Dynamic
Firing Order

Imagine a combustion engine where each cylinder can be fired variably or skipped to provide the torque demanded by the driver. The engineers at Tula Technology, headquartered in Silicon Valley, have made this vision a reality and are expecting a series production with several OEMs.

D ynamic skip fire (DSF) is the name of the new technology at Tula. By deciding if and when each individual cylinder is ignited, DSF enables a particularly efficient engine operation. The engine control determines the number and sequence of cylinders that need to be fired in order to provide the required torque (figure 1).

Dynamic Firing Increases Comfort and Efficiency

One of the biggest concerns involved in cylinder deactivation lies in deploying production-ready NVH (noise, vibration, harshness) powertrain behavior. To make this possible, intelligent algorithms developed by Tula avoid unfavorable resonances, thus ensuring a comfortable driving experience at any time. By continuously timing the firing order, the DSF avoids resonance frequencies, thereby obtaining noise suppression and vibration reduction (figure 2).

Cost-Effective Solution

Among the low-cost, fuel-efficient technologies available on the market, DSF is the most efficient with fuel savings of up to 20%. Another advantage is its compatibility with other fuel-saving technologies such as direct injection, turbocharging, start-stop systems, and mild or full hybrids. Several OEMs and Tula are working together to get the technology ready for series production.
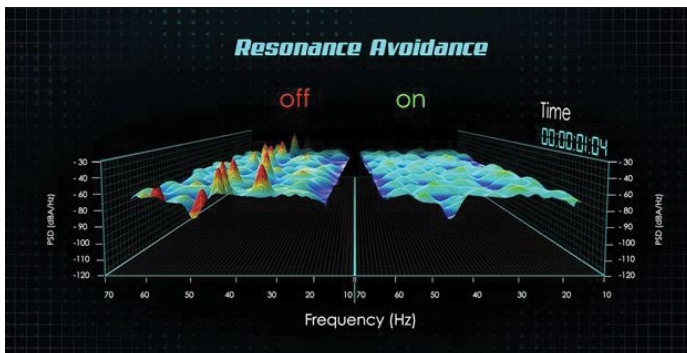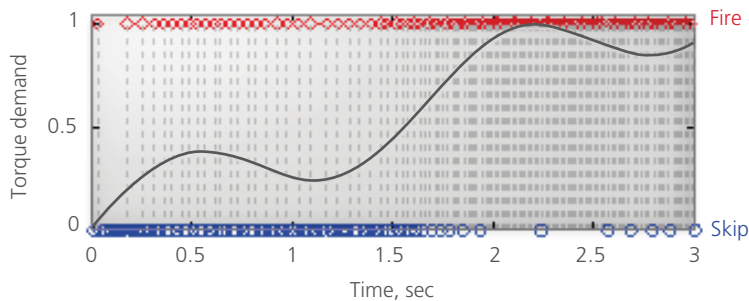
>>

Figure 1 (top): Dynamic Skip Fire decides when to fire each cylinder.
Figure 2 (below): Intelligent algorithms avoid resonance frequencies, thereby optimizing driving comfort.

Development Tasks for Practical Use

Tula's main strength is in algorithms for variable cylinder deactivation and NVH suppression. To demonstrate the practical feasibility of the algorithms, the algorithms need to be implemented and executed in a demonstration vehicle. This is done with a prototyping system that controls a GMC Yukon Denali V8 engine. But before the algorithm is used on a real engine, it must undergo functional validation first. An Hardware-in-the-loop (HIL) simulation system is the best test environment for this. To reduce the overall development time up to completion of the quality assurance (QA) tests, the HIL system should also be available for development tasks. Both the software developers and the testing and QA engineers must have access to the HIL test bench. In addition, the HIL test bench must be accessible from several locations.

Combined Tool Chain

To run the prototype of the DSF technology on customer platforms, Tula chose dSPACE's rapid prototyping tool MicroAutoBox II. To validate the functions of the controller software, Tula developed a HIL test bench that uses the current ASAM HIL API (ASAM = Association for the Standardisation of Automation and Measuring Systems). The HIL test bench receives the analog and digital I/O signals from the hardware and software of a third-party engine simulation. The engine simulation also transmits the crank and camshaft signals to dSPACE MicroAutoBox II, which in turn executes the DSF software. dSPACE AutomationDesk is the test management tool for performing the tests on the DSF software. All the components and AutomationDesk communicate through a common interface.

Challenge for the Combined Tools

Several different processes had to be combined to ensure a reproducible and reliable process: retrieving the source code and automation test scripts, compiling the source code for the test benches, loading executable files on the target HIL test bench, executing automated test scripts, generating summaries, and archiving the results for review and auditing.

