2008-01-2711

Open Architecture Solution for Hardware-in-the-Loop Testing

Larry Long and Gardell Gefke
Vehicle Systems Integration

Brian McKay
The MathWorks

ABSTRACT

Hardware-in-the-loop (HIL) testing has become an essential verification step in the development of vehicle electronics and software systems. New system concepts continue to drive the requirements for HIL systems. The use of an open architecture for HIL testing provides many benefits to meet these requirements quickly and cost effectively. In this paper we will discuss the development of an open architecture HIL system for a J1939 bandwidth study. We will show how this HIL system was used to test and validate that a heavily loaded networks can operate without compromising the performance of safety critical systems

INTRODUCTION

Both rapid prototyping and hardware-in-the-loop testing of embedded control systems have been a key part of algorithm development and Model-Based Design for almost 15 years. In most implementations of these real-time systems, the key benefit is that they greatly streamline the process of integrating the software and the hardware. As commercial off-the-shelf (COTS) hardware has evolved, it has created a new class of open architecture real-time solutions to evaluate.

OPEN ARCHITECTURE HARDWARE-IN-THE-LOOP SYSTEM FOR A HEAVY-DUTY TRUCK

DEPARTMENT OF TRANSPORTATION (DOT) TESTING OF J1939 NETWORK

The SAE J1939 network, onboard the vehicle, provides a high-speed communications network that a number of subsystems, both safety-critical and non-critical, currently utilize. It is likely that the number of subsystems that require this network will increase on future commercial vehicles.

The challenge the industry faces is in minimizing the potential risks associated with the coexistence of both critical and non-critical systems on the same data bus. Critical safety systems would include braking-related systems, collision warning systems (CWS), roll stability systems, adaptive cruise control, and engine and transmission systems. These systems must communicate in real time or near real time to operate properly. J1939 is important to the proper operation of several components on the vehicle. These components include the engine, automated transmission, antilock braking system (ABS), and CWS. Engines can experience a reduction in power if data is not received from the J1939 network.

As each of these types of systems operates on the network, it increases the bus loading and stress on the network. Therefore, the DOT desired a careful testing of SAE J1939 to ensure that the coexistence did not undermine or compromise major vehicle systems such as brake controls.

RATIONALE FOR HARDWARE-IN-THE-LOOP (HIL) SIMULATOR

Among the initial industry contacts, the J1939 component suppliers expressed a common interest that the network represent an accurate model of a truck-borne system. Many had hoped that an actual truck would be utilized for the setup, but if not, at a minimum the setup would incorporate as many actual electronic control units (ECUs) as possible instead of emulated ECUs. Due to this shared industry position, it was apparent that to provide the most credible and useful test of the J1939 network, actual ECU hardware must be integrated into the test bed setup.

For several reasons, the research team recommended using a full-scale HIL laboratory network for the test activity, rather than a production Class 8 truck tractor. The goal of the program was to test the performance of a nominal SAE J1939 network in a laboratory setting, as opposed to that of a particular instance of a production vehicle. Also, there was a need for accessibility to system components, specific locations on the network cabling and harnesses, and fabrication complexity beyond normal manufacturing processes. In addition, it is impractical (and it presents potential safety implications) to implement the network traffic loading,

