Intelligent Test-Case Generation and Automation for Real-Time Test Systems

Mugur Tatar
QTronic GmbH, Berlin

Automotive Testing Expo
ASAM Open Technology Forum
Stuttgart, May 17., 2011
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 cases automatically
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
TestWeaver - Test Generation Strategy

- Change sub-optimal scenarios to generate worst-cases
- Drive the system in states that were not covered before

TestWeaver Strategy

![Diagram of TestWeaver strategy](image)

- Control input
- Component fault
- Vehicle model
- ECU C code
- MiL/SiL/HiL simulation
- State
- Alarm
- Outputs y
- Reached state
- Alarm state
- Discrete state space
- State DB

*Image of a car and other relevant components.*
TestWeaver - Test Generation Strategy

TestWeaver - Test Generation Strategy
TestWeaver instruments: MATLAB/Simulink
TestWeaver: result analysis

Overview report for all scenarios

Detailed reports for individual scenarios

Replay, plot, debug
Example: Integer range violations

Monitoring the ranges of 6538 signals after time 1s

14 out of range signals -- names not containing "_L2":

<table>
<thead>
<tr>
<th>Name</th>
<th>A2L Min</th>
<th>A2L Max</th>
<th>Sim Min</th>
<th>Sim Max</th>
<th>Scen Min</th>
<th>Time Min</th>
<th>Scen Max</th>
<th>Time Max</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>accVehX_VSC</td>
<td>-10.24</td>
<td>10.16</td>
<td>-12</td>
<td>14</td>
<td>s538</td>
<td>27.515</td>
<td>s209</td>
<td>17.885</td>
<td>127.451</td>
</tr>
<tr>
<td>trqClPrev_CctcVSWA</td>
<td>-500</td>
<td>1000</td>
<td>-500</td>
<td>1145.44</td>
<td>s0</td>
<td>1.005</td>
<td>s1186</td>
<td>28.755</td>
<td>109.696</td>
</tr>
<tr>
<td>EngRPM_Max_Rq_DCT_OcptVUW</td>
<td>0</td>
<td>8190</td>
<td>8192</td>
<td>8192</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>EngRPM_Max_Rq_DCTcp_OcptVUW</td>
<td>122880</td>
<td>131070</td>
<td>122878</td>
<td>122878</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>EngRPM_Rq_TCM_OcptVUW</td>
<td>0</td>
<td>8190</td>
<td>8192</td>
<td>8192</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>EngRPM_Rq_TCMcp_OcptVUW</td>
<td>122880</td>
<td>131070</td>
<td>122878</td>
<td>122878</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>intrvtnMd_TCM_OcptVUC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>6.555</td>
<td>200</td>
</tr>
<tr>
<td>intrvtnMd_TCMcp_OcptVUC</td>
<td>254</td>
<td>255</td>
<td>252</td>
<td>254</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>200</td>
</tr>
<tr>
<td>TxNShiftMd_OcptVUC</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>TxNShiftMdCp_OcptVUC</td>
<td>254</td>
<td>255</td>
<td>252</td>
<td>252</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>TxShlRcmd_DispRqTCmcp_OcptVUC</td>
<td>0</td>
<td>252</td>
<td>254</td>
<td>253</td>
<td>s0</td>
<td>6.875</td>
<td>s0</td>
<td>1.005</td>
<td>0.396825</td>
</tr>
<tr>
<td>rpmTqRefNom_SccpVSW</td>
<td>0</td>
<td>9999</td>
<td>0</td>
<td>10129</td>
<td>s0</td>
<td>1.005</td>
<td>s292</td>
<td>7.715</td>
<td>101.3</td>
</tr>
<tr>
<td>prctAccpUphp_SdgpVUW</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>107.313</td>
<td>s0</td>
<td>1.005</td>
<td>s205</td>
<td>35.49</td>
<td>107.313</td>
</tr>
<tr>
<td>trqBrkEsp_RcesVUW</td>
<td>0</td>
<td>12285</td>
<td>0</td>
<td>59151</td>
<td>s0</td>
<td>3.305</td>
<td>s0</td>
<td>1.005</td>
<td>481.49</td>
</tr>
</tbody>
</table>

3 out of range signals -- names containing "_L2":

<table>
<thead>
<tr>
<th>Name</th>
<th>A2L Min</th>
<th>A2L Max</th>
<th>Sim Min</th>
<th>Sim Max</th>
<th>Scen Min</th>
<th>Time Min</th>
<th>Scen Max</th>
<th>Time Max</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>intrvtnMd_TCM_L2siVUC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>6.555</td>
<td>200</td>
</tr>
<tr>
<td>TxNShiftMd_L2siVUC</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>s0</td>
<td>1.005</td>
<td>s0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>TxShlRcmd_DispRqTCmcp_L2siVUC</td>
<td>0</td>
<td>252</td>
<td>254</td>
<td>255</td>
<td>s0</td>
<td>6.875</td>
<td>s0</td>
<td>1.005</td>
<td>0.396825</td>
</tr>
</tbody>
</table>

Example: Problem found and corrected

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

- improved control software
  - run regression test
  - problem solved

Example: Debugging a found problem
TestWeaver: What can be found

Control software
- division by zero, access violation
- integer or index out of range (A2L)
- oscillating signals
- overheating of clutches
- wrong state estimation
- 'unreachable' state reached
- bad health mgmt. (e.g. wrong diagnosis)
- bad function quality (e.g. long shifts)

Vehicle simulation
- modeling problems, solver failure

every problem reported by TestWeaver comes with one or more reproducible examples!

