Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 837540, 15 pages
http://dx.doi.org/10.1155/2015/837540

Research Article

Tool-Body Assimilation Model Based on Body Babbling and Neurodynamical System

Kuniyuki Takahashi,[1] Tetsuya Ogata,[2] Hadi Tjandra,[1] Yuki Yamaguchi,[3] and Shigeki Sugano[1]

[1]*Graduate School of Creative Science and Engineering, Waseda University, Tokyo 1698555, Japan*
[2]*Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo 1698555, Japan*
[3]*Graduate School of Informatics, Kyoto University, Kyoto 6068501, Japan*

Correspondence should be addressed to Kuniyuki Takahashi; takahashi@sugano.mech.waseda.ac.jp

Received 7 March 2014; Revised 15 June 2014; Accepted 15 June 2014

Academic Editor: Yi Chen

We propose the new method of tool use with a tool-body assimilation model based on body babbling and a neurodynamical system for robots to use tools. Almost all existing studies for robots to use tools require predetermined motions and tool features; the motion patterns are limited and the robots cannot use novel tools. Other studies fully search for all available parameters for novel tools, but this leads to massive amounts of calculations. To solve these problems, we took the following approach: we used a humanoid robot model to generate random motions based on human body babbling. These rich motion experiences were used to train recurrent and deep neural networks for modeling a body image. Tool features were self-organized in parametric bias, modulating the body image according to the tool in use. Finally, we designed a neural network for the robot to generate motion only from the target image. Experiments were conducted with multiple tools for manipulating a cylindrical target object. The results show that the tool-body assimilation model is capable of motion generation.

1. Introduction

Humans are capable of expanding their ability by using tools. Robots are expected to become more useful to society through the use of tools. With the development of robotics technology, robots have become very complex, with increasing numbers of applicable sensors and degrees of freedom. Therefore, complicated calculations are required to build conventional robot tool use models. Modeling robot tool use based on human cognitive development has been proposed as an approach to mitigate this problem [1]. Among the many factors of human cognitive development, tool-body assimilation, studied in the field of neuropsychology, has begun to gather attention [2]. Tool-body assimilation occurs when humans use a tool and treat it as an expansion of their own bodies. Iriki et al. recorded neurons called "bimodal neurons" before and after monkeys were trained to use a tool. Bimodal neurons respond both to tactile stimulation on the hand and to visual stimulation. Through tool use training, the visual receptive field of bimodal neurons expands from the monkey's hand to the surroundings of the grasping tool. This result shows that tool-body assimilation occurs at the neuron level. We aim to achieve robot tool use by this approach; by modeling tool-body assimilation, it is possible to alter the behavior of a posttrained body model by adding new neurons and expressing a "body model that is using a tool."

Tool use with tool-body assimilation is also gathering attention in the field of robotics. Nabeshima et al. [3] used visual and touch stimuli to connect bodily and sensory information. After training the relationships between visual and touch stimuli, dynamic touch was performed to predict the inertial parameters of the tool. The resulting simulation model allowed the robot to perform a pulling task with a target object located in an invisible area. However, the inertial parameters used as tool features were determined in advance, and the model was incapable of adapting to nonrigid bodies. Hikita et al. [4] treated tools as an expansion of the robot's

body in the form of saliency maps, basing their research on the experimental results of Iriki et al. However, no actual motion was generated.

There are various studies regarding tool manipulation, which is limited to not only focusing on tool-body assimilation, but also affordance. Affordance is the value that the environment provides to animals [5]. For example, a solid horizontal plane at knee height affords sitting. In the field of robotics, affordance research is divided into two approaches. One is the use of human affordance knowledge, while the second is the acquisition of affordance by the robot itself. Montesano et al. [6], Saxena et al. [7], and Song et al. [8] took the former approach, while Stoytchev [9], Detry et al. [10], and Tikhanoff et al. [11] adopted the latter approach. Montesano et al. trained a Bayesian network by giving the network information about the relations between actions and the resulting effects. Saxena et al. taught a robot where to grasp during tool use and generalized the model by categorizing unknown tools with similar shapes, allowing the grasping of unknown tools. Song et al. succeeded in training a model that grasps differently for different objectives. For example, this allows the robot to hold a cup from its side instead of grasping the top; open side of the cup when drinking is set as the objective. However, because it is very difficult for humans to directly teach a robot about the affordance of the environment, it is unrealistic to use this approach to teach about various numbers of tools. In addition, depending on the robot's bodily structure, it may be impossible to execute actions based on the trained affordance. In the study by Stoytchev, a robot learned the relationships between movements, tool shapes, and target objects by generating motions based on predetermined motion sets. Each tool was recognized by its respective colors, and the target object was moved to the target position by referring to the learned relationships. However, although the robot successfully learned the affordance of the tools, the generated motions were limited to the predetermined motions used during training. Detry et al. succeeded in achieving almost all possible graspable parts of various tools by testing almost all graspable positions. However, this is an unrealistic approach because large amounts of time will be required when this is done under real conditions, and the resulting model will not be able to adapt to unknown tools. Tikhanoff et al. performed object pulling tasks and taught pulling motions to the robot. In this case, the robot decided what tool and which part of the tool to use, based on the position of the target object. Because humans define the decision algorithm by using mathematical equations, appropriate modeling is required for new tasks.

