

by Randy Frank

The Newest Car Models— SOFTWARE

To cope with the growing complexity of vehicle electronics, car manufacturers and suppliers have been transitioning from a test it at the end to simulate it at the beginning mindset. Complete simulation requires a well-defined process and development environment, an extensive number of tools and cooperation at the tier-one module suppliers down to component suppliers. Thus, car models today include plant models, signal delivery models, and algorithm models. In short, simulation and testing will drive future vehicle electronics.

	1979	2005
Microprocessor	Custom Motorola 6800 (8 bit)	Motorola PowerPC (32 bit)
Microprocessor Clock Speed	1 MHz	56 MHz (56x)
Memory	4 kBytes	2000 kBytes (500x)
Programming Language	Assembly code	C code
Calibration	0.256 kBytes	256 kBytes (1000x)
Sensors and Actuators	5 each	20 each (4X)
ECM Pins	Less than 50	More than 200 (4X)

Saying that today's vehicles are extremely complicated electronic assemblies would be a gross understatement. To cope with this growing electronics complexity, carmakers and tier-one suppliers have been going through a transition. From a "test it at the end" to a "simulate it at the beginning" mindset, simulation will be playing a crucial role in the design and development of auto electronics. Complete simulation will require a well-defined process and development environment, an extensive number of tools, as well as cooperation with tier-one module suppliers down to component manufacturers. Consequently, car "models" today include plant models, signal delivery models, and algorithm models.

POWERTRAIN COMPLEXITY

As the oldest control system in the vehicle, the powertrain control provides an example of how today's most complex control system has

Figure 1. (a) Changes in MCU hardware and software from 1979 to 2005 have meant a 500x increase in memory and a 1000x increase in calibrations. (b) Increase in major transmission control functions. Courtesy of General Motors.

grown in complexity. As shown in Figure 1a, in 1979, General Motors used an eight-bit microcontroller (MCU) operating at 1 MHz clock speed with 4 Kbytes of memory, and 256 bytes of calibrations. In 2005, a 32-bit MCU operating at 56 MHz with 2000 Kbytes of memory (a 500x increase) has 256 Kbytes of calibrations (a 1000x increase) in the powertrain control module (PCM).

During the past 25 years, GM's transmission control (Figure 1b) has increased from a single function to 60 controlled items. Today's engine controller functionality includes fuel and ignition calculations that are updated every cylinder; 10,000 CAN serial data signals transmitted or received per second; and more than 300 continuously monitored diagnostics. In addition, the controller provides driver display information that includes speedometer, tachometer, fuel level, odometer, PRNDL, coolant temperature gauge, and oil life. Table 1 shows a summary of the key developments that GM engineers have made in powertrain controls and the future challenges that lie ahead. To implement today's and future functions, GM is making a transition that they call Road to Lab to Math (RLM). The goal is a math-based design effort at the front end of the design process. The transition requires a well-defined methodology and a complex testing environment.

DESIGN PROCESS AND TESTING ENVIRONMENT

The engineering process that GM uses (Figure 2) is at a high enough level that it does not change through the transition for RLM. Increased implementation puts

emphasis upfront on the math process.

Starting either with the engine or transmission controller, software engineers must have visibility into the workings of the microprocessor for model-based design, noted Dennis Bogden, director of Powertrain Electronics Engineering, General Motors Corp. This requires observing how the code is executing to verify that what was expected to occur is actually being executed by the MCU. In addition, calibration engineers need visibility into the system: (1) to verify that the correct information is stored in memory addresses; (2) to observe that the proper loads, such as injectors, are turned on or off based on specific event or data input; and (3) to write new values to set the calibration.

To access this information, a development tool and special interface are required to communicate with the MCU in the powertrain module. GM uses ETAS Group's tool and ETAS' ETK interface to connect to one of the members of Freescale Semiconductor's Oak family of processors in its PCMs. ETAS also has interfaces for other companies' MCUs. For development, controllers with the ETK interface connect the instrumentation, the logic analyzer, and the Ethernet network. The interface is an add-on used for development and configured for GM by their control module

Key Developments

- Brake torque management
- Vehicle theft control
- Airbag crash recorder
- Generator control
- OnStar
- Oil Life System
- Shift mode select

Key Developments

- Remote start
- Adaptive cruise control
- Stability control
- Alternator-starter system
- Virtual sensors
- Parallel hybrid
- Electronic range select
- Displacement on Demand

Future

- Fuel cell control systems
- Advanced controlled collision avoidance
- Advanced remote diagnostics
- Remote electronic system
- Increased vehicle dynamics interaction

Table 1. Key developments and future additions to powertrain control for General Motors.

