Application of AI control technology using reinforcement learning and Sim2Real to manufacturing equipment.

Takahiro Nitawaki,[1] Ryo Shiratori,[1] Yohei Kasai,[1] Yoshihiro Tabata,[1] and Kiminori Kurosawa[1]

In recent years, AI technology has made remarkable advancements and has been increasingly utilized in various sectors of society. As a result, expectations for the application of AI technology in the manufacturing industry have also grown. Our company has been actively pursuing the implementation of AI technology using deep learning since 2015, and we have achieved successful integration in inspection processes, among other areas. This paper aims to describe a case study where AI technology was introduced in the field of device control, expanding the scope of its application.

1. Introduction

The recent advancements in AI technology have been remarkable, and its rapid integration into general society is evident. The manufacturing industry is no exception, with active efforts to incorporate AI technology into business activities, such as automating manufacturing processes. Our company began activities related to applying AI technology, particularly deep learning, to production processes in 2015. By 2017, we had successfully implemented an AI-based visual inspection system into our production process. This AI system has been effective in reducing the flow of defective products into subsequent processes and minimizing the labor required for inspections.

To successfully apply AI technology to production processes, our company developed an AI system development roadmap (Figure 1) and has been gradually implementing AI based on this clearly defined strategy. In Figure 1, the horizontal axis represents the types of information handled by AI, and the vertical axis lists the key technologies necessary for handling this information. The bar graph within the figure indicates the possible application areas that can be realized by combining this information with the key technologies. The information handled by AI is broadly classified into four categories: "Images," "Numerical Data/Symbols/Sounds," "Motion Control," and "Language Concepts." These categories are further broken down; for example, images are divided into "Single Still Image," "Single Video," and "Multiple Images," while numerical data/symbols/sounds are broken down into "Graphical Representation" and "Time Series." These categories are listed in order of increasing technical difficulty. Our company has followed this AI system development roadmap, progressively applying AI to production processes starting with the areas of lower difficulty.

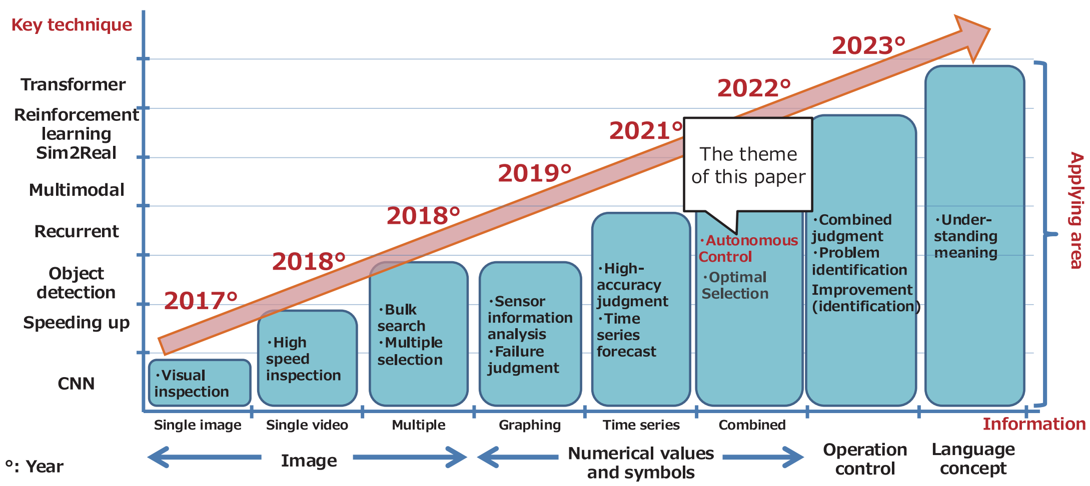

Fig 1: AI system development roadmap.

1 : Fujikura Ltd.

Abbreviations, Acronyms, and Terms.

AI—Artificial Intelligence

CNN—Convolutional Neural Network
a type of deep learning used for image and video recognition.

RL—Reinforcement Learning
a machine learning method that deals with the problem of deciding the ac

Agent—Agent
The agent that learns and tasks action in reinforcement learning.

