Using LabVIEW, NI VeriStand to Create DYNACAR, a Model-Based Dynamometer With Full Vehicle Simulation

"Instead of creating a test architecture from scratch, we used NI VeriStand to configure our test system, DYNACAR, which significantly reduced our development time."

Authors:
Miguel Allende (miguel.allende@tecnalia.com),
Iñaki iglesias (inaki.iglesias@tecnalia.com),
Adrian Martin (adrian.martin@tecnalia.com),
Alberto Peña (alberto.pena@tecnalia.com), Tecnalia Research & Innovation

The Challenge:
Developing an easily configurable, modular vehicle model and test system to use in a model-based dynamometer and to rapidly prototype vehicle control systems.

The Solution:
Creating a comprehensive test system based on NI LabVIEW software as a simulation tool and NI VeriStand software for stimulus generation, data logging, and report generation.

Introduction:
Tecnalia, one of the largest private research, development, and innovation groups in Europe, wanted to develop a tool for the following tasks:

1. Smooth transition between the design and rapid control prototyping stages of embedded software development
2. Advanced control strategy testing for a vehicle's power train

We started from the idea of generating a “virtual vehicle” embedded in a real-time system that could accurately calculate real-world variables present in a vehicle when driving. (Figure 1)

We decided to divide the vehicle model into subsystems such as the drive train, power train, wheels, chassis, and brakes. This distribution has several very important advantages. It helps us assign development for each model subsystem to the person that has the most knowledge of the associated vehicle subsystem, and we can update the model in stages as new subsystems are completed. With this modular approach, we can easily replace simulated components with real components as they become available, and the user can swap subsystems that we created with subsystems that were developed in other environments.

Our test software required the following capabilities:
- Integrate various subsystem models to generate a real-time compatible system model (DYNACAR)
- Write subsystem models in a variety of programming languages
- Easily swap models in the system
- Easily replace subsystem models with real components as they become available
- Perform data logging and report generation

We chose NI VeriStand for our test software because it supports all of these requirements out of the box. Instead of creating a test architecture from scratch, we used NI VeriStand to configure our test system, which significantly reduced our development time. It also helped us focus on testing and improving the model.
Figure 2 shows a summarized diagram of the system implementation with the previously explained concepts.

**Application Example**
The objective of the project was to update an existing test bench, which included a complete power train consisting of an electric traction drive connected to two wheels via a differential and two transmission axles. We used Dynacar software to create real-time “vehicle-in-the-loop” and “human-in-the-loop” simulations. These simulations subject the vehicle’s mechanical components to strains and speeds that are equivalent to those experienced by the real car, and it helped us to accurately test our embedded software in the lab, which reduces the amount of time required for field testing.

Figure 3 shows an outline of the project’s basic installation components. EM3 represents the system’s traction drive and EM1 and EM2 are the motors that emulate the resistant force (contact between wheel and tarmac) that the traction drive must overcome based on the driving and environmental parameters.

The following table shows the requirements for this test system:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Update Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute the car model software (DYNACAR)</td>
<td>Update Rate: 1 ms</td>
</tr>
<tr>
<td>Control the three motors via controller area network (CAN) communication</td>
<td>Update Rate: 5 ms</td>
</tr>
<tr>
<td>Carry out auxiliary logic control</td>
<td>Update Rate: 100 ms</td>
</tr>
<tr>
<td>Manage key alarms</td>
<td>Update Rate: 1 ms</td>
</tr>
<tr>
<td>Generate stimulus for automatic tests</td>
<td>Update Rate: 10 ms</td>
</tr>
<tr>
<td>Generate reports</td>
<td>No update rate</td>
</tr>
<tr>
<td></td>
<td>requirement</td>
</tr>
</tbody>
</table>

We used NI VeriStand for our test platform because of its native support for multicore execution. For example, with NI VeriStand we can use three cores for model execution, which ensures a time of 1 ms, while the fourth core is used for the other functions such as alarm management and stimulus generation. Separating this data processing ensured real-time system performance.

**Communication and Tests**

We implemented all communication with the three drives through CAN. We chose the NI PXI-8512/2 module as our I/O device for the CAN messages. This module uses the NI-XNET protocol, which includes an intuitive API that can be extended for custom messages.

Our test system can be used to perform manual testing and it can also automate real-time tests. To perform manual testing, the operator drives the car in a virtual environment to test system response to a variety of scenarios according to the operator’s driving habits. When the system performs automated testing, the model navigates via a virtual driver that follows a predefined driving pattern. Both types of tests allow engineers to test specific operation points that would be difficult or impossible to reproduce during field testing.

**Setting Parameters**

We developed a GUI with LabVIEW (Figure 4). We use the GUI to configure model parameters based on the needs of the test. The model includes up to 150 configurable parameters so you can emulate a range of vehicles and prepare driving scenarios and automatic driving cycles.
**Conclusion**

Tecnalia chose LabVIEW and NI VeriStand as the development environment for the model vehicle and for its implementation in the final test system. This environment offers easy programming and interaction with the hardware, as well as native support for various protocols available in the market. NI VeriStand has a run-time editable UI that provides users with the ability to add controls, indicators, graphics, and records in real time, without the need to stop the execution of the engine running on the PXI controller. Combining Dynacar with NI tools helped us create a state-of-the-art system to develop and validate a vehicle’s components and controllers.

DYNACAR was awarded in 2011 by the Automotive Testing Technology International, CAE Innovation of the year. Figure 5.

Authors:
Miguel Allende (miguel.allende@tecnalia.com),
Iñaki iglesias (inaki.iglesias@tecnalia.com),
Adrian Martin (adrian.martin@tecnalia.com),
Alberto Peña (alberto.pena@tecnalia.com), Tecnalia Research & Innovation
Figura 1: DYNACAR technology features a highly configurable animation environment that provides engineers with visualization and Human in the loop.

Figura 2: DYNACAR implementation diagram developed with NI VeriStand
Figure 3: Bench control diagram

Figure 4: DYNACAR configuration GUI
Figure 5: DYNACAR, CAE Innovation of the year 2011. 