Technology Explanation

Enhancing the Accuracy of SA Damping Force Simulation Implemented with AI Technology and Building an AI Operational Management Platform

MANTANI Hiroaki・SAGEHASHI Ryota

Abstract

The future trend of Artificial Intelligence (AI) includes the industrial application of combining AI with Computer Aided Engineering (CAE). By integrating AI and CAE, several benefits emerge, such as the acquisition of data through CAE that would be impossible to obtain in the real world for training AI models, reducing CAE computation time by degenerating Finite Element Method (FEM) tasks for AI.

Additionally, in recent years, the automotive industry has witnessed an increased adoption of Model-Based Development (MBD), leading to collaborative development where CAE models replace physical prototypes between departments or companies.

To facilitate MBD and enhance the performance prediction of our flagship shock absorber products, we developed a technology that combines Machine Learing Model(ML model) with CAE. This allows us to predict damping force performance metrics rapidly and accurately. Furthermore, a system was developed to automate the operation and management of machine learning models.

In this report, we provide a technical explanation of the implementation of machine learning for shock absorber damping force calculations and the construction of an operational management platform for machine learning models.

1 Introduction

1. 1 Target Product

This report introduces a Shock Absorber (SA) for automobiles shown in Fig. 1. SAs play the role of damping the vibration of the vehicle. They can extend or contract according to the bumps and dips of the road surface and changes in the position of the vehicle. Hydraulic fluid is displaced from the SA cylinder according to the operating speed. The flow of the displaced fluid is reduced by a small-area orifice or a laminated leaf valve, creating a pressure differential to provide a damping force. The SA performance is evaluated in terms of the damping force-velocity characteristics. The laminated leaf valve is a particularly important component for performance evaluation. It consists of a thin steel plate that is installed to vary the oil passage area depending on the pressure during operation. By changing the lamination specifications (outer diameter, plate thickness, the number of leaves), the damping force is tuned to the vehicle.

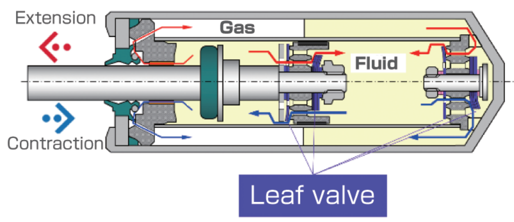

Fig. 1 Shock absorber structure

1. 2 KYB's CAE Initiative

Since the introduction of Computer Aided Engineering (CAE) in 1968, KYB has introduced various performance prediction technologies as shown in Fig. 2 to support the development of products and technologies. There are two types of CAE. One is 1DCAE[Note 1)], which develops product functions at the product planning stage (system simulation). The other is 3DCAE[Note 2)], which studies product geometry in detail (FEM[Note 3)]). KYB has accumulated prediction technologies for both types. In 1985, we put into operation the CAE Standard Execution System[1)], which is KYB's original system for easy technical calculation. The system allows users to easily utilize advanced prediction technologies from anywhere by using standardized input, execution, and output methods on various applications for either 1DCAE or 3DCAE. Approximately 2,000 programs are currently registered in the system.

KYB established the basic theory of SA damping force simulation, which is the main subject of this report, in 1981. Since then, the company has addressed the calculation as mathematical 1DCAE to be calculated on the CAE Standard Execution System.

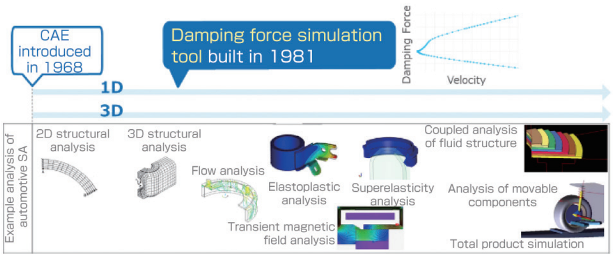

Fig. 2 Example of analysis of automotive SA

1. 3 Outline

With the recent widespread use of Model-Based Development (MBD[Note 4)]), it is necessary to develop products by jumping between 1DCAE and 3DCAE to run a cycle of studying both functions and geometry upstream in the product development stage. In addition, the need to circulate models instead of prototypes through internal departments and related companies has increased. In our initiative to address MBD, the technical challenges to be solved were as follows:

[1] In some cases, the coordination between the system study and geometry study was not well established. This was because the work of CAE specialists is divided into separate tasks due to the differences in modeling concept and skills between 1DCAE, which deals with simple and transparent models, and 3DCAE, which aims to reproduce actual machines more accurately.

[2] Geometry studies using 3DCAE tend to be expensive to analyze and take a significant amount of time to complete.

[3] Models to be circulated among related companies and internal departments must allow for high-speed computation while being highly detailed to represent actual products.

In order to solve these technical challenges and smoothly promote MBD, we needed to establish a technology that degenerates 3DCAE, which can handle detailed geometry, into a machine learning model (ML model) and implement it in 1DCAE (Fig. 3).