Sim2Real—Simulated to Real
A class of machine learning method that uses simulations to solve real-world problems

Table 1. Supervised learning, Unsupervised learning, Reinforcement learning.

Type of learning	①Supervised learning	②Unsupervised learning	③Reinforcement learning
Overview	Learning from data with correct labels(supervised information)	Learning from data without using correct labels	Learning to maximize rewards based on actions
Advantages	It is easier to obtain the expect-ed results	No need for labeling, making it easier to approach	Achieving results beyond expectations through trail and error
Disadvantages	The effort required for labeling is substantial	The quality of the analysis results depends on the analyst's capabilities	Reward setting is challenging, and learning time is substantial
Main applications	Visual inspection	Data analysis	Games

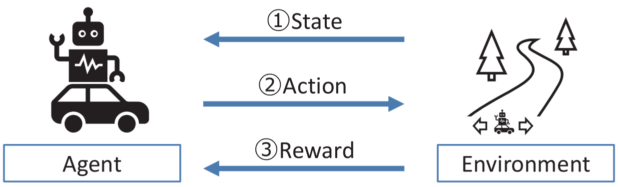

Fig 2. Image of reinforcement learning.

In this paper, we will introduce a case study of applying AI to autonomous control of manufacturing equipment, which is still rare globally. In Chapter 2, we will explain the key technologies of reinforcement learning and Sim2Real. Chapters 3 and 4 will describe the content and results of our efforts, and finally, Chapter 5 will summarize the paper.

2. Explanation of the Technologies Used

Generally, AI technologies can be classified into supervised learning, unsupervised learning, and reinforcement learning (Table 1). This paper will focus on AI technology based on reinforcement learning, which is considered suitable for device control.

2.1. Reinforcement Learning

Reinforcement learning is a learning method in which an agent acquires an appropriate action policy through interactions with its environment. The concept is illustrated in Figure 2. First, the environment provides the agent with a state (①). The agent then takes an action based on the state, which alters the state (②). Based on the result of this change, the agent receives a reward (③). Through this process, the agent engages in trial and error, gradually learning to adopt an optimal action policy that maximizes the reward.

If rewards are appropriately set for actions, reinforcement learning can proceed without prior knowledge. Because of this, reinforcement learning is increasingly being applied in fields such as gaming, where actions and rewards can be clearly defined. In device control, where actions and rewards can also be clearly defined, reinforcement learning is considered a suitable approach. However, there are many challenges in practical implementation, and cases of applying reinforcement learning to manufacturing equipment are still rare globally.

2.2. Challenges in Learning

There are two major challenges in applying reinforcement learning to manufacturing equipment. The first challenge is that the learning phase requires the use of actual manufacturing equipment for extended periods, which necessitates limiting production. The second challenge is the risk of unintended actions during the trial-and-error process, which could result in equipment malfunction or wear. Generally, users wish to avoid production limitations and equipment malfunctions or wear, making it difficult to

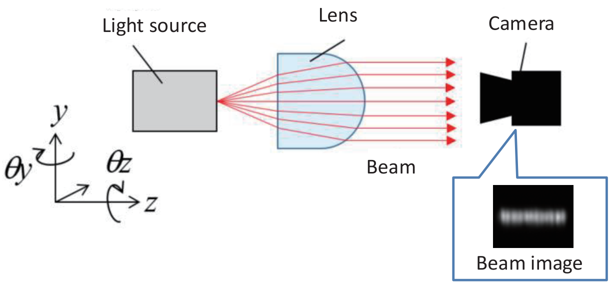

Fig 3. Overview of lens alignment.

