

Received January 14, 2021, accepted January 22, 2021, date of publication February 2, 2021, date of current version February 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054960

# Motion Generation Using Bilateral Control-Based Imitation Learning With Autoregressive Learning

AYUMU SASAGAWA<sup>1</sup>, (Student Member, IEEE), SHO SAKAINO<sup>2,3</sup>, (Member, IEEE), AND TOSHIKI TSUJI<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Graduate School of Science and Engineering, Saitama University, Saitama 338-8570, Japan

<sup>2</sup>Department of Intelligent Interactive Systems, University of Tsukuba, Ibaraki 305-8577, Japan

<sup>3</sup>JST PRESTO, Saitama 332-0012, Japan

Corresponding author: Ayumu Sasagawa (a.sasagawa.997@ms.saitama-u.ac.jp)

This work was supported in part by the JST PRESTO, Japan, under Grant JPMJPR1755.

**ABSTRACT** Imitation learning has been studied as an efficient and high-performance method to generate robot motion. Specifically, bilateral control-based imitation learning has been proposed as a method of realizing fast motion. However, the learning approach of this method leads to the accumulation of prediction errors during the prediction process and may not generate desirable long-term behavior. Therefore, in this paper, we propose a method of autoregressive learning for bilateral control-based imitation learning to reduce the accumulation of prediction errors. A new neural network model for implementing autoregressive learning is also proposed. Three types of experiments are conducted to verify the effectiveness of the proposed method, where the method is shown to have improved performance over those of conventional approaches. Due to the structure and method of autoregressive learning employed by the developed model, the proposed method can generate desirable long-term motion for successful tasks and has a high generalization ability for environmental changes based on the human demonstrations of tasks.

**INDEX TERMS** Bilateral control, imitation learning, motion planning, robot learning.

## I. INTRODUCTION

Robots that can execute various tasks automatically are becoming an increasingly important focus of research in the field of robotics. Recently, machine learning has exhibited high performance in various fields, including image recognition [1], machine translation [2], expression recognition [3], human activity recognition [4], and robotics [5]–[8]. Also, approaches based on end-to-end learning for motion generation have also recently shown high performance [9]–[12]. Where approaches based on reinforcement learning require a considerable number of trials [10], end-to-end learning reduces the effort required for programming, and complex robotic motion can be easily generated. Moreover, end-to-end learning methods have a high generalization ability for situation-related changes.

Among these approaches, imitation learning (IL) and learning from demonstration have attracted attention as methods for efficiently learning robot motion [13], [15]–[18]. These learning-based methods use datasets derived from

human demonstrations of tasks. Yang *et al.* realized autonomous robot operations using neural network (NN) models [17], whereas another study proposed a method that combines reinforcement learning and IL [18]. IL using force information has also been proposed [19]–[25], where force control improves the robustness against position fluctuations. More specifically, force control increases the possibility of adapting to complex tasks that require force information and thus enables the accomplishment of a greater number of various tasks. References [19], [20] used haptic devices to collect force information during human demonstrations of tasks. Rozo *et al.* realized cooperative work between a human and robot using a Gaussian mixture model [21], and Ochi *et al.* used NN models to integrate visual, position, and force information to perform tasks [23]. In addition, [25] used dynamic movement primitives to model human demonstrations of tasks and realized the automated task of writing letters. A common problem with these approaches is that robot motion is extremely slow compared to that of humans.

We previously proposed a bilateral control-based IL as a method that uses force information [26], [27]. Bilateral

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

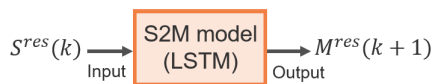


FIGURE 1. Network model of our bilateral control-based IL.

control is a remote-control system that uses two robots, namely, a master and slave [28]–[31]. We employed bilateral control in a series of human demonstrations of tasks, where a human operated the master robot, with the slave being teleoperated and conducting tasks within the workspace. As Fig. 1 shows, the NN model for motion generation predicted the master state from the slave state. The NN model included long short-term memory (LSTM) [32] to predict sequence data. Here,  $S$  and  $M$  represent the slave and master, respectively, the superscript  $res$  indicates the response values, and  $k$  represents the step of the sequence data. Our bilateral control-based IL can execute tasks requiring a force adjustment and realize fast motion that a conventional IL [19]–[25] cannot. As Fig. 2(a) shows, in conventional IL, the response values collected during the demonstrations are given as the command values for autonomous operations, and delays among the command and the response values are not strictly considered. However, in robot control, eliminating control delays is virtually impossible. In addition, when performing tasks, including contact with the environment, delays due to physical interactions occur. For this reason, a conventional IL can achieve only slow motion where delays can be ignored. Based on the foregoing, the following two conditions must be satisfied to realize fast motion in the IL:

- (i) command values must be predicted during autonomous operation, *i.e.*, collected during the demonstrations,
- (ii) the same control system must be implemented during the demonstrations and autonomous operations.

Our bilateral control-based IL can satisfy these two conditions for the following reasons. First, in bilateral control, the command values of the slave are the response values of the master, and both of them can be measured independently. As a result, the slave's command values of the slave can be predicted during autonomous operations. As Fig. 2(b) shows, in our bilateral control-based IL, delays that occur during the demonstrations also occur during autonomous operations. Second, as Fig. 3 shows, in a bilateral control-based IL, the system is designed to reproduce bilateral control during an autonomous operation. Thus, the control system of the slave can be the same during the demonstrations and autonomous operations. In demonstrations using bilateral control, humans collect data while considering delays, *i.e.*, humans demonstrate skills to compensate for delays. If the control system is different during demonstrations and autonomous operations, this compensation skill will be lost.

Although our bilateral control-based IL can achieve a fast and dynamic motion, it does have a drawback. Specifically, the learning method of this approach is unsuitable for long-term predictions because the NN model is trained without autoregressive learning. This learning method is called teacher forcing [33]. When the NN model is trained using

teacher forcing, if prediction errors occur during the prediction process, the errors will accumulate, and the robot will not realize a desirable behavior. Autoregressive learning is a method to solve this problem, in which the output at the previous step is input to the model in the subsequent step. This method is called free running [34]. Because autoregressive learning predicts a series of motions continuously, the model is learned to minimize the total errors of the long-term prediction. To implement autoregressive learning, the input and output of the model must be the same variables. In general, the implementation of autoregressive learning is simple [35] because the input and output of the model are the same variables, *i.e.*, response values. By contrast, in our bilateral control-based IL, the output of the model cannot be used as the next input because the input and output of the model are different variables, *i.e.*, the response values of the slave and of the master (Fig. 1). Therefore, we propose a model in which the input and output of the proposed model have both master and slave response values to implement autoregressive learning in a bilateral control-based IL. In summary, the main contributions of this paper are as follows:

- (i) A new NN model for autoregressive learning in our bilateral control-based IL is proposed.
- (ii) The proposed model improves the ability to generate long-term behavior, and it leads to improve the adaptability to environmental changes.

In this study, the proposed model was compared with conventional models. For the experiments, three tasks were conducted to clarify the effectiveness of the proposed method. The success rates of the tasks were used to evaluate the performance of the method. In all experiments, the proposed method showed an excellent performance equal to or greater than that of previous conventional methods.

The remainder of this paper is organized as follows. Section II introduces the control system and bilateral control, and Section III describes the method of the bilateral control-based IL. Section IV describes autoregressive learning and the NN models for the proposed method and previous conventional methods. Section V describes the experiments and presents the results of the three tasks. Section VI provides concluding remarks regarding this study and discusses areas of future research.

