ISSN 1345-3041

Mar.2026 / Vol.193

Mitsubishi Electric

ADVANCE

Digital Technologies for Sustainable Business Growth (The second part)

MITSUBISHI ELECTRIC
Changes for the Better

March 2026 / Vol. 193
Mitsubishi Electric

ADVANCE

Digital Technologies for Sustainable Business Growth (The second part)

• **Editorial Inquiries**
Fumitoshi Yoshikawa
Corporate Productivity Planning & Engineering Dept.
Fax: +81-3-3218-2465

Mitsubishi Electric Advance is published on line quarterly (in March, June, September, and December) by Mitsubishi Electric Corporation.

CONTENTS

Technical Reports

Precis

The Mitsubishi Electric Group positions sustainability as a cornerstone of its management and is transforming itself into an “Innovative Company” that creates new value without fear of taking on risks. The evolution of digital technologies is the key to transforming into an Innovative Company. We will introduce specific initiatives related to advanced digital technologies in two parts: Part 1 (previous issue, Mitsubishi Electric ADVANCE Vol. 192, December 2025) and Part 2 (this issue, Vol. 193, March 2026).

Solution Creation Methods Utilizing Data Analysis for Social Infrastructure Systems

Authors: *Yohei Ueno*, Ken Imai**

**Advanced Technology R&D Center*

Abstract

We focused on supervisory control systems of social infrastructure and developed a method that promotes understanding of operations and the creation of solutions, by analyzing the vast amount of measurement data recorded in such systems. Operations of social infrastructure involve multiple overlapping goals such as safety, economic efficiency, and environmental performance, and it is difficult to systematize them. Accordingly, we developed a method by drawing on operators' cognitive approaches that constructs plant state models from measurement data. By using this method, we can create patterns of a vast amount of value data and more easily discover operational goals and challenges from the data. We also believe that using the analysis results in system requirements analyses will promote shared understanding between operators and developers and lead to the co-creation of solutions.

1. Introduction

Water and wastewater, roads, rivers, railways—social infrastructure operations are not only vital foundations supporting the lives of people, but, due to their scale, are characterized by having a large impact on society through energy use, environmental load, and other factors. Therefore, in these social infrastructure operations, in addition to safe and reliable operations, appropriate operations are also required from economic and environmental perspectives.

Moreover, unlike typical factory production processes, social infrastructure is characterized by being significantly affected by external factors such as weather and citizen dynamics. To handle various potential situations, buffers are provided for each equipment, and operations are carried out based on the flexible judgments of operators; however, this situational judgment and degree of freedom are factors that make the transfer of skills in operations.

To address such challenges, efforts are being considered that leverage rapidly evolving IoT technologies, and initiatives to simultaneously improve the efficiency and enhance safety of infrastructure operations are being examined[(1)(2)]. Various proposals exist, such as operation automation, predictions and anomaly detection, but in this paper we focus in particular on operation data and measurement data accumulated in supervisory control systems for social infrastructure, and describe a technology that, by leveraging this information, quantitatively identifies the decision criteria for operations that had been tacit knowledge of operators. By organizing this information—which had traditionally been accumulated as individual know-how—we can identify important operational conditions and issues. We believe this will lead not only to more advanced operations but also to the creation of a wide range of solutions, such as training young operators.

2. Supervisory Control Systems and Measurement Data

2.1 Characteristics of data accumulated in supervisory control systems

Social infrastructure operations often have extremely large areas as the scope of supervisory control. To collect information efficiently, monitoring systems are installed, and methods are adopted that aggregate and display the value for each equipment; in large plants, the number of display signals can reach into the thousands. Operators set control command values while inferring current conditions occurring at the plant and the device status, based on the wide variety of values displayed on the monitoring system. In general, the evaluation metrics for plant operation include multiple types, such as operating cost, safety and environmental impact. Many of these metrics involve trade-offs, where improving one adversely affects another, so operations are carried out while balancing the whole. Furthermore, when weather changes, disasters or device failures occur, safety may be prioritized more than in normal times; in other words, the priority of evaluation metrics can change depending on conditions.

As outlined above, data accumulated in supervisory control systems encapsulates various operational goals and evaluation metrics that change from moment to moment. When analyzing and interpreting such data, it is difficult to treat them as a single processing or function; only by correctly classifying and organizing the operational goals according to conditions does it become possible to extract information that reflects the reality of operations. This approach can be applied across various operations, and analyses and evaluations are being conducted in fields such as water and wastewater, river management, electric power, railways, and building management.

2.2 Classification and modeling of measurement data

In interpreting the time-series data accumulated in supervisory control systems, we conducted modeling while referencing the cognitive methods operators use through system operation. Using reservoir water level in water and wastewater as an example, operators not only read the value displayed by the system, but simultaneously add interpretations such as “water level is high” or “water level is low.” Here, by applying statistical methods to each value, we automatically set ranges such as “high water level” and “low water level,” and we developed a method that defines plant states by combinations of those ranges[3]. With this method, even for the same event of the water level dropping by 1m, if the change occurs within the “high water level” range, the change in the plant state is small, whereas a change from “high water level” to “low water level” represents a large change in state—enabling a nonlinear expression. This kind of plant state model is particularly effective when observing changes before and after an operational input, making it easier to identify the intent and purpose of the operation from an overarching perspective.

Figure 1 shows the method for classifying measurement data. When the operator intentionally controls the reservoir water level, the distribution is not uniform; it often splits into several clusters. Thus with this method, we automatically determine boundary values between clusters using statistical techniques, and by labeling each cluster, we classified the plant’s states. By treating changes in time-series data not merely as numerical differences but as state transitions, it becomes easier to express even complex plant operations.

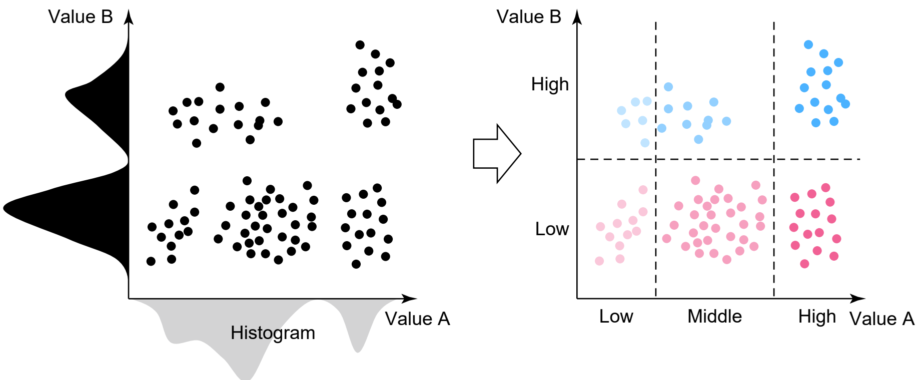


Fig. 1 Measurement data classification method

3. Understanding Operations and Extracting Issues

3.1 Data visualization

Data recorded in supervisory control systems of social infrastructure—such as water and wastewater, electric power, and buildings—often exhibit patterns synchronized with the life rhythms of the end users, namely citizens. This data is characterized by being based on a 24 hour periodic pattern, while fluctuating under factors such as weather and day of the week. By applying the analysis method described in 2 to long-term data recorded in the supervisory control system, it becomes possible to extract typical patterns, such as weekdays against holidays, or fair weather versus rainy weather. Moreover, since there are patterns that do not belong to the typical pattern classifications, investigating them makes it possible to comprehensively find events that occurred as exceptions. Such exceptional patterns often include cases where operation differs from normal due to device inspections or failures, as well as responses to sudden weather changes such as localized heavy rain.

Figure 2 shows the results of pattern classification at our office, using 8 months of electric power usage data. The same color in the figure indicates similar power-consumption states, and X-axis direction represents dates, Y-axis direction represents times, in a 2 -dimensional graph. In almost all periods, 0:00

to 5:00, and 21:00 to 23:00 are the same color, which indicates that employees have left the office. 5:00 to 21:00 shows increased power consumption; in particular, 8:00 to 17:00 shows higher consumption, which matches working hours. Holidays are indicated with 24 hours across the Y axis where the same color continues, with every 7 days appearing as similar patterns.

From a long-term perspective, the appearance pattern of colors also changes over spans of several months, indicating seasonal variation.

Indeed, the electric power consumption of air-conditioning units varies greatly between summer and autumn, and these changes are reflected as changes in the color patterns.

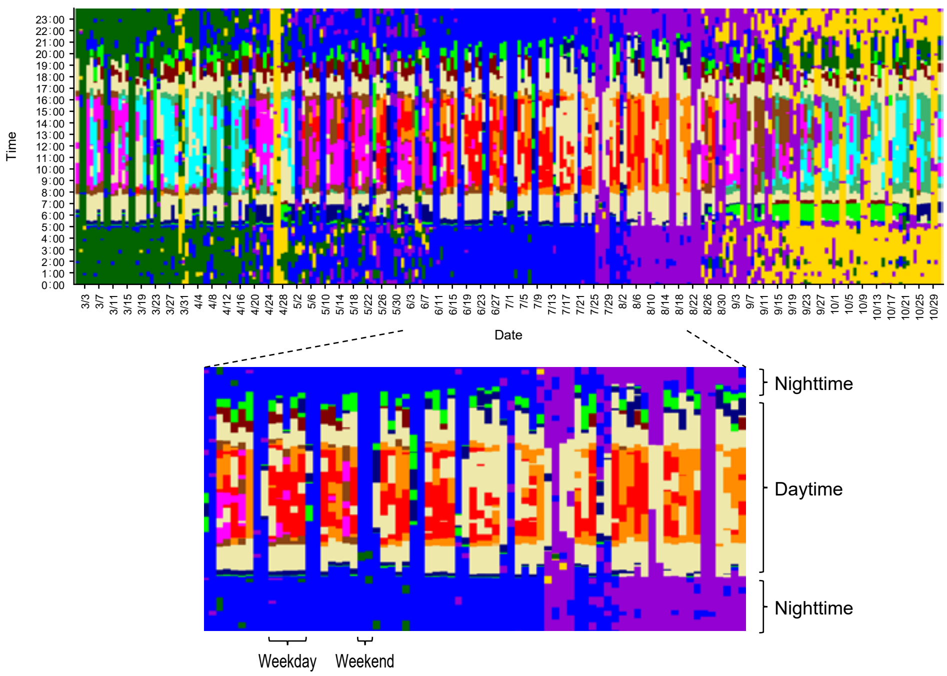


Fig. 2 Visualization of office building power consumption

3.2 Understanding operational objectives

In social infrastructure operations, evaluation indicators include not only cost reductions such as power and chemical costs, but also multiple items such as impacts on water quality and the environment, and safety. And depending on various conditions such as weather and events, the objectives considered particularly important also change. When interpreting operations from data, the goal settings for each period need to be understood correctly. Traditionally, to understand social infrastructure operations, investigations were conducted by conducting interviews with operators. However, it was difficult to comprehensively identify goal settings that change by period, and, given differences in individual viewpoints, it was also difficult to consolidate all opinions.

Therefore, to understand operations, we decided to use the visualization method outlined in 3.1. Using this method, we can aggregate and display changes over time and modifications to operating methods due to seasons. By classifying patterns on a daily basis, we were also able to quantitatively determine what operations were carried out, and how frequently. Figure 2 uses an office building as an example, allowing identification of when weekday–holiday differences and seasonal changes occurred. Given that periods such as “summer” can be defined from operation patterns, it has become possible to accurately estimate how much effect might be obtained from operational improvements during “summer” and what impacts there would be.

