

Problem solver

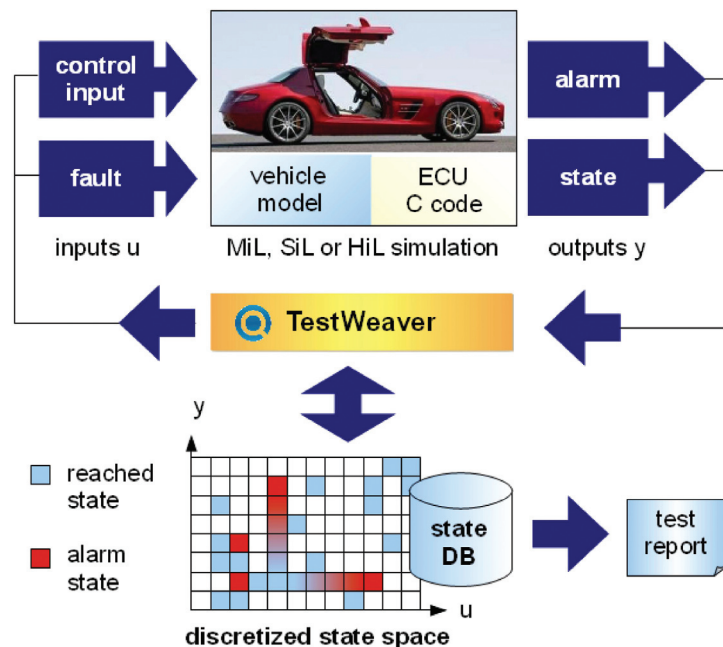
Automated testing of transmission control software is never easy, but there's a product on the market that helps transmission designers and car makers to ensure that no serious bugs or design flaws occur

■ The size of the code running in a typical transmission controller doubles every few years. However, the typical budget spent to test such a controller does not. This puts increasing pressure on transmission makers and OEMs to ensure that no serious bugs or design flaws occur and leak into the final product. Testing a TCU in real life requires the transmission to be exposed to all relevant driving situations repeatedly during the development cycle. Established methods based on either test-driving on the road or driving a HiL simulation using hand-written test scripts do not scale well in this instance, as there are simply too many situations to consider. Checking all of them one by one is very time consuming or not feasible. Consequently, new ideas, methods and tools to achieve a much higher degree of automation are required.

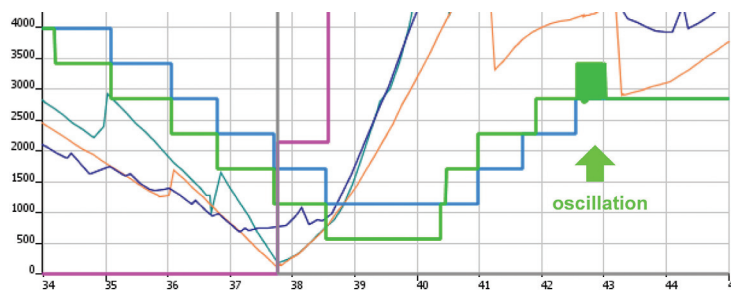
Ten years ago, a Daimler R&D team took up the challenge and started to experiment with intelligent test case generation systems based on what they later called the computer chess principle. Their key idea was simple: chess computers are very powerful and beat virtually every human player, so why not exploit this power to automate system testing? After all, testing and playing chess are similar activities. For example, a chess player searches for a sequence of moves to checkmate his/her opponent in much the same way as a tester searches for a sequence of events to drive the transmission into a state where it fails, therefore violating the specification.

This idea led to the tool known today as TestWeaver. In 2006, the R&D team founded QTronic. Today, transmission developers at Mercedes and AMG use TestWeaver on a regular basis to test all software releases of the 7G-Tronic AT-family as well as various DCTs. Other established companies such as GIF and ZF have integrated TestWeaver into their development processes. The main reason for the success of TestWeaver is its excellent cost/benefit ratio, achieving high-test coverage with little effort for engineers.

With TestWeaver, the test process proceeds as follows: first, build a simulation of the system under test. This can be a MiL, SiL or HiL setup implemented using tools such as MATLAB/Simulink, QTronic's Silver, or a mid-size HiL from dSPACE. In a model-based process, such a setup should be built for test



Left and below: TestWeaver creates, runs and evaluates thousands of driving scenarios in order to find the weak points of a transmission controller unit



purposes anyway. The simulation should be deterministic to be able to reproduce found problems and the process should not run slowly, as it executes up to 10,000 test cases, with there being around 40 seconds for each test case.

Secondly, it's vital to mark the inputs that TestWeaver can control and the outputs that TestWeaver can observe. For a typical TCU, there are 10 inputs, including such aspects as acceleration and brake pedal, shift lever, slope of the road and about 150 outputs to monitor. TestWeaver is shipped with target-specific libraries to mark inputs and outputs. There is a Simulink block set used to instrument Simulink models for use with TestWeaver.

Thirdly, the developer must classify the values of certain outputs on a good/bad scale. This is the part of the test specification that tells TestWeaver what to look for. TestWeaver tries to find driving scenarios, for which these outputs take 'bad' values. For example,

if the engineer uses an output to measure shift duration, and marks long durations as 'bad', TestWeaver will search for long shifts. Likewise, if an engineer outputs friction losses during shift, TestWeaver will look for shifts with high losses.

Fourthly, the engineer needs to start the system test with TestWeaver. It takes about one night to reach a satisfactory test coverage for a typical TCU, therefore finding examples for all up- and down shifts using all kinds of shift programs for all values of engine torques. As a test result, TestWeaver presents a report that lists worst-case scenarios according to the good/bad classification provided by the test engineer. Typically, these are cases where the transmission performs badly or where bugs show up. As experience in transmission development has shown, the reported cases help to focus work on weak points of a design and to optimize the overall quality of the controller. **TTI**