control) or a simulated track (task: to choose the speed so that the car stays on the track). Many of these tools are based on Matlab®/Simulink® or at least have an interface to Simulink or a Simulink model export capability (e. g. IAV VeLoDyn [2], PSAT (Power-train System Analysis Toolkit) by Argonne Nat. Lab. [3], dSPACE ASM [4]). Other environments, such as Dymola, SimulationX, LMS AmeSim, CarSim etc. often use their own simulation environment and corresponding solvers or support code export to Simulink.

Independent from the simulation environment, a precise simulation requires accurate plant models and accurate ECU models (soft ECUs) i.e. controller models.

Plant Models

The variety of models for these kinds of applications is very wide, so a detailed analysis of the market would definitely exceed the focus of this paper. An overview on common modeling approaches for the non-real time and real-time simulation of engines and hybrid vehicle components can be found in the literature ([2], [5], [6]). Many of the models are based on Matlab/Simulink. A general structure of a plant model is shown in figure 3.1.1.

Controller Models (ECU Models or Soft-ECUs)

The controller models can be integrated as pure Simulink models in the offline/desktop simulation or as the real functional ECU code (often in C), wrapped as an S-function in Simulink or in other simulation environments (software-in-the-loop). In the early phase of the development, it is often not clear which software component will later be integrated in which physical ECU. Here, software architecture design tools such as DaVinci by Vector, SystemDesk by dSPACE [4], AutoSar Builder by Geensoft (and to some extent Intecri o by ETAS) are applied to define the software architecture or
the distributed control system independently from any physical device (ECU) and its implementation.

![Plant model components and their interaction for the simulation of HEV drivetrains](image1)

3.2). RAPID CONTROL PROTOTYPING: ON A TESTBENCH OR ON THE TRACK

After the architecture of the system is decided, the basic control strategy is developed, and control algorithms have been drafted in a graphical language such as Simulink, the next development challenge is to implement the draft algorithms in executable code into a prototype ECU. Many issues arise when real-time control demands from the ECU/microprocessor conflict with the intent of the draft algorithm if the system even runs at all. Several makers of Rapid Control Prototyping systems now offer hardware and software packages that take Simulink/StateFlow control models and translate them into code that can be tested with a prototype ECU containing a real microprocessor and I/O hardware.

![Hybridized ML350 by Magna Steyr](image2)

An example of this phase of the development cycle is the Hybrid Drive System developed by Magna Steyr in cooperation with its suppliers and partners to build a Mercedes M-class HySUV demo vehicle.

Magna Steyr, working with Magna Powertrain, and Siemens VDO (now Continental) replaced the automatic transmission and transfer case in the Mercedes ML350 (see figure 3.2.2) and installed two electric motors - one which exclusively powers the front wheels and one which works in concert with the IC engine to power the rear wheels. Four hydraulically actuated multi-plate clutches control the torque flow to each wheel. A 70 kW / 360V lithium ion battery system provides electrical energy storage and several engine accessories were converted to electrically operated components.

All hybrid drivetrain components are controlled with just one controller prototyping system (refer to Figure 3.2.3 below) using an auxiliary extendable driver hardware system (RapidPro by dSPACE) to provide actuator drivers, sensor I/ O and hardware diagnostic interfaces.

The control software was developed to contain the functions and interfaces of the entire driveline torque path including:

- Determine torque requested by driver based on accelerator pedal, brake pedal and gear level positions
- Control the distribution of torque to the axles based on driver's demand, optimum traction, and availability of the IC engine and 2 electric motors
- Consider efficiency, dynamics, battery charge status, comfort and thermal conditions to distribute torque and select the appropriate gear
- Manage torque sources and control transient driveline processes
- Control components in the automated manual transmission and the all wheel drive module (E4WD)

This was a complex control development project which involved a careful step-by-step process to ensure success. Following are some statistics from the software developed using the rapid control prototyping system:

- 257,000 lines of generated C code
- 44,000 model blocks
- 4.5 MB memory control application

One indication of success of this new control system is the dyno-generated charts of typical engine operating conditions versus efficiency. Refer to figure 3.2.5 (graphs of torque vs. engine speed) A comparison of the engine operating conditions between IC engine only and the hybridized powertrain indicates significant shift of operating points to
more efficient areas. Many of the low efficiency operating points of the IC engine have been replaced or shifted to either purely electric operation or electrical assist operation.

Rapid Control Prototyping has become a standard method for developing and optimizing algorithms in HEV control applications. Several other applications are included in the references ([7], [8]).