As outlined above, creating patterns using a plant state-transition model makes it possible to more clearly identify operational goals and issues. Furthermore, using these analysis results in interviews with operators is expected to help extract tacit knowledge and recall issues, enabling the discovery of issues that occur infrequently but have significant impact when they do occur, without overlooking them.

4. Exploring Solutions

Operational support solutions implemented in supervisory control systems generally often provide direct information, such as prediction functions and guidance functions. While these functions are effective for pre-assumed issues, for goals that need to be addressed in the future, such as population decline and environmental considerations, it has been difficult to guarantee output accuracy, and application and adjustment have been time-consuming.

For goals that require long-term efforts, the PDCA cycle is generally often used to drive continuous improvement. Going forward, as we transition from an era of infrastructure construction to one of maintenance and management, a similar approach is needed for supervisory control systems, and we believe that support functions will need to be implemented for each phase of the PDCA cycle. Examples of this are supporting the identification of issues through analysis of measurement data, or impact assessments when operations are changed—solutions that go beyond conventional guidance functions—can be considered; however, in developing these functions, rather than simply brainstorming, we believe that we can identify the issues that truly need to be solved and propose effective functions, by first understanding actual operations.

Using water supply as an example, the following outlines a case of exploring solutions using measurement data. Here, we applied the analysis method outlined in Chapter 2 to water supply data, visualized operation patterns, and classified them into patterns such as “fair weather” and “rainy weather.” Here, “rainy weather” refers to a period in which river water quality (turbidity) temporarily worsens due to rainfall, and differs from the actual rainfall period. Typically, increases in river turbidity continue for a certain period even after the rain stops; therefore, in water treatment processing, “rainy weather” is defined as “the period during which operations for rainy weather continue.” Using the analysis results, interviews with water supply utilities revealed a need for operational support for chemical injection control during rainy weather, leading to the start of concrete solution discussions.

Figure 3 shows examples of solutions created through discussions between our company and water supply utilities. In the functional study, we created various use cases and ideas, not limited to presenting guidance information, but also including functions for reviewing past cases and evaluating impacts associated with changes in operation patterns. All of these were obtained by sharing understanding between operators and developers based on information derived from value data analysis, and we found that measurement data analysis and patterning were also effective for facilitating communication among stakeholders.

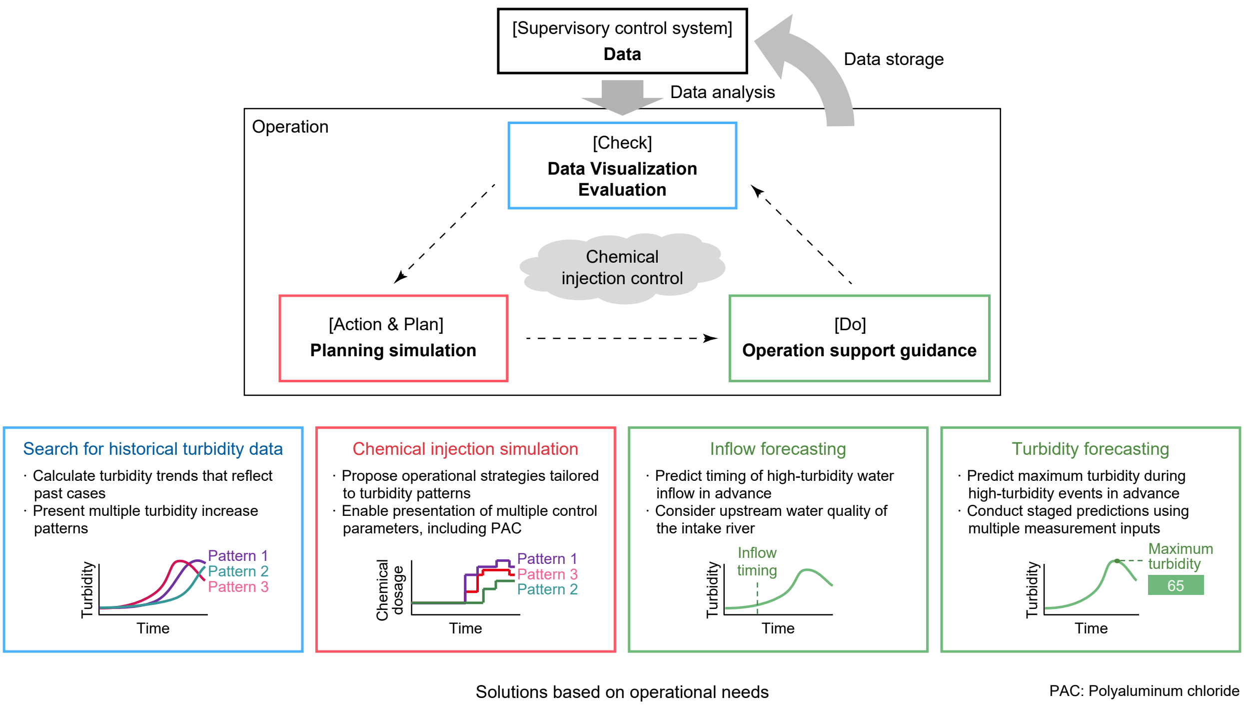


Fig. 3 Chemical injection solutions for water supply systems

5. Conclusion

In the social infrastructure business field, we introduced here a method for creating plant state-transition models based on time-series data accumulated in supervisory control systems, and case examples of solution creation using the state-transition models. Needs related to social infrastructure operations change daily, and we believe that quantitatively identifying and understanding the operations currently being performed through data analysis is effective for extracting new needs and creating solutions. We have also started examining solutions that apply this analysis method outside the water and wastewater business introduced here, and will continue to improve the method and develop tools as foundational technology for solution creation.

References

(1) Gotoda, K., et al.: Data Utilization Solutions for Water and Wastewater, Mitsubishi Denki Giho, 97, No. 7, 5-01-5-07 (2023)

(2) Nishida, K., et al.: Development of a Raw-water Turbidity Prediction and Operational Support System During Heavy Rain, 2024 National Conference (Water Supply Research Presentations) Proceedings, 190-191 (2024)

(3) Imai, K., et al.: Study on the Applicability of Plant State Classification Technology to Supervisory Control Data at Wastewater Treatment Plants, Proceedings of the Sewerage Research Presentation Conference, 60, 1192-1194 (2023)

Performance Prediction Technology for Rubber Materials Utilizing Machine Learning

Authors: *Yutaro Hanawa*, Koji Nakatani**

**Advanced Technology R&D Center*

Abstract

Various rubber components are used in electrical products. To identify rubber components that can be applied to products from among the many varieties available, multiple performance evaluation tests need to be conducted, which requires an enormous amount of time. Therefore, to improve the efficiency of performance evaluation tests for rubber components, we examined performance prediction technology for rubber components using a regression model trained via machine learning on component performance and analytical data(1). In this study, by utilizing computer-based virtual experiments, including Finite Element Method (FEM) analyses, for collecting training data, we reduced training-data collection time to one-tenth of that required for repeated physical experiments. The developed model was able to predict the nonlinear load–displacement relationship of rubber components with an average accuracy of 84%.

1. Introduction

In recent years, advances in machine learning have enabled faster and more efficient development of materials, and higher-performance materials are expected to be commercialized in shorter time frames. For materials users such as Mitsubishi Electric, it will be important to quickly find and utilize materials that can be applied to their own products. As such, we are developing technology that uses machine learning to predict the performance of resin-based materials such as plastics and rubber, as well as components made from these materials. Because resin-based materials have drawbacks such as rapid degradation under environmental condition, a variety of time-consuming performance evaluation tests are needed before being applied in products. If the performance of resin-based materials and components can be estimated, it becomes possible to prevent rework due to evaluation test failures and to narrow down the candidates for testing, thereby improving the efficiency of performance evaluation.

This paper outlines the development of performance prediction technology for rubber components that contributes to such efficiency in performance evaluations.

2. Investigations to Improve Efficiency of Training Data Collection

This chapter outlines the investigations carried out to improve the efficiency of training data collection.

2.1 Collection of training data leveraging virtual experiments on a computer

A feature of this development is that the training data required for machine learning were collected through computer-based virtual experiments. An overview is shown in Fig. 1. The performance of rubber materials is known to be influenced by micro-structure, such as the degree of dispersion of carbon black mixed in as a reinforcing material(2)(3), and we examined extracting features from Transmission Electron microscopy (TEM) images to use as explanatory variables. However, prototyping materials, evaluating performance, and acquiring TEM images require an enormous amount of time. Therefore, using CAD, we created a model that represents TEM images, and we extracted features related to the micro-structure from this model. Furthermore, we calculated the performance of rubber components to use target variable by FEM analysis.

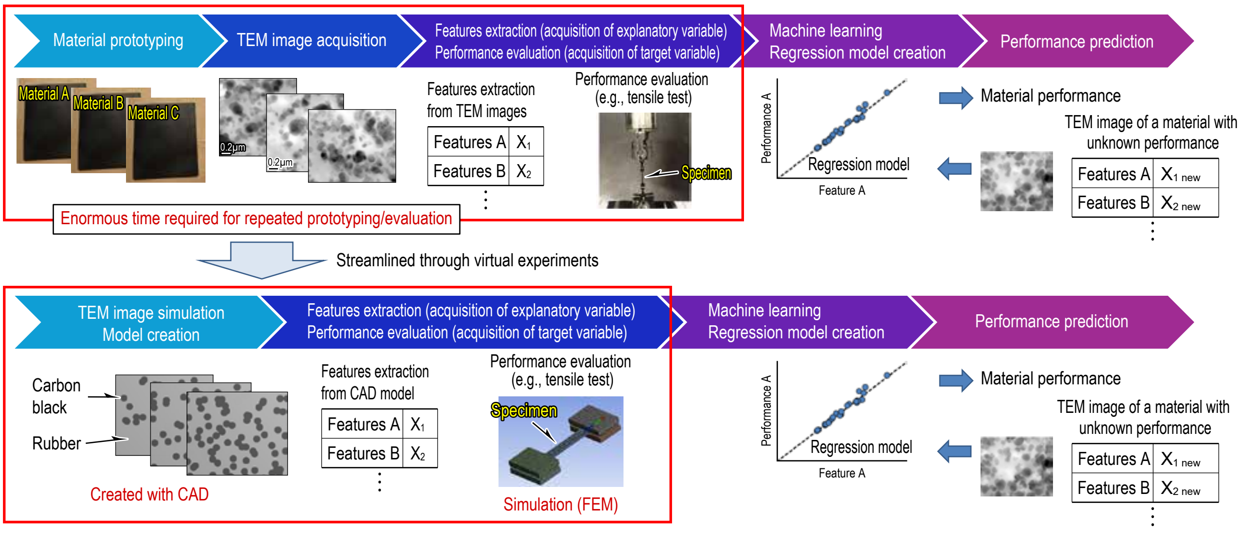


Fig. 1 Overview of the performance prediction technology for rubber components

2.1.1 Creation of a CAD model simulating TEM images and extraction of explanatory variables

An overview of the CAD-created model that simulates TEM images is shown in Fig. 2. The model was created by randomly placing circles to represent carbon black on a 2D plane. It is known that at the interface between the carbon black mixed into the rubber and the rubber matrix, there exists a middle layer with higher elasticity than the rubber[(4)], and a model of this middle layer was created in a similar manner. In this paper, we created 100 models in which the number and diameter of circles varied, and from each model we extracted features such as the fraction of area occupied by carbon black.

