# Improvement in development efficiency through time reduction of model-in-the-loop simulation

Ayumu OSHIMA* Kentaro MATSUNAGA* Jaeho KIM*

**Abstract**

Model-based development has been widely used for the efficient development of control software. In the model design phase, verification of the virtual environment is important. However, model-in-the-loop simulation (MILS), which is used to verify transmission-control software, requires extremely long simulation times. Therefore, we introduced methods and tools that reduce simulation time to increase development efficiency. This paper introduces the details of these approaches.

## 1. Introduction

Model-based development has been widely used in the development of control software, with blocks representing each control function. In model-based development, control specifications are described by a model directly used for the simulation. This process allows designing and verification to be performed in a short cycle (Fig. 1). These verification methods and environments using models are known as model-in-the-loop simulations (MILS). For example, if a design error is discovered during the actual vehicle verification phase, which is set in the latter half of the development process, the process must return to the design phase for redesign. However, if the MILS is used for verification in the design phase, the control software can be modified with a small amount of rework, even if errors are discovered. Thus, efficient development can be achieved[1].

One approach is to perform verification using a model created before implementing the control software in the transmission control unit (ATCU). We prepared a transmission-controlled MILS (TM-control MILS). However, the simulation time for the TM-control MILS was extremely long. Therefore, we attempted to reduce the MILS simulation time. This paper presents the details of these efforts.

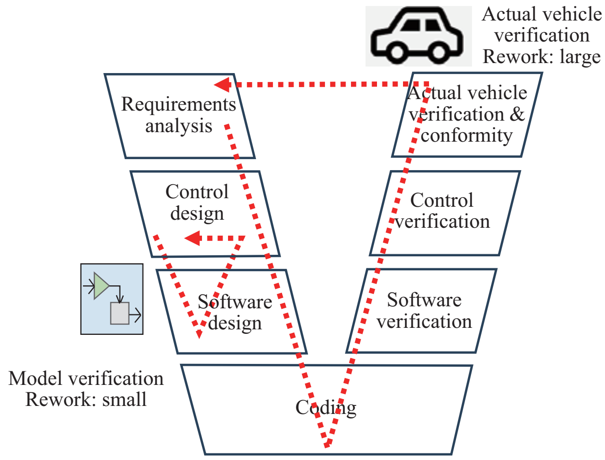


**Fig. 1 V process of control development**

## 2. Current state and challenges of TM-control MILS

The configuration of the TM-control MILS is as follows. The TM-control MILS comprised three models: a driver model simulating actual vehicle operation, a plant model representing the actual vehicle and transmission equipment through physical equations, and a controller model that includes the transmission control. The models were connected to form a single model for simulation (Fig. 2).

Among these models, the driver and plant models were limited to minimum necessary functions. However, the controller model had more blocks and a larger volume than the other models because it incorporated all the control

*Control System Development Department

software implemented in the ATCU. This configuration resulted in a long simulation time for the entire TM-control MILS. Specifically, a simulation using a 30s test case took approximately 600s, which was 20 times longer, to complete the calculation.

Therefore, we started to consider whether the simulation time could be reduced. We aimed to reduce the simulation time within twice of the real-time period.

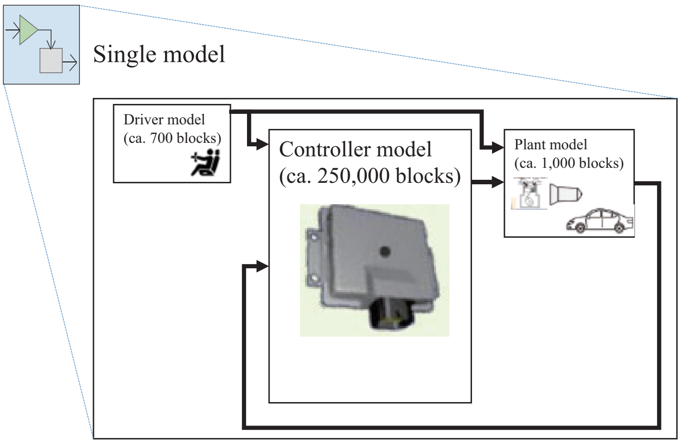


**Fig. 2 Overview of the TM-control MILS**

## 3. Methods to solve the problem

We employed the following two methods to reduce the simulation time for the TM-control MILS.