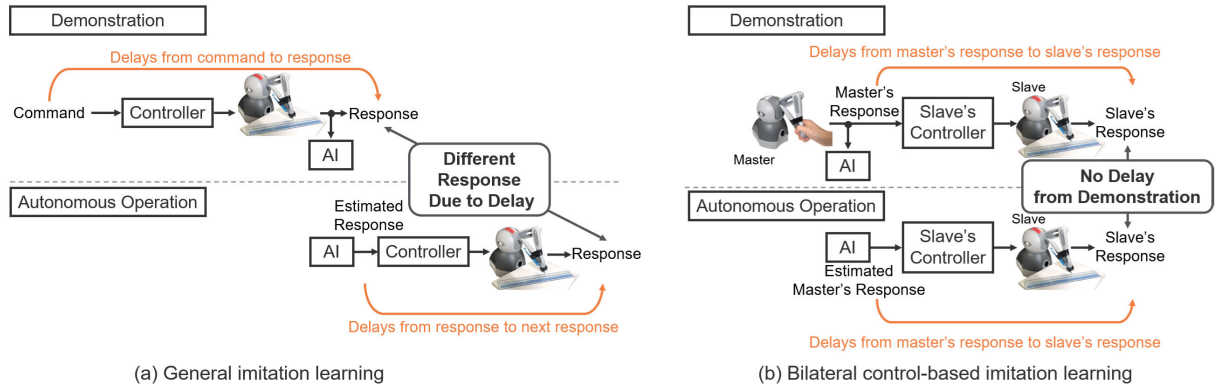
## II. CONTROL SYSTEM

### A. ROBOT

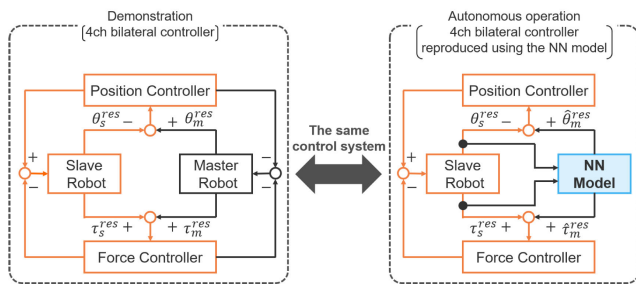
Two Touch<sup>TM</sup>, which are haptic devices manufactured by 3D Systems, were used in the experiments. Two robots were used as master and slave, respectively. The robots had three-degrees-of-freedom (DOF), as shown in Fig. 4. The robots could measure only the joint angles  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  with the encoders. Here, the subscripted numbers represent each joint shown in Fig. 4.

### B. BILATERAL CONTROL

Bilateral control is a type of remote-control system that uses two robots, *i.e.*, a master and slave [28]–[31]. In this study,



**FIGURE 2.** Overview of general IL and our bilateral control-based IL. In general, the delays caused during the demonstration and autonomous operation are different. Therefore, a general IL can realize only slow motion, thus ignoring delays. In the bilateral control-based IL, the delays caused during the demonstration and autonomous operation are the same. Thus, in our bilateral control-based IL, fast motion with delays can be achieved.

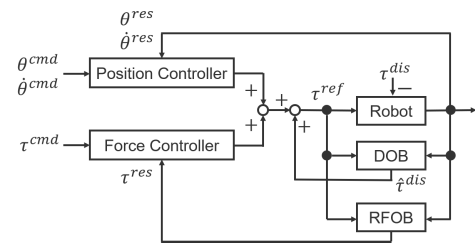


**FIGURE 3.** 4ch bilateral controllers during the demonstrations and the autonomous operation. The figure on the left side shows a 4ch bilateral controller. This controller was used during the human demonstrations of tasks. As the right side of the figure shows, the master robot and master's controllers were replaced with the NN model to virtually reproduce the 4ch bilateral controller during an autonomous operation. With the method proposed in this study, both the master and slave response values were input to the NN model. The same control system was applied during the human demonstrations of tasks and autonomous operation. Note that the system including the slave robot and slave's controllers (orange-colored lines) must be the same because the slave was used during the autonomous operation.



**FIGURE 4.** Robot (Touch™).

4ch bilateral control [36], [37] was used from among various types of bilateral control because it offers the highest performance and excellent operability, and the slave and master consist of both position and force controllers. Therefore, 4ch bilateral control is suitable for IL [27]. In bilateral control, when the operator operates the master, the slave is teleoperated. The control goal is to synchronize the position and satisfy the law of action and reaction forces between the two



**FIGURE 5.** Controller.

robots. The reaction force caused by the contact between the slave and environment is presented to the master. Thus, the operator can feel the interactions between the slave and environment. The control law of the 4ch bilateral control is expressed through the following equations using the angle response values  $\theta^{res}$  and torque response values  $\tau^{res}$  of the robots. A block is given on the left side of Fig. 3. Note that the subscripts  $s$  and  $m$  represent the slave and master, respectively, and the superscript  $res$  represents the response values.

$$\theta_m^{res} - \theta_s^{res} = 0, \tag{1}$$

$$\tau_m^{res} + \tau_s^{res} = 0. \tag{2}$$

### C. CONTROLLER

The control system consisted of position and force controllers, as shown in Fig. 5. Here,  $\theta$ ,  $\dot{\theta}$ , and  $\tau$  represent the joint angles, angular velocity, and torque of each joint, respectively, the superscripts  $res$ ,  $cmd$ , and  $ref$  indicate the response, command, and reference values, respectively. In addition,  $\theta^{res}$  was measured by the encoders of the robots, and  $\dot{\theta}^{res}$  was calculated using pseudo-differentiation. The disturbance torque  $\tau^{dis}$  was estimated using a disturbance observer (DOB) [38] as  $\hat{\tau}^{dis}$ . Furthermore, a reaction force observer (RFOB) [39] was used to calculate the reaction force  $\tau^{res}$ . Details of the RFOB are provided in Section II-D. The position controller also included proportional and derivative controllers, where the force controller comprised a

**TABLE 1.** Gains and identified system parameters for the robot controller.

| Parameter  |   | Value |
|------------|---|-------|
| $J_1$      | Joint 1's inertia [mkgm <sup>2</sup> ]                  | 2.55  |
| $J_2$      | Joint 2's inertia [mkgm <sup>2</sup> ]                  | 4.30  |
| $J_3$      | Joint 3's inertia [mkgm <sup>2</sup> ]                  | 1.12  |
| $G_1$      | Gravity compensation coefficient 1 [mNm]                | 79.0  |
| $G_2$      | Gravity compensation coefficient 2 [mNm]                | 55.0  |
| $G_3$      | Gravity compensation coefficient 3 [mNm]                | 33.0  |
| $D$        | Friction compensation coefficient[mkgm <sup>2</sup> /s] | 4.55  |
| $K_p$      | Position feedback gain                                  | 121   |
| $K_d$      | Velocity feedback gain                                  | 22.0  |
| $K_f$      | Force feedback gain                                     | 1.00  |
| $g$        | Cut-off frequency of pseudo differentiation [rad/s]     | 40.0  |
| $g_{DOB}$  | Cut-off frequency of DOB [rad/s]                        | 40.0  |
| $g_{RFOB}$ | Cut-off frequency of RFOB [rad/s]                       | 40.0  |

proportional controller. The torque reference values  $\tau^{ref}$  of the slave and master were calculated as follows:

$$\tau_m^{ref} = -\frac{J}{2}(K_p + K_d s)(\theta_m^{res} - \theta_s^{res}) - \frac{1}{2}K_f(\tau_m^{res} + \tau_s^{res}), \quad (3)$$

$$\tau_s^{ref} = \frac{J}{2}(K_p + K_d s)(\theta_m^{res} - \theta_s^{res}) - \frac{1}{2}K_f(\tau_m^{res} + \tau_s^{res}), \quad (4)$$

where  $s$  represents the Laplace operator,  $J$  is the inertia, and  $K_p$ ,  $K_d$ , and  $K_f$  represent the position, velocity, and force control gain, respectively. The gain values and cut-off frequencies used in the experiments are listed in Table 1.

Also, robot dynamics are represent as follows:

$$J_1 \ddot{\theta}_1^{res} = \tau_1^{ref} - \tau_1^{dis} - D\dot{\theta}_1^{res}, \quad (5)$$

$$J_2 \ddot{\theta}_2^{res} = \tau_2^{ref} - \tau_2^{dis} - G_1 \cos \theta_2^{res} - G_2 \sin \theta_3^{res}, \quad (6)$$

$$J_3 \ddot{\theta}_3^{res} = \tau_3^{ref} - \tau_3^{dis} - G_3 \sin \theta_3^{res}. \quad (7)$$

where  $D$  and  $G$  represent the friction compensation and gravity compensation coefficients, respectively, and the subscripts represent the different joints of the robots. Also,  $\ddot{\theta}$  means angular acceleration. The off-diagonal term of the inertia matrix was ignored because it was negligibly small.