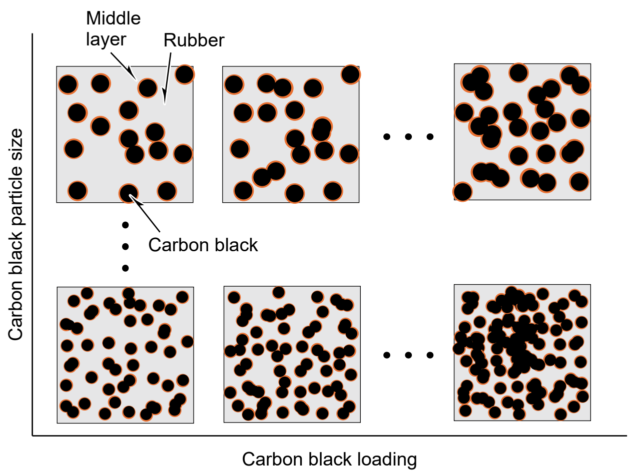


Fig. 2 Overview of CAD-simulated models of TEM images

2.1.2 Acquisition of target variables by FEM

In this paper, we obtained the performance of the target variable of rubber components through FEM analysis. An overview of the FEM analysis is shown in Fig. 3. Component performance can be evaluated using metrics such as the load–displacement relationship, creep deformation, and sealing performance. In this paper, as a simple example, we outline results predicting the load–displacement relationship of a strip specimen. Because the micro-structure such as degree of caebon black dispersion of rubber, affects the performance of macro-scale components, we conducted a Multi-Scale Simulation that couples the properties and behavior of structures at different scales. In the Multi-Scale Simulation, the load–displacement relationship of the strip-shaped component is calculated in FEM analysis 2. The stress–strain relationship required as material property in FEM analysis 2 was obtained by FEM analysis 1 using the CAD model created in section 2.1.1. With this method, differences in material properties arising from differences in the rubber's micro-structure can be reflected in the FEM analysis that simulates macro-scale performance evaluation tests. Note that the elastic modulus values of the rubber, carbon black, and middle layer used in

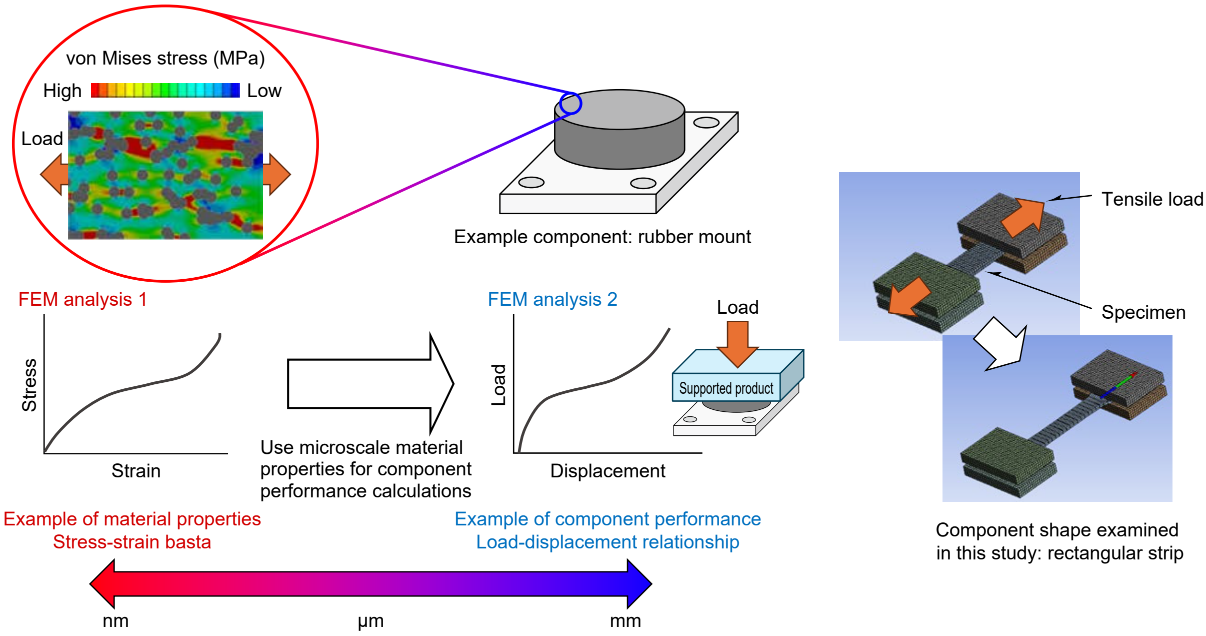


Fig. 3 Overview of the multi-scale simulation and the component shapes examined in this study

FEM analysis 1 were estimated from prototyping and evaluation of rubber without carbon black and from analyses such as Scanning Probe Microscopy (SPM) and Atomic Force Microscope (AFM).

2.2 Model training

We performed machine learning using the explanatory variables and the target variable collected in the virtual experiments of sections 2.1.1 and 2.1.2, and built a prediction model. If the training data were collected through actual experiments, it would be expected to take about 50 days; however, in this paper, it was completed in about 5 days. That is, by conducting virtual experiments on a computer, we were able to shorten the collection of training data to one-tenth of the time.

3. Accuracy Verification of the Prediction Model

In Chapter 2, we constructed a prediction model, into which we input features extracted from TEM images of acrylonitrile-butadiene rubber (NBR) with varying carbon black particle size and amount, and predicted the load displacement relationship. The results are shown in Fig. 4. Across all 12 conditions examined in this study, regardless of the particle size or amount of carbon black, the load displacement relationship of the rubber component could be predicted with an average accuracy of 84%. In addition to prediction, machine learning can evaluate the importance of each explanatory variable. Feature-importance analysis showed that carbon black content was the most influential feature, followed by the standard deviation of the contour lengths of the circles used to represent carbon black particles and the spatial dispersion of the centroids of those circles. This result suggested that, for the deformation of rubber components, not only the carbon black content but also the particle size distribution and the degree of dispersion contribute.

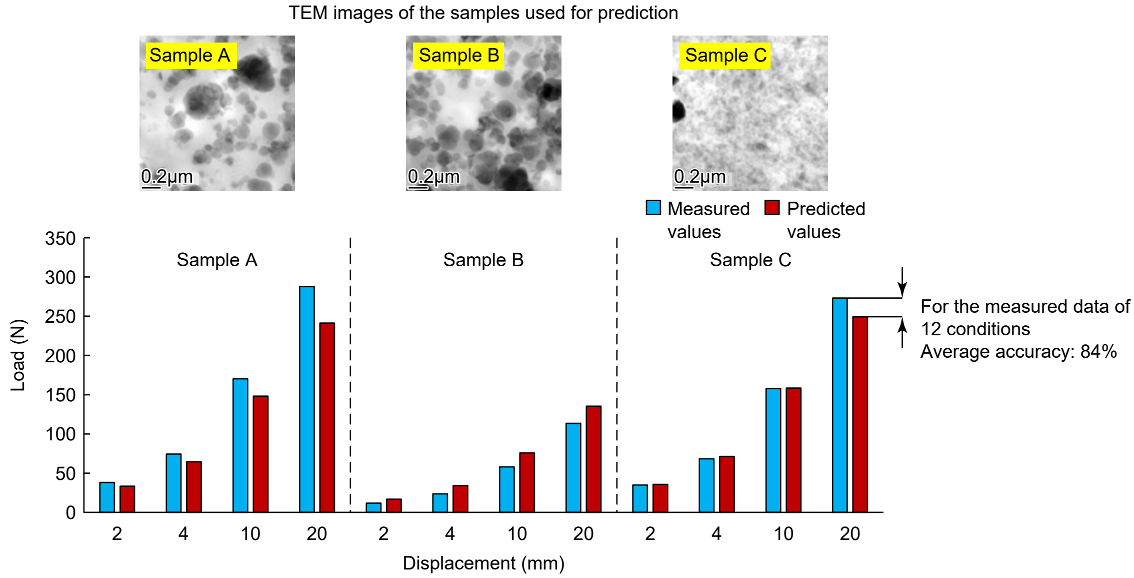


Fig. 4 Results of accuracy verification for the machine learning model

4. Conclusion

Using training data collected from compute-based virtual experiments, we developed a machine learning model that predicts the deformation of rubber components under load. In machine learning, acquiring training data for machine learning is often costly; however, in this paper we established foundational techniques for virtual experiments using CAD and the FEM to enable more efficient data collection, and verified their effectiveness.

Going forward, we will consider developing models capable of predicting various performance characteristics, such as vibration isolation performance, sealing performance and fatigue resistance, using not only features from TEM images, compositional data, and infrared spectral images.

References

(1) Hanawa, Y.: Development of Performance Prediction Technology for Rubber Materials Utilizing FEM and Machine Learning (2), 2023 Annual Meeting of the Society of Rubber Science and Technology, Japan, B-9 (2023)

(2) Nakada, T.: Carbon Black - Focusing on Rubber Reinforcement Performance, Journal of the Carbon Society of Japan, No. 106, 111-120 (1981)

(3) Sakakibara, A., et al.: Improvement of Rubber Properties by Carbon Black, Journal of the Society of Rubber Science and Technology, Japan, 73, No. 7, 385-391 (2000)

(4) Takano, N., et al.: 3D Morphology Analysis and Finite Element Modeling of Nanoparticle-dispersed Materials, Journal of the Society of Materials Science, Japan, 57, No. 5, 423-429 (2008)

Advancements in Material and Process Design through Machine Learning: Application to Epoxy-Based Adhesive Materials

Authors: *Keisuke Ando*, Yoshiyuki Kamo**

**Advanced Technology R&D Center*

Abstract

During the development of materials technologies supporting higher performance and reliability of electrical devices and reduction of environmental impact, design parameters such as the chemical composition of raw materials and manufacturing processes need to be adjusted. Yet changes in design parameters can dramatically alter the microstructure and state of materials, so developing the material with the desired structure and state requires tremendous time through trial and error.

Accordingly, we developed a method that first estimates promising design parameter regions via machine learning using known experimental data, and then selects as the next experimental point the condition with high predictive uncertainty and sparse experimental data, thereby efficiently clarifying promising design parameter regions. By applying this method to the development of epoxy-based adhesive materials, we also reduced the number of experiments by about 80% compared with conventional practices. This technique is anticipated to be applicable not only for the development of new materials, but also for various uses such as improving existing materials and manufacturing processes.

1. Introduction

Materials technology is a cross-disciplinary foundational technology that has the potential to lead to solutions for a wide range of industrial and societal challenges(1). In recent years, there have been active efforts into research and development of material technologies incorporating Materials Informatics (MI) across industry, government and academia, around the world(1)(2). At Mitsubishi Electric, we manufacture electrical device products that deliver value in every environment, from home appliances to space. Achieving greater added value for these devices, such as improved performance, enhanced reliability, and reduced environmental impact, requires the development of new materials. After generating candidate materials, design parameters such as the chemical composition of raw materials and manufacturing processes need to be adjusted in materials development so that they can be put to practical use in devices; however, even slight differences in conditions can lead to dramatic changes in a material's microstructure and state. Therefore, if promising design parameter regions that yield the desired material can be identified from the outset, efficient and innovative development is anticipated.