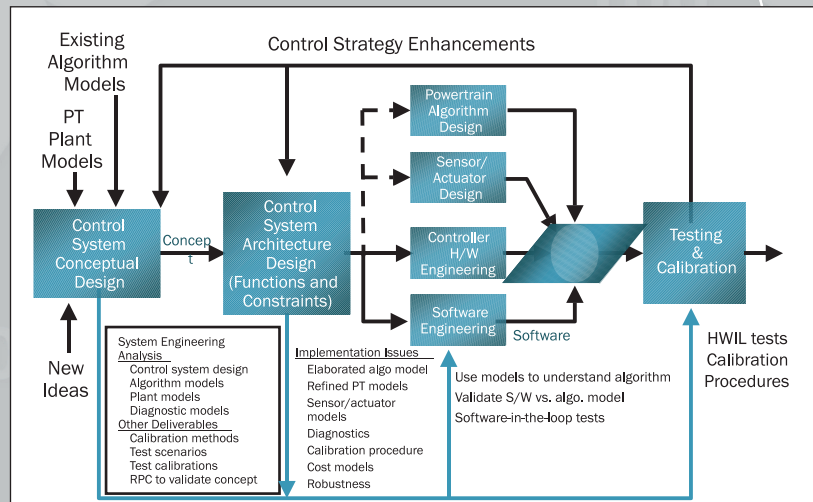


Figure 2. Model-based powertrain control design process. Courtesy of General Motors.

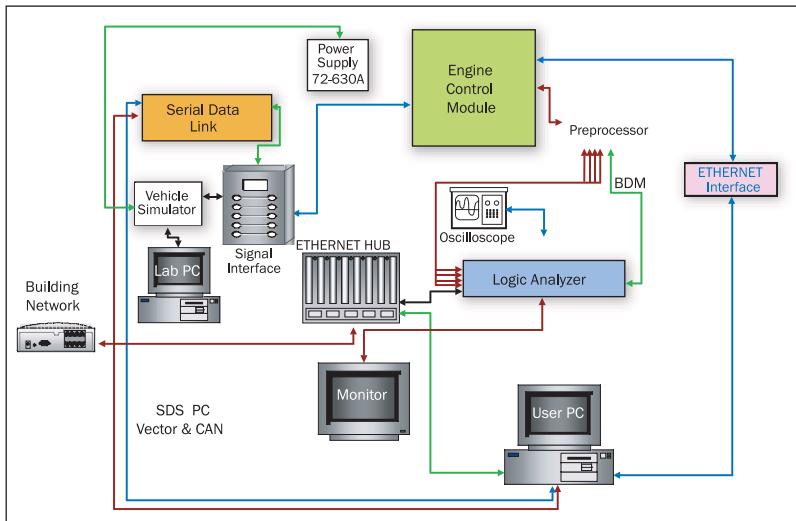


Figure 3. General Motor's testing environment. Courtesy of General Motors.

suppliers. This provides standardized communications even though GM has more than one module supplier. ETAS' INCA is used as the instrumentation software for calibration. Other bench development hardware comes from the ETAS tool chain for consistency. These tools provide the foundation for GM's testing environment as shown in Figure 3.

PEEKING INTO THE SYSTEM

In the test environment, the engineers who create the algorithms and the software need to look at the controller and observe its operation from a byte-by-byte execution viewpoint. They must verify: (1) that code works properly, (2) that controller input is read correctly, and (3) that the output changes occur as required. Bench analysis using the logic analyzer allows them to observe the execution of the C code in real time. GM has a group of software testing engineers that start testing at the unit level, the most elemental functions, to verify that the combination of inputs is causing the output to occur as desired. The oscilloscope in Figure 3 provides the hardware signals to observe the electrical pulses in the system. The Ethernet hub allows engineers to get the data from a server network and run the tests from their desk or at the bench location.

To program the specific engine parameters, calibration engineers assume that the software is working properly. The ETK connects to the INCA tool so calibrators can see any calibration parameter that they need to change. Software extracts the data in computer language from the controller and converts it to engineering units, so the calibration engineer does not have to work with binary numbers. The interface also takes the input in engineering units and converts it to binary numbers for use within the controller. INCA is the software interface that handles the conversion of information from computer language to engineer units.

Data acquisition is another part of the equipment so calibration engineers can log and store data. This allows the monitoring and storing of a parameter or set of parameters during a driving test for subsequent analysis on a PC, like an electronic data recorder.



