

Technology for Embedding Engine Models in ECU for Model-based Development

Shusuke AKAZAKI*

Tomoya KUBO*

Kohei CHIDA*

Haruki SHIMIZU**

Fumitake YAMAZAKI***

ABSTRACT

Consistency in the control subject model from the top level to the lower level of the V-process used in model-based development is an important factor when designing internal combustion engine control. For this reason, parameter calibration was used to provide relevance between the 1D engine model in the top level and the mean value engine model in the middle level so that the deviation is within a certain level. In addition, an environment was constructed for embedding the mean value engine model used by Hardware-in-the-Loop Simulation as is in an ECU. The model is described using a formula simulator and converted to C code. A tool was developed that automatically converts this code into C code for mounting in an ECU. The converted C code was actually operated using a CPU evaluation board, and was confirmed to operate in 10 ms or less, which is the calculation cycle for engine device control. This made it possible to link the models in the top and middle levels of model-based development and also enabled model mounting to an ECU in the lower level, which increased the reusability of V-process engine models.

1. Introduction

Increasing attention is being given to fuel economy enhancement technology that focuses on increasing thermal efficiency, such as downsizing turbocharged engine technology and research on Low Pressure Loop (LPL)-EGR⁽¹⁾ and lean burn.

High-efficiency engines call for dedicated sensors, devices, and software. However, systems are becoming more complex overall due to factors such as enhancement of catalysts used to process emissions. Calibration has been performed thus far by relatively simple work such as mapping with unit bench tests, but the parameter settings for mass production currently take an enormous amount of time. Attempts are being made to enhance optimization tools and simulations to achieve optimization with the goal of minimizing deviation by means such as the use of efficient data sampling methods and response surfaces⁽²⁾. In addition, it is predicted that exhaust emission regulations will become stricter around the world, and there are concerns that demands on engine control will increase, such as more accurate air-fuel ratio control and EGR control to support fuel economy and exhaust emission regulations.

Meanwhile, due to the push for model-based development, Hardware-in-the-Loop Simulation (HILS)⁽³⁾ is becoming popular as a verification tool. In the field of control system design, attempts are being made to develop control logic using HILS and actual ECU. To increase the effects of model-based development, it is desirable to promote advances in models by increasing model reusability. Therefore, this paper mounted an accuracy-assured mean value engine model⁽⁴⁾ created for HILS and calibrated for control design and verification as is in a control ECU, and investigated the reuse of this model as an actual engine-state predictor in an ECU. CPUs have thus far lacked sufficient calculation performance, so while there are cases of embedding partial engine models⁽⁵⁾, there are few known cases of embedding a full engine model including from the intake system to the combustion and exhaust systems. The mean value engine model used in this paper was a direct fuel injection turbocharged engine model created by a formula simulator, and was optimized beforehand using a mathematical model. This model was executed using a commercially available CPU evaluation board to verify mounting in an ECU, and the calculation times and calculation results were compared.

* Automobile R&D Center

** RYOMO SYSTEMS CO., LTD.

*** Cybernet Systems Co., Ltd.

2. HILS and ECU-embedded Model Sharing and Issues

2.1. Engine Control HILS for Model-based Development

When performing function design in the middle level of model-based development, control logic that satisfies the function requirements is designed using Model-in-the-Loop Simulation (MILS) or HILS as shown in Fig. 1. An engine model is used as the control subject model, and the model parameters are calibrated so that the actual engine and steady-state behavior match. In addition, the transient-state behavior should also be reproduced in the same manner as the steady state. The use of a common model by MILS during function design and by HILS during verification helps prevent differences between design and verification. However, although the MILS model does not call for real-time execution, HILS uses an actual ECU, so real-time execution is needed.