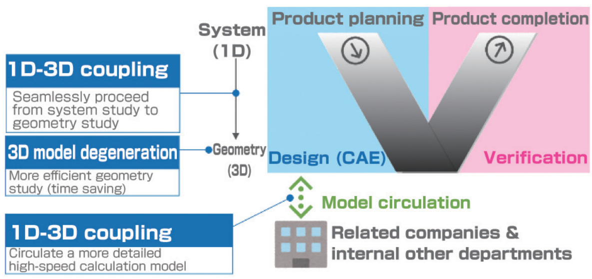

Fig. 3 Technical challenges of MBD

In order to establish technology for predicting the SA damping force accurately and quickly, we have built a technology for implementing the machine learning[Note 5)] model, which was degenerated from an FEM model of the SA laminated leaf valve, into a 1DCAE simulation tool (the 1D-CAE[Note 6)]). Its outline is shown in Fig. 4.

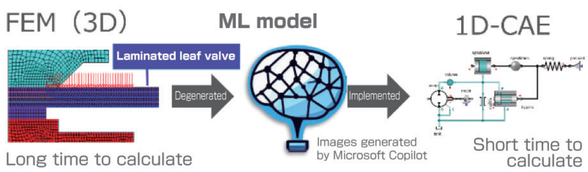

Fig. 4 Technology building for laminated leaf valve

Note 1) A CAE field that predicts product performance through mathematical system simulation.

Note 2) A CAE field that uses FEM to study product geometry in detail.

Note 3) Finite Element Method is one of the numerical analysis methods that can be used to deal with complex geometry by dividing the geometry of the product into elements (mesh).

Note 4) A design and development methodology that uses model-based simulation.

Note 5) A computer technology that learns regularities and patterns in data to determine the current state and predict the future.

Note 6) A simulation tool for 1DCAE. The hyphen is used to distinguish the tool from the CAE domain.

2 FEM Models

The FEM analysis of the laminated leaf valve, which is the target of this technology building, can be characterized as follows (Fig. 5):

[1] Allows evaluation of the deformation and stress of the laminated leaf valve during assembly (applied with an axial force due to screw tightening) and during operation (applied with a pressure due to hydraulic resistance of the orifice and other parts).

[2] Can take into account the initial deformation caused by the axial force due to screw tightening and the partial contacts inside the laminated leaf valve, which cannot be included in the theoretical calculation of the disc stiffness used to determine the SA damping force.

[3] Allows structural analysis not only with the microdeformation theory, but also with the large deformation theory.

[4] Due to its long computation time, the analysis has rarely been applied to the calculation of the SA damping force. Rather, it has been mainly used to study the geometry of parts around the laminated leaf valve.

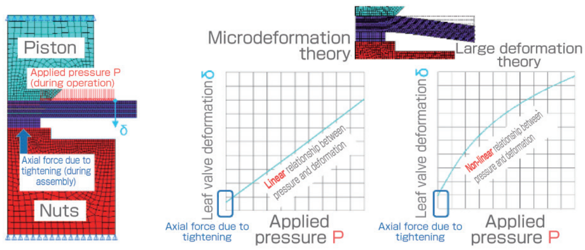

Fig. 5 FEM analysis of laminated leaf valve

3 Data Set Generation

We continuously performed the FEM analysis of a portion of the SA design series by changing the lamination specifications (outer diameter, plate thickness, the number of leaves) and the applied pressure, creating data sets necessary for building an ML model (Table 1). These data sets can also be used to calculate complex laminated leaf valves, called "preload valves", as shown in Fig. 6. A preload valve is a type of laminated leaf valve with its leaves at different heights to have an initial deformation, generating a preload to provide a high damping force.

For the continuous calculation, the FEM program codes were subjected to string processing by Python® to generate a program in advance with the lamination specifications (outer diameter, plate thickness, the number of leaves) of the laminated leaf valve randomly changed. KYB's standard CAE system was used for continuous automatic calculation 24 hours a day, seven days a week.

Table 1 Training data sets

Outer diameter of laminated leaf valve	5 levels or more
Plate thickness of laminated leaf valve	4 levels or more
Number of leaves of laminated leaf valve	3 levels or more
Preload valve	2 levels (yes/no)
Applied pressure	21 conditions
Deformation theory	Large deformation theory
Number of data sets	1,045,380

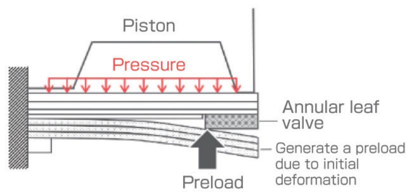

Fig. 6 Preload valve

4 Building an ML Model

After experimenting with the implementation of various machine learning algorithms, we were able to build an ML model that can infer the results of FEM analysis with high accuracy by using FLAML, a library in Python® Note 7) recently released by Microsoft Corporation. Fig. 7 shows the inference accuracy of the ML model for unknown data. The machine learning inference results (the vertical axis) and the FEM computation results (the horizontal axis) are plotted on a straight line with a slope of 1 (45 degrees). This verifies that the machine learning provides highly accurate inference.