#### D. SYSTEM IDENTIFICATION

The parameters of the control system were identified based on [40]. Friction  $D$  and gravity  $G$  were identified under free motion, assuming  $\tau^{dis} = 0$ . The DOB calculated the estimated disturbance torque  $\hat{\tau}^{dis}$  as follows:

$$\hat{\tau}^{dis} = \tau^{ref} - J\ddot{\theta}^{res}. \quad (8)$$

When the RFOB is used, each force can be measured without using a force sensor, and the torque response values of each joint were calculated as follows:

$$\tau_1^{res} = \hat{\tau}_1^{dis} - D\dot{\theta}_1^{res}, \quad (9)$$

$$\tau_2^{res} = \hat{\tau}_2^{dis} - G_1 \cos \theta_2^{res} - G_2 \sin \theta_3^{res}, \quad (10)$$

$$\tau_3^{res} = \hat{\tau}_3^{dis} - G_3 \sin \theta_3^{res}. \quad (11)$$

All identified parameters used in the experiment are listed in Table 1.

### III. METHOD OF BILATERAL CONTROL-BASED IMITATION LEARNING

In our experiments, the robots learned behaviors from human demonstrations of tasks and then conducted the tasks autonomously. In the demonstrations, the desired tasks were conducted using 4ch bilateral control. A human operated the master, and the slave performed the tasks in the given workspace. The joint angles, angular velocity, and torque response values of the two robots were used as the dataset for model training. Both the control and data-saving cycles were 1 ms long.

The NN model was then trained using the dataset derived from the demonstrations. The NN model consisted of LSTM and fully connected layers to learn the time-series data. The number of LSTM layers was set for each task by trial and error. Details are described in Section V. The activation functions for the LSTM and fully connected layers were a hyperbolic tangent function and identity mapping, respectively. Essentially, the model was trained to input the state at time  $t$  and output the state at time  $t + 20$  ms. Whether either the input or output were the master or slave state depended on each model described in Section IV. The state consisted of the joint angles, angular velocity, and torque response values. The loss function is the mean squared error (MSE) between the model output values and true values of the dataset. The model was learned to minimize the loss function with Adam optimization [41]. The dataset values were normalized to  $[0, 1]$  before the input to the model.

Finally, the trained model generated the motion, and the robot autonomously conducted the tasks. The control system was designed to reproduce 4ch bilateral control during the autonomous operation. The joint angle, angular velocity, and torque response values of the slave were measured in real time and input to the learned model. The command values predicted by the model were normalized before the input to the slave controller. Note that the prediction and control cycles of the model and robot were 20 and 1 ms, respectively.

### IV. NEURAL NETWORK MODEL

#### A. AUTOREGRESSIVE LEARNING

Fig. 6 shows the LSTM model developed in the time direction. Here,  $x$  represents an arbitrary value used for the input and output, the superscript  $tr$  represents teacher data, and  $\hat{\circ}$  represents the predicted values of the model. Fig. 6(a) shows the learning method without autoregressive learning. With this method, the teacher data are input at each step during the learning process, *i.e.*, the input values are completely unaffected by the prediction in the previous steps. More precisely, the output error used to train the NN model is based only on a one-step prediction.

However, as Fig. 6(c) shows, the model's prediction values are used as the next input during the prediction process. Therefore, without autoregressive learning, the input probability distributions are different between the learning and prediction processes. If prediction errors occur because of

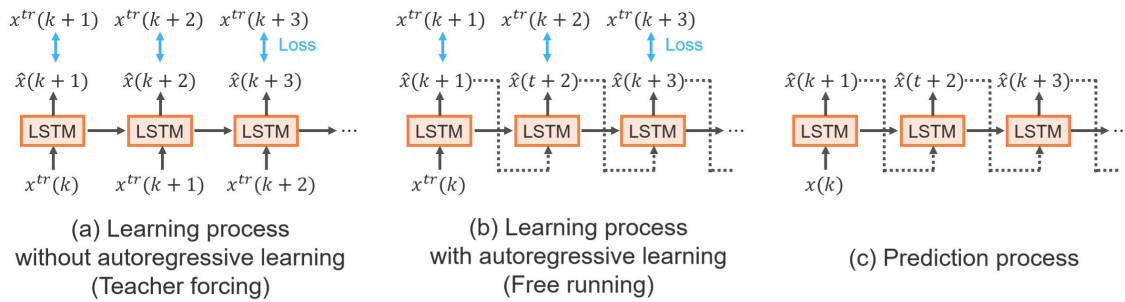


FIGURE 6. Learning and prediction methods using LSTM with and without autoregressive learning.

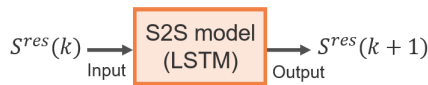


FIGURE 7. Network version of the S2S model (conventional method).

this difference, they will accumulate step by step. Therefore, although the model is learned with high accuracy during the learning process, it cannot generate desirable behavior during the prediction process. This problem similarly occurs in the field of natural language processing when using a recurrent NN [42]–[45]. It is known that the gap in input distributions can be filled by autoregressive learning, which uses the model’s predicted values as input during the learning process, as shown in Fig. 6(b). In addition, during autoregressive learning, prediction errors accumulate step by step during the learning process, and the model is optimized to reduce these errors. Therefore, the model is learned to behave properly throughout the continuous-time series. Autoregressive learning prevents error accumulation during the prediction process, and the model is more likely to generate desirable behavior in the long-term to execute tasks.

In a conventional bilateral control-based IL [26], [27], autoregressive learning cannot be implemented. This is because the input and output of the model are different variables, *i.e.*, the slave and master have different response values (Fig. 1). In this study, the SM2SM model is proposed to solve this problem. In addition, the performances of three models are compared with that of the proposed method. Summary information about the different models is provided in Table 2. The general IL [19]–[25] predicts the next response values from the current ones. Therefore, the S2S model that predicts the next slave state from the current one is used as a comparison method that replicates the general IL. The S2M model is used as the conventional bilateral control-based IL [26], [27] and the SM2SM model is used as the proposed method. Details of each model are provided in the following sections.

### B. S2S MODEL (CONVENTIONAL MODEL)

As Fig. 7 shows, the S2S model predicts the next slave state from the current one. The input and output consist of the joint angles, angular velocity, and torque response values of the slave with three DOFs. In other words, the S2S model has nine inputs and nine outputs.

### 1) LEARNING

During the learning process, the slave’s response values were input, and the slave’s response values 20 ms later were output. The S2S model was trained without or with autoregressive learning. The cases without and with autoregressive learning were called S2S-w/o-AR and S2S-AR, respectively. The number of autoregressive steps was set to 10 to allow the prediction errors to converge quickly. In other words, every 10 steps, the values of the training dataset were input instead of the predicted values from the previous step.

### 2) AUTONOMOUS OPERATION

The model predicted the response values of the slave. The predicted values of the model were used as the command values of the slave.

### C. S2M MODEL (CONVENTIONAL MODEL)

As Fig. 1 shows, the S2M model predicted the next state of the master from the current state of the slave. The input consisted of the joint angles, angular velocity, and torque response values of the slave with three DOFs. The output consisted of these response values of the master with three DOFs. Therefore, the S2M model had nine inputs and nine outputs.

### 1) LEARNING

During the training, the response values of the slave were input, and the model output the response values of the master 20 ms after the input was applied. In the case of the S2M model, the model was trained without autoregressive learning because it could not be implemented. The S2M model without autoregressive learning is called S2M-w/o-AR.

### 2) AUTONOMOUS OPERATION

The model predicted the response values of the master; these predicted values were used as the command values of the slave.

### D. SM2SM MODEL (PROPOSED MODEL)

SM2SM was the proposed model applied to adapt autoregressive learning to a bilateral control-based IL. As Fig. 8(a) indicates, the SM2SM model predicted the next state of the

TABLE 2. Details of the NN model.

| Model                     | Neural network model        |                             | Autoregressive learning |
|---------------------------|-----------------------------|-----------------------------|-------------------------|
|                           | Input                       | Output                      |                         |
| S2S-w/o-AR                | Slave (9 dims.)             | Slave (9 dims.)             | -                       |
| S2S-AR                    | Slave (9 dims.)             | Slave (9 dims.)             | ✓                       |
| S2M-w/o-AR                | Slave (9 dims.)             | Master (9 dims.)            | -                       |
| SM2SM-w/o-AR              | Slave and master (18 dims.) | Slave and master (18 dims.) | -                       |
| SM2SM-AR (Proposed model) | Slave and master (18 dims.) | Slave and master (18 dims.) | ✓                       |