2.2. Setting Targets for ECU-embedding

Models that can be executed in real time on HILS consist of comparatively simple calculations, so it was

thought that these calculations could also be performed by an ECU. Therefore, it was investigated whether models could be reused to predict temperature, pressure, and other characteristics at points that are a challenge to observe. Unlike an actual engine, the sensors needed for measurement can be freely mounted in a model inside an ECU. This paper performed development with a focus on torque calculations using a combustion model. Driving force control calls for a torque calculation deviation of 10 Nm or less, which is 5% of the full-throttle torque, so the torque calculation deviation of the model was also set to 10 Nm or less to satisfy this requirement. The control cycle for devices such as the throttle is 10 ms, so the control cycle was set to 10 ms or less. The development targets are listed below.

- (1) The HILS mean value engine model used for control system design and when verifying controllability is a full model including all systems from intake to exhaust, and can be mounted in an ECU.
- (2) Development of a tool that automatically modifies the C code generated automatically from a mathematical model into code that can be mounted in an ECU.
- (3) The ECU calculation cycle for the engine model is 10 ms or less, and the torque calculation deviation is 10 Nm or less.

3. Calibration of Mean Value Engine Model with an Actual Engine

3.1. Embedded Engine Model

Figure 2 shows the flow of engine model parameter calibration in sequential system identification⁽⁴⁾. Different tools are used in each of the top, middle, and lower levels of the V-process flow, so the architecture performs calibration to match the setting parameters of each model with the simulation results. A 1D model is calibrated using a current engine in the initial stage of development, and in the next

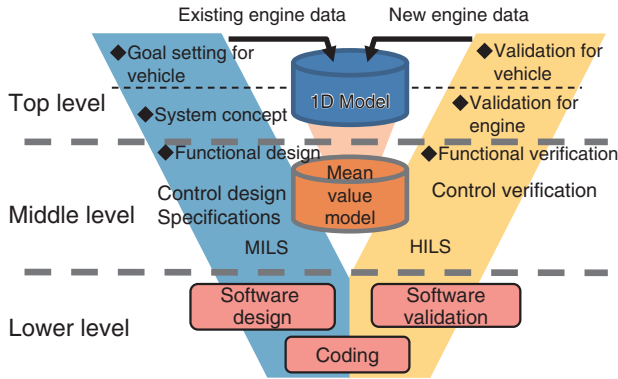


Fig. 1 Standard V-process

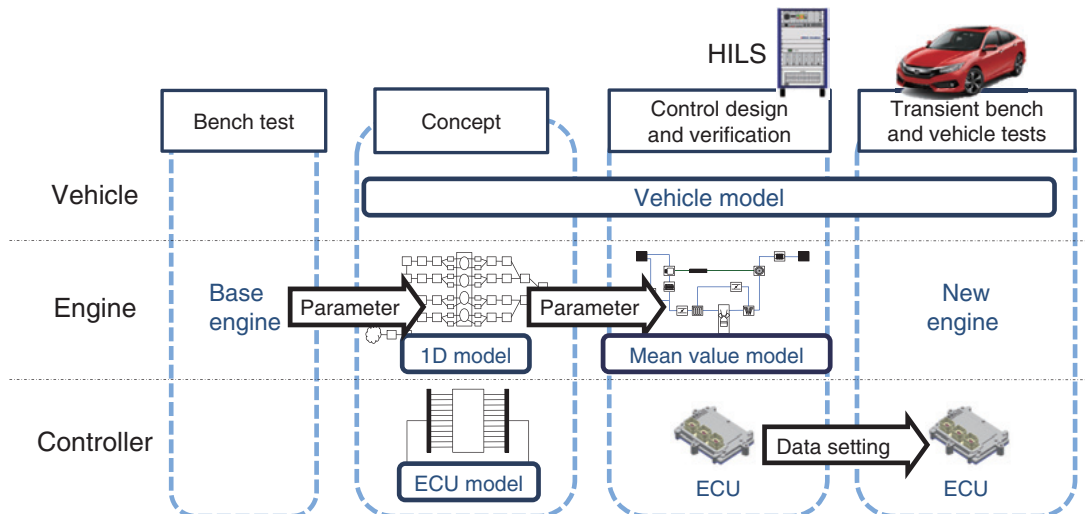


Fig. 2 Model parameter identification flow for engine plant model

step calibration is performed to approach the mean value engine model. The final stage is the development step where the model is mounted in the control ECU. This step is then repeated to enhance the developed engine and also raise the system development level.