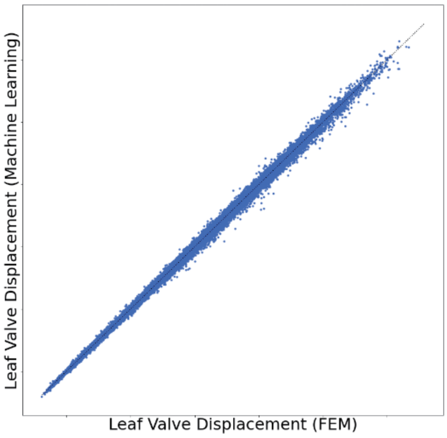

Fig. 7 ML model inference accuracy

FLAML is AutoML that can automatically select decision tree Note 8) machine learning algorithms or hyperparameters (parameters set by the model creator to control algorithms before learning). By using FLAML, we finally decided to use the LightGBM algorithm. LightGBM, which is a decision tree algorithm characterized by lightness and high speed, has been widely used in machine learning competitions in recent years.

5 Implementing the ML Model in 1D-CAE

Together with NewtonWorks Corporation, we researched how to implement the Python® ML model in 1D-CAE SimulationX®. As a result, we decided to use coupled analysis via communication, which is easy to implement. Fig. 8 shows how SimulationX® and Python® communicate with each other to send/receive coupled analysis data. For the purpose of verifying the effectiveness of the SA damping force simulation with the ML model mentioned in Chapter 6, Fig. 8 reflects communication between applications within the same personal computer. However, such a system is difficult to use for model circulation among related companies and internal departments. Finally, we introduced the cloud communication described in Section 7.4.

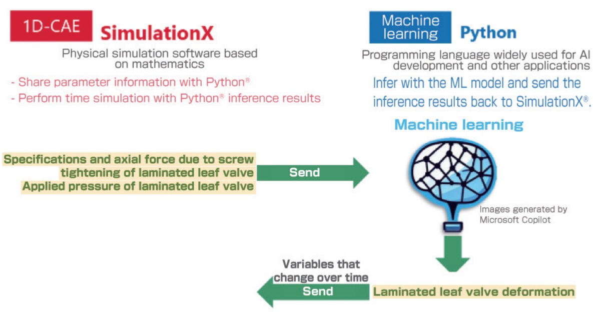

Fig. 8 Data exchange between SimulationX® and Python®

It was decided to transfer the information of variables that do not change with time, such as the outer diameter, plate thickness, and number of leaves of the laminated leaf valve, from SimulationX® to Python® only once immediately after the start of the calculation, and to transfer the information of variables that change with time, such as the applied pressure, to Python® via telecommunication at each communication time step. Based on the information of variables from SimulationX®, Python® infers the deformation of the laminated leaf valve and sends the results back to SimulationX® via telecommunication.

The computation for the coupled analysis by SimulationX® and Python® via telecommunication cannot be performed unless both tools are started to perform their own computation. In order to complete the computation by SimulationX® only, we created our own SimulationX® customized block shown in Fig. 9 to execute the Python® codes of the ML model in synchronization with the computation by SimulationX®. The customized block was created using the Modelica language Note 9) and can be characterized as follows;

[1] Allows parameterization of ML models.
[2] Allows computation via telecommunication between SimulationX® and Python® (Fig. 8).
[3] Allows setting up the Python® virtual environment and Python® codes to be operated for calculation via telecommunication.

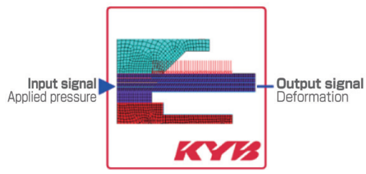

Fig. 9 SimulationX® customized block

Note 7) Python® is a trademark of Python Software Foundation.

Note 8) A machine learning algorithm that has a tree structure in which data is conditionally branched. Non-linear relationships can be identified.

Note 9) A multi-domain language for physical modeling.

6 SA Damping Force Simulation

We implemented the ML model built in Chapter 4 in the SimulationX® customized block (Fig. 9) to verify the effect of the ML model on the accuracy of predicting the damping force-velocity characteristic, which is an SA performance indicator. For this purpose, we built a SimulationX® model by linking the ML model with KYB's original hydraulic library in the Modelica language (Fig. 10). In this model, the pressure applied to the leaf valve calculated by the hydraulic library is used to calculate the deformation of the laminated leaf valve using the laminated leaf valve ML model. The resulting deformation is converted into an area of the hydraulic orifice and then returned to the hydraulic library.