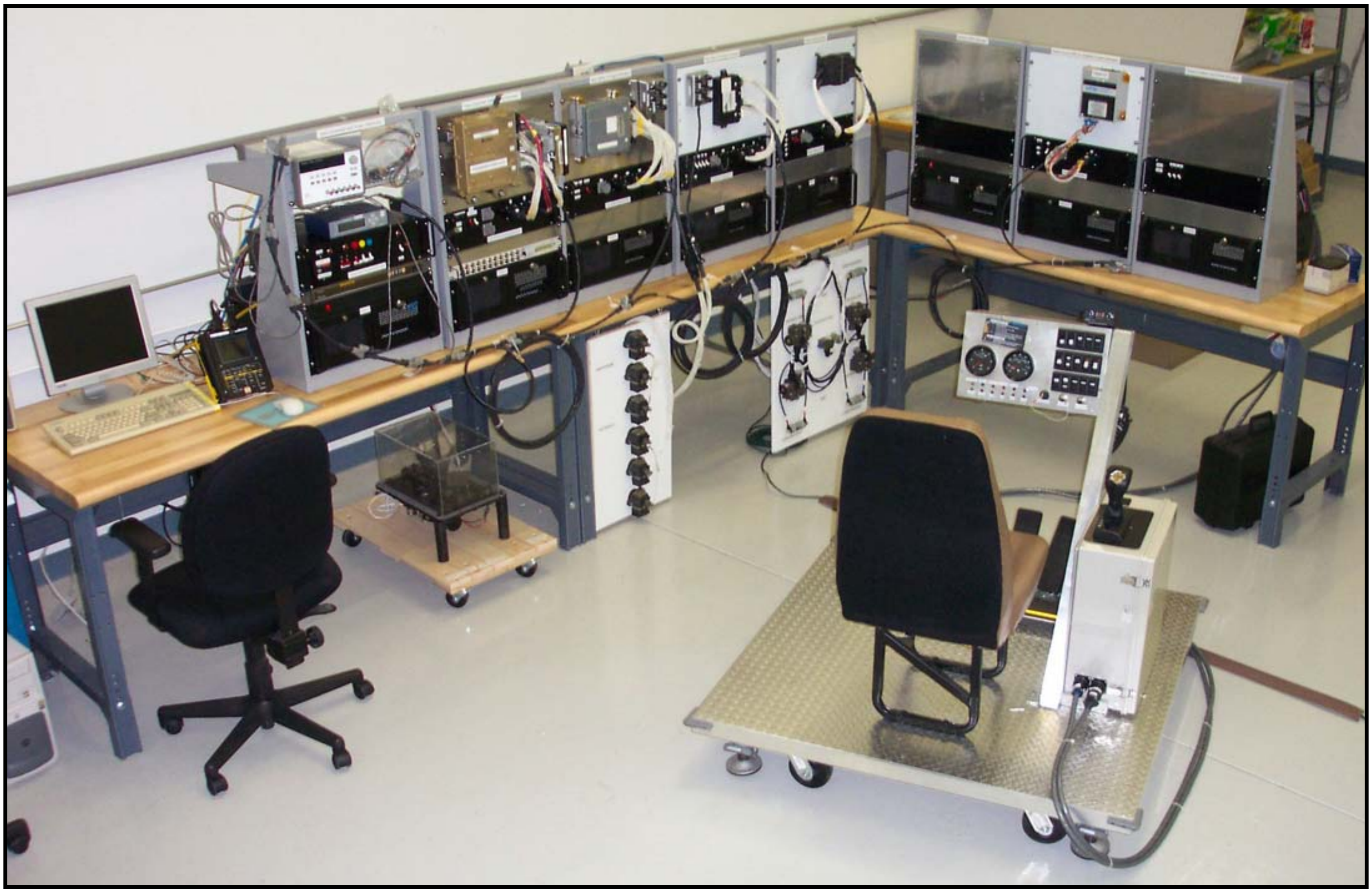

Exhibit 1 – HIL Simulator Test Bed

physical failures, driving scenarios, and test monitoring equipment while actually operating a vehicle.

OVERVIEW OF TEST BED SETUP

Simulator Design

One of the early design decisions facing the team was the selection of the HIL platform. A survey of available HIL platforms was performed and the solutions fell into one of two categories. Either a single vendor closed architecture solution (e.g. ETAS or dSpace) or an open architecture system (e.g. Labview or Simulink with xPC Target). The closed systems limit the expandability of the system but offer tightly integrated software/hardware solutions. The open systems offered a wide range of interface options, however the software was less well developed for Automotive HIL applications. However the tools were available in the open systems to add the software functionality required. Another major advantage of the open architecture system was the hardware cost. The closed architecture systems were expensive to acquire and set up. Furthermore, they would require a continuing financial relationship with the vendor for maintenance and enhancements. As a result of the cost issues and the availability of Simulink vendor models the decision was made to develop the HIL on the Simulink/xPC Target open architecture platform.

The simulator was intended to emulate a well-equipped, state-of-the-practice Class 8 truck currently The simulator was developed with a high degree of modularity to facilitate the simulation of different truck hardware configurations. This modularity was present in both the hardware and software relationships of the simulator. The simulator utilized a distributed computing environment in which each ECU was connected to a single target computer.