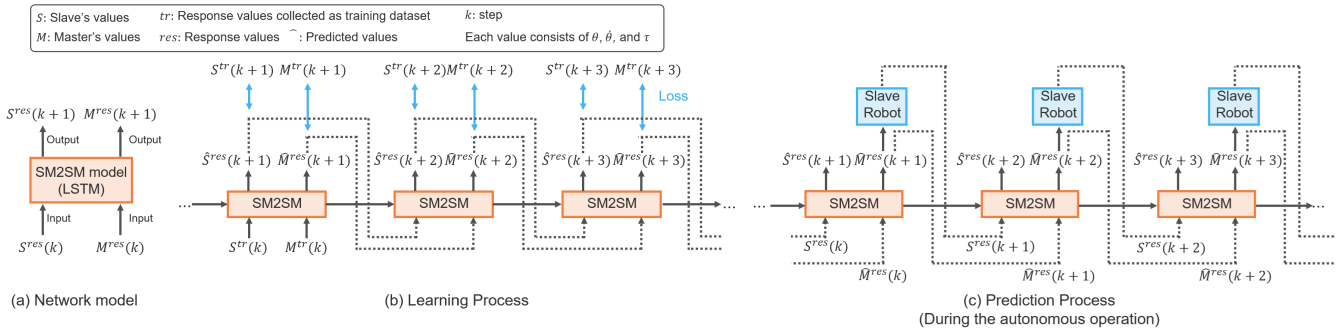


FIGURE 8. Learning process and prediction process of the SM2SM model (proposed method).

slave and master from their respective current states. In contrast to the S2M model, the input and output of the SM2SM model consisted of both slave and master states. Therefore, autoregressive learning could be implemented. In addition, the structure of this model was shown to have another advantage. In bilateral control-based IL, the relationship between the slave and master must be learned accurately to reproduce the demonstrations. During the learning of the S2M model, only one relationship was used, which was the prediction from the slave to the master state. By contrast, in the SM2SM model, four relationships were used: predictions from the slave to the slave state, from the slave to the master state, from the master to the slave state, and from the master to the master state. These improved the learning of the dynamics of bilateral control. Based on the foregoing, because interactions between master and slave robots could be implicitly learned by the SM2SM model, it was expected that the SM2SM model would be a suitable model for bilateral control-based IL. The input and output for the SM2SM model consisted of the joint angles, angular velocity, and torque response values of the slave and master with three DOFs. In other words, the SM2SM model had 18 inputs and 18 outputs.

1) LEARNING

An overview of the learning process of the SM2SM model is shown in Fig. 8(b). During the learning process, the response values of the slave and master were input, and the response values of the slave and master 20 ms later were output. The SM2SM model was learned without or with autoregressive learning. The cases without and with autoregressive learning were called SM2SM-w/o-AR and SM2SM-AR, respectively. In this study, the number of autoregressive steps was set to 10 to ensure that the prediction errors converged quickly.

2) AUTONOMOUS OPERATION

Overview during the autonomous operation is shown in Fig. 8(c). The slave state among the input to the model was the slave response values measured in real time. By contrast, the master state among the inputs of the model was that predicted by the model one step before. The states of the master predicted by the model were used as the command values of the slave.

V. EXPERIMENT

In the experiment, three types of tasks were conducted to clarify the effectiveness of the proposed method. The following abilities were verified in the three tasks:

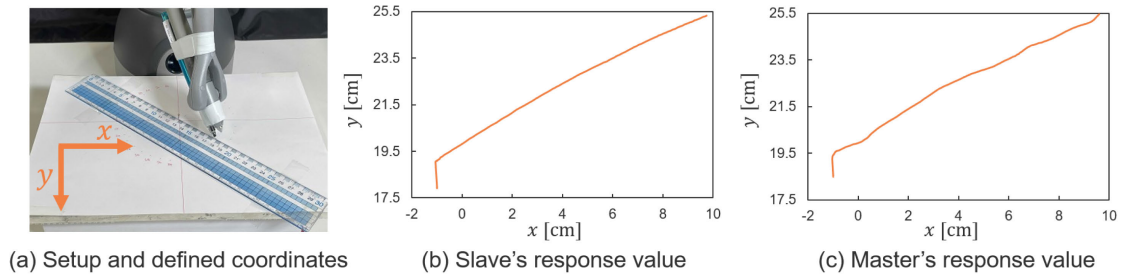
- (i) Adaptability to environmental changes (confirmed in experiment 1 [V-A]).
- (ii) Ability to realize fast motion (confirmed in experiment 2 [V-B]).
- (iii) Ability to generate desirable long-term behavior (confirmed in experiment 3 [V-C]).

Three types of NN models were compared for each experiment. Only the S2M model was conducted without autoregressive learning, whereas the S2S and SM2SM models were compared with and without autoregressive learning, thus resulting in five types of models being compared. The success rates of the tasks verified the performance of each model.

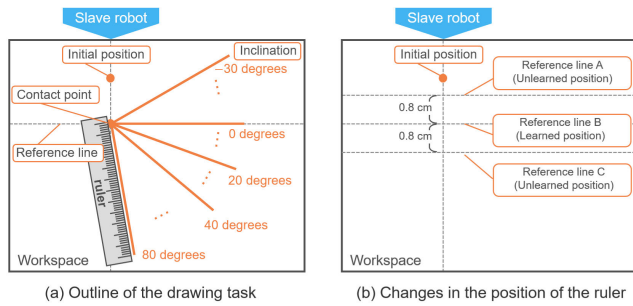
A. EXPERIMENT 1 (DRAWING A LINE USING A PEN AND RULER)

1) TASK DESIGN

Fig. 9(a) shows the setup of this experiment. A mechanical pencil was fixed to the slave. Initially, the slave moved from the initial position toward the ruler. After touching the ruler,



**FIGURE 9.** Setup and training data of the drawing task when drawing a 40 degree line: (b) The response values of the slave did not include large fluctuations. (c) By comparison, the response values of the master included large fluctuations (Fig. c).



**FIGURE 10.** Outline of the drawing task.

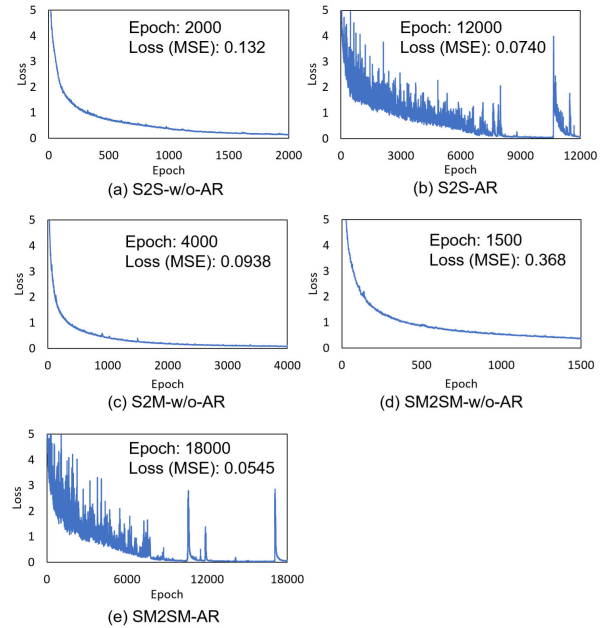
the slave then drew a straight line to the right along the ruler. The goal of this task was to draw lines according to various inclinations. As Fig. 10(a) shows, the inclination was defined by the angle at which the ruler was rotated around the point where the pen first contacted the ruler. Zero degrees is represented by the “reference line” in the figure. To succeed in this task, a proper adjustment of the contact force between the pen and ruler or paper was required. In addition, adaptability to unknown inclinations or unknown positions of the ruler was required.

2) HUMAN DEMONSTRATIONS AND DATASET FOR LEARNING

We collected data using ruler inclinations of zero, 20, and 40 degrees, as shown Fig. 10-(a). Eight trials were conducted for each inclination, and the total number of trials was 24. The time for one trial was 3 s. The slave began moving from the initial position and, within 3 s, drew a line of 5 cm or longer along the edge of the ruler.

3) NURAL NETWORK ARCHITECTURE

The NN model consisted of six LSTM layers, followed by a fully connected layer. The unit size of all layers was 50. The mini-batch consisted of 100 random sets of 150 time-sequential samples corresponding to 3 s. The loss graph for each model is given as Fig. 11. As Figs. 11(b) and (e) show, models with autoregressive learning required many learning epochs to converge the loss function. Note that the performance of the autonomous operation did not necessarily depend on the value of the loss function. Although



**FIGURE 11.** Relationship between loss and epoch number in the drawing task. The vertical axis shows the prediction loss calculated by MSE. The maximum value was set to 5 to enlarge the graph. The horizontal axis shows the number of learning epochs, and the maximum value of each model is the number of epochs used for the validation. In addition, the epoch and loss in the graph represent the number of epochs used for the validation and the value of the MSE at each epoch, respectively.

the loss value was useful for roughly identifying the high-performance model, we identified the optimal number of learning epochs by comparing the performance of the autonomous operation in trial and error.

