VIRTUALISATION
VALIDATION OF OVER 200 TRANSMISSION VARIANTS ON PC

Transmission control units are often developed and calibrated for dozens or hundreds of vehicle variants. Not every possible variant is physically realised during development. Even the simulation of all variants on a HiL becomes infeasible with growing number of variants, since at least the control unit must then be physically realised and integrated into the HiL system. In such variant-rich cases, simulation of the entire system, including the control unit, becomes particularly attractive. Daimer, QTronic and Vector describe how Mercedes-Benz currently uses virtual ECUs to validate transmission control software for about 200 variants of the Sprinter series in a highly automated way on Windows PC.

AUTHORS

STEFAN GLOSS
is an Engineer at Daimler AG
in Stuttgart (Germany).

MILAN SLEZÁK
is an Engineer at QTronic GmbH
in Stuttgart (Germany).

ANDREAS PATZER
is an Engineer at Vector Informatik
GmbH in Stuttgart (Germany).

VALIDATION OF 200 VARIANTS

Mercedes-Benz Vans of the Sprinter series are available as box wagon, station wagon and pickup, for four- and six-cylinder diesel engines also with seven speed automatic transmission 7G-Tronic Plus. Maximal vehicle payload and transmission ratio of the rear axle can be configured by the customer. This results in more than 200 variants of the transmission control unit VGS3, with corresponding variant-specific calibrations. Since 2010, Mercedes-Benz develops a novel tool chain and process to test and calibrate these variants. Both is presented in this article, ❶. The main idea is to shift development and test tasks from road and HiL test rigs to PC as much as possible. This way, the problematic dependency on expensive and less available resources, such as prototype vehicles and HiL test rigs is reduced. On a technical level, this requires to "virtualise" the transmission system, i.e. to simulate the system on PC.
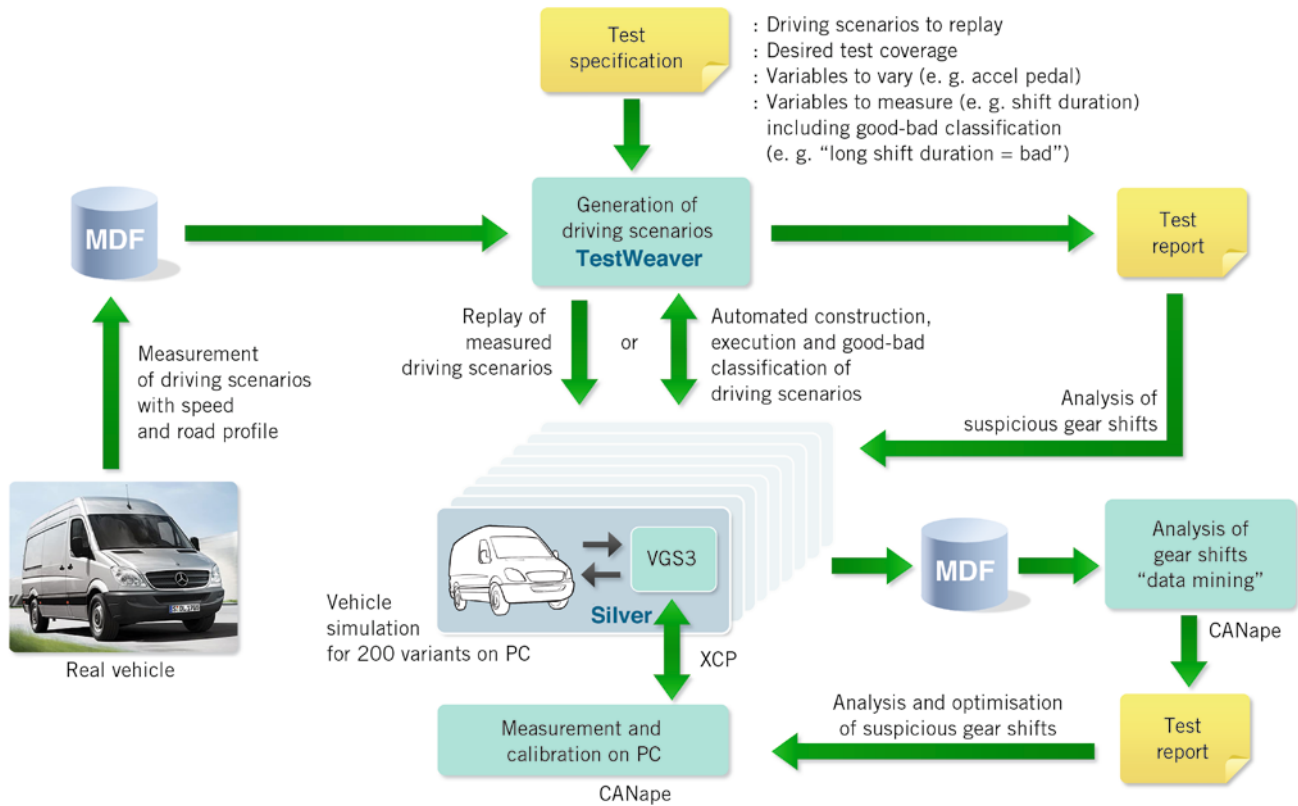
Since a few years, the OEM uses virtual ECUs for the development of its transmission control software, based on the virtual ECU tool Silver [2, 4, 5]. For that reason, high-fidelity simulation models used to develop the control software for 7G-Tronic for passenger cars are readily available for reuse during calibration and test of the Sprinter variants.