Figure 3 shows the developed engine model. A mean value model of a downsizing turbocharged direct fuel injection 1.5 L 4-cylinder gasoline engine was created using MapleSim, which is a modeling simulation tool with formula processing functions. The model parts configuration used the actual engine parts combination. The layout took into consideration modularization in order to minimize the changes to the model when changing the parts configuration or specifications. This enabled to support changes to parts specifications by modifying part unit bench measurement data and changing parameters to achieve the expected parts model simulation results. Standard parts components are provided for MapleSim, but non-standard models such as the combustion model were expressed mathematically and created using custom components. The LPL-EGR and combustion models created using custom components are described below.

3.2. LPL-EGR Model

When the exhaust gas introduction delay is not reflected in LPL-EGR, the issue arises that the control system design using HILS and the transient state behavior of an actual engine do not match. Simulations were therefore performed to confirm whether the actual EGR gas delay behavior could be reproduced by parameter settings. Figure 4 shows the MapleSim simulation results on a PC. These are the results when

step response is performed by LPL-EGR valve-opening control. EGR_rate indicates the ratio of the EGR gas entering the intake pipe. The EGR gas ratio when closing the LPL-EGR valve was compared for rate 1 when the

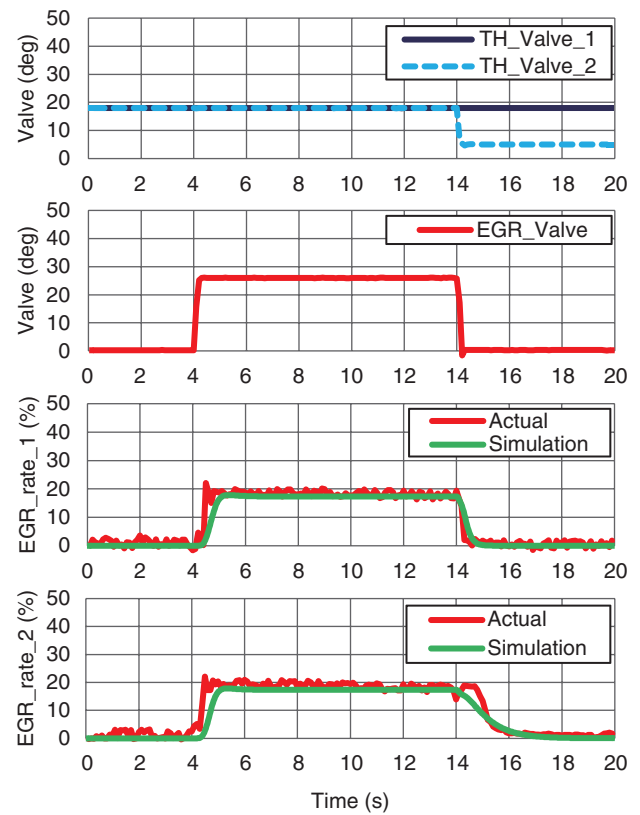


Fig. 4 LPL-EGR simulation results

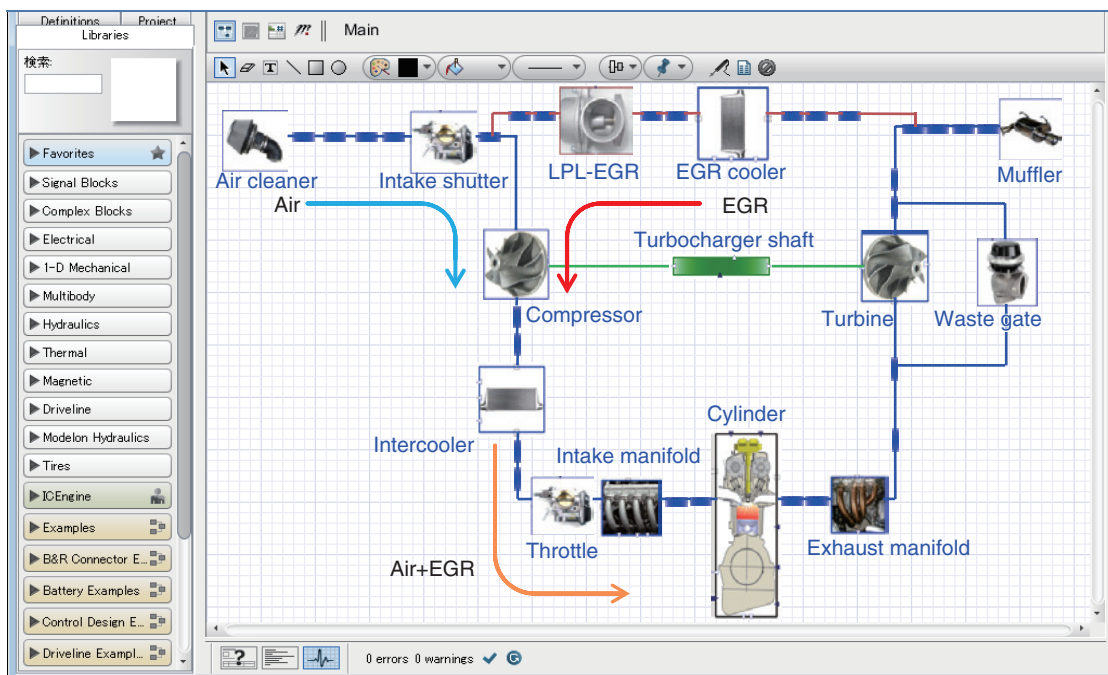


Fig. 3 Mean value engine model with turbocharger and LPL-EGR

throttle at the entrance to the intake pipe is open and rate 2 when the throttle is closed. These results show that the EGR gas responds immediately in the rate 1 (throttle open) condition. However, in the rate 2 (throttle closed) condition, the time to replace the gas inside the chamber such as the intercooler before the intake pipe increases, and an extended time of 2 s is needed to completely reduce the LPL-EGR gas. The actually measured response waste time and response delay was the same 2 s as in the simulation, so it was determined that the LPL-EGR model can accurately express the gas delay.