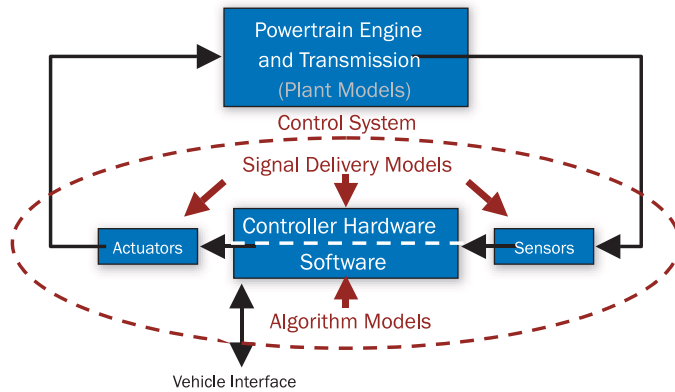


Figure 4. Controls modeling, simulation and analysis. Courtesy of General Motors.

HARDWARE IN THE LOOP

This existing process, shown in Figure 3, is a global process with the same tool chain for a global family of controllers that GM is rolling out for Europe and Asia. The upper left corner shows the signal interface. The vehicle simulator is the hardware in the loop (HIL) portion. The simulation makes a bench controller think it is connected to an actual transmission. Plant modeling engineers create the algorithms to simulate engines and transmissions and their various subsystems such as ignition, and fuel delivery so the HIL appears as real units for bench analysis. Bogden said this is the first crude step toward math-based modeling. With the original HIL, design engineers could never calibrate a real engine or transmission from the HIL because the models did not contain sufficient detail.

In the RLM transition, GM is working to improve the fidelity of the plant models that run on the HIL system. The intent is to make the accuracy representative of a real engine or transmission. It will take computing power that does not currently exist and working knowledge to get to the point that the calibration can be accomplished at a 90% confidence level. Bogden said that GM's goal within the next year is to obtain a 65% level and then improve on a year-to-year basis. A new Pontiac, MI lab facility will be the key to getting GM in the 70% to 75% range.

In addition to HIL, GM engineers are spending a lot of time modeling the algorithms and the signals of the sensors and actuators—the signal delivery models. Using Mathworks Simulink, a math-based tool, engineers can show how the algorithm is supposed to work and run an executable model before it is coded. This lets engineers to see at an early level if the algorithms are conceptually performing as expected before spending much time writing the code and installing it in an electronic module.

With these signal delivery models, engineers can apply mathematical techniques, simulate and then code, and put this code into the engine control module. A lot of effort is occurring to develop direct coding from Simulink by linking math libraries to the tool. Bogden expects the tool eventually to be able to generate code that will go directly into the engine control module without requiring anyone to write the code. Today, rapid prototyping tools, such as dSpace and ETAS, allow autocoding into rapid prototype controllers but not to production controllers. The rapid prototype tool can be connected to an engine or transmission and run the

system, but they are expensive, costing thousands of dollars. Writing directly to a production controller is happening in pieces now and the pieces are being integrated with handwritten code.

One of the goals in GM's RLM transition is early simulation of ideas in the design process before connecting them to the actual hardware. GM uses another tool from Synopsis for the modeling of sensors and actuators. All of these tools must connect together in the test environment.

Getting the tools to communicate with each other and share information is a big part of Bogden's team's job. As the tools improve, their capabilities improve, but this requires users to stay on top of what has changed and determine if that change could in uence integrating the tool into the existing environment.

According to Bogden, the engineering process (shown in Figure 2) will meet GM's needs for the near future. However, as the RLM is implemented, more emphasis is placed up front. This is in contrast to the industry 10 to 20 years ago, where more emphasis occurred at the end of the process. As pieces of software were working, they would be evaluated on a dynamometer or vehicle and then revised after the evaluation. The effort was on the back end of the process until the system worked. It took a lot of time and many vehicles to make that process work. Today and in the future with the math-based process, more emphasis will be placed on the modeling and simulation in the upfront process to get things to work before going to the vehicle and "get it right the first time." This will minimize the time to production after vehicle evaluation. "That's where the