The target computers exchanged data and timing information on a private, secondary CAN network, operating independently of the J1939 network. Using modular components facilitated the addition or replacement of ECUs and their associated hardware.

For example, replacing the ABS ECU portion of the simulator with an electronically controlled braking system (ECBS) ECU only required the hardware exchange of the ABS interface box for an ECBS interface box. Similarly, the software changes needed only required that the ABS plant target model be swapped for the ECBS plant target model. The remainder of the simulator hardware and software remained unchanged.

Open Architecture Hardware

Exhibit 2 presents a block diagram of the simulator. The yellow boxes denote truck hardware. These components were purchased from truck parts suppliers in order to be as representative of hardware on the road as possible. The blue boxes denote the hardware that was purchased as COTS equipment. The green boxes denote the hardware that was custom designed and built for this program. Physical faults were inserted into the network at either physical fault test point “A” or “B” in the test bed setup.

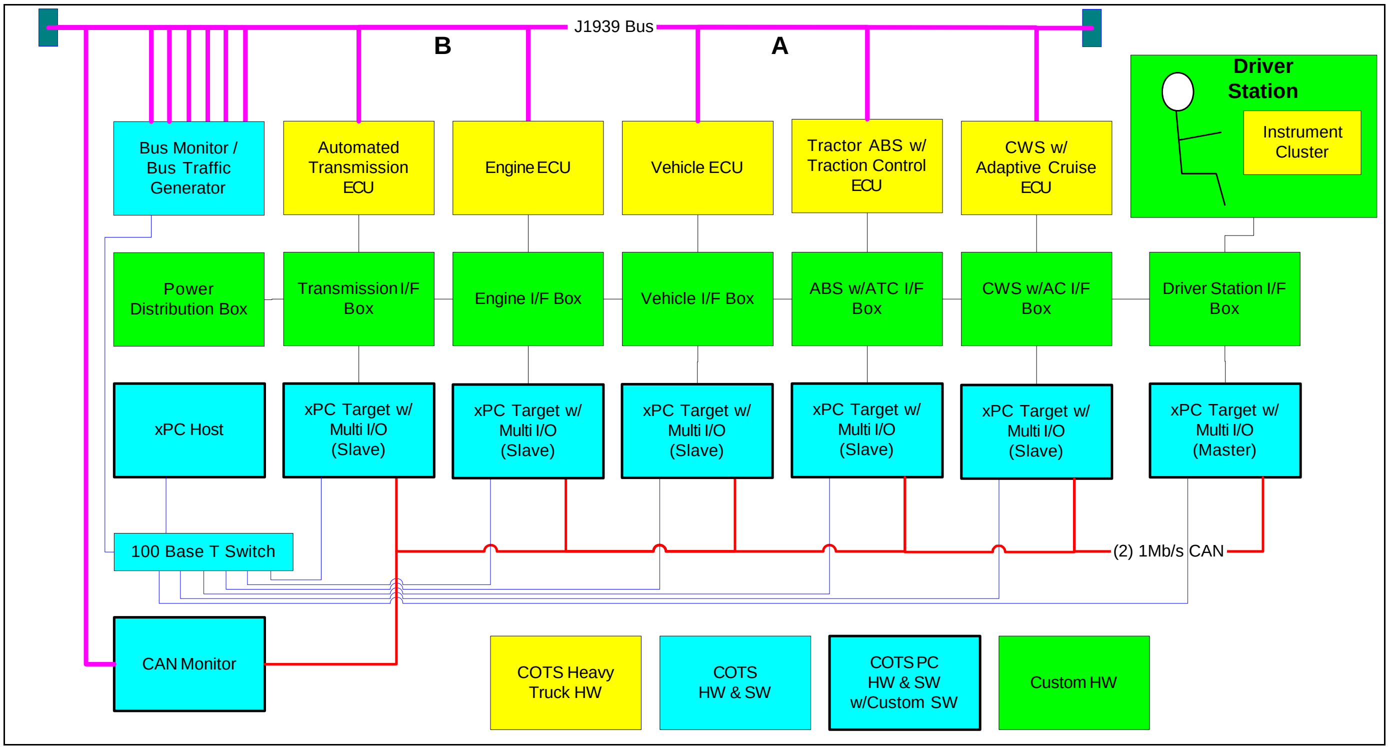
J1939 Bus
B
A
Driver Station
Instrument Cluster
Bus Monitor / Bus Traffic Generator
Automated Transmission ECU
Engine ECU
Vehicle ECU
Tractor ABS w/ Traction Control ECU
CWS w/ Adaptive Cruise ECU
Power Distribution Box
Transmission I/F Box
Engine I/F Box
Vehicle I/F Box
ABS w/ATC I/F Box
CWS w/AC I/F Box
Driver Station I/F Box
xPC Host
xPC Target w/ Multi I/O (Slave)
xPC Target w/ Multi I/O (Slave)
xPC Target w/ Multi I/O (Slave)
xPC Target w/ Multi I/O (Slave)
xPC Target w/ Multi I/O (Slave)
xPC Target w/ Multi I/O (Master)
100 Base T Switch
(2) 1Mb/s CAN
CAN Monitor
COTS Heavy Truck HW
COTS HW & SW
COTS PC HW & SW w/Custom SW
Custom HW