Here, we examined a method leveraging MI that actively and efficiently clarifies promising design parameter regions from known experimental results, taking an epoxy monolith sheet used for novel epoxy-based adhesive materials as a specific example. In an epoxy monolith sheet, the microstructure changes depending on the weight ratio of three raw materials, and there are also weight ratios of raw materials at which an epoxy monolith sheet cannot be obtained. Formation of the epoxy monolith sheet is also strongly influenced by the reactivity of the raw materials, so it also depends on differences in the chemical structure of the raw materials. Therefore, identifying design parameter regions such as the raw material weight ratios at which an epoxy monolith sheet can be obtained has in the past required experimental trial and error and a tremendous amount of time.

In this paper, we developed an MI method that first estimates promising design parameter regions using machine learning with known experimental data, and then selects as the next experimental point as the next experimental point the condition with high predictive uncertainty and sparse experimental data, thereby efficiently clarifying the promising design parameter regions—we then outline a case study applying the method to an epoxy monolith sheet.

2. Method for Estimating Promising Design Parameter Regions

For objectives like this one, under the premise that conditions such as the weight ratio of multiple raw materials are variable, one approach of clarifying parameter regions that yield the desired material is to use active learning to estimate phase diagrams that show thermodynamic equilibrium according to the temperature and composition of the material. Dai et al. showed that by using Bayesian optimization based on posterior prediction with a Gaussian process, the number of sampling points required in estimating a phase diagram with two phases can be greatly reduced(3). Terayama et al. showed that uncertainty sampling, which focuses exploration on uncertain regions near phase boundaries, makes efficient sampling possible even for phase diagrams with multiple phases(4). The phase diagram estimation addressed by these prior studies handle the thermodynamic equilibrium, so thermodynamic calculation can often be leveraged, and consequently a relatively large number of samplings can be tolerated. On the other hand, for materials like the epoxy monolith that are characterized by changes in microstructure depending on the manufacturing process, experimental verification is required, so there are severe constraints on the number of samplings. Therefore, to efficiently estimate promising design parameter regions, for uncertainty sampling, we introduced a procedure that preferentially proposes conditions that are far from the training data that have been tested to date.

2.1 Estimation framework

Figure 1 shows the framework for estimating promising design parameter regions. Here, we illustrate the case where the ratios of three raw materials A, B, and C are selected as the design parameters. The ternary diagram in Fig. 1 represents the raw material weight ratio; the closer to the vertex for raw material A, the greater the amount of raw material A.

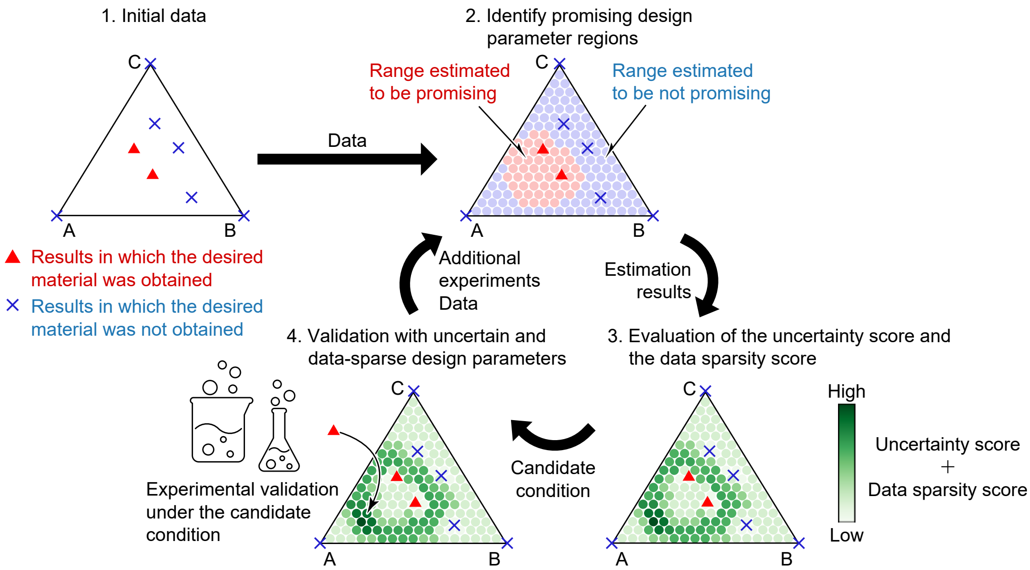


Fig. 1 Framework for identifying the promising design parameter regions

First, we prepare two types of outcomes as the initial training data: cases in which the desired material was obtained and cases with undesirable outcomes. Here, given that it is obvious that weight ratios that lack the required raw materials yield undesirable results, we added to the training data that undesirable results are obtained at the vertices of the ternary diagram.

Next, by performing machine learning using this training data, we built a classification prediction model to determine whether the design parameters are promising, and estimated promising design parameter regions. Based on these estimation results, we evaluated the uncertainty score and, as a penalty term, calculated the proximity to the training data, thereby evaluating the data sparsity score. The uncertainty score is a metric that evaluates the uncertainty in the prediction model's classification of whether the design parameters are promising, and given that the boundary between the region estimated as promising and the region estimated as undesirable is calculated to be the most uncertain, the entire perimeter of the boundary is evaluated as equally uncertain. Therefore, designing an objective function that adds the data sparsity score to the uncertainty score makes it possible to uniquely propose candidate conditions that are both uncertain and have not yet been experimentally tested. Using the candidate conditions proposed by this method, we carried out experimental validation and checked whether the desired material could be obtained. By adding these results to the training data and repeating the cycle of estimation, evaluation and verification, we also clarified the promising design parameter regions.

2.2 Preliminary verification of the estimation framework

To verify the operation of this method, we defined a virtual ground truth and tested whether the promising design parameter regions could be estimated. The interior of the ellipse shown with the black line in the figure is the region assumed as the ground truth, and Fig. 2 shows that the red region on the left denotes the region estimated as promising. It can be seen that in the first cycle, the estimated region does not coincide with the ground truth. Figure 2 on the right shows the distribution of the sum of the uncertainty score and the data sparsity score calculated from the estimation results. At the boundary between the region estimated as promising and the region estimated as undesirable, the sum of the uncertainty score and the data sparsity score is rated highly, and the asterisks are proposed as candidate conditions to be verified next. When a candidate condition falls within the region assumed as the ground truth, we added virtual data indicating that the desired material was obtained; when it falls outside, we added virtual data indicating that the desired material was not obtained.

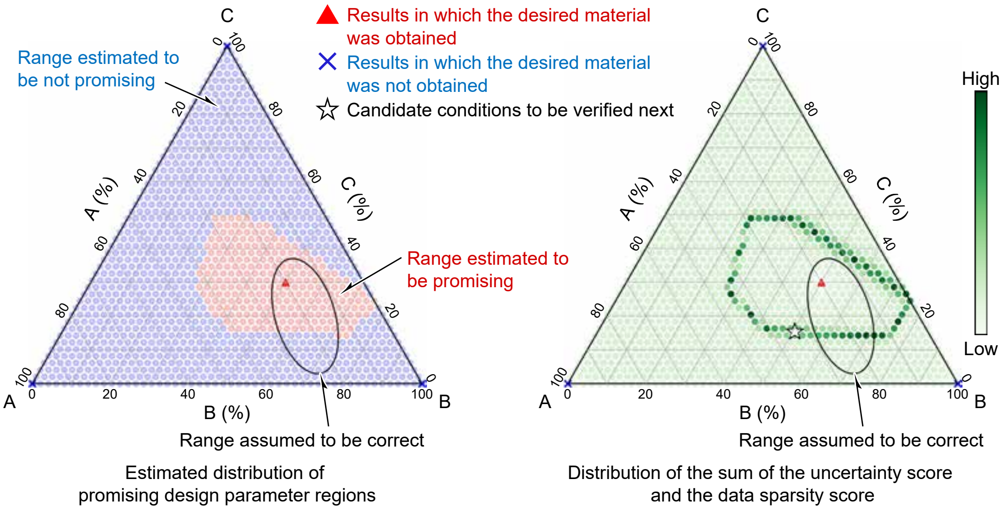


Fig. 2 Results of the estimation for the virtual data in the first cycle

The estimation results after repeating the estimation framework for 13 cycles are shown in Fig. 3. With the addition of virtual data 13 times, the region estimated as promising showed results very close to the ground truth, confirming the potential to derive promising design parameter regions with few experiments.

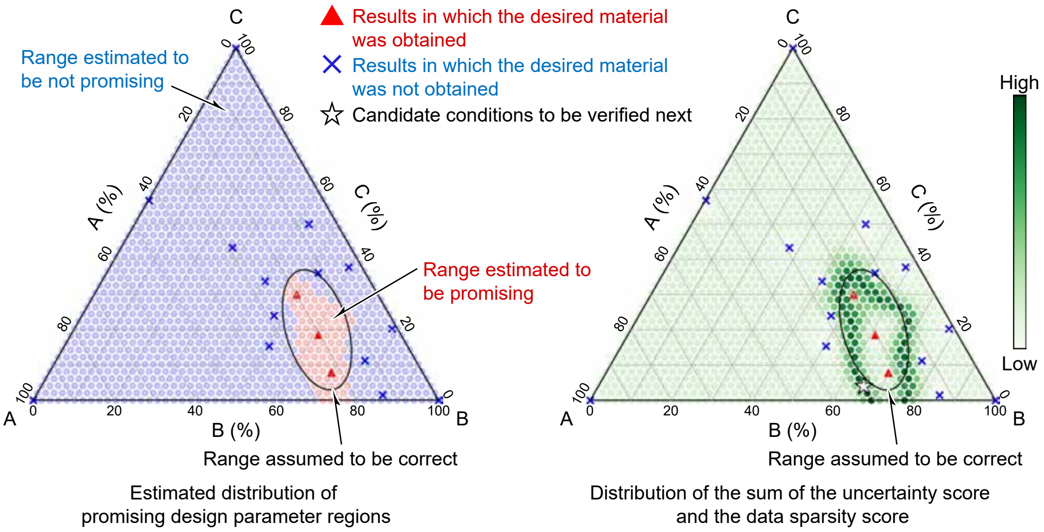


Fig. 3 Estimation results for the virtual data after 13 cycles

3. Application to the Development of New Epoxy-Based Adhesive Materials

In this chapter, we outline a case where this method was applied to an epoxy monolith sheet used in new epoxy-based adhesive materials.

3.1 Epoxy monolith sheet used in new epoxy-based adhesive materials

Epoxy-based adhesive materials, widely used in electric motors, semiconductor devices, and aerospace devices, are materials with high strength and excellent heat resistance, but they also have the drawback of being resistant to plastic deformation and brittle. In electrical devices, components repeatedly expand and contract due to temperature cycles during operation and shutdown, so greater flexibility of epoxy-based adhesive materials is required. We developed a sheet adhesive material in which an epoxy monolith sheet with internal continuous pore is impregnated with adhesive components, and have shown that by making the flexible epoxy monolith sheet function as a stress-relief layer at the bonded joint, it contributes to improving the heat cycle resistance of epoxy-based adhesive materials[(5)], and we are pursuing research and development toward further performance enhancement.

The epoxy monolith sheet is a thin film characterized by a co-continuous pore structure in which a mesh-like epoxy framework and voids are each three-dimensionally interconnected (Fig. 4)[(6)]. To achieve an epoxy monolith sheet, at least three raw materials need to be mixed: an epoxy resin, a curing agent and a porogenic solvent, and the microstructure varies depending on differences in the raw material weight ratio. If the raw-material weight ratio is not within an appropriate range, the epoxy framework becomes discontinuous and the sheet cannot maintain a self-supporting integrity, or the voids become discontinuous and cannot be impregnated with adhesive components, and it fails to function as an adhesive material. Therefore, clarification was needed of the raw material weight ratios that produce an epoxy monolith sheet, characterized by a mesh-like epoxy framework and voids that are each three-dimensionally interconnected.