4) TASK VALIDATION

To verify the autonomous operation, the performance of the ruler inclinations from  $-30$  to  $80$  degrees was verified every 10 degrees. Success was defined by those cases in which the robot drew a line of 5 cm or longer along the edge of the ruler. Verification was conducted through three trials for the inclination of each ruler. In addition, the performance was validated when the ruler’s position was shifted back and forth. Here, the position of the ruler was defined based on the point where the pen first contacted the ruler. As Fig. 10(b) shows, the validation was conducted by shifting 0.8 cm

TABLE 3. Success rates of the drawing task.

| Model                         | Reference line | Success Rate [%]  |     |     |      |     |     |     |     |     |     |      |      |      | Subtotal       | Total |
|-------------------------------|----------------|-------------------|-----|-----|------|-----|-----|-----|-----|-----|-----|------|------|------|----------------|-------|
|                               |                | Inclination [deg] |     |     |      |     |     |     |     |     |     |      |      |      |                |       |
|                               |                | -30               | -20 | -10 | 0*   | 10  | 20* | 30  | 40* | 50  | 60  | 70   | 80   |      |                |       |
| S2S-w/o-AR                    | A              | 0                 | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 91.7 | 88.0 (95/108)  |       |
|                               | B**            | 0                 | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 91.7 |                |       |
|                               | C              | 0                 | 0   | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 66.7 | 81.0 |                |       |
| S2S-AR                        | A              | 0                 | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 91.7 | 93.5 (101/108) |       |
|                               | B**            | 0                 | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 91.7 |                |       |
|                               | C              | 100               | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 66.7 | 97.2 |                |       |
| S2M-w/o-AR                    | A              | 0                 | 0   | 0   | 66.7 | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 66.7 | 69.4 | 66.7 (72/108)  |       |
|                               | B**            | 0                 | 0   | 0   | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 66.7 | 0    | 63.9 |                |       |
|                               | C              | 0                 | 0   | 0   | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 0    | 66.7 |                |       |
| SM2SM-w/o-AR                  | A              | 0                 | 0   | 100 | 100  | 100 | 100 | 100 | 100 | 0   | 100 | 100  | 100  | 75.0 | 81.0 (87/108)  |       |
|                               | B**            | 0                 | 0   | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 83.3 |                |       |
|                               | C              | 0                 | 0   | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 83.3 |                |       |
| SM2SM-AR<br>(Proposed method) | A              | 100               | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 100  | 100 (108/108)  |       |
|                               | B**            | 100               | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 100  |                |       |
|                               | C              | 100               | 100 | 100 | 100  | 100 | 100 | 100 | 100 | 100 | 100 | 100  | 100  | 100  |                |       |

\*: Learned inclination of the ruler      \*\*: Learned position of the ruler

and forth from the learned position. The learned position was “reference line B,” and the unlearned positions were “reference line A” and “reference line C.”

### 5) EXPERIMENTAL RESULTS

The success rates of each model are shown in Table 3. First, when the models without autoregressive learning were compared, S2S-w/o-AR had a higher success rate than S2M-w/o-AR and SM2SM-w/o-AR. As mentioned in Section III, the S2M model was more suitable than the S2S model for IL, including for fast motion with delays. However, this task was not particularly fast. In addition, during the drawing task, the motion of the slave was restrained by the ruler. The dataset of the slave’s response values was easy to learn because it did not include large fluctuations, as shown in Fig. 9. By contrast, fluctuations may have been contained in the master responses because the master was not restrained by anything, as shown in Fig. 9. Therefore, when the master responses were used in the input or output of the model, such as in the S2M and SM2SM models, learning was difficult. In addition, SM2SM-w/o-AR exhibited a higher performance than S2M-w/o-AR. As described in Section IV-D, the structure of the SM2SM model was more suitable than that of the S2M model because accurately understanding the relationship between the master and slave was necessary for bilateral control-based IL.

Furthermore, SM2SM-AR had a higher success rate than SM2SM-w/o-AR as well as the highest success rate among all models. In particular, compared to other methods, SM2SM-AR had a high adaptability to changes in the ruler’s position and extrapolation inclinations. As described in Section IV, autoregressive learning is a method that was evaluated not by the prediction error of only a single step, but by the prediction errors of all consecutive steps. Therefore, the model could properly generate a series of motions to perform a task even for unknown environments, and the effects of the fluctuation of the master’s responses were negligible. These results indicate that the proposed model’s structure and autoregressive learning improved the generalization performance for unknown environments, even with fluctuating responses. The generalization of the proposed method, which can achieve

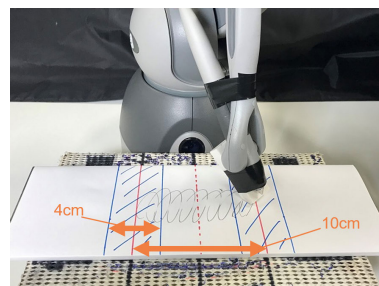


FIGURE 12. Experimental setup of the erasing task. During human demonstrations of tasks, training data were collected for the task of erasing the area inside the solid red line shown in the figure. In verifying the autonomous operation, success was defined as the case in which the movement was switched in the opposite direction in the area indicated by the blue diagonal line.

high success rates even in unknown environments, is expected to be applied to other tasks.

### B. EXPERIMENT 2 (ERASING A LINE USING AN ERASER)

#### 1) TASK DESIGN

Fig. 12 shows the steps of this experiment. An eraser was fixed to the slave. In the initial condition, the eraser was in contact with the paper at the solid red line on the left shown in Fig. 12. Then, the slave moved horizontally and erased a line written on the paper with the eraser. The goal of this task was to erase a line according to various paper heights. Adaptability to unknown paper heights was required. To succeed in this task, proper adjustment of the contact force between the eraser and paper was required. In this task, the robot had to operate quickly and utilize the inertial force, as considerable friction occurred between the eraser and paper.

#### 2) HUMAN DEMONSTRATIONS AND DATASET FOR LEARNING

We collected data with paper heights of 35, 55, and 75 mm. Five trials were conducted for each paper height, and the total number of trials was 15. The time for one trial was 10 s. The dataset was collected to erase the area inside the solid red line shown in Fig. 12. In the figure, the slave moved horizontally and repeatedly in the opposite direction at the solid red line.



TABLE 4. Success rates of the erasing task.

| Model                         | The number of LSTM layer | Success rate based on the success area [%] |      |      |      |      | Total               | Total rate of robot continued to perform the task during the trial [%] |
|-------------------------------|--------------------------|--|------|------|------|------|---------------------|--|
|                               |                          | Height [mm]                                |      |      |      |      |                     |  |
|                               |                          | 35*  | 45   | 55*  | 65   | 75*  |                     |  |
| S2S-w/o-AR                    | 2                        | 0  | 0    | 0    | 66.7 | 33.3 | 20.0 (3/15)         | 40.0 (6/15)  |
|                               | 4                        | 0  | 0    | 0    | 100  | 100  | 40.0 (6/15)         | <b>100</b> (15/15)   |
| S2S-AR                        | 2                        | 33.3                                       | 0    | 0    | 0    | 0    | 6.67 (1/15)         | 80.0 (12/15)   |
|                               | 4                        | 0  | 0    | 0    | 0    | 0    | 0 (0/15)            | <b>100</b> (15/15)   |
| S2M-w/o-AR                    | 2                        | 100  | 100  | 66.7 | 100  | 100  | <b>93.3</b> (14/15) | <b>100</b> (15/15)   |
|                               | 4                        | 100  | 66.7 | 66.7 | 100  | 100  | 86.7 (13/15)        | <b>100</b> (15/15)   |
| SM2SM-w/o-AR                  | 2                        | 0  | 0    | 0    | 0    | 33.3 | 6.67 (1/15)         | <b>100</b> (15/15)   |
|                               | 4                        | 100  | 66.7 | 100  | 100  | 100  | <b>93.3</b> (14/15) | <b>100</b> (15/15)   |
| SM2SM-AR<br>(Proposed method) | 2                        | 100  | 100  | 100  | 66.7 | 100  | <b>93.3</b> (14/15) | <b>100</b> (15/15)   |
|                               | 4                        | 100  | 100  | 33.3 | 66.7 | 100  | 80.0 (12/15)        | <b>100</b> (15/15)   |

\*: Learned height

The slave robot was teleoperated to reciprocate left and right within the area at approximately constant cycles.