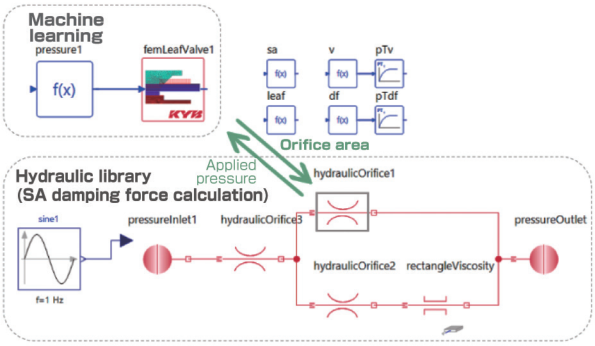

Fig. 10 SA damping force-velocity characteristics calculation model

As an example of verifying the prediction accuracy of the damping force-velocity characteristics compared to the experiment, Fig. 11 shows the results for the preload valve shown in Fig. 6. These results prove that, compared with the conventional theoretical calculation, the calculation model with the built-in ML model can predict the damping force-velocity characteristics with high accuracy.

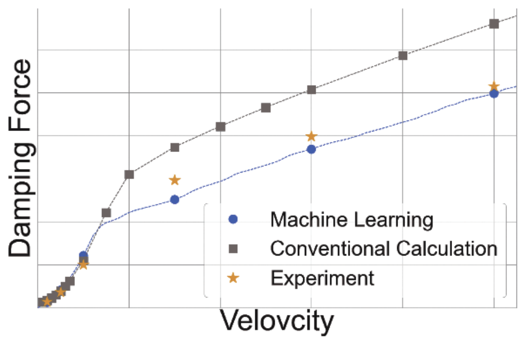

Fig. 11 Damping force-velocity characteristics prediction accuracy

By degenerating the FEM model for the laminated leaf valve to the ML model, the calculation time was significantly reduced from 93 seconds (with 23 pressure conditions) to 3 seconds (with 1,000 pressure conditions, including the time for the SA damping force calculation process).

7 Building an ML Model Operational Management Platform

7. 1 ML Model Operational Management System

In the process of operational management of the ML model technically built in the previous chapters, the following three major problems became obvious:.

[1] Complexity of building the environment and concerns about technology leakage

[2] Too many man-hours for operational management of the ML model

[3] Difficulty in managing the quality of the ML model

The environment mentioned in [1] refers to the user environment for the ML model. In general, individuals need to build a Python® programming environment to run the ML model on their own PC. In addition to the ML model data, the Python® program codes must be provided. This raised concerns about technology leakage when circulating the model among related companies and other departments of KYB. In addition, the program codes contain third-party libraries, which, depending on the user's PC environment, may disable the availability of libraries from target versions. Partly because of the possibility of this problem, it seemed likely to be complex to build the environment, raising concerns about too much man-hour burden on model users. In terms of [2], the ML model will be continually managed even after the model has been completed. The accuracy of the ML model may gradually deteriorate over time due to changes in the environment and other factors. In particular, its prediction accuracy may decrease with changes in the way the laminated leaf valve is used (design trend). To solve this problem, a model retraining process would be needed to ensure stable operation of the ML model. However, such continuous retraining usually requires a lot of man-hours for the ML model manager. If this operational process is not included in the workflow, there was a concern about too many additional man-hours. In [3], it was feared that the quality management of the ML model would be difficult even if the model was updated by retraining. This is because the ML model management itself depends on individual users as long as the complexity of the model environment mentioned in problem [1] persists.

To solve these problems, we internally developed an ML model operational management platform to realize MLOps [Note 10] as a system that allows relevant users to share and use a controlled