Automation to Reach the Goal

To implement the process steps, Tula created suitable software solutions. The first solution combines the engine simulation with dSPACE MicroAutoBox II via AutomationDesk as the framework for test automation. The implementation is done using the current version of the ASAM HIL API standard. With the HIL API, the Python-based test scripts can be written and executed as AutomationDesk test cases. All test cases are available as AutomationDesk projects, which can be executed on the HIL test bench in both closed and open loop. This is usually done when implementing existing fuel test profile (FTP, US Environmental Protection Agency Federal Test Procedure) vehicle cycles. In AutomationDesk projects, it is also possible to collect and record data for the test cases, compare this against predefined pass-fail criteria, and generate comprehensive reports on the performed test runs.

> "Our goal is complete automation for the tests. That is why we use dSPACE AutomationDesk with the ASAM HIL API."
>
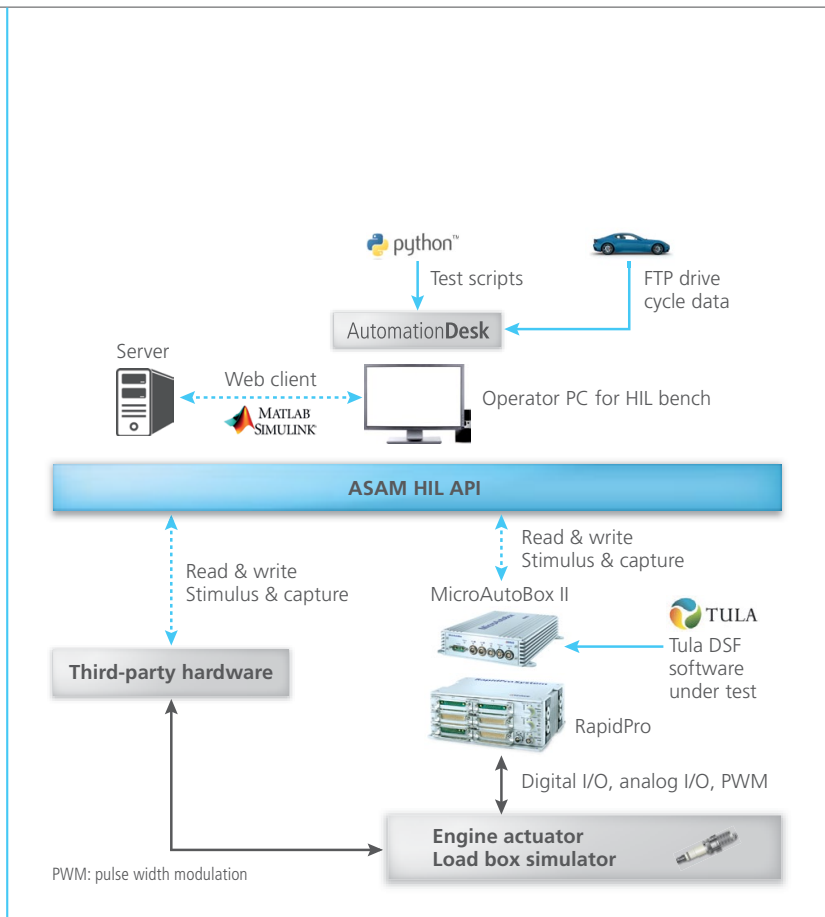> *James McKeever, Tula Technology*

Figure 3: The web-based tool WebCarLab allows remote access to the complex HIL test bench.

Conclusion

Tula is using an automated HIL test bench to develop innovative functions for dynamic cylinder deactivation in internal combustion engines. On the test bench, the controller software is validated for in-vehicle use. With MicroAutoBox II, Tula implements the functions in the demonstration vehicle, reaching fuel savings of up to 20% without sacrificing ride comfort.

WebCarLab – The Optimal Solution
To ensure that the tests are performed from a central source code storage and to enable multiple remote access to the HIL test bench, Tula developed the web-based test automation tool WebCarLab (figure 3). The application communicates with a software configuration manage- ment (SCM) system and provides a web interface. WebCarLab has an intuitive user interface that lets users run tests on the HIL test bench either interactively or in batch mode (batch processing). After the user has chosen a mode, WebCarLab checks the code from the SCM system and performs the selected test cases. In addition to generating test reports, WebCarLab creates all the artifacts that lead to the test results and archives the test reports for future audits. ◾
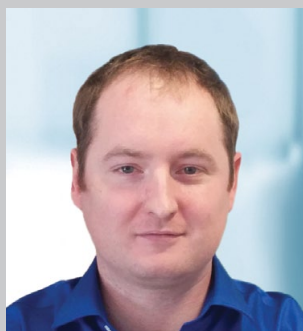
The video shows Dynamic Skip Fire in action.

Paul Liu
Paul Liu is a Manager and Embedded Software Engineer at Tula Technology, Inc. in San Jose, USA.

James McKeever
James McKeever is a Sr. Embedded Software Test Engineer at Tula Technology, Inc. in San Jose, USA.

Abhijit Bansal
Abhijit Bansal is an Embedded Software Controls and Test Engineer at Tula Technology, Inc. in San Jose, USA.