Exhibit 2 – HIL Test Bed Simulator Block Diagram

Each ECU was interfaced to a general-purpose PC with I/O cards through an interface box. The system utilized a number of different PCI I/O cards as listed below:

- National Instruments PCI-6602 counter/timer cards
- National Instruments PCI-6040E multifunction I/O cards
- National Instruments PCI-6503 24-channel digital I/O cards
- National Instruments PCI-6527 48-channel isolated digital I/O cards
- National Instruments PCI-6703 16-channel analog output cards
- Softing CAN-AC2-PCI dual CAN interface cards

The interface boxes were custom designed and contained all of the power, signal conditioning, and switchgear required to convert the signals in and out of the ECU to allow it to be interfaced with the target computer. In addition to the HIL hardware, the test bed included equipment for simulating an additional 6 ECUs (12 controller applications) as well as CAN analysis tools.

The interface boxes contained circuit protection for each of the DC power lines supplied to the ECU. A J1587/1708 connector was provided on the front of each interface box to permit the use of a J1587 service tool.

General-purpose computers were selected over proprietary HIL systems primarily due to lower hardware cost and increased flexibility. Rack mounted PCs with sufficient processor speed and a large number of PCI slots were readily available for under $1000 each. These PCs were easily tailored to the needs of each ECU through integration of available I/O boards. The use of common PC components also facilitates rapid, inexpensive upgrades to processors, storage, and memory that can be made with minimal impact on other interface hardware.

The general-purpose computers used in HIL testing had an Asus A7V8X motherboard with an AMD Athlon

XP2100 processor and 512 KB PC2700 memory in a rack mount case.

Software

The software environment chosen to develop the heavy-truck simulator was based on products from The MathWorks including:

- MATLAB® – Provided a high-level modeling/scripting language
- Simulink – Provided a visual modeling environment
- Real-Time Workshop – Compiled Simulink models for real-time applications
- xPC Target – Enabled use of PC hardware and COTS data acquisition cards as a real-time target

Using MATLAB and Simulink allowed the program team to utilize vendor-supplied models. Most of the ECU vendors use the MATLAB and Simulink environment for ECU development, regardless of the HIL platform they may utilize for internal testing. The use of common tools allowed the program team to leverage the plant modeling work performed by vendors in support of their development programs.