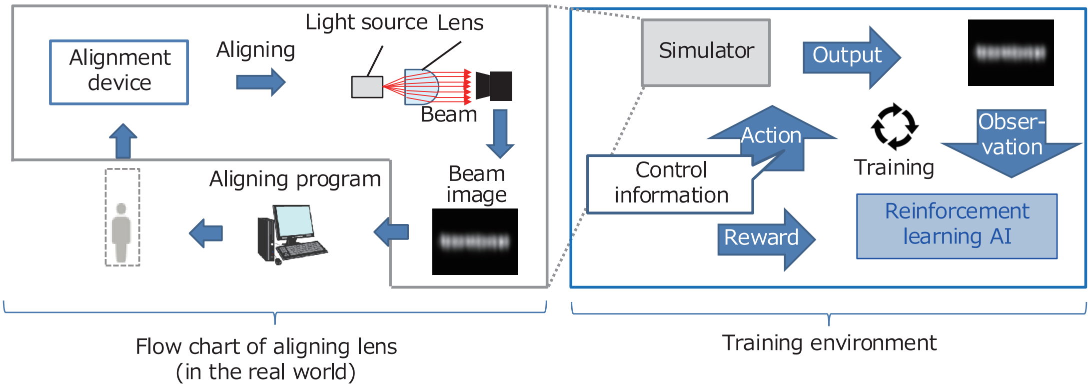

Fig 4. Conventional lens alignment workflow and the configuration of the simulator-based reinforcement learning alignment system.

conduct learning with manufacturing equipment. These challenges are considered major obstacles to the advancement of reinforcement learning in manufacturing equipment.

To address these challenges, this case study also applied Sim2Real, a method that allows learning without the use of actual manufacturing equipment.

2.3. Sim2Real

Sim2Real is a technology that applies reinforcement learning AI trained in a simulator to real-world environments. By using this technology, it is possible to overcome the challenges of needing to use actual manufacturing equipment for extended periods during the learning phase, as well as the risk of equipment malfunction or wear due to unintended actions during the trial-and-error process.

However, it is difficult to fully replicate the conditions and perceptions of the real-world environment—affected by factors such as material shapes, properties, and impurities on sensors and cameras—within a simulator. Typically, a gap exists between the real environment and the simulator. If this gap is large, directly applying the reinforcement learning AI to manufacturing equipment may result in unintended behavior. In this case study, a gap between the real environment and the simulator was encountered, and we worked to address this issue, as explained in Section 3.5.

3. Application of Reinforcement Learning to Manufacturing Equipment

In this chapter, we explain the application of reinforcement learning control to lens alignment, a process that adjusts the focus of light emitted from a light source, as an example of the application of reinforcement learning and Sim2Real discussed in Chapter 2.

3.1. Overview

Figure 3 shows an overview of lens alignment. The beam emitted from the light source spreads out. To efficiently use the beam, it is necessary to eliminate this spread and shape the beam into a parallel beam by adjusting the position and angle of the light source and the lens. In this process, an automatic alignment program (conventional automatic alignment) has been introduced, which adjusts four axes—Y-axis, θY-axis, Z-axis, and θZ-axis—so that the feature quantities obtained from the beam image captured by the camera meet the product specifications. The most important feature quantity is the beam width, which narrows as alignment approaches an optimal state.

In conventional automatic alignment, alignment is performed one axis at a time in a predetermined sequence. This method requires alignment of all axes, even those that do not require adjustment, resulting in extended alignment times. Additionally, slight differences in the lens shape,

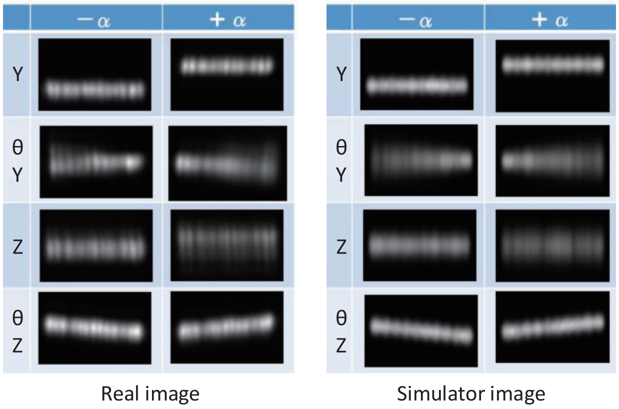

Fig 5. Comparison between real images and simulator images.