3.3. Engine Combustion Model

A Wiebe function is often used to perform combustion calculations for applications such as normal combustion analysis systems. However, a number of operations equal to the number of crank angle steps (approximately 0.5 deg to 1.0 deg/step) is needed, which makes real-time execution for use in HILS a challenge. Therefore, the calculation formula was simplified by making it possible to set the heat release pattern using typical points extracted from premeasured data. The model used by HILS employs functions of the engine speed, intake air volume, ignition timing, and other engine control parameters so that these setting values match the actual engine. Engine measurement used a standard Kistler sensor (305B) and an Indicom combustion analysis system made by AVL.

The red points in Fig. 5 represent the typical points set based on the heat release amount. A total of seven points were set, including the three points of the calculation start point, heat release start position, and heat release peak value, the two points of the second-order differential

value peak positions, and the two points indicated by triangles near the mass burnt fraction 5% position and the combustion end position.

Interpolation was performed by straight-line approximation from each point to obtain the heat release pattern relative to the crank angle. The cylinder pressure during the combustion period was calculated from the obtained heat release pattern and the stroke volume.

Equation (1) is used to calculate the heat release from the cylinder pressure, the stroke volume, and the specific heat ratio. Equation (2) is obtained by transforming Eq. (1), and expresses the cylinder pressure.

$$\frac{d}{d\theta} Q(\theta) = \frac{1}{\kappa - 1} \cdot V(\theta) \frac{d}{d\theta} \cdot P(\theta) + \frac{\kappa}{\kappa - 1} \cdot P(\theta) \cdot \frac{d}{d\theta} V(\theta) \quad (1)$$

The parameters used in the above relational expression of cylinder pressure and heat release are as follows.

