In these days, automobile control systems are rapidly increasing in size and being complicated due to various market needs concerning environmental regulations, advanced safety, information communication and other needs. Under this circumstance, simulation technology is widely used for efficient development and designing of electronic control units for automobile (hereinafter, referred to as ECU).

We at FUJITSU TEN also heavily use our original simulator "CRAMAS (Computer Aided Multi-Analysis System)" as HILS (Hardware in the Loop Simulation).

While the HILS is used at a downstream step in ECU development process, the use of SILS (Software in the Loop Simulation) that enables the evaluation of control software before the completion of ECU is now increasing. This paper introduces our newly-developed framework "CRAMAS-VF" that enables creation, within a short period, of the environment for SILS suitable for efficient and practical development of control software.

Abstract

In these days, automobile control systems are rapidly increasing in size and being complicated due to various market needs concerning environmental regulations, advanced safety, information communication and other needs. Under this circumstance, simulation technology is widely used for efficient development and designing of electronic control units for automobile (hereinafter, referred to as ECU).

We at FUJITSU TEN also heavily use our original simulator "CRAMAS (Computer Aided Multi-Analysis System)" as HILS (Hardware in the Loop Simulation).

While the HILS is used at a downstream step in ECU development process, the use of SILS (Software in the Loop Simulation) that enables the evaluation of control software before the completion of ECU is now increasing. This paper introduces our newly-developed framework "CRAMAS-VF" that enables creation, within a short period, of the environment for SILS suitable for efficient and practical development of control software.
Introduction

These days, automobile control systems are rapidly increasing in size and becoming more complicated due to various market needs concerning environmental regulations, advanced safety, information communication and other needs. On the other hand, the reduction of development costs and the shortening of developing period are also in demand. We at FUJITSU TEN widely use our original simulator "CRAMAS (Computer Aided Multi-Analysis System)" (Fig. 1) for efficient development and designing of electronic control units for automobile (hereinafter, referred to as ECU). CRAMAS is classified as a simulator that provides closed loop simulation with the use of HILS (Hardware In the Loop Simulation); hardware currently being used widely.

The greatest advantage of CRAMAS is that it can simulate, in real time, behavior of an engine and a vehicle with the use of mathematized programs (hereinafter, referred to as plant models). CRAMAS allows the development and evaluation of an ECU even under the environment with no functional vehicle parts, just by using the target ECU connected to CRAMAS. Because of such contribution to efficient development, CRAMAS is used in various fields including engine control, transmission control, hybrid power train control, integrity control, and advanced safety relevant to millimeter wave radars. CRAMAS also contributes to the shortening of evaluation period because CRAMAS can carry out automated tests (e.g. midnight unmanned driving), and surely carries out test issues that are difficult to generate in reality, such as a test involving danger in actual vehicle and fail-safe failure verification.

Expansion of Model-Based Development

In order to further shorten the development period, model-based development has been adopted, wherein the contents for control are expressed in a model for front-loading from an algorithm development stage.
Moreover, for the efficient and practical development of the control logic, control software has to be evaluated as a whole (application plus middleware for control). Therefore, interrupt processing must be simulated at the timing equivalent to microcomputer clock ([ns] order) so as to be able to properly simulate the middleware. Here, another problem is that this slows down the simulation speed.

This paper introduces CRAMAS-VF, newly-developed framework, which is free from these problems, and is capable of creating the MILS/SILS environment suitable for the model-based development.

3. Overview of CRAMAS-VF

Fig. 4 shows the whole structure of simulation environment for MILS with "CRAMAS-VF."

CRAMAS-VF is operable on a computer with Windows OS, and thus a user can create the whole environment with CRAMAS-VF in one personal computer. Table 1 shows CRAMAS-VF operating environment.

Table 1 CRAMAS-VF operation environment

<table>
<thead>
<tr>
<th>Item</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>WindowsXP Professional SP3 32bit</td>
</tr>
<tr>
<td></td>
<td>Windows7 Professional SP1 32bit</td>
</tr>
<tr>
<td></td>
<td>Windows7 Professional SP1 64bit</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel®Core™2 Duo 2.0GHz or more</td>
</tr>
<tr>
<td>Memory</td>
<td>2GB or more</td>
</tr>
<tr>
<td>HDD</td>
<td>5GB or more</td>
</tr>
</tbody>
</table>

As shown in Fig. 4, CRAMAS-VF uses a modeling tool, MATLAB®/Simulink® of MathWorks Inc., as a user interface. A vehicle model and control software are connected to Simulink to construct the simulation environment.

A user can easily assemble a whole model on Simulink just by drawing lines between the functional blocks provided in a row. Thus, Simulink is a de facto standard modeling tool for control development. In addition, Simulink can also link with the modeling tools created by other companies.

The connection between CRAMAS-VF and CRAMAS GUI application (hereinafter, referred to as CRAMAS application) allows the entry of the data concerning accelerator operation or other data into the simulator, the monitoring of the simulator, and the analysis of measured data. The same application used in HILS environment is used on CRAMAS, which allows contents (measurement data file, entry data file, etc.) to be made common. That is, evaluation using the common contents is available both in MILS/SILS environment and HILS environment.

The standard "ISO26262" for road vehicles was established as an international standard concerning functional safety for automotive electronic control functions, and every manufacturer is making efforts to follow the standard. The standard requires, even in the case of model-based development, the issues concerning the verification of the consistency between the code built in a model (specifications) and the code in a product (hereinafter, referred to as Back-to-Back test). The use of the contents common in MILS/SILS environment and HILS environment allows efficient implementation of the Back-to-Back test.

4. Features of CRAMAS-VF

CRAMAS-VF provides the following features so as to solve the problems in control development described in Section 2 and to extend the simulation environment.