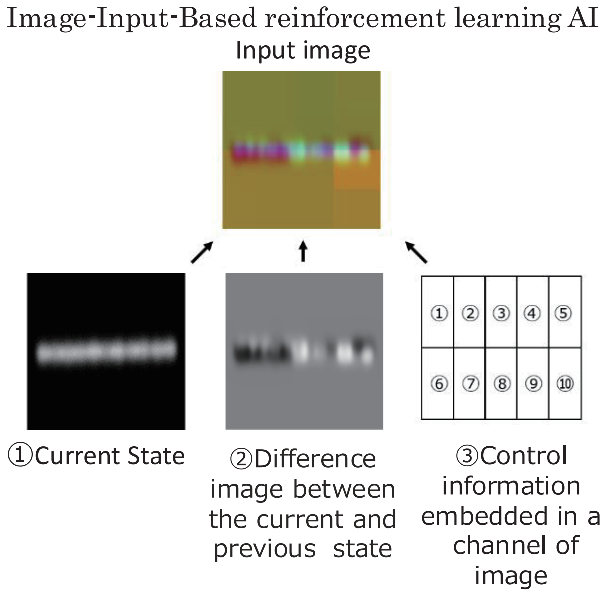

Fig 6. Input image for the reinforcement learning alignment system in the Image-Input-Based method.

equipment, or other factors can cause unintended axis shifts during alignment. If the sequence of alignment for that axis has already been completed, alignment may be considered complete even though the axis is misaligned, resulting in non-compliance with the specifications. If the specifications are not met, manual alignment is performed by personnel to bring the alignment within the specifications. However, manual alignment can be burdensome, as it prevents personnel from performing their primary tasks.

3.2. Simulator

Figure 4 shows the flow of conventional lens alignment workflow and the configuration of the simulator-based reinforcement learning alignment system developed in this study. A simulator was created to simulate conventional lens alignment and generate beam images corresponding to the alignment amounts for each axis—Y-axis, θY-axis, Z-axis, and θZ-axis. Figure 5 compares actual beam images with simulator-generated images.

3.3. Image-Input-Based Reinforcement Learning AI

Using the simulator images explained in Section 3.2, we developed reinforcement learning AI. The input state was a single image as shown in Figure 6. The inputs were (①) an image representing the current state, (②) a difference image between the current state and the previous state, and (③) control information represented as an image, designed to embed all necessary information in a single image. The control information was split into multiple regions within the image based on the number of control dimensions, with each region's brightness corresponding to the control information value. Past states, current states, and control information were each represented as grayscale images, which were then combined into a single RGB image containing multiple states and control information. The reward was based on the beam width, the most critical feature quantity in lens alignment. A narrower beam width resulted in a higher positive reward. Additionally, a negative reward proportional to the number of alignment attempts was applied, encouraging behavior that achieved a narrower beam width with fewer alignment attempts. The output was the next axis to align and the amount of alignment for that axis. The reinforcement learning AI targeted the Z-axis, θY-axis, and θZ-axis, as the Y-axis had no dependencies with other axes and could be easily aligned using the conventional algorithm, making it unnecessary to include in the reinforcement learning AI.

With this approach, we successfully developed reinforcement learning AI that operates in the simulator. This reinforcement learning AI is expected to perform alignment more efficiently than conventional automatic alignment, as it adjusts the axes by appropriate amounts based on the beam's state. However, when preparing to introduce the reinforcement learning AI into production, a problem arose concerning the gap between the real environment and the simulator, as mentioned in Section 2.3.

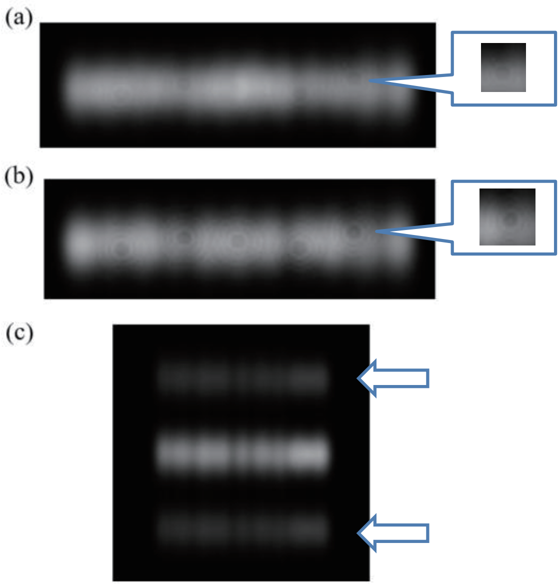

**Fig 7. (a) Interference fringes(Weak).
(b) Interference fringes(Strong). (c) Side lobes.**

Table 2. Effect of image appearance.