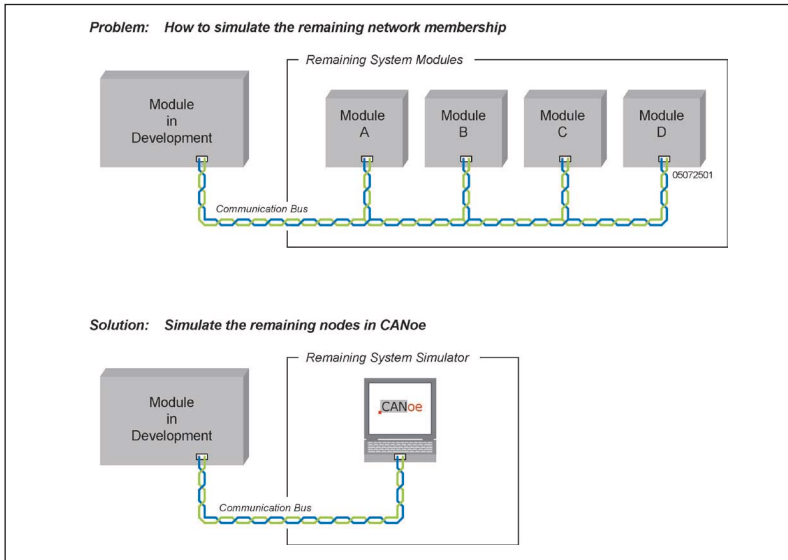


Figure 5. Simulating other system modules using CANoe. Courtesy of Vector CANtech.



Figure 6. CANoe panels for controlling ECUs. Courtesy of Vector CANtech.

whole industry is focusing their energies, on how do we get faster to market and at the same time how do we apply these math-based tools to do statistical studies, design for Six-Sigma, and robustness analysis," said Bogden.

DISTRIBUTED VEHICLE CONTROL

The testing and simulation discussed so far focused strictly on powertrain control. The same or similar approach could be used for other systems, but when the systems come together in the vehicle, the

complexity increases even further. To deal with the distributed aspects of vehicle control, one company, Vector CANtech, provides CANoe, a product that allows module suppliers to evaluate the impact of their module in the system and the impact of the system on their module. As shown in Figure 5, CANoe replaces real modules with simulated modules. CANoe can run up to six CAN buses and even LIN and manage the system modules. The process is somewhat independent of the communication bus and it even acts as the gateway between the buses. Figure 6 shows a CANoe screen view of the control for console, door, dashboard, engine control, gateway, and network management. With modules available at different times during the vehicles development process, simulating those that are not available allows design evaluation and testing to proceed rather than be stalled waiting for complete hardware availability.

CANoe has been used in Europe where vehicle complexity is high but is just starting to be considered for U.S. development. "CANoe is fast enough to run all of the system in real time," said Bruce Emaus, president, Vector CANtech Inc.

IDENTIFYING PROBLEMS EARLY—THE PARTITIONING PROBLEM

To perform simulation at the vehicle system level, simulation tools suppliers, such as Vector CANtech are developing or have developed tools to verify performance and system aspects such as partitioning early in the development process. For example, installing a LIN bus in a body electronic network to reduce hardware

See Software continued on page 42

Software

Continued from page 18

costs could create a system problem requiring extensive engineering time to resolve—offsetting the hardware savings.

Consider the situation with a LIN node on a CAN bus, the LIN message goes through a gateway to CAN back through another gateway to LIN for a door-to-door network. The total amount of time for this process, including the transitions could create a delay problem. Simulation tools can reveal these types of problems early. In fact, this is a rather straightforward example. X-by-wire systems will raise the level of complexity even further, noted Emaus.

MODELING COST DECISIONS

With the new approach that GM is taking, cost decisions can be made with the models. For example, a choice of sensor suppliers could occur based on using the model that each supplier provided and their cost quote. The models could prove that the lowest cost sensor meets the design specifications. With sufficient fidelity, “what if” design decisions can be made at the simulation level. Some of the component models are at this level but others require more effort. For example, oxygen sensors and fuel injector models accurately represent how their real counterparts perform.

GM is working with its suppliers to ensure that they understand their role in GM’s process and that they know what is expected regarding models for their products. By creating accurate models, suppliers have found out more about their products. “Everybody is moving along this

path,” said Bogden. “I believe that in three to five years we are going to have very good models of all the sensors and actuators from suppliers.” At that point, when car companies want to make trade-off studies and cost decisions, they will be able to get good results.

REFERENCE

1. *Designing Electronic Powertrain Controls Symposium, SAE Symposium, Austin, TX, May 4-6, 2004.*

ABOUT THE AUTHOR

Randy Frank is president of Randy Frank & Associates Ltd., a technical marketing consulting firm based in Scottsdale, AZ. He is an SAE and IEEE Fellow and has been involved in automotive electronics for more than 25 years. He can be reached at r.frank@ieee.org.