$V(\theta)$: Volume (cm³)

$P(\theta)$: Pressure (kPa)

$Q(\theta)$: Quantity of heat (J)

κ : Specific heat ratio

$$P(\theta) = \left\{ \int_0^\theta \left[\frac{d}{d\theta'} Q(\theta') \right] \cdot (\kappa - 1) \cdot V(\theta') \cdot d\theta' + \frac{P_{ini}}{V(0)^{-\kappa}} \right\} \cdot V(\theta)^{-\kappa} \quad (2)$$

However,

θ' : Integral value from crank angle 0 to θ

P_{ini} : Initial pressure

IMEP is expressed by Eq. (3), NMEP by Eq. (4), and PMEP by Eq. (5).

$$IMEP = \frac{1}{V_d} \int_{V_{-360}}^{V_{360}} P \cdot dV \quad (3)$$

$$NMEP = \frac{1}{V_d} \int_{V_{-180}}^{V_{180}} P \cdot dV \quad (4)$$

$$PMEP = IMEP - NMEP \quad (5)$$

V : Volume (m³)

dV : Derivative of Volume

V_d : $V_{max} - V_{min}$ (m³)

P : Pressure (kPa)

IMEP: Indicated Mean Effective Pressure

NMEP: Net Indicated Mean Effective Pressure

PMEP: Pumping Indicated Mean Effective Pressure

NMEP during the combustion period is calculated by integrating the cylinder pressure over the stroke volume as shown in Eq. (4). Next, the pumping loss (PMEP) has a relatively small impact on the torque calculation, so it is obtained simply by determining the measured value from a map according to the pressure inside the intake manifold and the engine speed. This made it possible to shorten the IMEP calculation time.

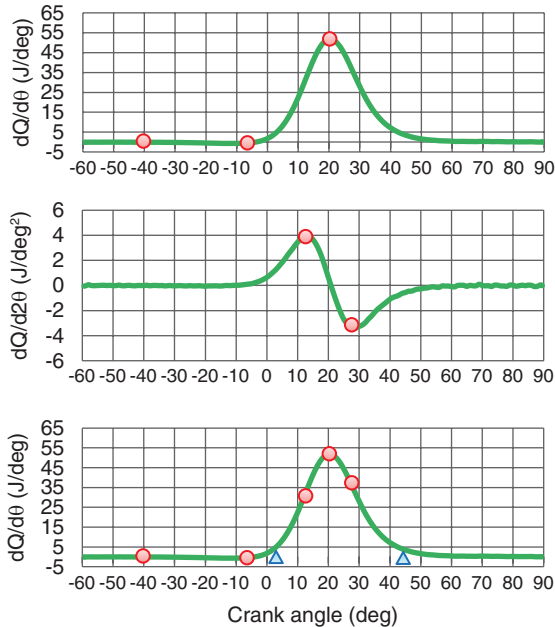


Fig. 5 Heat release rate of combustion and setting points for model calibration

3.4. Comparison with 1D Model

1D models are generally used from steady state to transient state as seen in cases of calibration with actual engines⁽⁶⁾. Calibration was performed so that the 1D model and mean value engine model parameters match the actual engine measurement results.

Calibration for the mean value engine model consisted of calibrating the setting values of the seven points of the combustion model described above to minimize the torque deviation. Figure 6 compares the calibration results of the two simulations with the measured engine torque. These results show that the two simulations exhibit a linear relationship with the measured torque from low to high power output, and the torque calculation value deviation of both the 1D model and the mean value engine model was 10 Nm or less. Based on these results it was determined that there are no issues for use as ECU-embedded models.