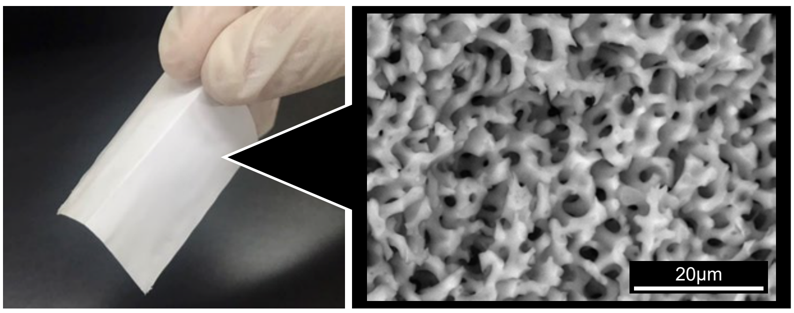


Fig. 4 Surface appearance and cross-sectional SEM image of the epoxy monolith sheets

3.2 Method for estimating promising design parameter regions to obtain an epoxy monolith sheet

In this paper, we focused on epoxy monoliths prepared using 4,4'-methylenebis (cyclohexylamine) (BACM) as the curing agent and poly (ethylene glycol) (PEG) as the porogenic solvent, with 2,2'-bis (4'-glycidyloxyphenyl) propane (BADGE) or 1,3-bis (N,N-diglycidylaminomethyl) cyclohexane (TETRAD-C) as the epoxy resin. For these two epoxy monolith systems, we set the weight ratios of the three raw materials as the design parameters, and estimated the promising design parameter regions for obtaining epoxy monolith sheets. First, we used as initial training data the result at a PEG concentration of 70 wt% where an epoxy monolith was formed in preliminary experiments, and the results for each raw material alone, where no monolith forms (the vertices of the ternary diagram). Based on the preliminary verification in section 2.2, while adding the results of additional experiments to the training data, we repeated the estimation framework for 13 cycles.

4. Estimated Results for Promising Design Parameter Regions where Epoxy Monolith Sheets are Obtained

Figure 5 shows the design parameter regions, estimated after 13 cycles, where epoxy monoliths are obtained. Focusing on the distribution of the experimental conditions, similar to the results of the preliminary verification in section 2.2, the experimental points that are not promising for monolith fabrication are distributed so as to surround the entire periphery of the ranges estimated to be promising for monolith fabrication, from which we consider that the promising design parameter regions have been estimated with high accuracy. In particular, given that the promising design parameter ranges differ for the monolith using BADGE as the epoxy resin and the monolith using TETRAD-C, we are able to estimate design parameter regions that reflect differences in the types of raw materials. By clarifying such promising design parameter regions, we became able to design materials after identifying the overall trends in epoxy monolith synthesis with a small number of experiments. Moreover, experiments through 13 cycles revealed that PEG concentrations of 40 wt% or lower did not yield epoxy monoliths. Because the number of experiments conducted at these concentrations was limited, even before it was definitively established that monoliths could not be obtained, we were able to focus on more promising experimental conditions. Whereas 66 additional experiments would traditionally have been required to test all conditions in 10 wt% increments, using this method we were able to identify the promising design parameter regions with 13 additional experiments. We reduced the number of experiments by about 80%, demonstrating that this method is effective for streamlining research and development (Fig. 6).

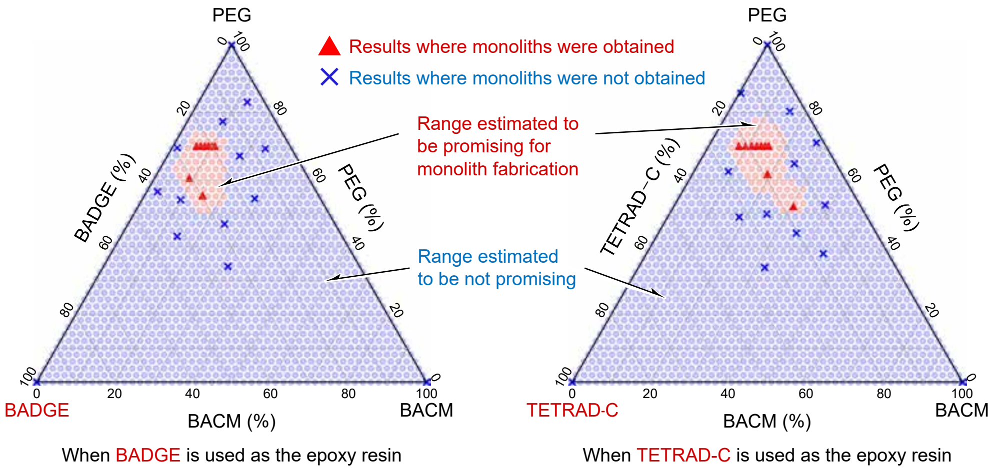


Fig. 5 Design parameter regions where the epoxy resin is obtained

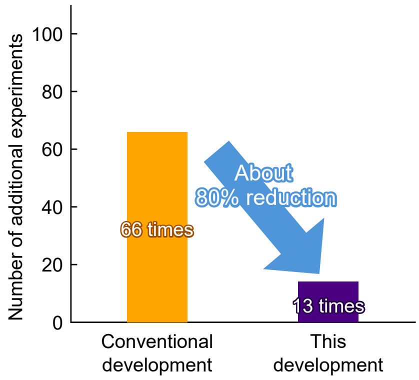


Fig. 6 Reduction in the number of experiments required to identify the promising design parameter regions

5. Conclusion

For efficient research and development of innovative materials, we developed a method that uses machine learning to efficiently select design parameters. Even in systems that include many design parameters, we identified a method to accurately estimate design parameter regions with a small number of experiments, and by applying this method to the development of new epoxy-based adhesive materials, we reduced the number of experiments by around 80% compared with conventional development. Given that this method can be extended to higher dimensions with three or more types of design parameters, we will proceed to apply it to even more complex and advanced materials development.

References

(1) Cabinet Office, Expert Panel on Materials Strategy: Issues and Directions for Consideration Toward Further Strengthening Materials Innovation Capability (2024)
https://www8.cao.go.jp/cstp/material/11kai/siryo1-1.pdf

(2) Japan Science and Technology Agency (JST), Center for Research and Development Strategy: Overview Report of Research and Development, Nanotechnology/Materials Field (2024)
https://www.jst.go.jp/crds/report/CRDS-FY 2024-FR- 04.html

(3) Dai, C., et al.: Efficient Phase Diagram Sampling by Active Learning, The Journal of Physical Chemistry B, 124, 1275-1284 (2020)

(4) Terayama, K., et al.: Efficient construction method for phase diagrams using uncertainty sampling, Physical Review Materials 3, 033802-1 to 033802-8 (2019)

(5) Kamo, Y., et al.: Stress relaxation and improved fracture toughness of metal bonding using flexible monolith sheets and an epoxy adhesive, Polymer Journal, 57, 203-214 (2025)

(6) Matsumoto, A.: Dissimilar materials joining using epoxy monoliths, Network Polymer, 38, No. 2, 93-102 (2017)

Domain-Specific Language Models for Manufacturing Industry

Authors: *Hayato Uchide*, Tatsuhiko Saito*, Shingo Oidate**, Masahiro Hiramori**, Shinya Taguchi**

** AI Transformation Innovation Center, ** Engineering and Development Center*

Abstract

With the advancement of Large Language Models (LLM), the practical use of Natural Language Processing (NLP) is progressing rapidly. Meanwhile, as models scale up, increases in computational cost and energy consumption, data privacy concerns, and constraints on real-time responsiveness among other issues are becoming apparent when operating them at manufacturing sites. In particular, in high-urgency situations directly tied to productivity—such as immediately identifying the cause of a production line stoppage and presenting response procedures—stable operation and high responsiveness of the model are required in restricted computing environments.

Against this backdrop, Mitsubishi Electric has developed a Small Language Model (SLM) specialized for the manufacturing domain, including the FA field. In this paper, in addition to continual pretraining and instruction tuning, we newly examined alignment capable of learning effectively even under limited data conditions. As a result, despite being a compact model with 1.8 billion parameters that can run on an edge device, it achieved an accuracy of 77.24% on a task that tests the correctness of knowledge in the FA field. This is anticipated to expand the scope of generative AI utilization in constrained environments such as manufacturing sites.

1. Introduction

With the advancement of LLMs, the practical use of NLP is progressing rapidly. In particular, OpenAI's ChatGPT[*1] and other general-purpose LLMs that run in the cloud deliver high performance across a wide range of language processing tasks, and are expected to be utilized in many industrial domains, including manufacturing. On the other hand, when using LLMs, the increase in computational cost and energy consumption that comes with larger models is becoming an issue.

Moreover, in operational settings, there are cases where simply using a general-purpose LLM in the cloud does not suffice. For example, at manufacturing sites, it is often difficult to send confidential information such as equipment manuals and incident response records outside of sites, and interactions with quick responses with on-site workers and equipment are frequently required. Specifically, these include immediately identifying the cause of a production line stoppage and presenting response procedures, providing support for confirmation during maintenance work, and making response decisions when abnormalities occur. In such cases, instantaneous information retrieval and responses based on confidential information such as equipment manuals and incident response records are directly tied to productivity and safety. In light of this backdrop, solutions that can be operated without relying on general-purpose LLMs in the cloud are needed. In particular, to complete secure, highly responsive processing on on-site terminals, it is important to develop technologies for high-performance SLMs that run on edge devices.

In the development and operation of SLMs, simply limiting the model size makes it difficult to achieve sufficient performance. Particularly in highly specialized domains such as manufacturing, there are many technical terms and expressions that are not included in general-purpose models. In highly specialized domains, appropriate domain adaptation is essential for generating appropriate responses tailored to user inquiries and on-site conditions. Therefore, continual pretraining using data from specific domains(1) and instruction tuning(2), and alignment to achieve responses that are natural and safe for users(3)—such fine-tuning methods—need to be appropriately combined.

*1 ChatGPT is a registered trademark of OpenAI OpCo, LLC.

In addition, to operate language models practically in constrained environments such as manufacturing sites, it is also important to achieve optimization while ensuring inference accuracy, considering the balance with response speed and resource usage. In use cases that require an instantaeous response in particular, designs that minimize response latency on site and techniques that ensure stable operation in restricted computing environments are important for achieving both model performance and practicality.

In this paper, we focus particularly on the FA field within the manufacturing industry and, using various documents related to our FA products, an SLM (Fig. 1) was built, and we outline this effort. Chapter 2 outlines representative fine-tuning techniques and practical examples of domain specialization using them. Chapter 3 outlines a practical example of running an SLM on an edge device using OSS. Finally, Chapter 4 summarizes the results of this work and future prospects.