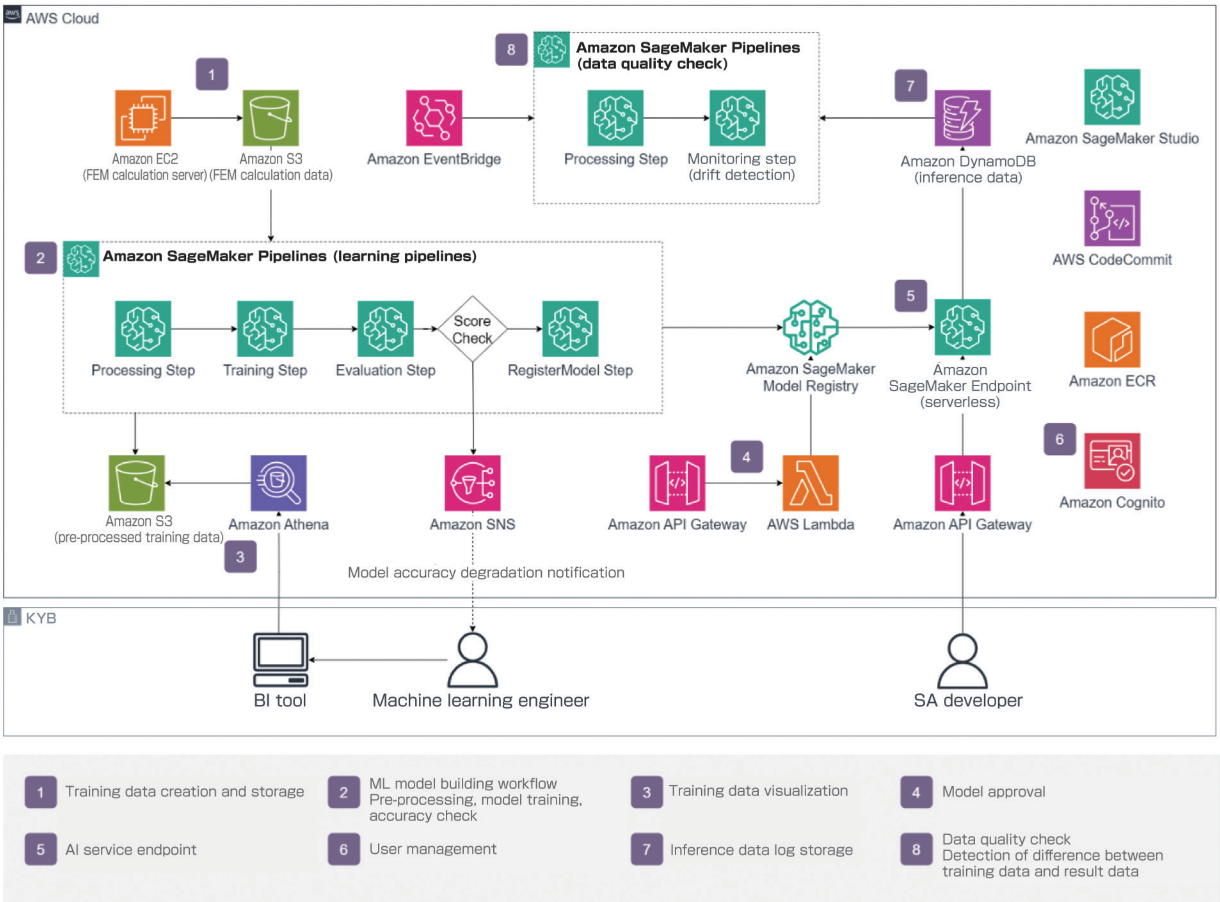

Fig. 12 Overview of the ML model operational management platform

high-quality ML model. Fig. 12 shows the overview.

The developed system can collectively manage all processes related to an ML model, including collecting, visualizing, and analyzing training data, as well as model training, accuracy checking, and developing AI services for SA damping force simulation by building an API system of the created model Note 11). This will greatly reduce the work of traditional ML model administrators. The system was built on the Amazon Web Services® cloud, which has advantages in functional extensibility and fault tolerance compared to the on-premises system Note 12). In addition, this system was developed by an internal cross-functional AI community[2] that brings together people with deep knowledge of machine learning and system development. The time required to complete the PoC Note 13) was only one year.

Note 10) An acronym for Machine Learning and Operations. It refers to an approach or concept for improving efficiency in the development, analysis, and operation of ML models.

Note 11) A scheme (interface) that connects different systems, including software and programs, with each other to allow users to share their functions.

Note 12) A type of system deployment where an organization owns and operates the servers and networking equipment needed to build the infrastructure.

Note 13) An acronym for Proof of Concept. It refers to a series of verification tasks to determine the feasibility of an idea or technology used to implement a product or service.

7. 2 Data Collection, Management, and Visualization

This system can be used to automate the series of processes from training data generation to uploading with a script, allowing the operator to perform the work with minimal operation. The FEM analysis data, which is the basis of the training data, is generated on the calculation server in the cloud managed by the CAE department with a training data generation script when KYB's relevant CAE expert deems it necessary to retrain the model. With this script, the FEM analysis data is processed to be used as training data for the ML model, and then automatically transferred to Amazon S3 Note 14) of this system.

The transferred data can be visualized using a BI[Note 15] tool Tableau®[Note 16](Fig. 13). In KYB, a company-wide data analysis environment using Tableau® is available, which enhances compatibility with the BI tool used in the company. This has enabled internal ML model managers and data scientists to quickly analyze data.

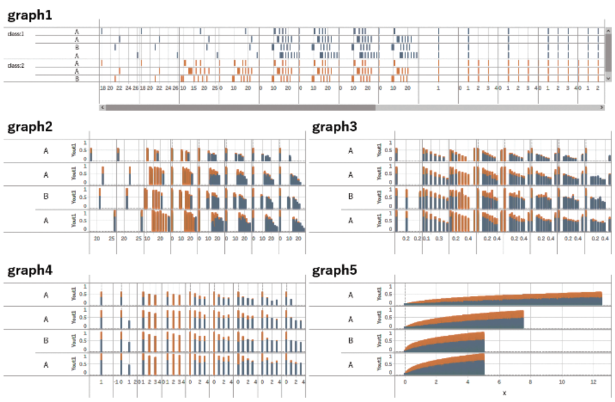

Fig. 13 Training data analysis screens

Note 14) Cloud storage that is highly fault tolerant and can store and protect data regardless of its type or capacity.