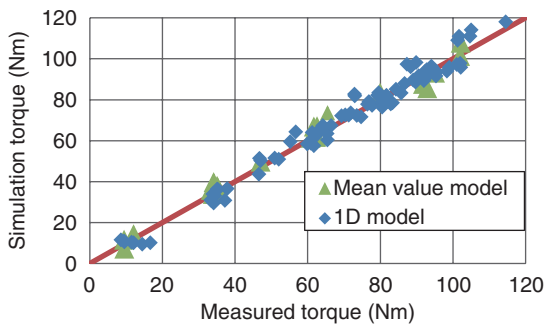


Fig. 6 Comparison of simulation results

4. Development of ECU Embedding Tool

4.1. Engine Model Code Conversion

The model introduced thus far is described using MapleSim, and C code is generated from the model using the auto C code generation function of MapleSim. This C code can be selected from MATLAB Simulink code and generic code, but generic code was used to minimize the impact of future tool version upgrades and other factors. Next, it is necessary to convert this code to the embedding program format in accordance with the program creation rules to enable embedding in the ECU. Embedding is then completed by using the target CPU compiler to create an HEX file and downloading the file.

Work efficiency was increased by developing a tool that automatically converts the code into C code for embedding. This enables provision of timely support for various engine models that are updated as needed in future development.

Figure 7 shows the execution screen of the developed code conversion tool. C code conversion can be executed simply by selecting the subject file on this simple screen. The development tool body was created using MATLAB

in consideration of continual version upgrades. The code conversion tool functions are listed below.

- (1) Application of program creation rules such as definition of variables

Variable name change, type name change, addition of comment text, limit processing, divided by zero prevention, type declaration of cast operators, function prototype declaration, addition of variables

- (2) Reduction of the processing load

Changing double precision to single precision, changing floating point division to private functions

- (3) Application of existing libraries

Header file load process

The C code conversion time using this tool was approximately 11 minutes, and the code size was 500 kbytes.

4.2. Operation Check Using a CPU Board

Figure 8 shows the appearance of the evaluation board used to confirm operation.

The Infineon TC297 evaluation board was selected due to its multicore CPU and the ability to set an arbitrary clock frequency. The program was mounted in this evaluation

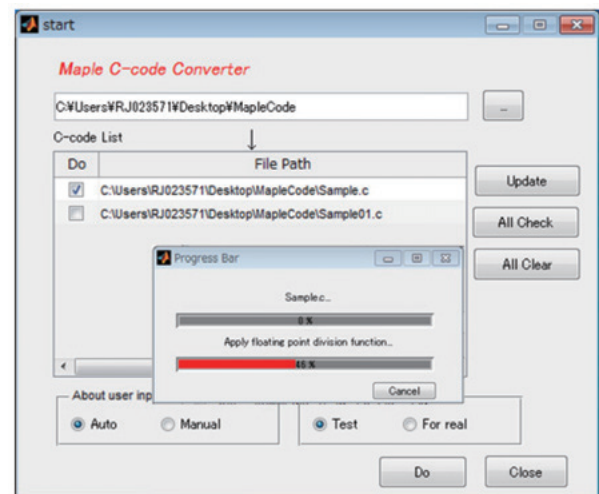


Fig. 7 File setting screen of conversion tool

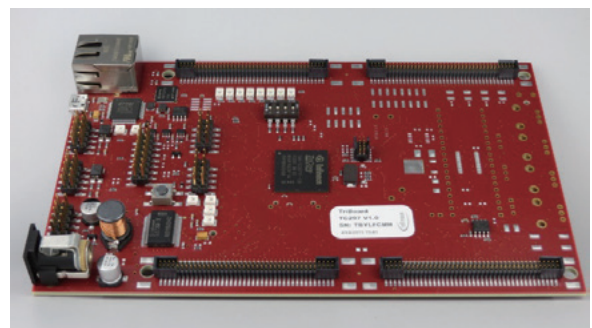


Fig. 8 Appearance of CPU evaluation board

board and the calculation time was measured. The performance of this board is approximately twice that of current mass-production ECU in terms of ROM and RAM capacity and processing capability.

These performance settings provide sufficient margin to support future expansion of the embedded model. Normal engine control and other calculation blocks are not mounted, so the impact of interrupts and other factors are not taken into account in the time measurements in this paper.