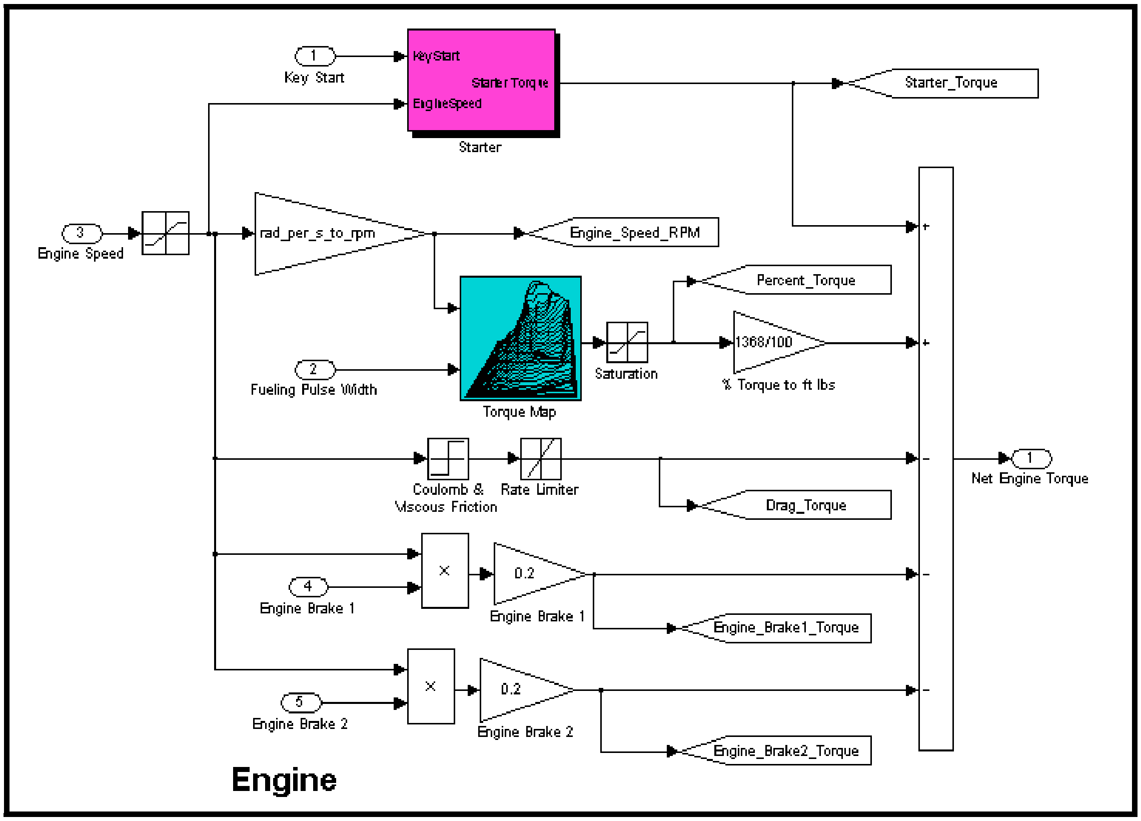

Exhibit 3 – Engine Plant Simulink Model

The computers that directly interfaced to each of the ECUs ran xPC Target, a real-time operating system that executes on the widely used x86 Intel architecture. The targets booted into xPC Target from a floppy disc, and the compiled models were downloaded from the host across the private Ethernet network. The remainder of the development environment resided on the host computer. This computer was similar to the target computers except that it ran Windows 2000.

The target computers were networked together with a 1 Mbps CAN network. At each 1 ms time step this network passed model parameters between the target computers, and timed the execution of the targets. The highest priority message was used to synchronize all of the targets. In addition, the CAN network was operated in a synchronous manner by scheduling each of the messages on the bus. This approach limited the impact of the inherent non real-time behavior of a CAN network. Using this approach, the variable latencies of CAN were minimized and timing jitter on the order of 10us, or 1% of each step time was achieved.

A separate software simulation model of each physical subsystem (plant model) was developed for each ECU's target PC. The plant models were developed in MATLAB and Simulink, and compiled in Real-Time Workshop with xPC Target in preparation for download to the individual target computers. Exhibit 3 illustrates the top level of the engine model used in the simulator.

The engine plant model utilized a time-based look-up-table architecture. This type of model was used in lieu of a more sophisticated physics-based model, since the engine simulation fidelity did not significantly affect the quality or quantity of J1939 traffic, the primary focus of this study. Most of each time step was dedicated to I/O and CAN latencies rather than model calculations. In fact, when operated in a single target configuration, plant model update rates were increased from 1 kHz to 10kHz without encountering a step time overrun error.

SIMULATOR CORRELATION AND VALIDATION

In any project in which simulation results are used to make qualitative or quantitative statements about the performance of a particular “real-world” system, it is critical to correlate the simulation environment and the real-world implementation of that same system. The extent to which this is possible increases the confidence the simulation is responding as the real system would under the same or different conditions. Fortunately for this testing program, a production truck tractor with similar system components and configuration to the HIL simulator was available to the team for data collection on a limited basis. Exhibit 4 shows the tractor test bed used to obtain correlation data. Due to the limited availability of the test truck, the correlation effort was directed at ensuring that the general characteristics of the J1939 traffic were similar between the truck and the HIL. One of the key measurements made on the HIL was bus loading, and much of the correlation testing was dedicated to measuring this parameter under a variety of conditions. This testing was performed on public roads so it was not possible to perform repeated safety-critical maneuvers.