Input noise	Image-Input-Based	
	Number of aligning times	Beam width [pixel]
No noise	27.4	13.2
Interference fringes (weak)	29.8	13.2
Interference fringes (strong)	29.6	13.3
Side lobes	73.1	13.4

3.4. Gap Between the Real Environment and the Simulator

When applying the system to actual manufacturing equipment, it became apparent that small impurities on the camera could case interference fringes, or side lobes occurring when the lens axis was misaligned caused a significant gap between the real environment and the simulator (Figure 7). When using the image-input-based reinforcement learning AI initially conceived, it was found that the alignment time was longer, particularly in the presence of side lobes, compared to the noiseless condition (Table 2).

To address this issue, a new simulator reflecting the interference fringes and side lobes would need to be created for further training. However, it was found that the appearance of these interference fringes and side lobes varied depending on the manufacturing equipment and conditions of the day, making it challenging to accurately replicate them in the simulator.

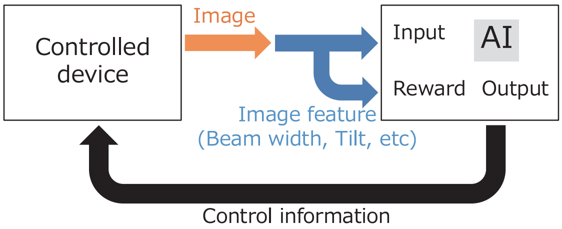

Fig 8. Learning models in reinforcement learning.

Table 3. Effect of image appearance.

Input noise	Image-Input-Based		Feature-Input-Based	
	Number of aligning times	Beam width [Pixel]	Number of aligning times	Beam width [Pixel]
No noise	27.4	13.2	28.8	13.3
Interference fringes (weak)	29.8	13.2	29.5	13.4
Interference fringes (strong)	29.6	13.3	33.3	13.5
Side lobes	73.1	13.4	29.7	13.1

3.5. Solution Using Feature-Input-Based Method

To resolve the gap issue between the real environment and the simulator, we devised a method of extracting features from the image and using them as input for learning (Figure 8). By inputting the state, such as beam width and beam tilt, as numerical values into the reinforcement learning AI, the impact of noise, such as interference fringes and side lobes, can be minimized. This approach is expected to absorb the differences between actual images and simulator images caused by various factors, such as the manufacturing equipment and the conditions of the day. The rewards and actions were the same as those used in the image-input-based reinforcement learning AI.

Table 3 shows the results of a comparison evaluation conducted under the same conditions. The table lists the number of alignments attempts and the average beam width for both methods. There was little difference between the methods in the cases without added noise and with interference fringes, but the feature-input-based method significantly improved alignment in the case of side lobes.

3.6. Application of Reinforcement Learning to Real-World Systems

This section describes the application of the feature-input-based reinforcement learning AI developed in Section 3.5 to actual control systems. Figure 9 compares the flow before and after the introduction of reinforcement learning. In the figure, "I/F" represents the interface. As mentioned in Section 3.1, conventional automatic alignment software exists for lens alignment. When introducing reinforcement learning AI, two interfaces in the conventional automatic alignment software were modified: the image input section and the alignment command section, adding functions to extract captured images and receive alignment information. As with the simulator, the Y-axis alignment was excluded from the reinforcement learning AI, as it had no dependencies with other axes and could be easily aligned using the conventional algorithm. The reinforcement learning AI targeted the Z-axis, θY-axis, and θZ-axis.