Figure 9 shows the calculation time results measured using Euler's method and the Runge-Kutta method. The ECU uses discretized⁽⁷⁾ signals when calculating the plant model, and typical Euler's and Runge-Kutta methods were compared as the discretization method. The CPU clock cycle was assumed to be the standard 200 MHz. The operation cycle of the engine model in the ECU should be as short as possible, but the torque control cycle using the intake air volume is 10 ms, so the operation cycle should be sufficiently shorter than this time. The comparison results show that the Runge-Kutta method, which has a high calculation load, had an execution time of 2.5 ms. The engine operating conditions during measurement were the vehicle operating conditions in the LA4 mode.

Discretization has the characteristic that the signal input and calculation values change in a step shape between the previous calculation and the current calculation. For this reason, when the difference increases, the calculation process easily becomes unstable due to overflow or other factors. In particular, when the calculation cycle becomes longer for a simple discretization process using Euler's method, the difference in the internal calculation results increases and produces a step in the calculations, so a short cycle is demanded. However, the overall calculation times increase, so when the engine speed increases, the number of event interrupts such as ignition timing and fuel injection also increase with the result that it takes time to finish calculations with a 10 ms cycle. The results of this study showed that even when using the Runge-Kutta method, the calculation time approximately doubled but was still within the target time, and stable operation could be obtained. Therefore, the Runge-Kutta method was selected.

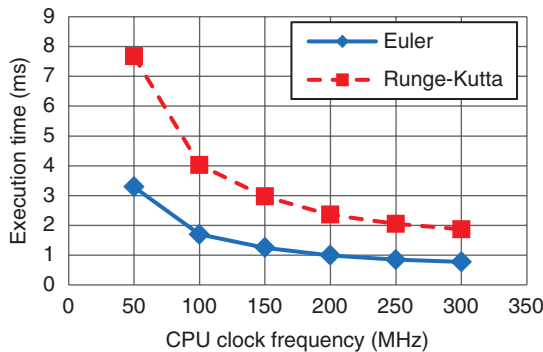


Fig. 9 Comparison of execution time with Euler's and Runge-Kutta methods

Table 1 Performance comparison between CPU and HILS

Execution condition	Evaluation board	HILS
Data length	16 bit	32 bit
Control cycle [ms]	10	0.1

5. Comparison of Calculation Results for HILS and ECU with Embedded Model

The difference between the calculation results when the engine model is mounted in HILS and when mounted in an evaluation board that simulates an ECU was verified. The driving pattern used for verification was the same LA4 mode used thus far. The calculation results were compared by simultaneously monitoring HILS and the evaluation board using CAN and measuring at 100 ms intervals.

Table 1 shows the difference in calculation performance between the HILS and ECU used.

The data lengths (16 bits, 32 bits) and the calculation cycles (10 ms, 0.1 ms) differed, so the monitored calculation results differed in terms of the monitor refresh cycle and the loss of significant digits due to the difference in the number of valid digits. As a result, difference waveform that resembled fine noise was observed when simply comparing the calculation results.

Figure 10 shows the results of comparing the torque calculation values at the 2nd peak of LA4 mode. Deviation such as fine noise also occurs, but this is due to the

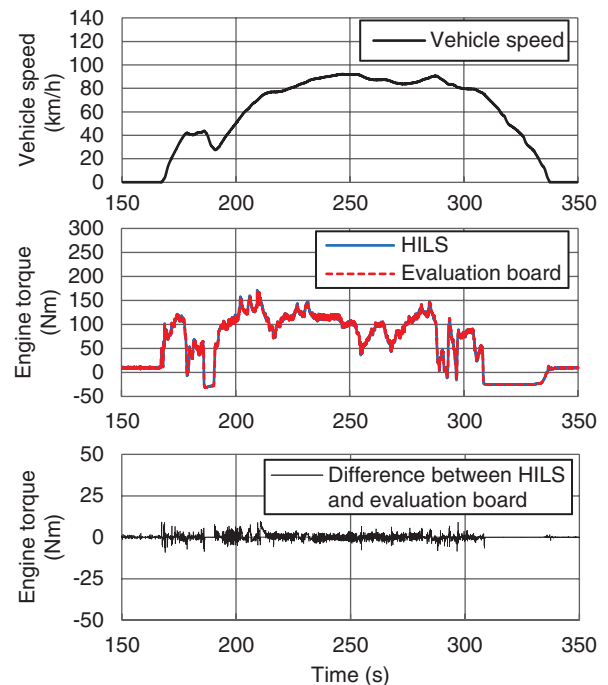


Fig. 10 Comparison of calculation results in LA4 mode

previously described difference in CPU performance.