3.3). TARGET CODE GENERATION

As ECU code becomes more complex and requires more memory and processor resource, one of the significant features of Model-Based Development (MDB) that brings great efficiencies is the use of Auto Code Generation (ACG) technology. Control Algorithms developed in model-based graphical languages such as Simulink can be converted to production code for implementation in an ECU very effectively. Even more importantly, as revisions to the algorithms due to design changes are developed, code can be regenerated in minutes versus weeks as is often the case where ACG is not used.

Modern ACG systems not only generate code faster but offer features to streamline code review and enhance overall code quality. Features such as the following are becoming more important in embedded software development projects:

---

**Figure 3.2.2.** Diagram of key components installed in the ML350 to hybridize the vehicle.

**Figure 3.2.3.** Diagram illustrating the control system architecture and its connections to the vehicle data network.
A large number of modular software components ensure that the hybrid drive implements the driver's wishes in an optimum manner.

Some critical features for this battery management system (BMS) are listed below; The BMS

- is contained in a separate dedicated ECU installed in the battery assembly,
- is networked to all hybrid-control ECUs including the climate control ECU (for battery temperature management),
- implements numerous safety checks to ensure high voltage contacts are live only when in operation,
- provides charge level and power flow data to the instrument cluster graphics display
- includes computation of current, voltage, and power limits, and monitors the battery aging,
- controls current flow and voltage limits to maintain battery temperatures between 30 to 50 deg C, and
- performs cell balancing to handle charge differences between the 35 high voltage cells to extend battery life. Each cell is recharged individually on the basis of a load charge analysis of that particular cell.

Automatic Test Case Generation and Code Validation
Model Validation by Formal Verification
Automatic Style Checking for Models

One recent example of the successful application of ACG is a joint venture between Johnson Controls and SAFT Advanced Power Solutions (JC-S). JC-S recently developed an efficient high voltage lithium-ion battery management system for the Mercedes-Benz S 400 HYBRID - with a mild hybrid drive system.

Battery assembly, mild hybrid starting motor and their installed locations.
Because this was the first control developed for lithium-ion battery applications in a mild hybrid, the controller software had to be developed from scratch, without adapting legacy code. Modeling guidelines were adapted from the recommendations of the ACG tool supplier, for the best possible implementation and most efficient production code. Figure 3.3.3, shows the flow of development phases in this project.

Some of the major successes realized in this project include:

- New mild-hybrid drive battery management controller model successfully developed from scratch
- Run-time behavior and the resource consumption tested by processor-in-the-loop (PIL) simulation. Real-time requirements fulfilled in all use cases
- Function algorithm designed with Simulink/TargetLink to easily work with other ECUs containing similarly developed algorithms
- Production code for fix point processor TriCore TC1796 generated with TargetLink
- approx. 25,000 lines of code generated
- Code validation per MIL, SIL, PIL simulation

3.4). HARDWARE-IN-THE-LOOP SIMULATION

When the ECU has been programmed, its functions can be tested manually or automatically by means of hard-ware-in-the-loop simulation (HIL). Unlike test drives in a real vehicle, these tests are performed in the lab, so that errors that occur can be reproduced at any time. The HIL test system replaces the real environment of the ECU and the tests can be executed in any conceivable test scenario, systematically and reproducibly with test automation. As a result, complicated and expensive test runs in the real environment (test track or test bench, proving grounds, public roads, summer and winter trials, country, town and mountain drives) can be reduced to a minimum. The key components of the HIL simulation are the models of the plant in combination with fast and precise I/O and communication interfaces. Figure 3.4.1 shows the typical signal flow in a real system compared to HIL simulation.
Like the aforementioned process steps, HIL simulation has been established as a standard E/E development step in the automotive industry. HIL simulator applications vary from single ECU testers over domain-specific simulators (powertrain, body, interior, infotainment) to full virtual vehicles with all the ECUs of a vehicle (platform).

**HIL in Hybrid Vehicle Applications**

EV and HEV applications are an important use case of HIL simulation. The simulation of electric machines and batteries are very challenging and call for some specialized hardware and software solutions. The basic structure of an electrical machine in a vehicle is shown in figure 3.4.2. The controller itself, with its current controller, sends pulse-width-modulated control signals to the power stages, which then switch the real voltages and currents for the electric motor. This generates torques on a vehicle component via mechanical shafts. For control, motor currents and the motor position are usually returned.