### 3) NEURAL NETWORK ARCHITECTURE

The NN model consisted of two or four LSTM layers, followed by a fully connected layer. During this task, two types of NN architectures were used because the robot behavior differed depending on the number of LSTM layers, and the different architectures affected the results. The unit size of all layers was 50. The mini-batch consisted of 100 random sets of 300 time-sequential samples corresponding to 6 s.

### 4) TASK VALIDATION

In verifying the autonomous operations, the performances for paper heights of 35, 45, 55, 65, and 75 mm were validated. The paper heights of 45 and 65 mm were the untrained heights. Success was defined as the case in which the robot erased the line within the specified area. We defined the area of success to exclude cases in which the robot movements were too narrow or too wide as compared to successful demonstrations. In Fig. 12, when the robot reciprocated to the left and right, the success was defined as the case in which the movement was switched to the opposite direction in the area indicated by the blue diagonal line. The ability to erase the line with the appropriate force according to changes in height was an essential condition for success. The robot executed the task for 8 s during each trial, and the case in which the robot stopped during the trial was defined as a failure. Verification was conducted through three trials for each paper height.

### 5) EXPERIMENTAL RESULTS

The success rates of each model are shown in Table 4. The rates in the rightmost column of the table were determined using a different evaluation criterion than the aforementioned. These results show the percentages of trials in which the robot could continue erasing the line without stopping, regardless of whether the success criteria were satisfied based on the success area previously described. During this experiment, the results differed with the number of LSTM layers. The performance was validated by changing the number of LSTM layers of each model. As the table shows, the S2S model generally had low success rates. Many of the failures were

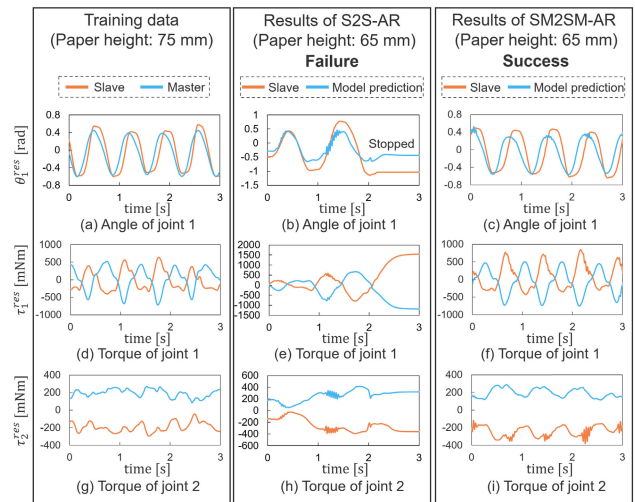
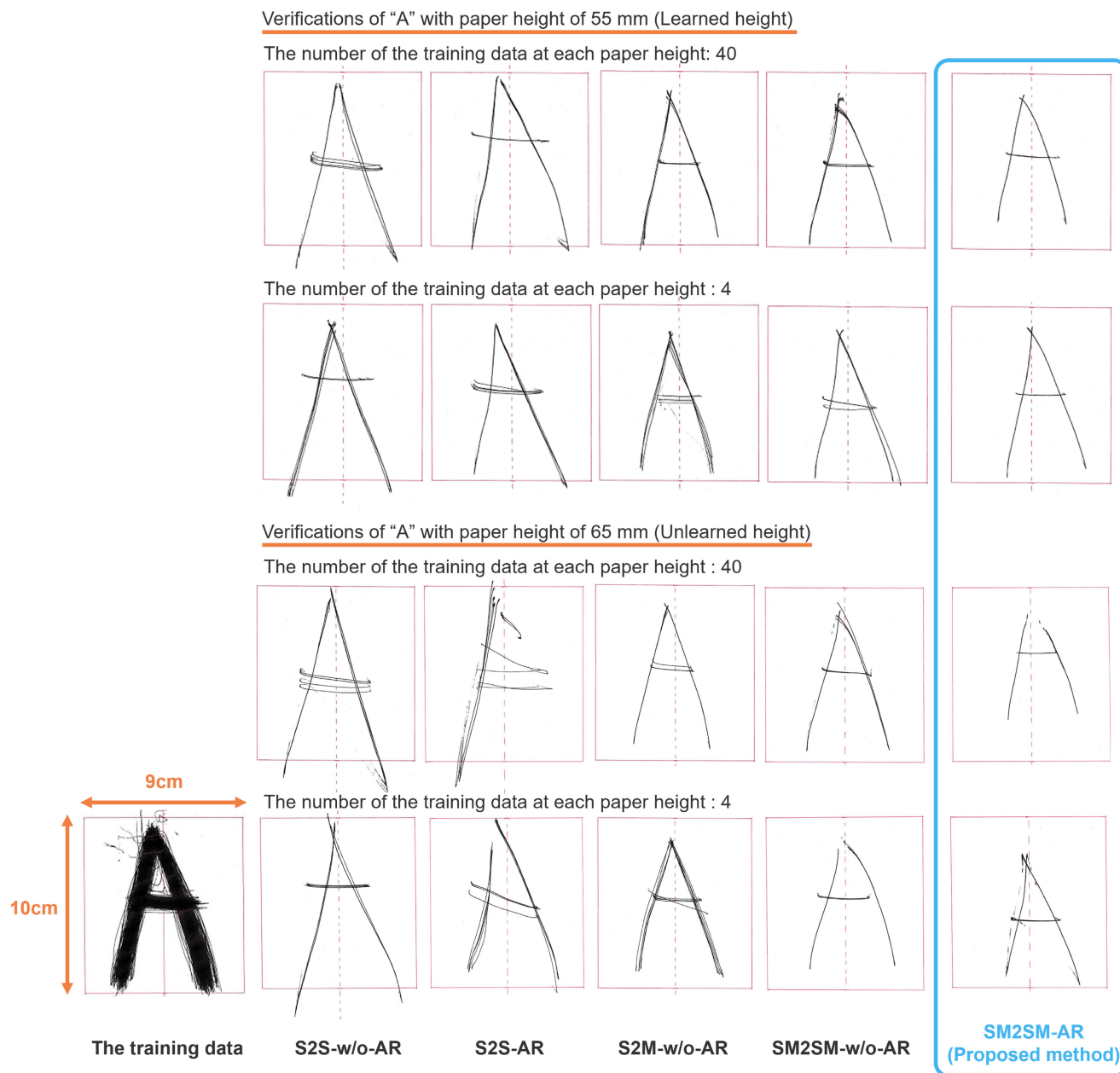


FIGURE 13. Training data and results of the erasing task. The results of the S2S-AR and SM2SM-AR are shown as examples.

cases in which the robot stopped due to friction between the eraser and paper, or the robot went outside the workspace. During this task, the robot had to move extremely quickly. In addition, considerable friction occurred between the eraser and paper. Thus, control delays and delays due to physical interactions occurred during the demonstrations. The training data and results are shown in Fig. 13. We focused on  $\theta_1^{res}$  because joint 1 moved mainly during the erasing task. In addition, the bottom of the figure shows  $\tau_2^{res}$ , which was the force exerted primarily in the vertically downward direction. Fig. 13(a) shows that a delay between the response values of the master and slave occurred. Therefore, the skill required to compensate for the delays performed by humans during the demonstrations had to be reproduced during the autonomous operation. The results of the S2S-AR model, as an example of task failure, are shown in the center of the figure. In the middle of the operation,  $\tau_1^{res}$  and  $\tau_2^{res}$  became greater than the appropriate force, and the robot could not move. The S2S model lost this compensation skill and could not complete this task requiring fast motion.

By contrast, both the S2M and SM2SM models showed high success rates. The SM2SM-AR model could reproduce the fast motion with delays while maintaining the same level



**FIGURE 14.** Training data and the results of the task of writing the letter "A." The figures on the far left are the letters written by a human during the demonstrations. Those from the training data appeared as thick because they were written on a single sheet of paper through all trials.

of appropriate force as the training data. In addition, the robot could properly erase the line without stopping during all trials. The robot applied the appropriate force even at unlearned heights. Most failures occurred when the movements were slightly beyond the success area. Although none of the models exhibited a perfect performance, as the success area was strictly defined, the S2M and SM2SM models achieved excellent performances in realizing fast motion while maintaining the proper force. Because this task involved a reciprocating motion with a short cycle, a long-term prediction was not required, and it was a relatively easy task for bilateral control-based IL. Therefore, even the conventional S2M model without autoregressive learning showed as high a success rate

as that of the proposed model. It was confirmed that the proposed method with autoregressive learning achieved an excellent performance even for a short-cycle task without adverse effects.