For the simulation-based validation of the VGS3, real driving scenarios (city traffic, cross road, autobahn) have been recorded using few physically realised Sprinter variants. For each scenario, vehicle speed and (using GPS) road profile have been recorded and stored as MDF measurement file. Based on these recorded scenarios, validation process proceeds as follows, ❷:

: configure 200 software-in-the-loop (SiL) setups in Silver on several PCs; each setup represents a variant using the original control software and the variant-specific calibration data, including adaptation values generated by a scripted training sequence

: drive all 200 simulation setups using the recorded driving scenarios and record the simulated gear shifts as MDF file; this exploits the multi-core architecture of the employed PCs (Intel Core i7) to parallelise simulation

: evaluate the simulated gear shifts using scripts based on the data mining function of CANape; the scripts have been developed to analyse recorded driving scenarios and are re-used here to analyse simulated scenarios as well. This way, the same report format is available through the entire development process

: generate, run and classify additional driving scenarios using the test case generator TestWeaver, to further increase test coverage.

: analyse suspicious scenarios found by CANape and TestWeaver; for analysis, scenarios are run again. During simulation, CANape measures via XCP on PC into the simulated VGS3, without any bandwidth limit; also calibration (online tuning of parameters) can be done this way.

SIMULATION OF THE TRANSMISSION SYSTEM

For simulation, a vehicle and transmission model created during model-based development of the VGS3 control software has been reused. The model has been developed using Modelica / Dymola [4]. The compiled model was then adapted to the Sprinter variants by changing the corresponding parameter files, which does not require access to the underlying Modelica source. The model of the transmission control unit VGS3 has been created by compiling the C code generated with TargetLink for Window PC. This model has been reused from previous development steps as well. System simulation uses the calibration data and model parameter files valid for specific Sprinter variants. VGS3 uses various functions for automated adaptation of control parameters to the transmission hardware (such as filling times for clutches). Optimal parameter values are learned during operation, and permanently adpated to follow the aging process. This learning procedure is simulated in Silver as well using a training script. A the beginning of a simulation, the parameter values learned this way for a specific variant are ‚flashed' into the simulated TCU, together with the calibration data. This way, the error of the transmission and vehicle model is mostly compensated. In effect, simulation matches the real driving behaviour quite well. The vehicle and transmission model used here is considerable more detailed than the plant model typically found on a HiL system. The use of a more detailed model is enabled here by the full virtualisation of the system, which eliminates all real hardware from the simulation loop. Therefore, simulation is not constrained by real-time any more. Variable step solvers can be used, and simulation might run slower (during gear shifts), or many times faster (during stationary operation) than real time. For measurement, CANape fully compensates these variations of simulation speed and records all signals as function of simulation time, not real time. In summary, the elimination of the real-time constraint allows us to use of more accurate simulation models, which in turn enables us to move the validation of calibration data from test rigs and road to PC.

