ANALYSIS OF AUTOMATIC TRANSMISSION CONTROL UNITS USING SELF LEARNING SYSTEM

Balázs SCHERER
Advisor: Gábor HORVÁTH

I. Introduction

In the automotive industry it is extremely important, to produce high quality, low failure rate electronic devices. Subsystem providers like Robert Bosch GmbH have to ensure the trustiness of their products for 5 to 10 years in a very harsh environment. The software development of Bosch automatic transmission control ECU-s (Electronic Control Units) is done in Budapest at the GS-TC/ENC-Bp department. To satisfy the above reliability goals this department employs more test engineers than software engineer. The goal of this PhD. work is to find a solution to improve the software quality of these TCUs (Transmission Control Unit) [1] and to achieve cost reduction in the test process by using a self learning test system.

II. Traditional test methods and phases

Traditionally there are two types of test processes at GS-TC/ENC-Bp, the so called regression test, which is done before every delivery to a car manufacturer, and the so called integration test performed after every modification of the software. Each of these test processes has the following main sub processes: I/O, CAN, Diag, Comm-diag, OMM, ROS.

The I/O test is responsible for testing the power supply and the analog and digital interfaces of the TCU. A general TCU has about 60 such I/O lines. Test measurements made on the power supplies of the sensor and actuator units, the proper working of the solenoids and ignition controls are also investigated.

The CAN test contains measurements for the CAN (Controller Area Network) [2] communication of the TCU. This test step checks the presence and periodicity of CAN messages (there are about 10 CAN messages containing more than 20 important signals), and whether the system is able to notice the absence of these messages.

Diag test deals with the so called filtering and de-filtering of different errors, this test checks whether the system is able to notice and store errors.

Comm-diag subprocess provides test steps for the diagnostic communication. In modern cars it is possible to read out some performance and malfunction related notes from the ECUs in a repair station using a simple diagnostic device. This step ensures that every diagnostic function (like KWP2000 [3] commands) works properly.

The OMM (Operating Mode Management) test step is responsible for checking the transition between the main operating states of the TCU, like initialization, drive, limp-home, shut-down etc.

The ROS (Realtime Operating System) tests the behavior of the RTOS by checking the stack usage, the periodicity of tasks and so on.

Each of the tests above is done in nearly the same environment. This environment contains a TCU, a so called Laborauto, which simulates the behavior of the car's other ECUs, and an ETAS INCA interface and software. The most important part of this set-up is the ETAS INCA [4] software and hardware interface. Briefly this tool provides a dual port RAM like interface to the TCU, and by using this interface every important internal variables of the TCU software can be examined on the fly. Most of the above tests are done by checking, whether these internal variables contain the right values.

III. Goal and application space of a self learning based test system

As the previous chapter has shown there are extremely complex test processes at GS-TC/ENC-Bp, and many of these processes are under automation. However these tests are rather static ones. Currently there isn't any drive cycle based test analyzing the complete system`s behavior, simply because in this environment it is impossible to carry out such a test, due to the numerous variables and signals.

The goal of the self learning system is to fill this hole and provide a dynamic drive cycle based test.

IV. Difficulties of test system set-up

As the first step to create such a self learning system the base signal and variable set of the knowledgebase should be identified. As it was mentioned, traditional tests are done using INCA based monitoring of the states of internal variables, but unfortunately the set of these variables and their mapping are highly TCU version dependent. At GS-TC/ENC-Bp TCUs are developed for more than 5 car manufacturers, and there are several TCU versions for each manufacturer, therefore it is impossible to create a self learning system for such huge and changing knowledgebase. So as a decision INCA based diagnosis won't be used, and the TCU will be considered as a black box.

The next step was to separate a base of properties and variables that are identical to every TCU. The signals transmitted through the CAN communication seemed to be the best choice for this base variable set. However, the CAN matrix (the message IDs, and signal encoding) differs for each product line, but the set of signals are nearly the same for every TCU (like engine RPM and momentum, wheel speeds, accelerator pedal states, etc). This base set could be complemented with the diagnostic communications like KWP2000 protocol, or OBD [5] (On Board Diagnosis). These diagnosis protocols can complement the set of the variables provided by the CAN communication with information about the system states, and error codes, in a standard TCU independent way. Fortunately this black box configuration has very low hardware and software resource requirement and a test set-up can be realized in the Bosch Embedded System Laboratory at DMIS BUTE.

V. Conclusions, problems to solve

However, this set-up is promising and the knowledge base of the self learning systems can be collected, but there are two main problems to solve in the future. First of all there are no definitions for the correct or incorrect working of a TCU. Secondly there are many TCU versions, but even the same TCU version could have different behavior based on its so called calibration. And unfortunately this calibration can be changed even on the fly.

References

[1] The Bosch Yellow Jackets: *“Electronic Transmission Control ETC”* 2004.

[2] Robert Bosch GmbH: *“CAN Specification Version 2.0”* 1991.

[3] ISO 14230-2:1999. Road vehicles -- Diagnostic systems -- Keyword Protocol Part 1-4

[4] ETAS, INCA homepage: http://www.etas.com/en/products/inca_software_products.php

[5] On Board Diagnosis: OBD II, home page: http://www.obdii.com/