Automated test of the AMG Speedshift DCT control software

M. Tatar
QTronic GmbH, Berlin

R. Schaich, T. Breitinger
Mercedes-AMG GmbH, Affalterbach

9th International CTI Symposium
Berlin, 2010
Outline

- Motivation
- Principle of the scenario generator
- Test of AMG Speedshift DCT control software
  - the AMG Speedshift DCT
  - the software test setup
- Results and conclusion
Motivation

Ever growing complexity of automotive controllers

How to validate and test?
- do more road tests?
- write more test scripts?

This does not scale well

Code size grows faster

New processes needed

Idea

- increase degree of automation
- generate and evaluate useful test scenarios automatically

- After initial coding you can expect one bug per 20 lines of code
- After thorough unit testing you can expect 1 bug per 1000 lines of code in the final release
  ➢ 1 line ~5 bytes, so 1 bug per ~5KB

<table>
<thead>
<tr>
<th>Application</th>
<th>Code Size</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steering Angle Sensor</td>
<td>32KB</td>
<td>7 Bugs</td>
</tr>
<tr>
<td>Low-end Sensor Cluster</td>
<td>128KB</td>
<td>26 Bugs</td>
</tr>
<tr>
<td>Airbag Controller</td>
<td>256KB</td>
<td>52 Bugs</td>
</tr>
<tr>
<td>EPS Controller</td>
<td>512KB</td>
<td>104 Bugs</td>
</tr>
<tr>
<td>Central Chassis Controller</td>
<td>1.5MB</td>
<td>308 Bugs</td>
</tr>
</tbody>
</table>

source: presentation by Hans Adlkofer, Infineon, 2009
TestWeaver – scenario generator

Idea
- intelligent generation of 1000s of differing test scenarios
- active attempt to:
  - maximize state coverage
  - drive the system in “difficult” situations

Benefit
- high coverage
- low efforts for test specification

Testing = playing against (simulated) system
Strategy for Test Generation

GOALS
- **find bugs:** Change sub-optimal scenarios to generate worst-cases
- **coverage:** Drive the system in states that have not been reached before

- **control**
- **input**
- **fault**

- vehicle model
- ECU C code

**Test Weaver**

- inputs $u$
- outputs $y$

- reached state
- alarm state

- discretized state space

- state DB

- test report
Virtual Integration – Software in the Loop

Developer

Simulink / TargetLink

module .mdl

object .obj

Modelica / Dymola

Interactive drive of the virtual vehicle

DCT control SW

vehicle model

A2L PAR DCM

Silver

Interactive drive of the virtual vehicle
SiL and debugging environment
DCT with mechanical locking differential

- Integrated locking differential
- Transmission
- CRP drive shaft
- Dual clutch
Hydraulic unit of the DCT

Source: Hart et.al., The function development and application of the DCT in the Mercedes-Benz SLS AMG. VDI-Berichte 2081: Getriebe in Fahrzeugen 2010, pp. 599-615
Drivetrain physical simulation model
Drivetrain simulation model – gear actuators
SiL and debugging environment
Setup of the software test

- **Simulink / TargetLink**
  - module .mdl
  - object .obj

- **TestWeaver**
  - drive the virtual vehicle
  - generate worst-case scenarios

- **24 hours**

- **Modelica / Dymola**
  - vehicle model
  - DCT control SW

- **TestWeaver instruments**

- **A2L PAR DCM**
TestWeaver Alarms: correctness / quality

- **Runtime exceptions:**
  division by zero, stack overflow

- **A2L range monitoring:**
  thousands of TCU signals

- **Shift durations:**
  average and maximal durations

- **Clutch overheating, overspeeds:**
  engine overspeed or stalled

- **DCT condition monitoring:**
  > 200 signals, fault codes

- **Oscillations and unexpected control sequences:**
  repeated up/down shifts, bad fault diagnosis, bad fault reaction

- **Code coverage** and system state coverage

---

every problem reported by TestWeaver comes with one or more reproducible examples!
Example: Problem found and corrected

- oscillation of target gear
  - found by TestWeaver
  - replay in Silver

- improved control software
  - run regression test
  - problem solved
Test of DCT control software
- generated and analysed over 3000 different driving scenarios, each 45 sec. for every software release
- systematic test and validation with many usual and many unusual driving conditions

Conclusion
- The presented approach seems extremely well suited for the validation of automotive transmission controllers
- Necessary complement to other QA measures, test benches, prototype driving
- Main benefit:
  - much higher test coverage
  - feasible work effort