❶ Simulation-based Validation of Variants

TEST-CASE GENERATOR AUTOMATES SEARCH FOR FAILURES

Testing a transmission system with recorded driving scenarios leads to good test coverage of typical routine situations. However, recorded scenarios tend to be redundant, repeating the same typical gear shifts again and again with little variation, while other more exotic situations, such as certain multiple gear shifts are rarely tested or not covered at all. In order to increase overall test coverage and also to take unforeseen behavior of the driver into account, we have also integrated the test case generator TestWeaver [4, 5] into the validation process. The recorded driving scenarios are then replaced by TestWeaver, who takes over the role of the driver. TestWeaver does however not drive the car in a randomly, but in a systematic way, trying to achieve two goals. On the one hand, TestWeaver attempts to drive the system into states that

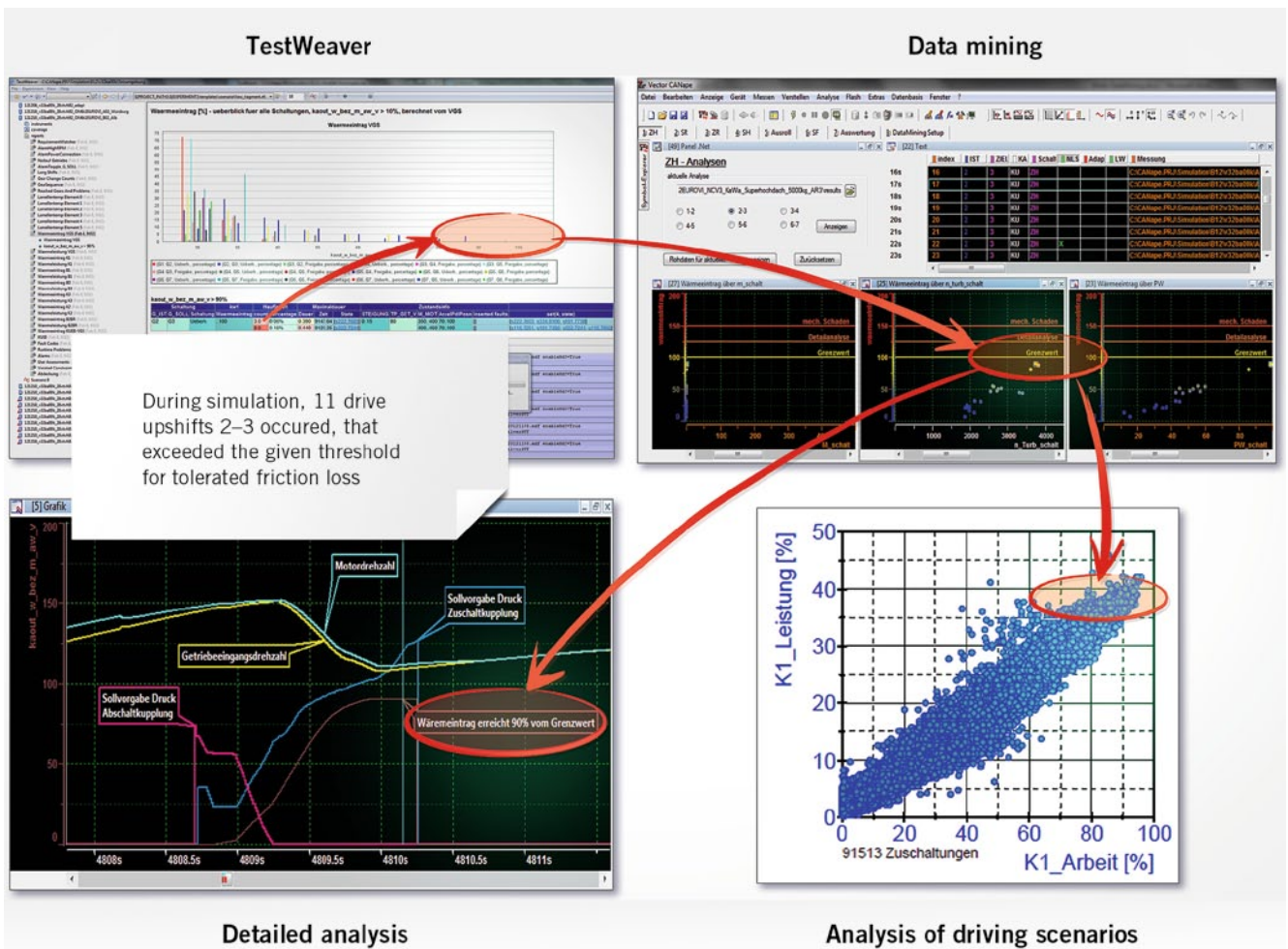With real-time enabled simulation models used on a HiL system, this would typically be impossible to achieve.