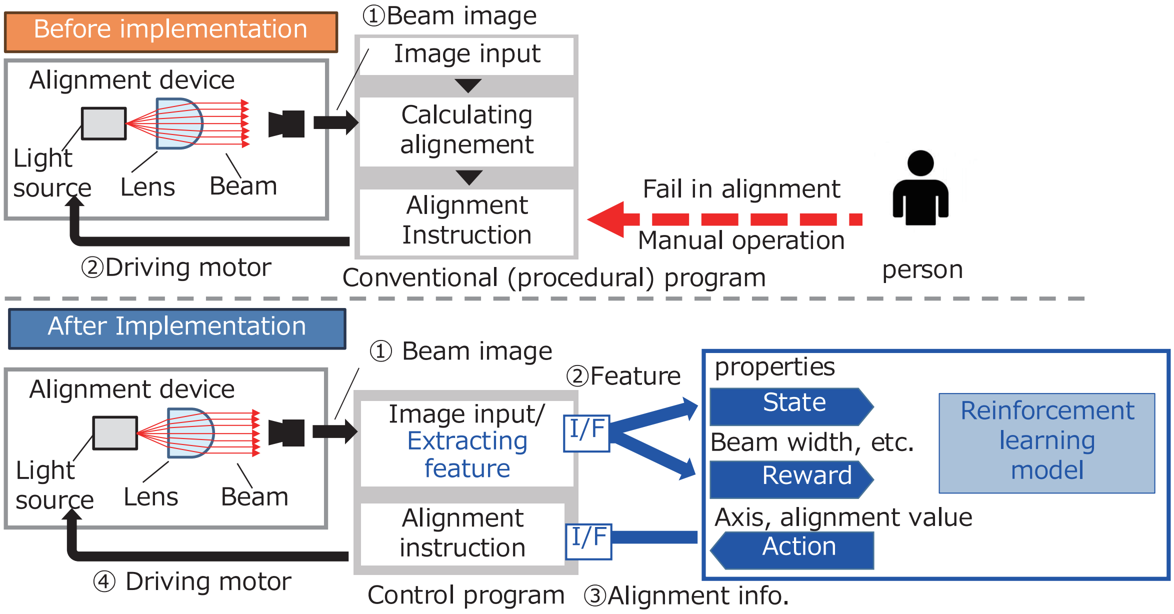

Fig 9. Comparison of workflow before and after introducing reinforcement learning.

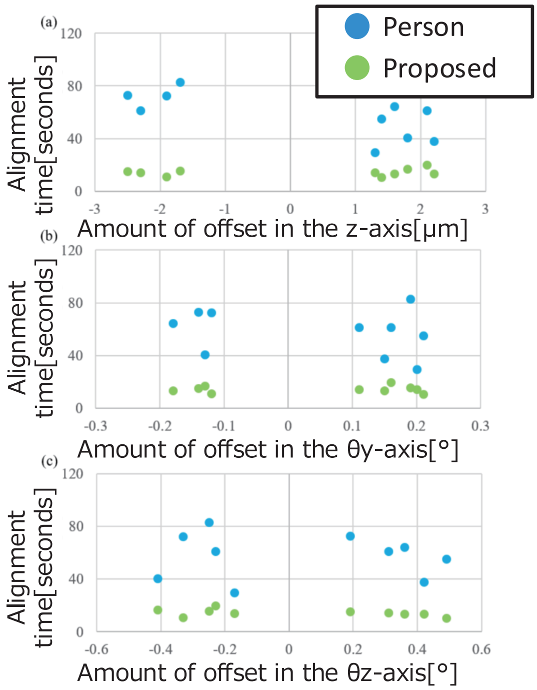

Fig 10. Amount of offset and alignment time for each axis: (a) z-axis, (b) θy-axis, (c) θz-axis, and alignment time.