Exhibit 4 – Tractor Test Bed

Network Data Collection from Tractor Test Bed

The test tractor ECU configuration was nearly identical to simulator ECU configuration. The tractor engine ECU was an updated version of the unit used in the HIL, and the vehicle ECU, which generates the instrument panel data, had a different software load. The HIL engine ECU was changed out to match the test tractor, and the vehicle ECU was reprogrammed to match the test tractor, thus eliminating the variations. The HIL modifications also allowed full use of the CWS system including adaptive cruise control (ACC), and updated the HIL to improve its fidelity. The ABS installed on the test truck was produced by a different manufacturer than the system integrated in the HIL; however, both vehicles used a four-modulator/four-sensor system with traction control.

The test tractor J1939 network was monitored and raw message data was collected while operating the tractor in a variety of situations. Step throttle inputs while stationary were used to derive the engine damping and inertial parameters to tune the engine model for a better real-world performance match. Shifting performance data was collected while driving at a variety of speeds. These included part-throttle and full-throttle upshifting and braking-induced downshifting. The ABS, ATC, and CWS/ACC were used during a series of subsystem driving events to evaluate the bus loading while the tractor was driven in conditions likely to stress the network (e.g., heavy loading, several ECUs requiring coordinated operation). The ABS was activated on dry, wet, and icy pavement conditions, as was the ATC. A sufficient number of events was recorded in which the ABS was activated to provide a comprehensive database for correlation. On the road, CWS ACC events were also recorded.

Correlation of HIL Simulator Versus Actual Truck

The focus of the correlation activity was to validate that the characteristics of the J1939 messaging traffic were similar to that of the test truck.

With the data collected from the tractor test bed, it was possible to compare the quality and quantity of J1939 message traffic between an actual heavy truck and the HIL simulator. Exhibit 5 compares the network data loading for the first 20 seconds of a full-throttle acceleration for both the HIL simulator and an actual tractor. Both were accelerated from zero to near 39 miles per hour. However, the tractor took approximately 21 seconds, while the HIL took 25 seconds. The difference was primarily the result of the HIL simulator having a lightly loaded trailer vs. no trailer for the test truck. From this data, it is possible to make several observations.

The vehicle and HIL simulator performance characteristics were similar. The acceleration profiles, with the slight speed dip during and just after a gear change, correlated well. The Torque/Speed Control 1 message (TSC1) rates and intervals showed similar characteristics and trends. There was comparable functionality of the automated transmission, the collision warning system, the adaptive cruise control, the antilock braking system, and the automatic traction control.

The J1939 network loading during the acceleration showed very good correlation between the actual versus the simulated tractor for similar events. Exhibit 5 and Exhibit 6 provide a comparison of the J1939 network busload between the HIL simulator and the actual test tractor during the vehicle acceleration profile. These exhibits show that both networks peaked at 25 to 26 percent of maximum capacity during shifting, and both run a similar 18-percent load during non-TSC1 event periods.

Together, this data confirmed that the HIL simulator was accurately utilizing the J1939 network. This meant that testing results on overall network loading for specific driving event scenarios would be an excellent indicator for the overall network loading levels of actual trucks under similar circumstances.