**(1)Simulation method using executable forms produced by converting models**

A method for converting a model into an executable form is as follows. Conventionally, a model is simulated by interpreting the described operations and translating them individually into a machine language. Therefore, for a model in which the operation results are switched by conditional branches, the results are not produced until all the operations are completed. The time required for a one-by-one translation into a machine language is considered. These factors result in longer simulation times.

By contrast, a simulation method that converts a model into an executable form converts first the entire operation into a machine language. In this case, only the minimum number of necessary operations are performed. The parts, including the conditional branches, are extracted from the entire operation and optimized. This process eliminates unnecessary operations, thereby reducing the simulation time (Fig. 3).

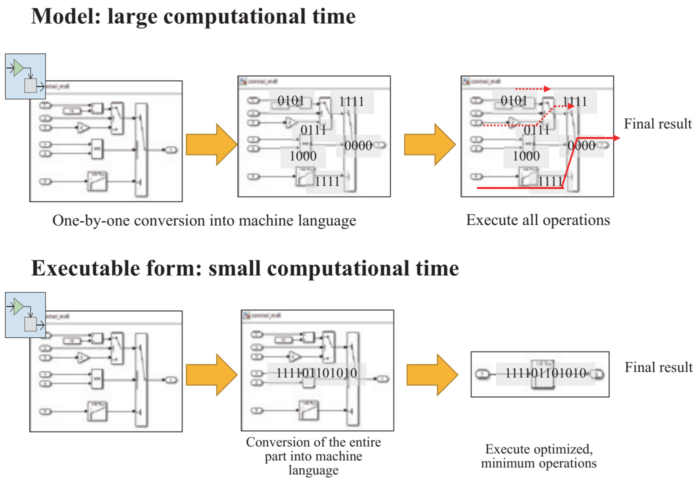


**Fig. 3 Difference between a model and an executable form**

The software tool VEOS from dSPACE was used to apply the method to the TM-control MILS (Fig. 4). VEOS has the following functions: model import, conversion of models into executable forms, and simulation with executable files.

When building a model with a large volume in VEOS, an error can occur. To avoid this error, a single large model must be preliminarily divided into multiple models with smaller volumes before being imported into the VEOS. Because the TM-control MILS models are large, they can be split into multiple smaller models.

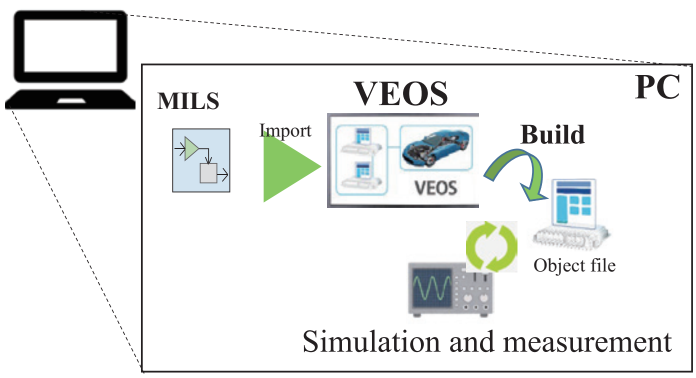


**Fig. 4 Simulation environment using VEOS**

**(2)Model simulation performed on a dedicated computer**

The method for performing model simulations on a dedicated computer is described as follows. The hardware dedicated to model simulation was used independent from the PCs. The model was downloaded to the processor core of the dedicated computer for the simulation. A SCALEXIO processor unit (hereinafter referred to as SCALEXIO) from dSPACE was used as the dedicated computer (Fig. 5). It is a hardware tool equipped with a processor core with high computing power and a real-time OS.

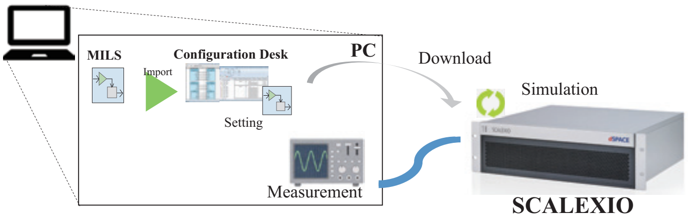


**Fig. 5 Simulation environment using SCALEXIO**

## 4. Results of simulation time reduction

The results of simulation time reduction obtained using the tools is described below.

**(1)Results obtained by converting models into executable forms**

The models were converted into executable forms in VEOS for TM-control MILS simulations. The simulation time was reduced to approximately four times that in real time.