A basic difference between electric machines and other vehicle components is that the machine is controlled at a very high clock rate (typ. 10-20 KHz). This places significant new demands on the HIL simulation. I/O sampling rates in the millisecond range and mean value models, which are used for combustion engines, are not sufficient for fast current and position control. Special FPGA-based hardware is commonly used for this purpose, for measuring the PWM control signal or for simulating incremental encoders or resolvers. Moreover, small model parts can be simulated directly on the FPGA to achieve simulation step sizes in the sub-microsecond range.

**Figure 3.4.1. Signal flows in a real system (top) and in HIL simulation (bottom left and right) [9].**

**Figure 3.4.2. Basic structure of electric drives in vehicles and the three possible interfaces to an HIL system.**

Figure 3.4.2 shows the possible interfaces between an HIL system and the electric machine's ECU. Depending on the application and the modeling depth, the interfaces can be defined and implemented either on the mechanical level (e.g. with mechanical load motors), on the electrical power level (by means of electronic load simulation), or on the signal level (without relevant power). HIL solutions for all these cases are available on the market and have been published [6].

**HIL System for a Battery Management System (BMS) for LiIon Batteries (BMW)**

This case study describes the special challenge in simulating the real-time behaviour of a LiIon battery necessary to test the corresponding BMS. The main tasks of a BMS are temperature management, charge control, several safety functions under high voltage conditions, isolation monitoring and - in case of any failures - onboard diagnostic functions. Additionally it has to supervise the cell voltages to calculate the state-of-charge (SOC) of the battery and to perform cell
balancing by (dis)charging certain cells. The structure of a BMS is shown in Figure 3.4.3.

In this particular application, the HIL system behaves like a real battery and allows testing all the aforementioned features. It contains an electrical failure simulation unit to check the diagnostic and safety features of the BMS. Additionally, variable resistors can be switched between the simulated battery poles and the vehicles GND or Vbatt lines to test the isolation monitoring etc. Finally, the cell voltage simulation has to support cell balancing. This means, a cell module emulator, that acts like a current source to the BMS, simulates the cell voltage which supports the discharge of individual cells. Figure 3.4.4 shows the structure of the HIL system that BMW established successfully in their E/E development processes. More details of that application can be found in the literature [10].

HIL system for the drive system of the Mitsubishi i-MiEV

One successful example for the usage of HIL in the EV domain is the test of the electronics of Mitsubishi's new electric vehicle, i-MiEV (Figure 3.4.5 and [11]). The i-

**Figure 3.4.3.** The BMS performs battery management, in conjunction with the cell ECUs (CEs), which directly monitor the battery modules.

**Figure 3.4.4.** The BMS, some cell module emulators, and real parts are integrated into the HIL simulator setup.
MiEV’s electronic platform is characterized by a distributed control system that uses five dedicated ECUs. Four ECUs (one for each function) are for a special battery unit, an individual cell monitoring unit, an electric motor control unit, and an onboard recharging unit. These four ECUs are connected to the fifth ECU, the overall electric vehicle ECU (EV ECU) to offer the various controls an electric vehicle needs. Mitsubishi ran rigorous tests to ensure the quality of the software, yet still managed to cut the time-to-market.

In executing the testing, it was necessary to adapt the test design content for the test patterns quickly and accurately. For the sake of analysis, a high level of replication was needed when software bugs were detected. Therefore, a dSPACE Simulator Mid-Size was used to simulate user operation and the inputs to the ECUs accompanying it during the design process. By supervising the inter-ECU communication (via CAN-Bus), a large number of input test patterns were applied to systematically verify the drive system of the i-MiEV. An actual vehicle was used for the real load on the HIL simulator and the test automation software AutomationDesk was applied as a means of creating test patterns quickly. The HIL simulation proved to be extremely effective in ensuring the quality of the software delivered on schedule. According to Mitsubishi, the verification of the software quality within the time available would have been very problematic without HIL simulation.

Another interesting application is an HIL system to test the entire ECU network of the 2010 VW Touareg Hybrid, a parallel hybrid vehicle. Volkswagen applied the HIL technology successfully for full network testing of all the ECUs of their high-end SUV. For details, please refer to the references [12].

3.5). ECU CALIBRATION, PARAMETER FINE TUNING, AND OPTIMIZATION

The calibration of software parameters is a major milestone in the development of ECUs. Whether in test drives, on a test bench, or in earlier phases of the development, ECU calibration, measurement and diagnostics are indispensable. It is often the last step in the development before start of production of the vehicle and thus, before the vehicle arrives at its customers.