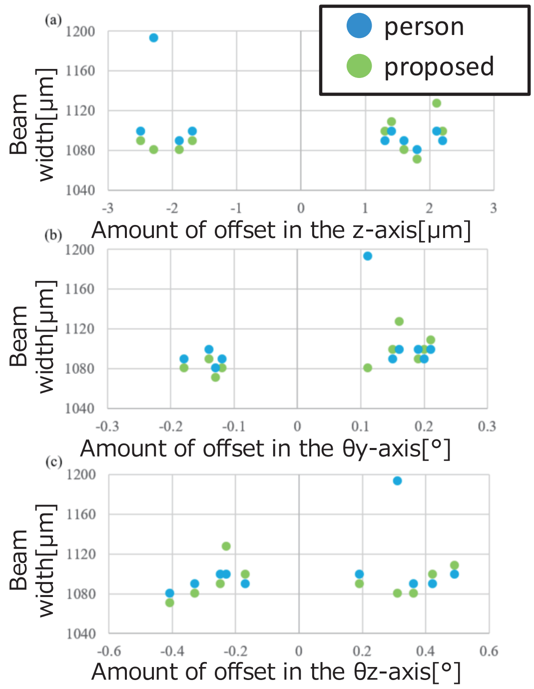

Fig 11. Amount of offset and beam width after alignment for each axis: (a) z-axis, (b) θy-axis, (c) θz-axis, and beam width.

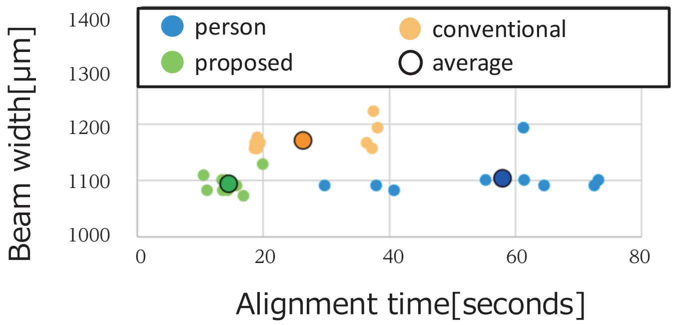

Fig 12. Relationship between alignment time and beam width in lens alignment experiments on the actual system.

4. Evaluation

(1) Comparison Between Manual Alignment and Reinforcement Learning-Based Alignment

To confirm the superiority of reinforcement learning-based alignment in actual manufacturing equipment, a comparative evaluation was conducted between manual alignment and reinforcement learning-based alignment, assessing alignment time and results. In this evaluation, the initial state was randomly offset from the best alignment state by ±2.5 mm on the Z-axis, ±0.25° on the θY-axis, and ±0.5° on the θZ-axis. The time required to complete alignment and the results after completion were evaluated for both manual and reinforcement learning-based alignment. Ten different initial states were prepared, and conditions were set to be the same for both manual and reinforcement learning-based alignment.

Figure 10 shows the relationship between the amount of offset in each axis in the initial state and the alignment time. The horizontal axis in each graph represents the offset amount in each axis at the initial state, with 0 indicating the best alignment state. The vertical axis represents the alignment time, with values closer to 0 indicating shorter alignment times. The plots show the relationship between the offset amount in each axis at the initial state and the alignment time. Manual alignment is indicated by blue circles, while reinforcement learning-based alignment is indicated by green circles. From this figure, reinforcement learning-based alignment completes alignment in a shorter time compared to manual alignment.

Figure 11 shows the relationship between the amount of offset in each axis at the initial state and the beam width at the completion of alignment. The horizontal axis represents the offset amount in each axis at the initial state, while the vertical axis represents the beam width at the completion of alignment, with values closer to 0 indicating a narrower beam. From this figure, the alignment result of reinforcement learning-based alignment is as good as or better than manual alignment. From these two results, it was confirmed that reinforcement learning-based alignment completes alignment in a shorter time and with results that are as good as or better than manual alignment.

Table 4. Average results of lens alignment experiments on the actual system.