As described above, embedding the mean value engine model mounted in HILS in the ECU enabled to standardize the engine model used to design the control system and verify operation by HILS with the engine model used for actual engine control. Accurate combustion calculations on the same level as a combustion analysis system can be performed by using a device such as a direct fuel injector integrated with a cylinder pressure sensor⁽⁸⁾. In addition to further enhancement of the engine combustion model, use of the embedded model in combination with this type of sensor can be expected to enable applications such as lean burn control using detailed cylinder information, and engine vibration suppression.

6. Conclusion

An environment was constructed for embedding the same engine model used for control design and verification in an ECU. As a result, when deviation or imperfections in the engine model are corrected anywhere in the V-process from upstream to downstream, the corresponding imperfections in other processes are also corrected. This can be expected to reduce reworking and increase the overall process efficiency.

These contents are summarized below.

- (1) The HILS mean value engine model used for control system design and when verifying controllability was the full model including all systems from intake to exhaust, and could be mounted in an ECU.
- (2) A tool was developed that automatically converts the C code generated automatically from MapleSim into code that can be mounted in an ECU.
- (3) The engine model calculation time using a CPU evaluation board was 2.5 ms, which met the target of 10 ms or less. The torque calculation error compared to HILS was 10 Nm or less, which is within 5% of the maximum output value.

References

- (1) Siokos, K., Koli, R., Prucka, R., Schwanke, J., Miersch, J.: Assessment of Cooled Low Pressure EGR in a Turbocharged Direct Injection Gasoline Engine, SAE Int. J. Engines 8(4), p. 1535-1543, (2015)
- (2) Kaji, H., Kita, H.: Experiment-Based Evolutionary Multi-objective Optimization of Engine Control Parameters, YAMAHA MOTOR TECHNICAL REVIEW, No.43, (2007)
- (3) Kodo, S., Fujimoto, S., Ono, T., Suzuki, T.: Driving Simulator for Power Plant Controller Development, Honda R&D Technical Review, Vol. 26, No. 2, p. 117-124
- (4) Guzzella, L., Onder, C.: Introduction to Modeling and Control of Internal Combustion Engine Systems, Springer, (2010)

- (5) Fujimoto, S., Ohtaka, Y., Kodo, S.: Development of High-performance Engine Plant Model and Sequential System Identification Environment for Realization of Model-based Development, Honda R&D Technical Review, Vol. 24, No. 2, p. 126-133
- (6) Kuboyama, T., Yamada, T., Moriyoshi, Y., Otaki, Y., Hatamura, K., Morikawa, K.: Development of 1D Numerical Simulation Model of a Supercharged Gasoline Engine, 2015 JSAE Annual Congress Proceedings, No.3-15S, p.63-67, (2015)
- (7) Yamamoto, Y.: Sampled-Data Control Theory—The Past and the Future, Journal of the Society of Instrument and Control Engineers, Vol. 40, No. 1, p. 76-82, (2001)
- (8) Akazaki, S., Inoue, T., Suzuki, T., Hayashi, T., Takaoku, A.: Improvement of Direct Fuel Injector Integrated with Cylinder Pressure Sensor and Development of Feedback Type Engine Torque Control, Transactions of Society of Automotive Engineers of Japan, Vol. 47, No. 1, p.17-22, (2016)

■ Author ■



Shusuke AKAZAKI



Tomoya KUBO



Kohei CHIDA



Haruki SHIMIZU



Fumitake YAMAZAKI