As a result, ECU calibration is still one of the most important and also most expensive steps of the development of modern vehicles. Hundreds of engineers and tons of vehicles and equipment are shipped around the world to find the right environment and the optimum test conditions for calibration work. Well established calibration tools are available from, for example ETAS (Inca), Vector (Canape), ATI (Vision) and dSPACE (CalDesk).

Even if the calibration work can not be avoided completely, the trend to do more development virtually and to solve problems earlier in the process (frontloading) can also be applied to calibration. In the engine development area, calibration and optimization methods have been established, that allow a fully automated pre-calibration of an engine ECU on a dynamic engine test bench by means of online DOE (design of experiment) methods. Concepts and tools for this have been presented among others, by AVL, IAV (“rapid calibration”) and Ricardo.

Adapting this approach to full HEV drivetrains, combining it with real-time simulation of the vehicle dynamics and with an (online) optimization and DOE system (for example AVL Cameo), leads to an automated optimization procedure that can be performed on dynamic test benches (figure 3.5.1).
tools is the EcoCar project. *EcoCar: The NeXt Challenge* is a vehicle technology engineering competition for students in advanced collegiate programs established by the United States Department of Energy (DOE) and General Motors (GM), and managed by Argonne National Laboratory (www.ecocarchallenge.org). It is a three-year project which began in late 2008 and is now entering year 3.

Over 60 universities applied for this competition, each with an extensive proposal detailing among other things how the university would support the competing team and how the lessons learned would be applied in future curricula. Currently, 16 universities across North America (including 3 Canadian) having been selected on the basis of outstanding proposals, are in the competition. A major goal of the project is to reduce the environmental impact of an automobile by minimizing the vehicle's fuel consumption and reducing its emissions without sacrificing its performance, safety and consumer appeal. Students learn a real-world engineering process (figure 4.1) to design and integrate their advanced technology solutions into a GM-donated compact cross-over SUV. All Universities started with the same vehicle, but have progressed to develop different powertrain architectures based on virtual simulations using a variety of modeling systems.

Students are designing and building advanced propulsion solutions that are based on vehicle categories from the California Air Resources Board (CARB) zero emissions vehicle (ZEV) regulations. They explore a variety of cutting-edge clean vehicle solutions, including full-function electric, range-extended electric, hybrid, plug-in hybrid and fuel cell technologies. Students also had the freedom to choose alternative fuels such as ethanol, biodiesel and hydrogen in their new architectures. Figure 4.2 is a matrix describing some of the major features of each team's project.

GM has provided all teams with instruction and mentoring in a version of GM's global vehicle development process (GVDP, figure 4.1). There have been previous student engineering competitions which focused on vehicle hardware modifications. EcoCAR, is the first competitive collegiate project which featured a strong emphasis on system and controls modeling and simulation, as well as subsystem development and testing.

This project is a collaborative effort of major proportions. General Motors is providing vehicles, vehicle components, seed money, technical mentoring and operational support. The U.S. Department of Energy and its research and development arm, Argonne National Laboratory, is providing competition management, team evaluation and technical and logistical support. Additionally over 25 corporate and governmental sponsors provide components, money, developmental tools and mentoring to the competing universities.

---

**Figure 3.5.1. Test bench with parallel hybrid powertrain (engine, electric drive, battery, transmission, and corresponding controllers) in combination with a real-time simulation (HIL) environment. This environment allows dynamic optimization procedures and online calibration of a hybrid drivetrain's ECUs.**

This test bench system is a very powerful tool that allows automated or semi-automated studies and optimization procedures, incorporating closed-loop dynamic driving maneuvers, for example, to:

- develop and optimize smooth engine start procedures and engine kick-in,
- use objective driveability evaluation to qualify and quantify the dynamic responses of the drivetrain
- evaluate the influence of start-stop, engine kick-in, regenerative braking etc. on the vehicle dynamics, with the real ECUs,
- evaluate the influence of vehicle dynamics control interventions (ABS, ESP, steering) on the behaviour of the hybrid vehicle drivetrain,
- optimize drive comfort in combination with minimized fuel consumption and emissions (for example during gear changes) to minimize jerk, accelerations, …

**4). UNIVERSITY STUDENTS DEVELOPING ALTERNATE HYBRID VEHICLE ARCHITECTURES**

An interesting example of the automotive industry emphasizing the importance of modern software development
Figure 4.3 contains a diagram illustrating the overall project broken down by the main areas of development focus: Mechanical, Electrical and Controls domains. In the first year of EcoCAR, participating teams used math-based design tools such as Argonne's Powertrain Systems Analysis toolkit (PSAT) or similar vehicle models research to compare potential architectures and select an advanced vehicle powertrain that meets the goals of the competition. Teams used CAD software to ensure that their chosen components fit into their vehicle and that the electrical, mechanical and software systems function properly. All teams are employing software-in-the-loop (SIL) and hardware-in-the-loop (HIL) technologies to develop controls and subsystems.