### C. EXPERIMENT 3 (WRITING LETTERS)

#### 1) TASK DESIGN

A ballpoint pen was fixed to the slave. The goal of this task was to write the letters "A" and "B" on paper. The initial position was the start point of the first stroke of each letter. The letter was written according to specified stroke order, and then the robot went back to the first stroke again and wrote it repeatedly. Compared to the erasing task, this writing

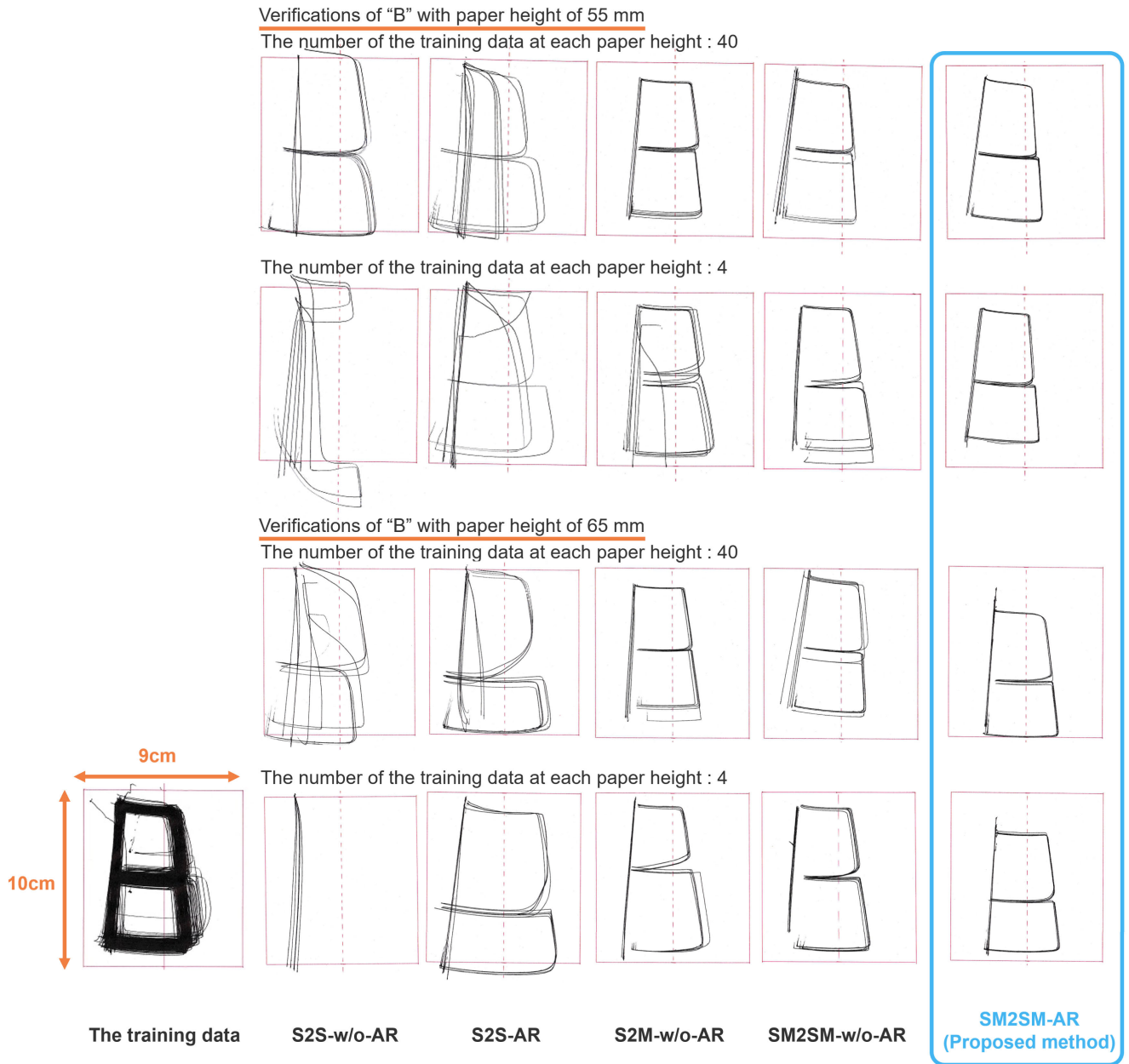


FIGURE 15. Training data and the results of the task of writing the letter "B"

task was a long-term operation and required a correct long-term prediction. To succeed in this task, reproducing the proper force between the paper and pen was necessary. In addition, the robot had to reproduce the stroke order learned from the human demonstrations. In other words, the ability to generate correct behavior based on past and current states was necessary.

2) HUMAN DEMONSTRATIONS AND DATASET FOR LEARNING

We collected data with paper heights of 35, 55, and 75 mm. The letters "A" and "B" were collected as separate trials. One 15-s trial included writing the same letter four times in

a row. In other words, the number of training data in one trial was four. Ten trials were conducted for each paper height. In total, 120 sets of training data were collected for each letter (40 sets of training data × three heights). The letters were written inside the solid red lines shown in Figs. 14 and 15. The letters were written by humans in demonstrations such that the shape would be roughly the same during all trials without using any restraining tools including a ruler.

3) NEURAL NETWORK ARCHITECTURE

The NN model consisted of six LSTM layers, followed by a fully connected layer. A unit size of 50 was used for all layers.

**TABLE 5. Success rates of the writing task.**

| Model                      | Success Rate [%] |                  |                  |
|----------------------------|------------------|------------------|------------------|
|                            | Letter "A"       | Letter "B"       | Total            |
| S2S-w/o-AR                 | 0 (0/4)          | 0 (0/4)          | 0 (0/8)          |
| S2S-AR                     | 0 (0/4)          | 0 (0/4)          | 0 (0/8)          |
| S2M-w/o-AR                 | 75.0 (3/4)       | 75.0 (3/4)       | 75.0 (6/8)       |
| SM2SM-w/o-AR               | 75.0 (3/4)       | <b>100 (4/4)</b> | 87.5 (7/8)       |
| SM2SM-AR (proposed method) | <b>100 (4/4)</b> | <b>100 (4/4)</b> | <b>100 (8/8)</b> |

The mini-batch consisted of 100 random sets of 200 time-sequential samples corresponding to 4 s.

#### 4) VALIDATION OF THE TASK

For autonomous operations, the performances for paper heights of 55 and 65 mm were verified. In addition, verification was conducted for the cases in which 4 and 40 sets of training data were used for each paper height. Success was defined as cases in which the robot wrote the letter five times continuously inside the solid red lines shown in Figs. 14 and 15 and with the correct stroke order. Verification was conducted for each height and each number of training data. Therefore, four verifications were performed (two heights  $\times$  two training datasets).

#### 5) EXPERIMENTAL RESULTS

The success rates of the different models are listed in Table 5. Only the proposed method was successful for all validations. The results of continuously writing the letter five times are shown in Figs. 14 and 15. With the conventional methods, the trajectory of the letters was unstable every time. By contrast, with the proposed method, the letters were written with the same trajectory every time. In particular, in the case in which the training data were few, the difference from the other methods was noticeable. Only the proposed method could generate a trajectory with little fluctuation. This result indicated that the proposed method generated motion with little fluctuation in the long-term thanks to autoregressive learning, *i.e.*, the model learned to minimize the total errors of the long-term prediction.

## VI. CONCLUSION

In this study, we proposed a method of autoregressive learning for a bilateral control-based IL. Due to the structure and autoregressive learning of the proposed model, the performance was improved compared to the conventional methods. During the experiments, three types of tasks were performed, and the proposed method had the highest success rate. In addition, the proposed method improved the generalization for unknown environments not included in the training dataset.

However, the proposed method must be improved in terms of model structure. In the proposed SM2SM model, the master state predicted by the model in the previous step was input to the model during an autonomous operation, and thus the master state used in the input could be regarded as a virtual master state. This is because the slave executes tasks alone during the autonomous operations. If sudden environmental changes occur during the execution of the task, this state

of the virtual master is likely to differ from the that of the actual master. For example, a sudden environmental change is a situation in which the paper height is changed during the operation of the writing task. This error affects the prediction of the model, and therefore this issue has to be solved in the future. We plan to further improve the performance of autoregressive learning for bilateral control-based IL by changing the structure of the SM2SM model or implementing systems to correct the error.