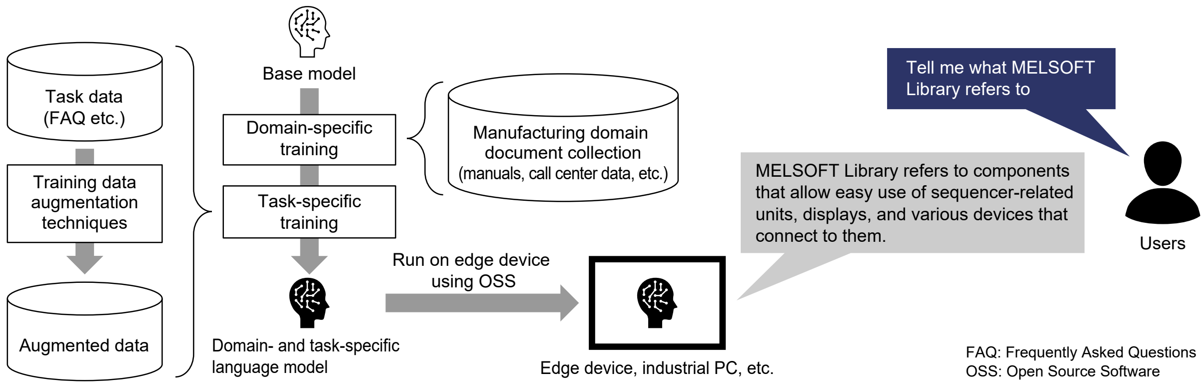


Fig. 1 A domain-specialized SLM for the manufacturing industry running on edge devices

2. Development of an SLM for Domain Specialization for the Manufacturing Industry

LLMs are pretrained using large amounts of general text data, thereby acquiring broad applicability across a wide range of language processing tasks. In practical applications, however, the model needs to be adjusted so that it fits specific languages, domains and use cases. For such post-training, there are several representative methods available, which are used selectively according to their objectives and targets.

In this chapter, we outline representative fine-tuning methods for maximizing language model performance, and then outline the domain specialization for the manufacturing industry approach undertaken in this development.

2.1 Continual pretraining

Continual pretraining is a technique that performs pretraining again on an existing pretrained model using a new corpus. This enables the model to additionally learn vocabulary, expressions and contextual knowledge in specific languages (e.g., Japanese) and domains (e.g., manufacturing industry, healthcare, law).

Because continual pretraining is based on self-supervised learning, it has the advantage that explicit task definitions and annotations are not required, making it easier to leverage unstructured data such as internal documents, equipment manuals and troubleshooting records.

For Japanese and highly specialized domains in particular, there are many linguistic phenomena, vocabulary and variations in notation that general-purpose LLMs do not cover sufficiently, so continual pretraining is an effective approach. On the other hand, to avoid degrading the performance of the existing model, careful balancing with the pretraining data and adjustments to the tokenizer are important.

2.2 Instruction tuning

Instruction tuning is a method for training models to acquire the ability to understand tasks and respond according to instructions. Specifically, supervised learning is performed using pairs of “instructions” and their corresponding “responses”. For example, for an instruction such as “Rewrite the following sentence in polite form,” the model is trained to generate a natural polite sentence without altering the meaning.

This technique is indispensable for enabling users to intuitively leverage the model through chat-style and prompt-style interfaces, and it plays a major role in making interactive use of LLMs possible.

Furthermore, to improve the level of general versatity across diverse tasks, devise ways are needed for collecting and constructing instruction data in various formats.

While there is abundant data for English instruction tuning, high-quality datasets are limited for Japanese, making the use of translated data and the design of Japanese-specific tasks important.

2.3 Alignment

Alignment refers to a set of techniques that align the model's responses with users' values, social norms, and safety considerations, aiming not only to output "correct" responses but to generate responses that are "desirable and appropriate for the user." Responses are also prioritized not only for their quality, but also for safety and ethical considerations. To achieve alignment, there are two main approaches available: learning based on human preferences, and safety tuning.

In learning based on human preferences, pairs of a "more preferred response" and a "non-preferred response" assessed by humans are used, enabling the model to learn a tendency to choose responses that are appropriate for the user. This method is used to evaluate model outputs and reinforce responses with higher quality and consistency. For example, Direct Preference Optimization (DPO)[4] and Proximal Policy Optimization (PPO)[5] are widely used; adjusting model outputs via reward signals makes it possible to improve response quality.

Safety tuning also aims to suppress outputs that include aggressive, inappropriate or dangerous content. In particular, when applied to Japanese, safety needs to be ensured with consideration for cultural backgrounds and social norms that differ from those of English. For example, establishing filtering criteria that account for the linguistic characteristics and social customs of Japanese is required to avoid unintended misunderstandings and inappropriate responses. Such efforts directly improve reliability when integrating models into business systems and public services, and are indispensable in domains where safety is paramount, such as the manufacturing industry.

Thus, the alignment process is an important means of adapting the model's responses to users' expectations and requirements. By achieving appropriate alignment, it is possible to enhance the model's reliability and practicality, enabling the construction of safer and more effective systems.

2.4 Methods for domain specialization for the manufacturing industry

Here we will outline domain specialization for the manufacturing industry using the methods discussed so far. The workflow for domain specialization for the manufacturing industry is shown in Fig. 2. In this initiative, we used various documents related to our factory automation (FA) products and applied the three approaches of continual pretraining, instruction tuning and alignment. This enabled the model to efficiently learn the specialized knowledge and terminology unique to the manufacturing industry, improving performance.

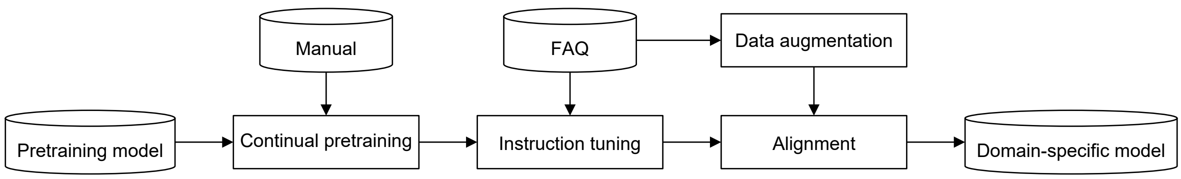


Fig. 2 Flow of domain specialization for the manufacturing industry

As the base pretraining model, a model with 1.8 billion parameters[6] (Model architecture: Llama2[7]) was adopted. In selecting the model size, we considered the balance between computational cost and performance to efficiently conduct training specialized for the manufacturing domain. In manufacturing in particular, building a high-accuracy model while taking resource constraints into account is required, so we selected a model of this size. We also place an importance on increasing model transparency by handling the model's training data in an appropriate manner so that we can explain the model's outputs in the future. From this perspective, we decided to adopt this base model, for which the training data are also publicly available. In the subsequent processing, we applied various domain-specialized methods using this base model.

For continual pretraining, we used manual data related to FA products. This manual data covers a wide range of content, including product specifications, operating procedures and troubleshooting. By retraining the pretrained model on this specialized corpus, we aimed to efficiently incorporate FA product–specific vocabulary and expressions, as well as structured information.

For instruction tuning, we used FAQ data related to FA products. This data is structured in a format that includes questions about product usage and troubleshooting along with their answers. Instruction tuning enables the model to improve its ability to generate appropriate responses to user questions. In particular, tuning at this stage is essential to provide natural responses to inputs in a question format. Furthermore, to address inquiries specific to manufacturing, we also reviewed the content of the FAQ data and organized it into an appropriate format.

For alignment, as with instruction tuning, we leveraged FAQ data related to FA products. The goal of alignment is to enhance the model's ability to generate preferred responses for users. However, while FAQ data includes questions and their "appropriate answers," we needed to address the absence of "non-preferred responses." Therefore, from the set of answer texts within the existing FAQ data, we extracted answer texts with high text similarity that are similar but not the same, and regarded them as "non-preferred responses" to the same questions as a newly considered data augmentation method. With this method, without performing additional manual annotation or designing "non-preferred responses," we were able to efficiently prepare alignment data pairs from existing domain data.

By combining these methods—continual pretraining, instruction tuning and alignment—we made it feasible to build a model specialized for the manufacturing domain. For the model's performance evaluation, we adopted LLM-as-a-judge using Anthropic Claude-3.7 Sonnet as the evaluation model(8) for reference-guided evaluation. For the evaluation data, we used a set of questions that test the correctness of knowledge about our FA products, and for comparison we selected OpenAI GPT-4o, a representative general-purpose LLM on the cloud. For the evaluation data, we compared the responses of the model we developed and those of the comparison model GPT-4o against reference answers and determined correctness. This comparison allows us to evaluate the effectiveness of the developed model with a certain degree of objectivity.

As a result of the evaluation, we confirmed that the model we developed achieved an accuracy rate of 77.24%. GPT-4o, the comparison target, had an accuracy rate of 52.03%. Because this evaluation uses two-choice questions that ask about the correctness of knowledge regarding our FA products, the accuracy rate (chance rate) when answering randomly is expected to be 50%. The accuracy rate of GPT-4o was close to the chance rate. In contrast, the model we developed exceeded the chance rate by 27.24%, demonstrating the effectiveness of the domain-specialized methods examined in this development.

3. Implementing Edge AI by Leveraging OSS

Unlike the conventional approach of running LLMs in the cloud, a domain-specialized SLM is expected to run on edge devices (edge AI) to meet on-site needs for lightweight operation and high responsiveness. By operating the SLM as edge AI, low-latency and privacy-conscious processing becomes possible, and its value is significant across various fields such as smart factories, edge robotics and energy control.

To run a domain-specialized SLM efficiently on an edge device, optimization that balances performance and accuracy within limited hardware resources is essential. As a means of optimization, leveraging a flexible and highly capable OSS is crucial. Our company participates in development alongside world-class engineers in OSS communities that underpin AI development, simultaneously improving our technical capabilities and contributing to society. Specifically, in Apache TVM, an AI compiler OSS*2 (9), our engineers are active in a central role as "committers" with source code edit rights, and in PyTorch*3, which has become the de facto industry standard as an AI framework, our contributions have also been recognized, and we were selected as finalists for the PyTorch Contributor Awards 2024(10).

In this chapter, drawing on our knowledge of these OSS, we describe our proof-of-concept efforts to run an SLM on edge devices. The target devices are the GPU-equipped Jetson*4 Orin Nano 8GB and the Radxa ROCK 5B 16GB equipped with an NPU (Neural Processing Unit); both were selected with industrial applications in mind. In this study, we examined a method to run an SLM with 3.7 billion parameters (model

*2 Apache TVM is a registered trademark of the Apache Software Foundation.
*3 PyTorch is a registered trademark of the Linux Foundation.
*4 Jetson is a registered trademark of NVIDIA Corp.

architecture: Llama2) on both devices in a memory-efficient and high-speed manner by leveraging OSS. Prior to the evaluation, we confirmed that with common execution methods such as PyTorch's Eager mode, the above SLM could not be run on the target devices due to insufficient memory (required memory: 14.9 GB), underscoring the importance of this effort.

To run an SLM on an edge device, an appropriate OSS for LLM inference first needs to be selected. In selecting OSS, Chapter 2 outlined two requirements: support for the SLM (model architecture: Llama2) outlined there and support for the target devices; we evaluated the usefulness of leveraging each OSS that met these.

On the Jetson Orin Nano 8GB, MLC-LLM (Apache TVM-based), ExecuTorch, ollama, vLLM, IREE, and llama.cpp[11] were selection candidates, and we ultimately adopted llama.cpp. llama.cpp takes a "handcrafting" approach to implementing inference—that is, manually optimizing code for each hardware platform—which offers high flexibility in early development stages and relatively easy adoption; these points led us to choose it. As a result, we were able to run the SLM on the Jetson Orin Nano 8GB and confirmed practical inference speed. The generation speed and memory usage at runtime are shown in Table 1.

