



# *Designing and Testing* **Electronic Body Systems** *at* **Jaguar**

It is a mistake to take good electronic body system (EBS) design for granted. Functions such as exterior lighting, windscreen cleaning, and locking are highly visible to the customer. A superbly engineered locking system is unlikely to convince them to buy your product, but a poorly engineered one will certainly reduce the chances of their buying another.

While EBS systems that are intuitive, logical, and consistent reinforce the image of a high-quality vehicle, they are becoming increasingly challenging to design. Nowadays, more and more of the control is performed automatically, to increase driver convenience, and controls are becoming increasingly distributed, to reduce the amount of wiring in the vehicle and free up interior space.

At Jaguar, design requirements like these were placing a strain on our traditional development approach and making it dif-

ficult for EBS development engineers to deliver their designs on time. We found our solution in a virtual integration and test automation laboratory.

## **The Traditional Design Process at Jaguar**

EBS software development at Jaguar typically begins with system engineers developing requirements, which they distribute as a paper specification to the suppliers who develop the module hardware and software. After testing the module against the speci-

fication the supplier delivers it to Jaguar, where engineers test the system on a 'breadboard' and then on a prototype vehicle.

This approach has three fundamental problems:

- Written specifications are incomplete and subject to interpretation errors.
- Meaningful system testing cannot begin until late in the project.
- System testing is complex and extremely labor-intensive.

While the first of these problems is well appreciated throughout the engineering industry, the second and third are largely a result of the limitations of our traditional breadboard test platform.

## **Breadboard Limitations**

The breadboard (Figure 1) consists of all the vehicle's electrical modules, sensors, switches, and actuators connected together using the production-intent wiring harness on a metal frame.

The breadboard has several inherent problems. First, meaningful testing cannot begin until all the components and the vehi-



Figure 1. A traditional breadboard.

cle wiring harness are available. (Typically, the wiring harness is the last item to be delivered—perhaps only two weeks before building of the prototype vehicles begins.) Second, testing must be performed manually using the actual switches, sensors, and actuators. Third, it is difficult to test system response under failure conditions, such as a blown fuse or wiring short circuit, as the engineer must physically introduce the fault. Finally, breadboard testing can cover only static vehicle conditions, making it

very difficult to test systems, such as drive-away door locking, that alter behavior as the vehicle is driven.

These problems place a heavy load on engineering resources toward the end of the development cycle. Inevitably, flaws in the written specifications are rarely found until just before the prototype vehicles are built. From that point on the engineers are playing catch-up, caught up in multiple software bug-fix loops and weeks of tedious manual verification.

## Introducing Model-Based Design

The Simulation and Control Group within Jaguar Cars is responsible for supporting and promoting the use of model-based software development and testing throughout the electrical engineering departments of Jaguar and our sister company, Land Rover. Previously, most of our efforts focused on powertrain and chassis applications, particularly hardware-in-the-loop testing.

In March 2003, we were tasked with supporting the adoption of Model-Based Design within the EBS area. Central to this methodology is the concept of the EBS Virtual Integration and Test Automation Laboratory (EBS-VITAL). With the EBS-VITAL (Figure 2), a simulation-based version of the traditional breadboard, we can begin automatic testing of the EBS software very early in the development cycle.

## Automated Testing with EBS-VITAL

The EBS-VITAL consists of all EBS control modules connected to a custom-built real-time simulator (RTS) supplied by dSPACE. It enables us to run Simulink®