	Alignment time [Second]	Beam width [mm]
Manual alignment	57.9	1103.6
Conventional automatic alignment	26.3	1171.2
Reinforcement learning-based alignment	14.5	1093.2

(2) Comparison Between Manual Alignment, Conventional Automatic Alignment, and Reinforcement Learning-Based Alignment

Next, a comparison was conducted between manual alignment, conventional automatic alignment, and reinforcement learning-based alignment in terms of alignment time and results. Figure 12 shows the results of a two-dimensional plot of alignment time and beam width. A shorter alignment time and a narrower beam width indicate better alignment results, so points closer to the lower left of the graph represent better performance. The dark markers for each alignment method represent the average values, and the results are shown in Table 4. The alignment times were 57.9 seconds for manual alignment, 26.3 seconds for conventional automatic alignment, and 14.5 seconds for reinforcement learning-based alignment. The beam widths were 1103.6 μm for manual alignment, 1171.2 μm for conventional automatic alignment, and 1093.2 μm for reinforcement learning-based alignment. Reinforcement learning-based alignment was shorter in time and higher in performance compared to both manual and conventional automatic alignment. From these results, the superiority of reinforcement learning AI was confirmed compared to manual and conventional automatic alignment.

Through this effort, we successfully introduced reinforcement learning AI into manufacturing equipment, addressing the challenges related to learning with manufacturing equipment discussed in Section 2.2 and the gap between the real environment and the simulator discussed in Section 3.4. Additionally, we resolved the issues of conventional automatic alignment discussed in Section 3.1, leading to expected improvements in alignment time and results.

5. Conclusion

As a case study of applying AI to the autonomous control of manufacturing equipment, we introduced an example of applying reinforcement learning and Sim2Real to lens alignment. The application of AI technology to manufacturing equipment is rare worldwide and a first for our company. We compared manual alignment, conventional automatic alignment, and reinforcement learning-based alignment and confirmed that reinforcement learning-based alignment is faster and more efficient. Our company has now completed the introduction of this technology to manufacturing equipment and is conducting operational maintenance. In the future, we aim to apply even more AI technologies to achieve further business benefits.

Reference

1) Kiminori Kurosawa, Takahiro Nitawaki, and Tran Duc Son, "Methods for Promoting AI and Technological Developments for the Realization", Fujikura Technical Review, No.51, pp. 39 – 51, February 2022.
2) Ryo Shiratori,Takahiro Nitawaki,Tran Duc Son,Yohei Kasai,and Kiminori Kurosawa," The application of AI control in the manufacturing industry using reinforcement learning and Sim2Real techniques", SSII 2022. Japanese.
3) Tatsuo Akiyama,"Practical Use Case of Reinforcement Learning and Sim-to-Real Transfer for Manufacturing", NVIDIA GTC 2023.
4) D. Silver, et al., "Deterministic policy gradient algorithms," Proceedings of the 31st International Conf. on Machine Learning, vol. 32, pp. 387 - 395, June 2014.
5) V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529 - 533, Feb. 2015.
6) Go Takami, "AI-based Plant Control", Yokogawa technical report., Vol.63, No.1, pp. 33 – 36, 2020. Japanese .
7) S. Daftry, J. A. Bagnell, and M. Hebert, "Learning transferable policies for monocular reactive mav control," in International Symposium on Experimental Robotics, 2016.
8) J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks fromsimulation to the real world," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017.
9) B. Balaji, S. Mallya, S. Genc, S. Gupta, L. Dirac, V. Khare, G. Roy, T. Sun, Y. Tao, B. Townsend, E. Calleja, S. Muralidhara and D. Karuppasamy, "DeepRacer: Autonomous Racing Platform for Experimentation with Sim2Real Reinforcement Learning," in IEEE International Conference on Robotics and Automation, 2020.
10) T. Mingxing and Le. Quoc, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Proceedings of the 36th International Conference on Machine Learning, vol. 97, pp. 6105 – 6114, 2019.
11) Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot and N. Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," Proceedings of the 33rd International Conference on Machine Learning, vol. 48, pp. 1995 – 2003, 2016.
12) C. Xiao, P. Lu and Q. He, "Flying Through a Narrow Gap Using End-to-End Deep Reinforcement Learning Augmented With Curriculum Learning and Sim2Real," IEEE Transactions on Neural Networks and Learning Systems, 2021.
13) L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," ICLR 2020.