CVT control software testing

Lionel Belmon, Liu Fei – Global Crown Technology, Beijing
Zeng Weihua - Jianglu Rongda, Changsha
议程

1. Motivation
2. Vehicle plant modeling
3. TCU embedded software
4. Simulation validation
5. TCU Testing
Motivation
Develop and test TCU control software

High quality vehicle model
Realistic simulation

Interaction road - vehicle - hydraulics - TCU

Cover large number of test cases
Vehicle plant model
Vehicle model overview

Platform: SimulationX

Driver

CAN

Hydraulics

Engine/EMS

Torque converter

RND clutch

Variator Belt

Longitudinal Vehicle dynamics with left/right tyre slip

GlobalCrown Technology (BeiJing) Co., Ltd.
Hydraulic module

- Torque converter lock-up
- RND clutches

Extraction of CAD parameters

Main flow pressure dynamics

Spool valves characteristics

Solenoid valve - pilot pressures

Solenoid pilot valves

Primary / Secondary pistons for variator
Hydraulic Unit tests and validation

Adjustment of solenoid characteristics

Primary cylinder pressure vs. Solenoid current
Variator belt model concept

Variator ratio change by controlling primary/secondary pressure

\[
\frac{di}{dt} = K_i n_{pri} (F_{pri} - F^*_{pri})
\]

Actual force

steady-state force required

\[
P_{secmx} = \frac{T_{pri} \beta \cos \alpha}{2 \mu_{sec} R_{pri} A_{sec}}
\]

Max torque limit depends on pressure!
Real-time simulation

Adjustment of hydraulic volumes and mechanical stiffness

Fixed-step solver – 0.5 ms

~80 ODE state variables

~200 inputs/outputs variables (CAN, sensors, actuators)

Code generation from SimulationX to FMU cosimulation

Execution in QTronic Silver

Simulated time: 98.38s; Actual time spent computing: 18.62s; Speedup factor: 5.28
TCU embedded software
TCU software generation process

1. Functional requirements
2. ETAS ASCET model
3. Embedded c-code
4. A2L and calibration data
5. ETAS INCA
6. ECU hardware
Virtual TCU with Silver

Functional requirements

ETAS ASCET model

Embedded c-code

A2L and calibration data

Silver virtual ECU

ETAS INCA
Virtual TCU platform

Vehicle model
SimulationX

Configurable GUI

Attach to Process
Debug
Microsoft Visual Studio

Python scripts
Glue
Tests

Read/Write
MDF CSV

ETAS ASCET CVT Control Model

Virtual ECU
PAR
DCM
Task scheduler
DBC
CAN

CVT Control Software:
Tasks
Fixed-point C code

Code generation
Validation of vehicle simulation
Test cases for simulation validation

~20 test drive cases to test various model characteristics:

10% pedal acceleration – coasting
30% pedal acceleration – coasting
…

Lever position changes: N-R-N-D-N-R-P-R-N…

Heavy braking with ABS

Tip-in, Tip-off

…
Example of test drive validation

- Engine speed: measurement - simulation
- Car speed: measurement - simulation
- Turbine speed: measurement - simulation

Hydraulic primary – secondary pressures
TCU embedded software testing
Test requirements

Usual test drive cases
NEDC, 100km/h acceleration…

Calibration parameters
Software functions activation/deactivation

Various environment conditions :
- slope, snow/water…

Fault monitoring and reaction
Fault insertion

- Solenoid and hyd faults
- Wheel sensor faults
- Bus faults
- Temperature sensors faults
- Speed sensors faults
Fault insertion – how it works

Example on solenoid valve fault

TCU application software

Target current

Feedback current

Vehicle model

Fault flag

TCU Basic software
Fault insertion – how it works

Example on solenoid valve fault

TCU
application software

Current target

Python

Vehicle model

Fault flag from BS

TCU BS emulation
python

Fault mode selection
Short to GND
Short to UBat
Open line
Fault insertion – how it works

Example on solenoid valve fault

```python
fault_sdi_tcs1_BOB = Chooser(
    # tw must be a WeaverConnection instance
    tw,
    # Sil/Hil Variable instance, or None if none
    Variable('fault_sdi_tcs1_BOB'),
    # instrument name shown by TestWeaver; if None use Variable name
    'fault_sdi_tcs1_BOB',
    # instrument unit shown by TestWeaver
    '',
    # instrument description shown by TestWeaver
    '',
    # partition definition
    # min and max, or value; name of partition; optional: severity/occurrence rate, comment
    [[-2, 'hyd0', Occurrence.FaultLow, 'hyd0'],
     [-1, 'hyd1', Occurrence.FaultLow, 'hyd1'],
     [0, 'no', Occurrence.OK, '0'],
     [2**1, 'SG', Occurrence.FaultLow, 'bit1'],
     [2**2, 'SB', Occurrence.FaultLow, 'bit2'],
     [2**3, 'OP', Occurrence.FaultLow, 'bit3'],
     [2**6, 'UBATT', Occurrence.FaultLow, 'bit6']],
    # type of the instrument: Default, Real, Integer, String
    Type.Integer)
```
Use of QTronic TestWeaver

- Formal requirements monitoring
- Hand written test scripts
- Automatic test cases generation
- Automatic test report generation
ABS simulation example

Clutch locked + Strong braking → Wheel block → **strong engine deceleration** → Transmission torque exceeded → **belt Slip !**

→ Belt slips are reported automatically in TestWeaver reports
Test case generation

QTronic TestWeaver automatically generates drive sequences
Example of Belt Slip found by TestWeaver

Graphs showing:
- Vehicle speed
- Wheel speed
- Belt slip
- High traction force
- Wheel slip
- Low mu
- high mu
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>362</td>
<td>fve_StandStill__F01_15_STANDSTILL_IMPL_p01_15_StandStill = jvr__F01_15_STANDSTILL_IMPL_p01_15_StandStill &gt;= _jvru_StandStill_buff &amp; &amp; jvrs__F01_15_STANDSTILL_IMPL_p01_15_StandStill &gt;= _jvru_StandStill_buff &amp; &amp; vve__F01_15_STANDSTILL_IMPL_p01_15_StandStill &lt; _vveu_StandStill_C &amp; &amp; hap__F01_15_STANDSTILL_IMPL_p01_15_StandStill &lt; _hapu_StandStill_C &amp; &amp; npm__F01_15_STANDSTILL_IMPL_p01_15_StandStill &lt; _npmu_StandStill_C &amp; &amp; _fbp_StandStill_C</td>
</tr>
<tr>
<td>363</td>
<td></td>
</tr>
<tr>
<td>364</td>
<td></td>
</tr>
<tr>
<td>365</td>
<td></td>
</tr>
<tr>
<td>366</td>
<td></td>
</tr>
<tr>
<td>367</td>
<td></td>
</tr>
</tbody>
</table>

```
1: T && T && T && T && T && T
2: T && T && T && T && T && F
3: T && T && T && T && T && F && _
4: T && T && T && T && F && _ && _
5: T && T && F && _ && _ && _ && _
6: T && F && _ && _ && _ && _ && _
7: F && _ && _ && _ && _ && _ && _
```
Conclusion
Vehicle model + TCU + calibration:

validated simulation, reproduce very well the actual execution of TCU in the real vehicle

Fault insertion and validation of TCU fault management

Automatic **generation** and **evaluation** of test cases
TestWeaver + requirements modeling

Very efficient support for TCU development and testing!
谢谢！
Thank you！