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

Mugur Tatar
QTronic GmbH, Berlin

Automotive Testing Expo Europe
ASAM Open Technology Forum
Stuttgart, June 12, 2012
Challenge: System Test and Validation

- Size of control software doubles every few years
- Budget for testing does not

![Graph showing the growth of code lines over time with various software and control systems](image-url)
Manually written test scripts

Test #31
start_car();
shift_lever = D;
while(gear<2)
    accelPedal = 20;
if(time>2)
    error("no shift")
...

Problems

- **High costs** for development and maintenance of scripts
- **Low coverage** of tests
**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
TestWeaver - Test Generation Strategy

- **Controllable input**
- **Component fault**

**Vehicle model**

**ECU controller**

**Requirement monitoring**

**MiL/SiL/HiL simulation**

**TestWeaver**

**State**

**Alarm**

**Inputs u**

**Outputs y**

**State space**

- Reached state
- Alarm state

**Test report**

- Reactive scenario generation
- Each scenario depends on history of generated scenarios
- All cases can be reproduced

Change sub-optimal scenarios to generate worst-cases

Drive the system in states that were not covered before
TestWeaver - Test Generation Strategy

Root

- Input a
- Fault 2
- Fault 1
- OK

Input a

Input b

Input c

Alarm!

Violation of defined requirement

Time

T0
TestWeaver with MATLAB/Simulink
TestWeaver - Result Analysis

Overview report for all scenarios

- Detailed reports for individual scenarios
- Replay, plot, debug
TestWeaver - Monitored requirements

- **SW problems:**
  division by zero, access violations...

- **Range monitoring (A2L signals):**
  thousands of ECU signals

- **Component overheating, overspeeds:**
  engine overspeed or stalled, ...

- **ECU condition monitoring:**
  > 100 signals, fault codes

- **Oscillations, illegal control sequences:**
  repeated up/down shifts
  violation of user-specified conditions

- **Code coverage and system state coverage**
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>0</td>
<td>27.515</td>
<td>0</td>
<td>28.755</td>
<td>109.696</td>
</tr>
<tr>
<td>trqCIPrev_CctcVSWA</td>
<td>-500</td>
<td>1000</td>
<td>-500</td>
<td>1145.44</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.186</td>
<td>28.755</td>
</tr>
<tr>
<td>EngRPM_Mx_Rq_DCT_OcptVUW</td>
<td>0</td>
<td>8190</td>
<td>8192</td>
<td>8192</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>EngRPM_Mx_Rq_DCTp_OcptVUW</td>
<td>122880</td>
<td>131070</td>
<td>122878</td>
<td>122878</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>EngRPM_Rq_TCMp_OcptVUW</td>
<td>0</td>
<td>8190</td>
<td>8192</td>
<td>8192</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>EngRPM_Rq_TCMp_OcptVUW</td>
<td>122880</td>
<td>131070</td>
<td>122878</td>
<td>122878</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>intrvlMd_TCM_OcptVUC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>6.555</td>
</tr>
<tr>
<td>intrvlMd_TCMp_OcptVUC</td>
<td>254</td>
<td>255</td>
<td>252</td>
<td>254</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>200</td>
</tr>
<tr>
<td>TxDnShiftMd_OcptVUC</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>TxDnShiftMd_Cp_OcptVUC</td>
<td>254</td>
<td>255</td>
<td>252</td>
<td>252</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
</tr>
<tr>
<td>TxShlRcmd_DispRqTCMp_OcptVUC</td>
<td>0</td>
<td>252</td>
<td>254</td>
<td>253</td>
<td>0</td>
<td>6.875</td>
<td>0</td>
<td>7.675</td>
<td>0.396825</td>
</tr>
<tr>
<td>rpmTqTrRefNom_ScppVSW</td>
<td>0</td>
<td>9999</td>
<td>0</td>
<td>10129</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>7.715</td>
</tr>
<tr>
<td>prctAccpUpsh_SdgpVUW</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>107.313</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>1.005</td>
<td>35.49</td>
</tr>
<tr>
<td>trqBrkEsp_RcesVUW</td>
<td>0</td>
<td>12285</td>
<td>0</td>
<td>59151</td>
<td>0</td>
<td>3.305</td>
<td>0</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>intrvlMd_TCM_L2siVUC</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1.005</td>
<td>0</td>
<td>6.555</td>
<td>200</td>
</tr>
<tr>
<td>TxDnShiftMd_L2siVUC</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>1.005</td>
<td>0</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>0</td>
<td>6.875</td>
<td>0</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

source: Schaich, Breitinger, Tatar,
Automated test of the AMG DCT
Speedshift control software.
9th International CTI Symposium
Innovative Automotive Transmissions,
TestWeaver - Supported Platforms

- **TestWeaver runs on a Windows PC**
  supported simulation platforms:

- **Simulink (MiL)**
  Simulink instrumentation library (blockset)

- **Dymola, SimulationX (MiL)**
  Modelica instrumentation library

- **Silver virtual ECUs (SiL)**
  Python instrumentation library

- **dSPACE, NI, ETAS (HiL)**
  Python instrumentation library

- **other (MiL/SiL, e.g. ZF SoftCar)**
  C/C++ instrumentation library
TestWeaver for HiL Platforms

- **TestWeaver runs on the host PC under Windows**  
  SiL and HiL use the same instrumentation syntax in Python supported platforms:

- **dSPACE**  
  using dSPACE Real-Time Testing

- **National Instruments**  
  using NI VeriStand

- **ETAS**  
  using LABCAR OPERATOR
TestWeaver for dSPACE HiL

generated test cases run in real-time
accurate timing on a milli sec scale

host PC

Python test

generate test run each test and record result

TestWeaver for HiL

recorded test results

Experiment Specification

Test Report

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

CAN, XCP, UDS

CANape
- measure during replay

Systematic Test-Case Generation for Real-Time Test Systems - ASAM Open Technology Forum Europe 2012
DCT Speedshift for AMG SLS

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:
Mercedes Crosswind Stabilisation Function

Crosswind Stabilisation

• Function exported from Simulink
• Co-simulated with Mercedes in-house vehicle model, wind and road model using Silver
• Test of the stabilisation function with TestWeaver
• Generated and analysed 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
ZF integrated TestWeaver into the development and test process

- virtual integration and test of control software for hybrid powertrain using ZF Softcar and TestWeaver.
- modeling of given system requirements (DOORS) as system invariants. Validation using TestWeaver. For validation, thousands of driving scenarios are systematically generated and analyzed, all system requirements are permanently supervised.

ZF’s 8-Speed Automatic Transmission

details in:
Martin Neumann, Mugur Tatar:
Absicherung von Steuerungssoftware für Hybridsysteme 5. Fachtagung AUTOREG 2011
Systematic Test and Validation

- Intelligent generation of thousands of test scenarios
- High coverage of system states
- Reactive test execution using SiL, MiL or HiL
- Tests: requirements, safety, robustness, regression

Benefit

- Find more problems earlier

<table>
<thead>
<tr>
<th>cost and time savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ millions</td>
</tr>
</tbody>
</table>