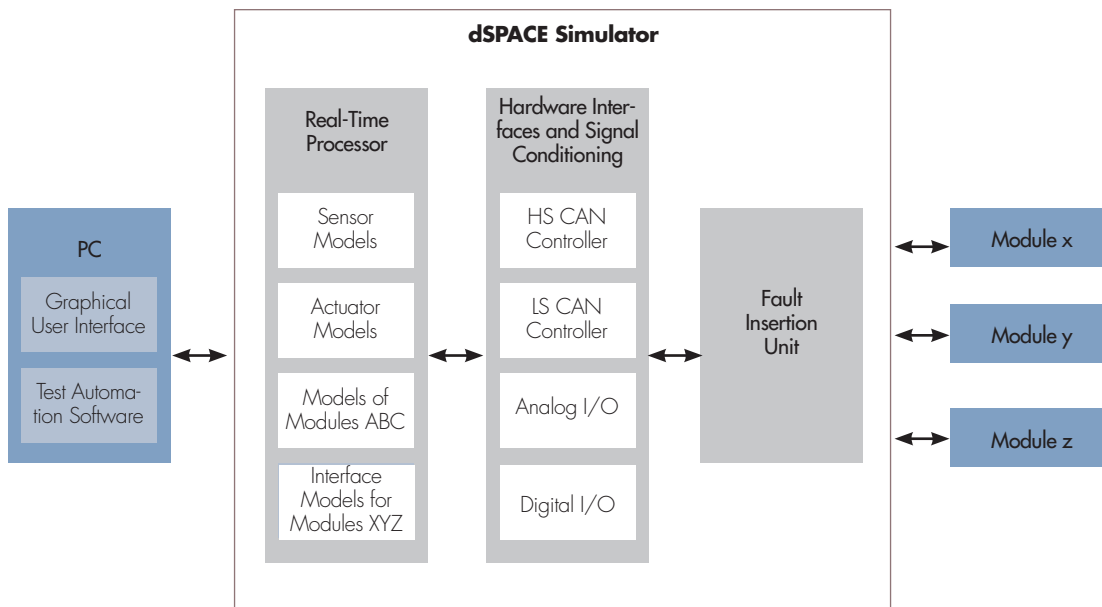


Figure 2. The EBS-VITAL topology.

Producing models during the very early stages of development increases the chance of uncovering issues in the system design or errors in the module specifications.

models of the EBS sensors and actuators, as well as simple models of the vehicle's powertrain and entertainment systems, in real time. The modules are connected to the EBS-VITAL using a simple wiring harness that links all the module inputs and outputs (I/O) directly to the RTS.

The EBS-VITAL offers many advantages over the traditional breadboard. We can run the full security software test plan automatically in 36 hours (typically, over a weekend), plus two engineer-days to review the results. Performed manually, these tests would take more than four engineer-weeks. Testing can begin much earlier, as the EBS-VITAL wiring harness is not subject to the same packaging constraints as the real harness, and unavailable modules can be replaced by a real-time simulation of a Simulink model. We can easily test dynamic functionality by simulating the vehicle's behavior during driving. The EBS-VITAL makes it easier to test system behavior under fault or on-limit conditions. We can modify sensor and actuator models, and all RTS I/O is linked to a fault insertion unit, enabling us to introduce faults during automated testing.

One of the biggest benefits provided by the EBS-VITAL has been its ability to help us modify functional behaviour during the project. We can update the module models to reflect the new functionality and then perform thorough system testing automatically on all functions, not just the one

being modified. As a result, we can identify and modify unforeseen side effects of the modifications before the suppliers begin their software changes, drastically reducing the time and cost of successfully introducing new functionality.

### Revamping the Development Process

Using the EBS-VITAL and Model-Based Design, testing takes place in three phrases.

In phase one, we replace all EBS modules with Simulink and Stateflow® models that capture the system requirements. Module suppliers use the models to develop hardware and software and test the module against the specification.

Phase two testing begins as soon as a module is delivered. In this phase, the EBS-VITAL becomes a hybrid of real EBS modules and models. Phase three begins when the last module has been delivered and the EBS-VITAL contains a full set of actual EBS modules.

In practice, it is rare for all EBS control modules to be replaced by models. Most of our systems include at least one module from a previous project. Where such a carryover component exists, there is little benefit in producing a model. Instead, we introduce the real component at the earliest opportunity. The EBS-VITAL then immediately enters phase two, the hybrid stage.

### Lessons Learned

Producing models during the very early stages of development increases the chance of uncovering issues in the system design or errors in the module specifications. We can use the models to demonstrate the new features to key decision-makers at the start of the program. By obtaining their feedback early on, we reduce the number of last-minute changes. In some cases, a model has been developed to a sufficient standard to enable us to automatically generate the module's application software using Stateflow Coder.

While Model-Based Design provides many benefits, it also introduces challenges. Considerable engineering effort is required to develop the module models. This is particularly challenging in an environment where the model-based project is running in parallel with projects following a traditional process, with a high demand for engineering resources toward the end of the program. In addition, it takes time and coaching for systems engineers with little software development experience to become proficient Simulink and Stateflow users, particularly when the models they are producing will be used to automatically generate production software.

Fortunately, these roadblocks become less significant on each subsequent project, and in our opinion, the effort and investment required to overcome them is far outweighed by the benefits of adopting Model-Based Design. ◀◀

#### RESOURCES

▶ **Jaguar Cars Limited**  
[www.jaguar.co.uk](http://www.jaguar.co.uk)

▶ **International Automotive Conference**  
[www.mathworks.com/res/iac06](http://www.mathworks.com/res/iac06)