Table 1 Results of running the domain-specialized SLM on edge devices (Jetson Orin Nano 8GB)

Target device	Jetson Orin Nano 8GB (GPU-equipped)
SLM to be executed	3.7B-class SLM (architecture: Llama2)
Candidate OSS	MLC-LLM (Apache TVM), ExecuTorch, ollama, vLLM, IREE, llama.cpp
Adopted OSS	llama.cpp
Generation speed	22 tokens/sec
Memory usage	2.4GB

On the Radxa ROCK 5B 16GB, MLC-LLM (Apache TVM-based) and rkllm were candidate selections, but because MLC-LLM did not sufficiently support the target device, we adopted rkllm. The generation speed and memory usage at runtime are shown in Table 2.

Table 2 Results of running the domain-specialized SLM on edge devices (Radxa ROCK 5B 16GB)

Target device	Radxa ROCK 5B 16GB (NPU-equipped)
SLM to be executed	3.7B-class SLM (architecture: Llama2)
Candidate OSS	MLC-LLM (Apache TVM-based), rkllm
Adopted OSS	rkllm
Generation speed	6.8 tokens/sec
Memory usage	4.5GB

In this way, by leveraging optimal OSS for the two target devices, even though general AI execution methods suffer from insufficient memory, we confirmed that an SLM with 3.7 billion parameters, previously impossible to run due to insufficient memory, is executable.

Broadly speaking, there are two approaches to the design of LLM inference frameworks. One is the handcrafting approach, as with llama.cpp adopted here, which achieves a high level of flexibility by directly adjusting the source code. The other is the compiler approach, exemplified by Apache TVM, which automatically generates optimized code tailored to the target device.

The handcrafting approach is flexible in supporting efficient fine-tuning methods such as LoRA (Low-Rank Adaptation), and is relatively easy to adopt. On the other hand, the platforms and devices it can support are limited, and implementation and optimization are required for each target device, leaving challenges in terms of scalability and maintainability.

In contrast, the compiler approach, although currently constrained in its support for LoRA, supports a wide range of platforms, including mobile and browser environments as with MLC-LLM, and is superior from the standpoint of future extensibility and scalability. Furthermore, because code generation can automate optimization for each environment, improvements in maintainability and development productivity can be expected over the long term.

We intend to advance a transition to the compiler approach in order to strengthen our ability to support increasingly diverse device environments and use cases going forward. In addition, regarding the implementation of training optimization features starting with LoRA, we are aiming to expand supported capabilities in collaboration with the OSS community. Leveraging the OSS utilization expertise we have cultivated to date, we will accelerate the practical use and adoption of domain-specialized SLMs in the edge AI field and work to achieve a more sustainable and highly scalable society.

4. Conclusion

These results show that, for developing SLMs specialized for expert and limited domains such as manufacturing, the combination of continual pretraining, instruction tuning and alignment is effective. This approach can also be applied to other manufacturing-related domains and to other industries, and by broadening its scope of application, further performance improvements can be expected.

Future challenges include collecting additional data to further improve model performance and improving alignment methods. In particular, there is a need to refine methods for the automatic generation of non-preferred response and to devise ways to reconcile response diversity and consistency. In addition, for model evaluation methods, approaches need to be considered that use more multifaceted metrics to quantitatively measure practical utility. Moreover, to address a wide variety of use cases, optimization on edge devices needs to be considered. Depending on the use case, higher responsiveness and lower resource usage than currently assumed may be required. Therefore, implementation-level ideas are needed to achieve stable model operation and high responsiveness in constrained computing environments. By continuously working on these challenges, we aim to further advance and put SLMs into practical use in the manufacturing domain.

References

(1) Gupta, K., et al.: Continual Pre-Training of Large Language Models: How to (re) warm your model?, The Thirty-Seventh Annual Conference on Neural Information Processing Systems (2023)

(2) Wei, J., et al.: Finetuned Language Models are Zero-Shot Learners.The Tenth International Conference on Learning Representations (2022)

(3) Bai, Y., et al.: Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, arXiv, 2204.05862 (2022)

(4) Rafailov, R., et al.: Direct Preference Optimization: Your Language Model is Secretly a Reward Model, Advances in Neural Infor- mation Processing Systems, 36, 53728-53741 (2023)

(5) Schulman, J., et al.: Proximal Policy Optimization Algorithms, arXiv, 1707.0634 7 (2017)

(6) Hugging Face: LLM-jp
https://huggingface.co/llm-jp

(7) Touvron, H., et al.: Llama 2: Open Foundation and Fine-Tuned Chat Models, arXiv, 2307.09288 (2023)

(8) Zheng, L., et al.: Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, Advances in Neural Information Processing Systems, 36, 46595-46623 (2023)

(9) Chen, T., et al.: TVM: An Automated End-to-End Optimizing Compiler for Deep Learning, 13th USENIX Symposium on Oper- ating Systems Design and Implementation, 578-594 (2018)

(10) PyTorch: Announcing the 2024 PyTorch Contributor Awards
https://pytorch.org/ecosystem/contributor-awards-2024

(11) GitHub: ggml-org/llama.cpp
https://github.com/ggml-org/llama.cpp

LLM-Based Program Analysis for Ladder Program

Author: *Naoki Sugawara**
**Information Technology R&D Center*

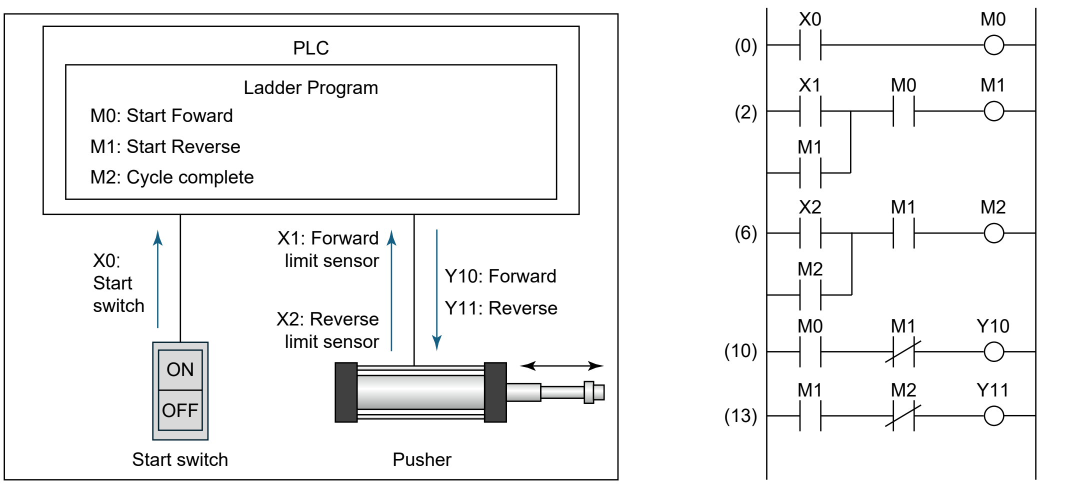
Abstract

Ladder logic is a programming language for describing control operations executed by a Programmable Logic Controller (PLC) and is used to control various equipment. When a control systems engineer modifies equipment, they need to analyze the control logic written in ladder logic (hereafter, “ladder program”) and understand the processing details, but this task takes a lot of work. To address this, Mitsubishi Electric developed a technology that uses Large Language Models (LLMs) to analyze ladder programs and generate processing description. It features analyzing not only the ladder program but also PLC log data, and explaining the ladder program’s processing in the order of signal changes. As a result, the sequence in which devices within the equipment are controlled is explained clearly, reducing the workload on engineers when it comes to decoding ladder programs.

1. Introduction

PLCs are used to control various equipment. When a PLC controls equipment, it repeatedly receives signals from devices within the equipment, processes the signals in the control logic, and outputs signals to devices within the equipment. There are several programming languages for writing control logic, but in Japan the programming language known as ladder logic is used particularly often.

Using Fig. 1, we outline equipment control using a ladder program. Figure 1(a) shows equipment composed of a PLC and devices (start switch, pusher), and X0, Y10, etc. in the figure are input/output signals exchanged between the PLC and the devices. In equipment comprising a PLC and devices like this, the PLC receives input signals from devices, processes those signals in the ladder program, and passes output signals to devices to control them.



(a) Equipment (b) Ladder program to control equipment

Fig. 1 Example of an equipment configuration and its ladder program

Figure 1(b) is an example of a ladder program that controls the equipment in Fig. 1(a). In a ladder program, horizontal lines are drawn between the left and right vertical lines; the condition is written on the left side of a horizontal line, and the processing on the right side, indicating what processing is performed under which conditions. For example, in item (0) of Fig. 1(b), the left portion indicates the condition “when X0 is 1,” and the right portion indicates the process “set M0 to 1.” Similarly, in item (10) of Fig. 1(b), the left portion indicates the condition “when M0 is 1 and M1 is 0,” and the right portion indicates the process “set Y10 to 1.” Combining the conditions and processes in items (0) and (10) of Fig. 1(b) yields control: “set Y10 to 1 when X0 is 1 and M1 is 0.” With this control, when the equipment’s start switch is turned ON, the pusher moves forward until it reaches the forward end.

In this way, ladder programs are difficult to analyze because the values of multiple input signals and the order in which they change affect the processing, and the processing related to a single output signal is written across multiple parts of the ladder program.

2. Challenges in Understanding Ladder Program Control

When a control systems engineer modifies equipment or estimates the cause of an equipment malfunction, they need to understand how the equipment is controlled by the ladder program. However, for equipment that has been in use for a long time, the control systems engineer who originally designed it may no longer be available due to retirement, etc., making it necessary to analyze the ladder program.

Figure 2 is an example represented in text the result of a human decoding the ladder program in Fig. 1(b). In this text, you can see in what order the equipment’s signals change and which parts of the ladder program cause those signal changes. The steps in the text correspond to the numbers written on the left of Fig. 1(b). With such an explanation of the processing, a control systems engineer can determine which parts of the ladder program to check and modify when renovating equipment or estimating the cause of malfunctions.

> When the user turns ON the start switch, X0 changes to 1, causing the process at step 0 of the program to set M0 (Start Forward) to 1. When M0 changes to 1, the process at step 10 sets Y10 (Forward) to 1, and the pusher begins to move forward.
>
> When the pusher reaches the forward end, X1 (Forward End Sensor) changes to 1, and the process at step 2 sets M1 (Start Backward) to 1. When M1 changes to 1, the process at step 13 sets Y11 (Backward) to 1, and the pusher begins to move backward. Additionally, the process at step 10 changes Y10 (Forward) to 0, stopping the pusher's forward motion.
>
> When the pusher reaches the backward end, X2 (Backward End Sensor) changes to 1, causing the process at step 6 to set M2 (Cycle Complete) to 1. When M2 changes to 1, the process at step 13 changes Y11 (Backward) to 0, and the pusher's backward motion stops.
>
> When the user turns OFF the start switch, X0 changes to 0, causing the process at step 0 to reset M0 (Start Forward) to 0. When M0 changes to 0, the process at step 2 resets M1 (Start Backward) to 0. Finally, when M1 changes to 0, the process at step 6 resets M2 (Cycle Complete) to 0.

Fig. 2 Example of a manual analysis of a ladder program

However, as stated in Chapter 1, decoding ladder programs is difficult to analyze a large ladder program and generate an explanation like the one shown in Fig. 2.

Therefore, we developed a technology that uses LLMs to generate a processing description for ladder program. Generated by this technology, the processing description for ladder program clearly shows the order in which devices within the equipment are controlled, which is a key feature.