violate given system requirements regarding e.g. tolerated jerk, duration of shift, or friction losses during gear shift. On the other hand, TestWeaver attempts to drive the system into many different system states, trying to reach all reachable points in state space. In TestWeaver, the notion of ‚state space' is conveniently defined by just naming the variables (dimensions) that span state space. For example, when choosing just the two TCU variables that represent current gear and target gear, state space becomes exactly the set of all combinations of current gear and target gear, i.e. the set of all gear shifts. Given just this definition, TestWeaver searches autonomously for driving scenarios that reach all gear shifts by autonomously varying given input variables for accel pedal, brake pedal and road profile. When adding engine torque as a third axis, TestWeaver tries in addition to also reach these gear shifts for all different engine loads. This way it is possible to achieve a high test coverage with little effort for test specification and without any manual scripting. This helps to close gaps not covered by testing with a set of recorded driving scenarios.

EVALUATION OF THE TEST CASES

Data mining with CANape and testing with TestWeaver generates report tables and diagrams which both offer means for interactive analysis of detected problem cases. This frequently requires to replay a problem case via simulation on PC. Silver supports this in a fully deterministic way: Driving scenarios can be repeated exactly and as often as needed to analyse a given problem. Simulation can also be paused, e.g. using a breakpoint on signal or even C code level, in order to inspect the system state at any point in time. Of course, debugging on C code level requires access to the source code of the TCU. In combination with CANape, it is also possible to change calibration data during simulation, or to run a simulation with different data sets, in order to explore and validate the corresponding behaviour on system level.

COSTS AND BENEFIT

When testing a software release with the process described above on four PCs

❷ Analysis of suspicious gear shifts with TestWeaver, Silver, and CANape

for seven days (test execution time), we achieve a test throughput of about 600.000 km. Given each PC is equipped with four cores, then each four PC runs four independent vehicle simulations in parallel. With respect to test throughput, a single PC is equivalent to four single ECU HiL systems then. In other words: To test a software release within the same time window and coverage on HiL, one would require 16 HiL systems. In addition, this would require to real-time enable the vehicle simulation model used on PC without sacrificing the required model fidelity.

With the presented process, it takes about two person days to setup the test for a given software release for VGS3. The 200 simulation setups for Silver are generated by a script. The required configuration setups for CANape are highly reusable and require little extra effort. The calibration data for all 200 variants

is contained in a single PAR or HEX file and assigned to the different simulation setups via variant coding in a fully automated way. A similar degree of automation would be hard to achieve by competing approaches, such as HiL or test with real prototypes. Compared to these alternatives, the presented process has a particularly good cost-benefit ratio.

SUMMARY

The presented validation process helped to detect a wide range of problems, ranging from errors in algorithms to sub-optimal calibration data, and provides an excellent cost-benefit ratio. Mercedes-Benz therefore plans to apply the process also to Vans of the Vito and Viano series in the near future. Also calibration of certain TCU functions on PC seems to be doable in the future.

REFERENCES
[1] Junghanns, A.; Serway, R.; Liebezeit, T.; Bonin, M.: Schneller zum virtuellen Steuergerät. In: ATZelektronik 7 (2012), No. 3, pp. 220 – 225
[2] Brückmann, H.; Strenkert, J.; Keller, U.; Wiesner, B; Junghanns, A.: Model-based Development of a Dual-Clutch Transmission using Rapid Prototyping and SiL. International VDI Congress Transmissions in Vehicles 2009, Friedrichshafen, Germany, 30.06.-01-07.2009.
[3] Mauss, J.; Simons, M.: Steuergeräte-Simulation auf PC mittels TriCore-Emulation. In: ATZelektronik 7 (2012), No. 6, pp. 460 – 464
[4] Rink, A.; Chrisofakis, E.; Tatar, M.: Automating Test of Control Software - Method for Automatic Test Generation. In: ATZelektronik 5 (2009), No. 6, pp. 36 – 40
[5] Tatar, M.; Schaich, R.; Breitinger, T.: Automated test of the AMG Speedshift DCT control software. 9th International CTI Symposium Innovative Automotive Transmissions, Berlin, 2010. http://qtronic.de/doc/TestWeaver_CTI_2010_paper.pdf