As described in Section 3, the large number of model blocks in the original model for TM-control MILS caused errors in the VEOS. Therefore, the original model was divided into several smaller models with equal number of blocks. The smaller models imported into the VEOS were successfully converted into an executable form without generating any errors (Fig. 6).

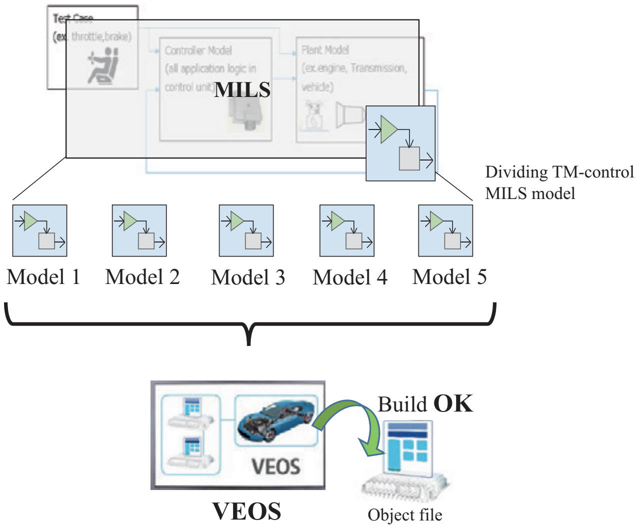


**Fig. 6 Model division for MILS**

Using the VEOS, we found a correlation between the number of model divisions and simulation time. The relationship between them is shown in Fig. 7. The results were obtained using 30s test cases. The numbers of divisions were up to 10. The results showed that the simulation time increased from 120s to 300s (4 to 10 times the real time) as the number of divisions increased.

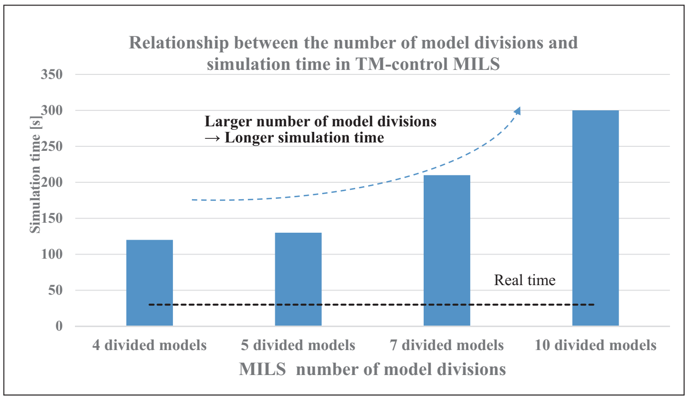


**Fig. 7 Relationship between the number of model divisions and simulation time**

The correlation between the number of model divisions and the simulation time was affected by sequence in which the model operations are completed and the subsequent operations are performed (Fig. 8). The larger the number of divisions, the more processes are required to specify the order of operations, resulting in an increase in the operation and simulation times. The order of operations must be specified and defined. If not defined, the order of operations might be different from that of the original model, and the expected result would not be obtained. However, specifying the order of operations among the models can produce the desired results similar with those of the original model.

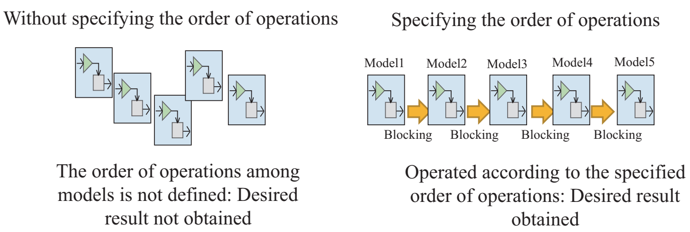


**Fig. 8 Specification of the order of operations**

Verification using MILS can target not only the entire TM control but also a single function with a small volume. This approach is called single-function MILS. The MILS is limited to the control functions of the controller and plant models, and is suitable for verifying only a certain control function. For example, for a single-function MILS limited to the shift control function, the simulation time was reduced to approximately half of the real time (Fig. 9). In actual development, the single-function MILS can be satisfactorily used for verification in a shorter simulation time when only a certain control function is verified.