1) Use of existing control software (C code)
2) Function for simulation of interrupt-driven processing
3) Linkage with a measuring tool created by another company

4.1 Use of Existing Control Software Asset (C Code)

CRAMAS-VF, as shown in Fig. 5, can perform simulation while concurrently operating the new control model to be developed and the existing control software in C code.

The existing control software is available for control other than the one to be developed. This allows the creation of development environment in a short time without expressing the entire control software in a model. As mentioned above, CRAMAS-VF allows the development of new functions to be developed as a model. This solves the problem in model-based development: "re-expressing..."
entire control in a model involves huge man-hours because the number of control functions in a larger and complicated control system is huge and the control functions have been conventionally expressed in C code.” That is, this allows smooth transaction to the model-based development.

Fig. 6 shows the configuration of CRAMAS-VF facilitating the use of existing control software.

CRAMAS-VF triggers interrupt-driven processing functions so as to carry out the same processing as the actual ECU. There is a method for executing simulation at the timing equivalent to microcomputer clock, like microcomputer emulation, so as to carry out control at accurate timing. However, if the control software is operated on CRAMAS-VF at the timing equivalent to microcomputer clock, simulation speed is reduced because the plant model and control software are operated at common time intervals. CRAMAS-VF creates and controls the list of interrupt generation timings (hereinafter, referred to as event list) to suppress the reduction of the simulation speed. CRAMAS-VF also has the function for specifying the timing for executing control software independently from the timing for executing a plant model, by separating arithmetic operating cycles and control software clock processing. (Refer to Fig. 8.)

4.2 Function for Simulation of Interrupt-Driven Processing

Fig. 7 shows the concrete configuration of interrupt-driven processing by CRAMAS-VF running on Simulink.

Here is a concrete description of the flow of interrupt processing based on Fig. 8. The example of the simulation to be described is executed under the following assumed conditions: use of the simulation arithmetic operation intervals of the plant model as the execution intervals of entry AD conversion processing (in this case, 4 [ms]); and use of the processing intervals of control software as the sampling time of the minimum interrupt intervals (in this case, 125[ms]).

The first step is to calculate, based on the engine speed and the crank angle at the current time (here, 0[ms]), the time of occurrence (1.280[ms]) of the synchronization interruption (hereinafter, referred to as 10degCA crank angle synchronization interruption) corresponding to the pulse (pulse signal every 10 degree) output by a crank angle sensor at the time (at 1.024[ms]) when the next 1ms period processing of the control software is executed. The next is to register, in CRAMAS-VF event list, the name of the interrupt function and the interrupt time, and the name of the next 1ms period processing function.
and its execution time. When the current time under simulation reaches the time registered at the top of the event list (in this case, at 1.024[ms]), CRAMAS-VF executes the 1ms period processing of the control software. Then, the next is to re-calculate the time for executing the next 1ms period processing of the control software (2.048[ms]) and then to register the time in the event list. In addition, the 10degCA crank angle synchronization interruption is also executed. The calculation processing of the plant model is transferred into MATLAB every simulation operation period, and then carried out by the solver of Simulink. The current time of the control software is advanced to the time for the next interrupt processing so as to provide efficient interrupt processing by skipping the period in which any interruption of control software is not generated.

The functions described so far concurrently allow the accurate execution of the interrupt processing and high-speed simulation.

Each of the conventional simulators creates the simulation environment by customizing a middleware part of the control software. However, CRAMAS-VF can execute the interruption processing function, which allows the use of the middleware part as is of the control software. Thus, CRAMAS-VF allows the middleware part to be debugged, unlike the conventional simulators.

These installed functions provide the simulation of the control software as a whole that executes operations in the same order at the same timing as the actual ECU. This solves the problem: “the interruption processing equivalent to the actual ECU can not be evaluated because the interrupt-driven processing in the MILS/SILS environment can not be simulated.”

4.3 Linkage with Measuring Tool Created by Another Company

CRAMAS-VF can work with Panel3 (created by Toyota Technical Development Corporation), which is the GUI application for measurement evaluation having been used for evaluation in actual vehicles and evaluation in bench tests [Product name: SMiL (Fig. 9)]

Due to the linkage between CRAMAS-VF and such evaluation tool that is familiar to users, the work of users can be more efficient and introduction of CRAMAS-VF can be easier.

5 Conclusion

The CRAMAS-VF we developed this time is the result of our tool development technology and our packaging technology of control software for mass produced ECU. CRAMAS-VF easily allows the creation of the MILS/SILS environment enabling the verification equivalent to actual ECU, which contributed to the promotion of the model-based development for vehicle control. However, there are various challenges that remain in the introduction of model-based development as a control development process, and also there are high expectations and many requests concerning tools. We will make efforts to level up the products in the MILS/SILS field based on CRAMAS-VF as a core technology, while following the evolution of computing technology, and will keep developing CRAMAS products including HILS so that they can function as toolchain in various steps of the model-based development.

6 Acknowledgment

Lastly, we would like to express our sincere gratitude to TOYOTA Motor Corporation and the companies concerned for cooperating in this development.

※CRAMAS is a registered trademark of FUJITSU TEN Limited.
※The proper nouns of product names and the like indicated in the paper are trademarks or registered trademarks of the respective companies.
Reference
1) "Development of Next Generation SILS," written by Yasutaka Fujiwara, Hisahiro Ito, Harunaga Uozumi, and Koji Fukuoka, on pp.1~4, of Proceeding No.122-12, published by Society of Automotive Engineers of Japan (JSAE) in 2012
2) "Development of Next Generation SILS" written by Koji Fukuoka on pp.2~3 of JSAE KANSAI BRANCH NEWS No. 42, published by JSAE KANSAI BRANCH in 2013