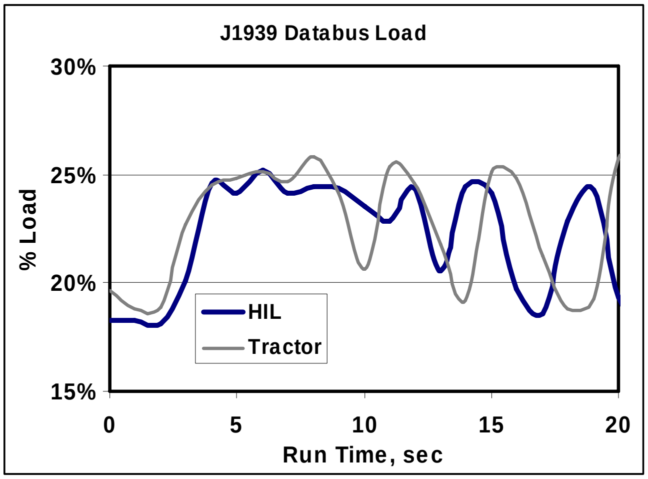

Exhibit 5 –
J1939 Network Load During Acceleration

While the overall network utilization correlation was very good, specific performance characteristics highlighted differences in the HIL simulator and the actual tractor performance. Exhibit 6 shows the engine speed in RPM, extracted from the Electronic Engine Control 1 message (EEC1), from both the HIL and the tractor. The overall RPM range and shift profiles are very similar. However, the HIL exhibits much sharper corners at the gear transition points. This is the result of slightly imperfect modeling of the end-to-end dynamics from the engine crankshaft to the actual wheels. The HIL does not fully capture all of the damping present in the real world, particularly at the moment of clutch plate release and engagement.

As shown in the J1939 network loading exhibits, these inevitable modeling errors did not meaningfully detract from the overall excellent correlation between the simulator and a real truck.

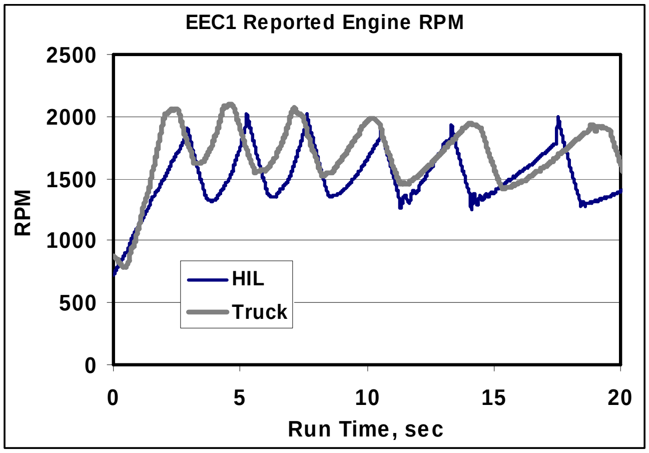

Exhibit 6 –
EEC1 Engine Speed During Full Throttle Acceleration

CONCLUSION

An open architecture HIL testing system was developed to analyze heavy-duty truck J1939 bus loading. The main purpose of the HIL system to was to study and ensure and validate that a heavily loaded J1939 network can operate without compromising the performance of safety-critical systems such as brake controls. The open architecture HIL test system, based on commercial off-the-shelf hardware, was able to satisfy all test performance, accuracy and cost objectives.

CONTACT

Larry Long, Vehicle Systems Integration (VSI)
Mr. Long is the president of Vehicle Systems Integration, a consulting firm specializing in vehicle control and communications. Prior to launching VSI, Mr. Long was a contractor for the consulting firm Booz Allen Hamilton developing controllers for a DOT sponsored fuel cell powered transit bus program. Mr. Long holds a B.S and M.S in Aerospace Engineering from the University of Maryland.
larry.long@vehiclesystems.biz

Gardell Gefke, Vehicle Systems Integration
Mr. Gefke is a systems engineer with Vehicle Systems Integration, a consulting firm specializing in vehicle control and communications. Prior to co-founding VSI, Mr. Gefke was employed at the University of Maryland Space Systems Laboratory where he served as the program manager for Ranger TFX, a shuttle-based flight experiment of a dexterous robot. Mr. Gefke holds an S.B. in Aeronautics and Astronautics from the Massachusetts Institute of Technology and an M.S in Aerospace Engineering from the University of Maryland.

brian.mckay@mathworks.com

Brian McKay, The MathWorks

Brian McKay is the rapid prototyping and HIL product marketing manager at The MathWorks. Before joining The MathWorks in 2005, Brian worked at Nuvera Fuel Cells developing control systems for automotive-based hydrogen reforming technologies. He has more than 10 years of automotive industry experience, including senior application engineering positions at dSPACE, Ricardo, and Ford Motor Company. Brian holds a B.S and M.S. in General Engineering, with an emphasis in control systems, from the University of Illinois at Urbana-Champaign.