In the last two years of the competition, students actually modify the supplied production vehicle and develop a working vehicle using their first year design to meet the competition's goals. The teams come together at the end of each academic year to compete against each other in more than a dozen static and dynamic events designed to measure the success of their project.

During each weeklong competition, student teams demonstrate the modified vehicles to industry expert volunteer judges to prove that their designs meet or exceed the following goals relative to the original production vehicle:
• Incorporate technologies that reduce petroleum energy consumption on the basis of a well-to-wheel (WTW) analysis of the total fuel cycle,
• Increase vehicle energy efficiency,
• Reduce WTW greenhouse gas (GHG) and regulated emissions, and
• Maintain consumer acceptability in the areas of performance, utility and safety.

This program has been highly successful in stimulating a better education in the competing universities in both electrical, software and automotive development processes, in addition to giving students first hand experience with the embedded software development tools that are the state of the art in this industry.

5). SUMMARY
Following the process described by the V-cycle metaphor, this paper described a series of development challenges that have been met using state of the art tools. The increasing complexity of networked controls in the trend to develop hybrid and fully electric vehicles has increased the focus of OEM and Tier One developers on technologies to ensure optimum quality of ECU software, and to minimize time to market. Specifically:

1). Plant Models and Controller Models are commonly designed using a combination of high level programming languages and graphical environments.
2). Draft control algorithms are developed and optimized using a process referred to as rapid control prototyping which utilizes a specialized development ECU containing many features to allow flexible, efficient, and quick prototype evaluations.
3). ECU Code can now be relatively easily converted from prototype algorithm to production ready C code using a process identified as Target Code Generation, or sometimes “Automatic Code Generation”.

4). ECU Code, no matter how developed, is being thoroughly tested in a number of configurations ranging from component control level to full vehicle network level using Hardware-in-the-Loop technology. Such technology offers numerous benefits including shorter time to market, and significantly improved quality.

5). The last and often most expensive step in embedded control development, is ECU Calibration, which incorporates the fine tuning of system and control parameters in the vehicle or on a testbench.

From the examples cited in this paper it is clear that the challenges posed by complex hybrid powertrain control adaptations in automobiles are being introduced more effectively and efficiently using the latest embedded software development tools. It is not much of a stretch to say that without these tools, hybrid vehicles would not be feasibly developed in an acceptable time frame with the necessary quality. Certainly as technology for electrification of vehicles advance, software development tools will continue to advance and play a critical part, ensuring quicker times to market and higher quality.

REFERENCES


10. “Virtual energy cells: dSPACE HIL simulators at the BMW group - testing lithium ion battery management systems” dSPACE Magazine 1/2010, page 7ff.


CONTACT INFORMATION

Kevin Kott
President
dSPACE Inc.
50131 Pontiac Trail, Wixom, MI 48393, USA
kkott@dspaceinc.com

Dr. Peter Waeltermann
Group Manager Engineering Applications/Engineering Dept.
dSPACE GmbH
Rathenaustraße 26, D-33102 Paderborn, Germany
pwaeltermann@dspace.de

DEFINITIONS/ABBREVIATIONS

ACG
Automatic Code Generation

AMT
Automated Manual Transmission

AWD, FWD, RWD
All-/Front/Rear Wheel Drive

BMS
Battery Management System
CAN
Controller Area Network

CARB
California Air Resources Board

DOE
Department of Energy, Design of Experiment

ECU
Electronic Control Unit

ESP
Electronic Stability Program

FPGA
Field-Programmable Gate Array

(H)EV
(Hybrid) Electric Vehicle

HIL, MIL, SIL, PIL
Hardware/Model/Software/Processor-in-the-Loop

HMI
Human Machine Interface

LiIon
Lithium Ion (Battery)

MBD
Model-Based Design

NiMH
Nickel Metal Hydride (Battery)

OEM
Original Equipment Manufacturer

PHEV
Plug-In Hybrid Electric Vehicle

RCP
Rapid Control Prototyping

SOC
State-of-Charge

ZEV
Zero-Emission-Vehicle