Note 15) Business Intelligence: A technology or approach that collects and analyzes necessary information from large amounts of accumulated data to be used for business management and operations.

Note 16) Tableau® is a BI tool and a registered trademark of Salesforce, Inc.

7. 3 Learning Pipelines

This system includes learning pipelines for ML model developers to develop an ML model with a low number of man-hours. Amazon SageMaker [Note 17] features are used to integrate the series of processes including data preprocessing [Note 18], model training, model accuracy verification, and inference endpoint [Note 19] into an automated workflow.

Using Amazon SageMaker allows the workflow to run on a virtual computing environment with the necessary specifications for processing. This eliminates the need to deploy physical computing servers with excessive specifications. Instead, the optimal processing resources can be quickly deployed to achieve cost-optimized processing.

The ML model is designed to be divided into two or more models according to specific computation conditions. This makes it easy to change the model design, thereby achieving flexible development that is easily adaptable to changes in demand. This model design will hopefully also help to identify data that has deteriorated the prediction accuracy, which could be caused by model retraining.

Thus, ML model developers can initiate the pipeline to train the target model by simply running a single command line that specifies the conditions of an ML model. In addition, this learning pipeline supports both the creation of a new ML model and its updating through re-learning. All processes related to model development with a view to long-term operation have been integrated into a single pipeline. On the other hand, the pipeline is designed to automatically select the workflow of creating a new ML model or the retraining workflow from the input to the pipeline. In addition, basic parameters related to model development have been given optimal values that have been previously collected, eliminating the cumbersome item setting that is otherwise necessary each time the pipeline is run. With these features, we have successfully built a pipeline that achieves the reduction of man-hours of ML model developers while increasing the reproducibility of model training.

Once the ML model training is completed, the generated learning model is automatically stored in Amazon S3. The results of the model performance evaluation are reported to the ML model administrators via the communication tool used in the company (Fig. 14).

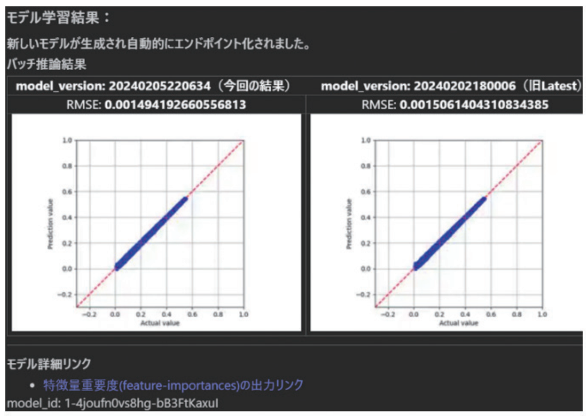

Fig. 14 ML model evaluation report screen

On the report screen, users can view the RMSE [Note 20] of the test results and the feature set importance[Note 21] of the generated model. In addition, for an ML model updated by relearning, a comparison with the results of the previous model is automatically displayed. In this way, the report screen is designed so that users can see the learning results at a glance.

Details of the ML model can also be viewed on

Amazon SageMaker. ML model developers can use the Amazon SageMaker Experiments feature Note 22) to visualize changes in the accuracy of the model being trained, changes in the optimal values of various parameters required by the learning algorithm, and the final model accuracy, and to compare the model's performance to that of any previous model (Fig. 15).

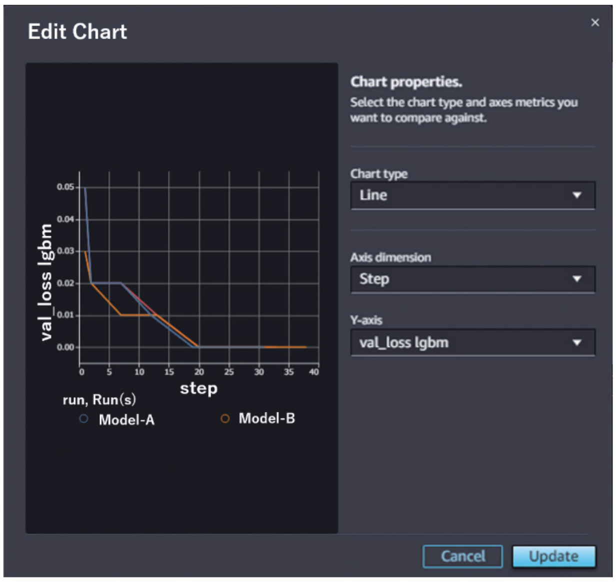

Fig. 15 Visualized changes in learning accuracy of ML model

Note 17) A cloud-based service that provides an implementation environment for rapidly developing and deploying ML models.

Note 18) A processing to integrate training data generated after data processing with a training data generation script.

Note 19) An interface that allows users to use ML models externally.

Note 20) The Root Mean Squared Error refers to the function used to determine the square root of the mean of the squares of the prediction errors. One of the general evaluation functions of ML models that focuses on regression issues.