## REFERENCES

- [1] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [3] T. Zhang, Y. Zong, W. Zheng, C. L. P. Chen, X. Hong, C. Tang, Z. Cui, and G. Zhao, "Cross-database micro-expression recognition: A benchmark," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 6, 2020, doi: 10.1109/TKDE.2020.2985365.
- [4] W. Qi, H. Su, and A. Aliverti, "A smartphone-based adaptive recognition and real-time monitoring system for human activities," *IEEE Trans. Human-Machine Syst.*, vol. 50, no. 5, pp. 414–423, Oct. 2020.
- [5] D. Yu, C. L. P. Chen, and H. Xu, "Intelligent decision making and bionic movement control of self-organized swarm," *IEEE Trans. Ind. Electron.*, early access, Jun. 4, 2020, doi: 10.1109/TIE.2020.2998748.
- [6] H. Huang, T. Zhang, C. Yang, and C. L. P. Chen, "Motor learning and generalization using broad learning adaptive neural control," *IEEE Trans. Ind. Electron.*, vol. 67, no. 10, pp. 8608–8617, Oct. 2020.
- [7] H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno, and E. D. Momi, "Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results," *Neural Netw.*, vol. 131, pp. 291–299, 2020.
- [8] H. Su, W. Qi, Y. Hu, H. R. Karimi, G. Ferrigno, and E. De Momi, "An incremental learning framework for human-like redundancy optimization of anthropomorphic manipulators," *IEEE Trans. Ind. Informat.*, early access, Nov. 9, 2020, doi: 10.1109/TII.2020.3036693.
- [9] R. Rahmatizadeh, P. Abolghasemi, L. Bölöi, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 3758–3765.
- [10] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, 2017.
- [11] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 1–8.
- [12] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 5074–5082.
- [13] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 3, pp. 297–330, May 2020.
- [14] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for robotic manipulation," *Int. J. Intell. Robot. Appl.*, vol. 3, no. 4, pp. 362–369, 2019.
- [15] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine, "One-shot imitation from observing humans via domain-adaptive meta-learning," in *Proc. Robot., Sci. Syst. (RSS)*, 2018, pp. 1–12.
- [16] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. B. Amor, "A system for learning continuous human-robot interactions from human-human demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 2882–2889.
- [17] P. C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata, "Repeatable folding task by humanoid robot worker using deep learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 397–403, Nov. 2017.

- [18] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," 2019, *arXiv:1910.11956*. [Online]. Available: <http://arxiv.org/abs/1910.11956>
- [19] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Adv. Robot.*, vol. 25, no. 5, pp. 581–603, 2011.
- [20] L. Rozo, P. Jiménez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *Intell. Service Robot.*, vol. 6, no. 1, pp. 33–51, Jan. 2013.
- [21] L. Rozo, D. Bruno, S. Calinon, and D. G. Caldwell, "Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2015, pp. 1024–1030.
- [22] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 177–184.
- [23] H. Ochi, W. Wan, Y. Yang, N. Yamano, J. Pan, and K. Harada, "Deep learning scooping motion using bilateral teleoperations," in *Proc. 3rd Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Jul. 2018, pp. 118–123.
- [24] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell, "Upper-body kinesthetic teaching of a free-standing humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3970–3975.
- [25] M. Tykal, A. Montebelli, and V. Kyrci, "Incrementally assisted kinesthetic teaching for programming by demonstration," in *Proc. 11th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, Mar. 2016, pp. 205–212.
- [26] T. Adachi, K. Fujimoto, S. Sakaino, and T. Tsuji, "Imitation learning for object manipulation based on Position/Force information using bilateral control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3648–3653.
- [27] A. Sasagawa, K. Fujimoto, S. Sakaino, and T. Tsuji, "Imitation learning based on bilateral control for human-robot cooperation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6169–6176, Oct. 2020.
- [28] S. Sakaino, T. Sato, and K. Ohnishi, "Multi-DOF micro-macro bilateral controller using oblique coordinate control," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 446–454, Aug. 2011.
- [29] T. Kitamura, N. Mizukami, H. Mizoguchi, S. Sakaino, and T. Tsuji, "Bilateral control in the vertical direction using functional electrical stimulation," *IEEE J. Ind. Appl.*, vol. 5, no. 5, pp. 398–404, 2016.
- [30] Z. Chen, F. Huang, C. Yang, and B. Yao, "Adaptive fuzzy backstepping control for stable nonlinear bilateral teleoperation manipulators with enhanced transparency performance," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 746–756, Jan. 2020.
- [31] Z. Chen, F. Huang, W. Sun, J. Gu, and B. Yao, "RBF-neural-network-based adaptive robust control for nonlinear bilateral teleoperation manipulators with uncertainty and time delay," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 2, pp. 906–918, Apr. 2020.
- [32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [33] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.
- [34] A. Lamb, A. Goyal, Y. Zhang, S. Zhang, A. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4601–4609.
- [35] K. Kase, R. Nakajo, H. Mori, and T. Ogata, "Learning multiple sensorimotor units to complete compound tasks using an RNN with multiple attractors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4244–4249.
- [36] W. Iida and K. Ohnishi, "Reproducibility and operability in bilateral teleoperation," in *Proc. 8th IEEE Int. Workshop Adv. Motion Control (AMC)*, 2004, pp. 217–222.
- [37] K. Tanida, T. Okano, T. Murakami, and K. Ohnishi, "Control structure determination of bilateral system based on reproducibility and operability," *IEEE J. Ind. Appl.*, vol. 8, no. 5, pp. 767–778, 2019.
- [38] K. Ohnishi, M. Shibata, and T. Murakami, "Motion control for advanced mechatronics," *IEEE/ASME Trans. Mechatronics*, vol. 1, no. 1, pp. 56–67, Mar. 1996.
- [39] T. Murakami, F. Yu, and K. Ohnishi, "Torque sensorless control in multidegree-of-freedom manipulator," *IEEE Trans. Ind. Electron.*, vol. 40, no. 2, pp. 259–265, Apr. 1993.
- [40] T. Yamazaki, S. Sakaino, and T. Tsuji, "Estimation and kinetic modeling of human arm using wearable robot arm," *Electr. Eng. Jpn.*, vol. 199, no. 3, pp. 57–67, 2017.
- [41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–15.
- [42] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, 2015, pp. 1171–1179.
- [43] W. Zhang, Y. Feng, F. Meng, D. You, and Q. Liu, "Bridging the gap between training and inference for neural machine translation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, p. 4334.
- [44] Y. Xu, K. Zhang, H. Dong, Y. Sun, W. Zhao, and Z. Tu, "Rethinking exposure bias in language modeling," 2019, *arXiv:1910.11235*. [Online]. Available: <http://arxiv.org/abs/1910.11235>
- [45] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," 2015, *arXiv:1511.06732*. [Online]. Available: <http://arxiv.org/abs/1511.06732>



**AYUMU SASAGAWA** (Student Member, IEEE) received the B.E. degree in electrical and electronic systems engineering from Saitama University, Saitama, Japan, in 2019, where he is currently pursuing the M.E. degree with the Department of Electrical and Electronic Systems. His research interests include robotics, motion generation, and neural networks.



**SHO SAKAINO** (Member, IEEE) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Yokohama, Japan, in 2006, 2008, and 2011, respectively. He was an Assistant Professor with Saitama University, from 2011 to 2019. Since 2019, he has been an Associate Professor with the University of Tsukuba. His research interests include mechatronics, motion control, robotics, and haptics.

He received the IEEE Industry Application Society Distinguished Transaction Paper Award in 2011 and 2020. He also received the RSJ Advanced Robotics Excellent Paper Award in 2020.



**TOSHIAKI TSUJI** (Senior Member, IEEE) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Yokohama, Japan, in 2001, 2003, and 2006, respectively. He was a Research Associate with the Department of Mechanical Engineering, Tokyo University of Science, from 2006 to 2007. He is currently an Associate Professor with the Department of Electrical and Electronic Systems,

Saitama University, Saitama, Japan. His research interests include motion control, haptics, and rehabilitation robots. He received the FANUC FA and Robot Foundation Original Paper Award in 2007 and 2008, respectively. He also received the RSJ Advanced Robotics Excellent Paper Award and the IEEE Industry Application Society Distinguished Transaction Paper Award in 2020.

• • •