Coverage
- which states (gears, torques, etc.) reached
- code coverage (which code parts are reached or not, CTC++)
TestWeaver for dSPACE HiL

host PC

generated test cases run here
accurate timing on a milli sec scale

Python test

generate test
run each test
and record result

r/w fault codes
r/w EEPROM (adapt. data)
measure
calibrate

CAN, XCP, UDS

record test results

TestWeaver for HiL

Experiment Specification

Test Report

recorded test results

CANape
- measure during replay

Silver
- emulate dSPACE API
- no dSPACE products required
- Simulation model
  and ECU code runs in Silver

Python test

generate test
run each test
and record result

r/w fault codes
r/w EEPROM (adapt. data)
measure
calibrate

CAN, XCP, UDS

record test results

TestWeaver for HiL

Experiment Specification

Test Report

recorded test results

CANape
- measure during replay

Silver
- emulate dSPACE API
- no dSPACE products required
- Simulation model
  and ECU code runs in Silver
TestWeaver for dSPACE HiL - detailed

**host PC**
- send instruments
- send test
- reset simulation
- run simulation
- retrieve results incrementally

**dSPACE API**
- TestWeaver for HiL
- e.g. ds1006
dSPACE Simulator

**TestWeaver on HiL**
- instruments.py
- hilConfig.py
- silConfig.py

**Real Time Testing Lib**
- version 1.2 to 1.7, Python
- part of AutomationDesk 3.0

**Real-time-capable**
- measure during replay
- on a milli sec scale

**Simulation model**
- test cases are
  - sequence of actions over time
  - reactive: may depend on time and thresholds
  - executed on the HiL board, not the host PC

**dSPACE API**
- ds1006
- example
- Python tests

**ControlDesk or PROVEtech**
- to display test (replay) results

**RT-Proxy**
- run for each test
- and record result

**CAN, XCP, UDS**
- measure
- calibrate

**CANape**
- - measure during replay

**CANape**
- - measure during replay

**Silver**
- emulate dSPACE API
- no dSPACE products required
- Simulation model
- and ECU code runs in Silver

**Python tests**
- instruments.py
- hilConfig.py
- silConfig.py

**recorded test results**
- use

**TestWeaver for HiL**
- run for each test
- and record result

**ControlDesk** or PROVEtech
- to display test (replay) results

**Real Time Testing Lib**
- version 1.2 to 1.7, Python
- part of AutomationDesk 3.0

**RT-Proxy**
- run for each test
- and record result

**Python tests**
- instruments.py
- hilConfig.py
- silConfig.py

**recorded test results**
- use

**TestWeaver for HiL**
- run for each test
- and record result

**ControlDesk** or PROVEtech
- to display test (replay) results

**Real Time Testing Lib**
- version 1.2 to 1.7, Python
- part of AutomationDesk 3.0

**RT-Proxy**
- run for each test
- and record result

**Python tests**
- instruments.py
- hilConfig.py
- silConfig.py

**recorded test results**
- use
AMG SPEEDSHIFT MCT
7-speed sports transmission

- released April 2008
- control software test with Silver and TestWeaver
- every software release: 24h test run in parallel on several PCs
- thousands of gearshifts generated and analyzed

SPEEDSHIFT MCT for SL63 AMG

details appeared in:
ATZelektronik, 6/2009
in German and English
Silver & TestWeaver at Mercedes-Benz

**Dual Clutch Transmission of Mercedes-Benz**

- control software test with Silver and TestWeaver
- every software release: 24h test run in parallel on several PCs
- thousands of gearshifts generated and analyzed
- code coverage measured using CTC++ and reported using TestWeaver

**details in:**

Model-based Development of a Dual-Clutch Transmission using Rapid Prototyping and SiL
TestWeaver for Heavy Truck Brake System

Brake Blending Function
- 2008 for Haldex
- together with Modelon AB
- test of brake system software with TestWeaver
- cosimulation Dymola and Simulink with Silver
- thousands of drive maneuvers generated and classified

details in:
9th International Symposium on Advanced Vehicle Control (AVEC2008), Kobe, Japan, 6. - 9.10.2008
TestWeaver for Dual-Clutch Transmission

DCT development
- 2009 by GIF, Alsdorf
- test of TCU control software with TestWeaver
- integrated into the Simulink development cycle
- database with 20,000 test cases generated and used for regression tests

details in:
Crosswind stabilisation

- function exported from Simulink
- co-simulated with Mercedes inhouse vehicle model, wind and road model using Silver
- test of the stabilisation function with TestWeaver
- generated and analyzed 100,000 different driving scenarios, each 45 sec. within 3 weeks

details in:
IFAC Symposium Advances in Automotive Control 2010, 12-14 July 2010, Munich, Germany
DCT Speedshift for AMG SLS

- control input
- fault
- vehicle model
- ECU code
- software-in-the-loop
- outputs y
- alarm
- state

TestWeaver

AMG DCT SPEEDSHIFT
7-speed sports transmission
- control software tested with Silver and TestWeaver
- every software release: 24h test run in parallel on several PCs
- thousands of gearshifts generated and analyzed

details in:
Automated test

- generate useful test cases
- evaluate on HiL, SiL, or MiL
- report results

Benefit

- low effort
- high coverage
- find more problems earlier

cost and time savings

- $ millions

Explored system states