Note 21) A metric that represents how much each feature set of the training data contributes to improving model accuracy.

Note 22) One of the Amazon SageMaker features that can track the learning record of the model.

7. 4 Deployment as an AI Service

The ML model generated by learning pipelines is automatically deployed as an inference endpoint. End users can use the ML model in real time.

To mainly enable internal SA developers to calculate SA damping force using machine learning directly from within their familiar SimulationX® tool, we have developed a custom library that can be built into SimulationX®. This custom library can be imported into the CAE tool to communicate with the inference endpoint in the cloud via WebAPI Note 23).

WebAPI is commonly used for back-end processing Note 24) of Web sites and other applications and allows the user to keep private certain processing, including model data and program code elements. Using WebAPI enables direct communication with inference endpoints from the 1D-CAE. The ML model inference results obtained through communication are reflected in the 1D-CAE. High accuracy SA damping force simulation is now available.

SA damping force simulation with this system involves communication via the Internet for inference processing with the ML model in the cloud. Therefore, we designed an original communication algorithm to optimize the number of communication times with the ML model to reduce the simulation time. As a result of applying the algorithm to the SA damping force simulation using machine learning, the communication time to determine the damping force-velocity characteristics shown in Fig. 11 was successfully reduced to about 1/10 of the time taken without the algorithm. End users can now use ML models to quickly and accurately calculate the SA damping force with a simple procedure without the need to individually build a machine learning environment.

The developed ML model is controlled for version by the system. Even if the system trains and updates the ML model to a new version, the previous version remains available (Fig. 16).

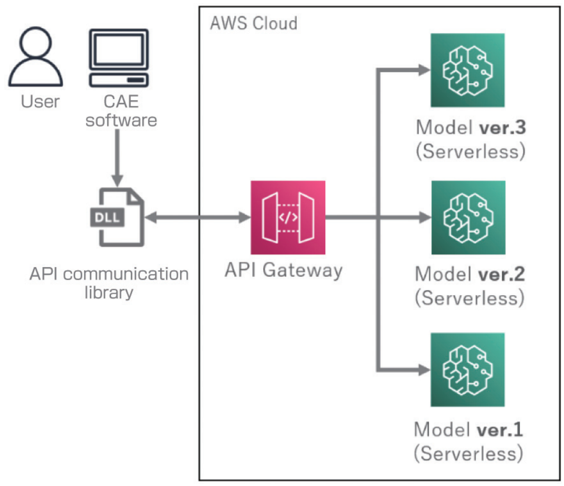

Fig. 16 Illustration of ML model version control

This version control ensures the reproducibility of the results of the SA damping force simulation using machine learning.

The custom library can be output after conversion to a data format according to the FMI standard Note 25), providing compatibility with other types of CAE tools that support the FMI standard. In particular, all models created in SimulationX®, including the original custom library, can be converted to royalty-free FMU Note 26). This means that this system can also be used by external SA developers using other types of CAE tools. The system also has a user account management function mainly for external end users to expand the range of users of the service. At the same time, we are considering providing a high-security API. External end users who have registered in advance can log in to their own user account from the user login screen provided by the system. After successful login, they can receive an authentication token (Fig. 17).

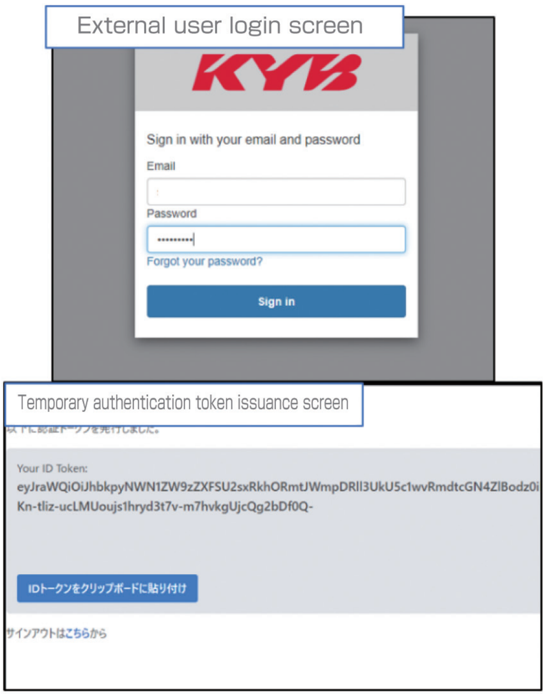

Fig. 17 External user login screen and authentication token issuance screen

External end users can enter the authentication token via the CAE tool by importing the pre-delivered FMU into the tool. The FMU only allows communication using the authentication token. Communication with the ML model cannot be established unless the access is allowed by the system. This feature allows all end users, whether internal or external, to access the SA damping force simulation using the ML model (Fig. 18).