3. Method for Generating the Processing Description for Ladder Program

When generating a processing description using an LLM, the explanation is basically generated in order from the top of the ladder program. Therefore, if only the ladder program is used as input data to an LLM to generate a processing description, it cannot generate an explanation like that shown in Fig. 2, where the order of processing carried out by the ladder program is clear. This technology addresses this issue by using not only the ladder program but also PLC log data as input. Table 1 shows the steps for generating the processing description for ladder program and the input and output data at each step.

Table 1 Procedure for generating a processing description using this technology

No.	Procedure	Input data	Output data	Explanation
1	Splitting the ladder program	Ladder program	The split ladder program	Ref. 3.1
2	Extraction of signal change order	Log data	Signal change information at each time	Ref. 3.2
3	Mapping signal changes to the ladder program	Output data No. 1 and No. 2	Relationship between signal changes and the ladder program	Ref. 3.3
4	Generation of processing description	Output data No. 3 and meaning of each signal	Program description for ladder program	Ref. 3.4

3.1 Splitting the ladder program

The analysis result in Fig. 2 first explains the processing at step 0 of the ladder program, and then explains that the processing at step 10 of the ladder program is performed next. In a ladder program, processing is performed only where the condition is satisfied; therefore, as in this example, the order in which processing appears in the program may not match the order in which processing is actually executed.

Accordingly, this technique splits the ladder program as shown in Fig. 3 to make reordering possible. Splitting is performed at the ladder block level, which is a combination of the processing content and the condition under which that processing is performed. Then, to allow reordering in the correct sequence at a later stage, the condition part and the processing part are also split.

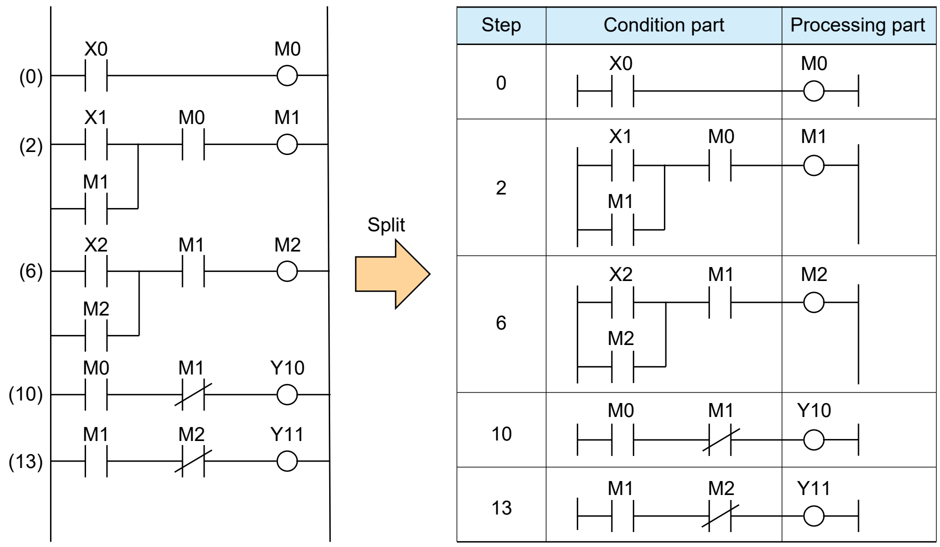


Fig. 3 Ladder program partitioning

3.2 Extraction of signal change order

The order of input signal changes is shown in the decoding results in Fig. 2—for example, X0 becomes 1 first, and then X1 becomes 1. Such changes in input signals occur due to device motion, and the ladder program alone does not indicate when or how input signals change.

Therefore, this technique uses PLC log data as input in addition to the ladder program. PLC log data records the state of each signal along with timestamps during equipment operation, and can be represented in tabular form as shown in Fig. 4. For example, in Fig. 4, the row where the time column is 00:01.0 shows a value of 1 in the X0 column, from which we can read that X0 is 1 at time 00:01.0.

In this technique, by extracting from the log data the parts where each signal's value has changed from the previous timestamp (as in Fig. 4), we create information about the signal change order—indicating when and how signals changed—which is required to generate the processing description.

: Sections where the value has changed since the previous time

Time	X0	X1	X2	Y10	Y11	M0	M1	M2
00:00.0	0	0	1	0	0	0	0	0
00:01.0	1	0	1	1	0	1	0	0
00:02.0	1	0	0	1	0	1	0	0
00:03.0	1	0	0	1	0	1	0	0
00:04.0	1	1	0	0	1	1	1	0
00:05.0	1	0	0	0	1	1	1	0
00:06.0	1	0	1	0	0	1	1	1
00:07.0	0	0	1	0	0	0	0	0

Log data

Extract by signal change order

Time	Change
00:01.0	X0 changes to 1 Y10 changes to 1 M0 changes to 1
00:02.0	X2 changes to 0
00:04.0	X1 changes to 1 Y10 changes to 0 Y11 changes to 1 M1 changes to 1
00:05.0	X1 changes to 0
00:06.0	X2 changes to 1 Y11 changes to 0 M2 changes to 1
00:07.0	X0 changes to 0 M0 changes to 0 M1 changes to 0 M2 changes to 0

Signal change order

Fig. 4 Signal change extraction process from log data

3.3 Mapping signal changes to the ladder program

In the decoding results in Fig. 2, the signal changes are explained by associating them with the corresponding parts of the ladder program. To generate this kind of explanation, the results of splitting the ladder program in Section 3.1 needs to be associated with the results of extracting signal changes in Section 3.2.

In this technique, we extract candidate associations and confirm whether those candidate associations are correct by verification using an LLM.

In the procedure for finding candidate associations, based on the signal changes extracted in section 3.2, we extract from the ladder program split in section 3.1 those programs in which both the signals in the condition part and the signals in the processing part change at the same timestamp. In the example in Fig. 5, at time 00:01.0 three signal changes occur; the ladder programs that include these changes in both the condition part and the processing part are the ladder programs in step 0 and step 10. In these two programs, one can predict that the processing was performed because the value of the signal serving as the condition changed, potentially causing other signal values to change.

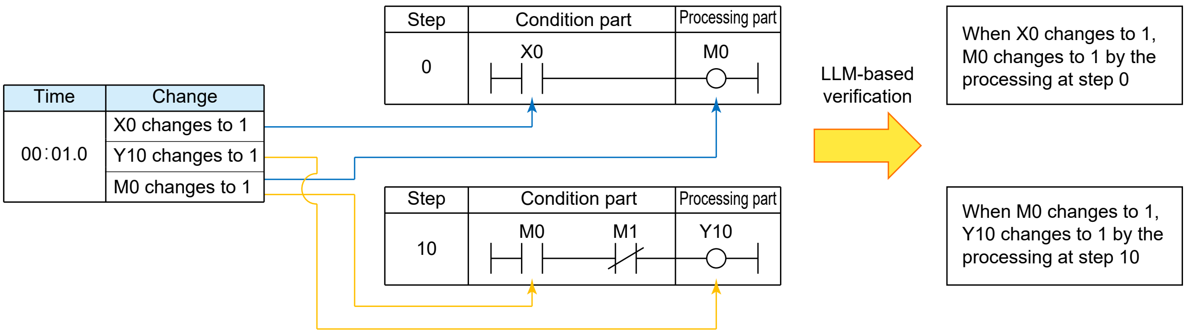


Fig. 5 Associating signal changes and corresponding ladder block

In the procedure for verifying candidate associations using an LLM, we provide the signal values and the ladder program to the LLM, and check whether the signal values change due to processing by the ladder program. The reason this procedure is necessary is that the candidate associations extracted above are not necessarily correct. For example, in the program of step 10 in Fig. 5, the condition part includes the signal M1, and depending on the value of M1, even if M0 becomes 1, Y10 does not become 1. Furthermore, if multiple candidate processing steps exist in the ladder program that could change the same signal value, this method is needed to determine which candidate actually changed the signal.

3.4 Generation of processing description

Finally, we combine the explanations obtained in section 3.3 to generate the processing description for ladder program. Here, by adding not only the signals but also the meaning of each signal—for example, "X0 (start switch)"—to the explanation, we make it easy to understand the relationship between the program and equipment operations.

4. Validation

We conducted a validation to confirm that this technique is able to generate explanations that correctly describe the processing content of the ladder program, and clear explanations that describe the processing in the order of signal changes. In the validation, we used gpt-4o (2024-05-13) as the LLM.

4.1 Validating that explanations correctly describing the processing contents can be generated

Using this technique, we generated the processing description for ladder program for the ladder program in Fig. 1(b) and compared the result with the human decoding in Fig. 2, to verify whether the processing description was correctly described in the generated explanations. Figure 6 shows the result of generating the processing description for ladder program using this technology. When comparing Fig. 2 and Fig. 6, causes of signal changes such as "pusher reaches the forward end" are not described in Fig. 6 because the input data lacks that information; however, from the meaning of each signal—such as "Start Forward" and "Forward End Sensor"—an approximate operation can be inferred. Otherwise, Fig. 2 and Fig. 6 present equivalent content, and the processing content is described correctly in the generated explanations.

When X0(Start Switch) changes to 1, the process at step 0 set M0(Start Forward) to 1.
When M0(Start Forward) changes to 1, the process at step 10 set Y10(Forward) to 1.

After 3.0 seconds, when X1(Forward End Sensor) changes to 1, the process at step 2 set M1(Start Backward) to 1.
When M1(Start Backward) changes to 1, the process at step 13 set Y11(Backward) to 1.
Additionally, the process at step 10 set Y10(Forward) to 0.

After 2.0 seconds, when X2(Backward End Sensor) changes to 1, the process at step 6 set M2(Cycle Complete) to 1.
When M2(Cycle Complete) changes to 1, the process at step 13 set Y11(Backward) to 0.

After 1.0 seconds, when X0(Start Switch) changes to 0, the process at step 0 set M0(Start Forward) to 0.
When M0(Start Forward) changes to 0, the process at step 2 set M1(Start Backward) to 0.
When M1(Start Backward) changes to 0, the process at step 6 set M2(Cycle Complete) to 0.

Fig. 6 Generation results of program descriptions using this technology

4.2 Explanations of processing content generated in the order of signal changes —validation

We compared the processing description generated using this technique, with the processing description generated using only the ladder program as input data, and confirmed the effectiveness of the procedure described in Chapter 3. Figure 7 shows the processing description generated using only the ladder program as input data.

In step 0, if X0 is 1, M0 is set to 1.
In step 2, if X1 is 1 or M1 is 1, and M0 is 1, M1 is set to 1.
In step 6, if X2 is 1 or M2 is 1, and M1 is 1, M2 is set to 1.
In step 10, if M0 is 1 and M1 is 0, Y10 is set to 1.
In step 13, if M1 is 1 and M2 is 0, Y11 is set to 1.

Fig. 7 Generation results of program descriptions using only ladder program as input

Given that the explanation shown in Fig. 7 is created in order from the top of the ladder program, it is not possible to determine which parts of the program perform processing in what order. Moreover, it is also unclear in what order the changes in each signal occur. On the other hand, as shown in Fig. 6, the description of the processing content generated using this technology resolves these issues and describes the ladder program's processing content in the order of signal changes in the log data. With such a description, it is possible to reduce the workload of engineers in analyzing the ladder program.

5. Conclusion

We described a technology that leverages LLMs and, by mapping signal changes extracted from log data to the ladder program, generates clear explanations of the control sequence of devices within the equipment. By utilizing this technology, the workload of engineers in analyzing ladder programs can be reduced.

Going forward, we will consider further developed technologies based on this approach, such as a chatbot that answers questions about the processing content and functions that explain differences from the specifications.

MITSUBISHI ELECTRIC CORPORATION