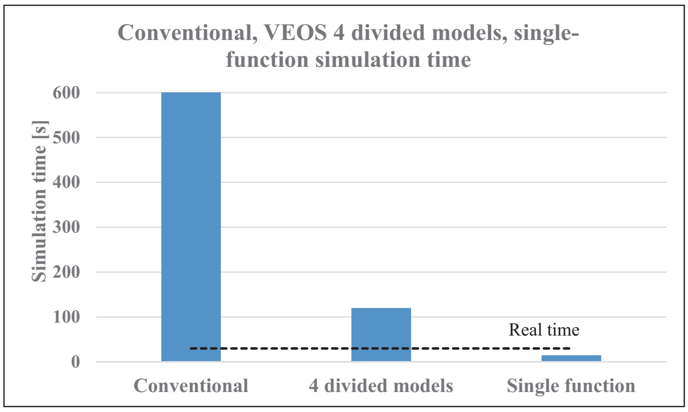


**Fig. 9 Comparison of simulation time between conventional MILS and VEOS**

**(2) Results of the simulation using a dedicated computer**

Using SCALEXIO, TM-control MILS simulations without model division in VEOS could be performed for a period of time comparable to real time (Fig. 10). This is because SCALEXIO executed the MILS simulations on a dedicated high-performance computer. In addition, SCALEXIO has a real-time OS that differs from OSs used in common PCs.

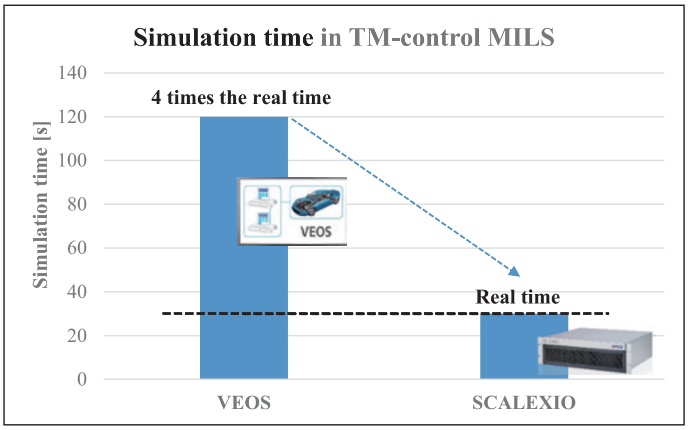


**Fig. 10 Simulation time of TM-control MILS**

Simulations using SCALEXIO require not only a PC but also dedicated hardware, which requires a large capital investment if the system is allocated to each designer. Therefore, we assume that it is ideal to operate one system as a shared resource used by several people. The ability to run high-volume TM-control MILS simulations in real time without dividing the models would be beneficial, even if the system is not used frequently.

## 5. Conclusions

We have achieved our goal to reduce the TM-control simulation time within twice of the real-time period.

A previous simulation took approximately 20 times longer than the actual time. When the models were converted to executable forms, the simulation times were reduced by a factor of 4–10. For a single-function MILS, the simulation time was reduced to less than the real time. For simulations using a dedicated computer, the time was reduced to the level of the real time.

## 6. Discussion

The two approaches described above were effective in improving the usability of MILS because they both reduced the simulation time. We believed that the control software development can be achieved more efficiently by utilizing these MILSs.

In actual verifications using the MILS, two approaches can be used depending on the application. When partial verification is sufficient, the approach of converting a single-function MILS model to an executable form is effective for simulation in a time shorter than real time. When the entire control software is to be verified, the approach of using a dedicated computer is effective because it can perform TM-control MILS simulations without dividing the model.

## 7. Future issues

Both approaches require tools. Therefore, to apply these tools within a company, it is necessary to examine the number of tools required and develop a capital investment plan that considers them. In addition, to use MILS and tools in the actual development process, the following are future issues: deciding a specific design phase, determining management and operation rules for MILS shared among developers.

For technological development, we are considering expanding the virtual verification area to enable control verification without an actual ATCU. In the current TM-control MILS, the OS is modeled in a simplified manner. We are considering building a virtual ECU simulation environment to simulate the entire ATCU in combination with the actual software. This will enable verifications during the design phase, even when actual ATCU equipment or actual vehicles are required.

We will continue to pursue further development efficiency by effectively utilizing the merits of model-based development.

## 8. References


(1)Ryohei FUJITSUKA, Daisuke HARADA, Yuki BABA: Enhancement of control logic validation using model-in-the-loop simulation (MILS), JATCO Technical Review No.18, pp.39 - 44, 2019.

(2)Daisuke YAMAGATA: Improvement of development quality by applying MBD to experiments -Application of a VRS system to CVT development-, JATCO Technical Review No.21, pp.85 - 88, 2022.



■ Authors ■

Ayumu OSHIMA

Kentaro MATSUNAGA

Jaeho KIM