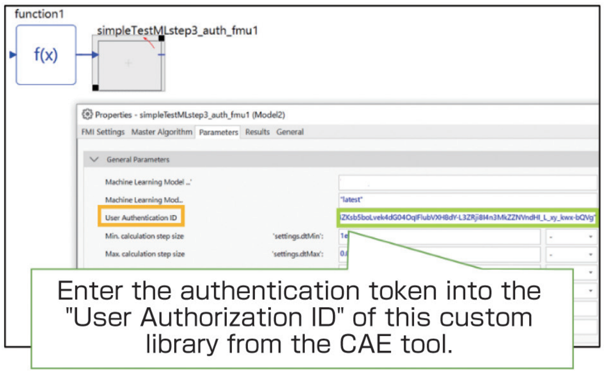

Fig. 18 API request with temporary authentication token screen

To reduce the risk of invalid use of the authentication token, the system is configured to disable the token itself after a certain period of time has elapsed since the user's login. Even if the authentication token is leaked externally, any request to the API with a token that has been disabled after a certain period of time will be rejected by the system. The system continuously keeps an access history and has a function to warn the system administrator about any suspicious unauthorized access.

Note 23) An API available on the Internet.

Note 24) Processing in the server zone that is invisible to users.

Note 25) Functional Mock-Up Interface: An open standard for exchanging and connecting dynamic simulation models between standardized tools of different types.

Note 26) Functional Mockup Unit: An execution module based on the FMI. The FMU can keep models private and use them regardless of the type of CAE tool.

7. 5 Monitoring the Quality of Training Data

To verify the accuracy of the ML model of this system, it is necessary to check the inference results of the model with the actual FEM calculation results. When specification information is input by an end user, the system outputs the prediction results of the ML model based on the inference endpoints. Obtaining accurate FEM calculated values for the results requires time and cost, so it is difficult to verify the accuracy of an ML model in operation. For the purpose of this system, we then focused on the quality of the training data that affects the model accuracy.

The SA specification information entered to use the AI service is recorded in the system as the usage history of the ML model. The system can compare the distribution of data used to train the model with the distribution of data actually used.

Statistics are automatically computed and periodically compared for verification. The results of the comparison are automatically reported to the ML model administrator via internal communication tools, so that they can be used to consider retraining the model (Fig. 19).

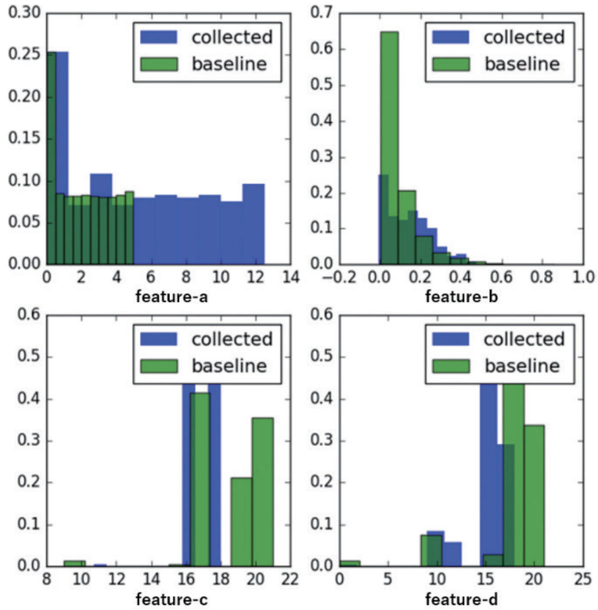

Fig. 19 Example of comparison of data distribution by feature set of training data and inference data

8 Concluding Remarks

By degenerating the detailed FEM model, which requires long computation time, into the ML model, we built the computational foundation to be implemented in 1D-CAE, which requires only short computation time despite its high accuracy. We also built the ML model operational management platform to enable staff to share the controlled high-quality ML model.

These technologies we have developed will help solve all MBD technical issues shown in Fig. 3 (seamlessly proceeding from 1D system study to 3D study, more efficient 3D geometry study, and model circulation among related companies and internal departments). We will actively apply the technologies to model development both inside and outside KYB, trying to maximize the effectiveness of the MBD.

Finally, we would like to take this opportunity to express our sincere gratitude to the CAE software vendors and cloud vendors, and staff of the related sections of KYB, who have cooperated in this project.

References

1) SHIMADA Minoru, MITSUSHIMA Koji, WATANABE Hirohito: "Introduction and Establishment of CAE in KYB Corporation", KYB Technical Review No.24 (April 2002).

2) MIYAUCHI Yuuki, ISASHI Ryosuke, TAKINO Shinsuke: "Initiatives to Develop Digital Human Capital", KYB Technical Review No.66 (April 2023).

Authors

MANTANI Hiroaki

Joined the company in 2009.
CAE Dept. (Gifu Office), Engineering Div.
Engaged in CAE mainly for performance prediction of automotive shock absorbers.

SAGEHASHI Ryota

Joined the company in 2019.
Digital Strategy Sect. (Stationed in Sagami Office), Digital Trans-for-mation Improvement Div.
Engaged in system development related to data analysis.