The concerns described above are summarized as follows:

(1) Limited motion owing to predetermined motion patterns,

(2) Inability to adapt to new tools owing to predetermined tool and motion features,

(3) Large calculations required in generating target motions owing to the need to fully search all available parameters.

To overcome these concerns, this research will undertake the following approach with tool-body assimilation:

(1) body babbling with a humanoid model,

(2) recurrent neural network and deep neural networks,

(3) Motion generation by minimizing the error of the final state predicted from initial states.

Our purpose is to suggest the new method for tool use by robots by applying the concept of tool-body assimilation in cognitive science with body babbling and a neurodynamical system rather than modeling human's cognitive mechanism.

First we will describe the first step of the approach. Many past studies have built motion models in advance for motion generation and used teaching data that are designed by humans to train robots. This approach can adapt to conditions and tool functions that are known. However, the approach cannot adapt to unknown conditions. In addition, the creation of the teaching data is strongly influenced by the creator's knowledge and intentions, causing bias in the data. Because of this, it is difficult to effectively make use of the bodily constraints of the robot. It is hypothesized that the body has a significant influence on the development of intelligence [12]. During the early days of infants, the relationship between motion and the nervous system has yet to be modeled; thus, movements and interactions between the body and the environment cannot be reproduced. The infant learns the reproducible motions and the accompanying sensory information from this interaction. As the infant learns, passive motions are replaced with autonomous motions, and as the infant interacts with the environment, relations between primitive movements and sensations are learned. By learning these relationships, various information is gained, such as the postures of the body that are easy to move. Even in the case of tool use, knowledge related to the tool is earned and generalized through trial and error [13]. In the field of robotics, a phenomenon called body babbling, which is one of the steps in the infant development process, has recently began to gather attention [14–17]. Humans are capable of unlimited numbers of movement patterns due to having bodies with many degrees of freedom. However, the actual movement patterns that are used are limited owing to the constraints of the body structure, causing some degrees of freedom to be easier to move than others. In the case of tool use, there are unlimited ways to use tools. However, the actual use of a tool is limited to very few methods because tool use is also influenced by bodily constraints. Thus, it can be said that the way humans use tools has a strong relationship with the body [18]. When considering a model with human-like superior tool use capabilities, by using a body model that is similar to a human body, it is expected that the model will generate a human-like motion. In addition, considering human cognitive development, many trials of movements must be performed, and a body that is able to withstand these trials is needed. If this is done with real robots, the robot will break down. Therefore, the body babbling will be performed by a robot in a simulator.

We will now describe the second step of the approach. Past efforts in robotics have required the model of robot

movement, object manipulation, and features to be designed in advance. This method is highly effective in fixed environments such as factory production management. However, when considering robots that work in changing environments such as human living spaces, it is unrealistic to assume that the environment does not change. It is possible to partially overcome this problem by predicting the changes in the environment. However, the model would still not be able to adapt to unpredictable environmental changes. To overcome this issue, we took an approach in which the robot autonomously acquires features of data and learns the relationships between input and output from the sensory data that do not require preset features. With this approach, the model can adapt to new situations. Specifically, in this research, tool and motion features in the form of image features from camera data are self-organized by using deep neural networks (DNNs) with an autoencoder. The autoencoder can generate feature values automatically. This is done by training the autoencoder to give output values that are equal to the input values. The relations between the robot motion and the earned features are used to train a multiple time-scale recurrent neural network (MTRNN). With these two methods, it is possible to build a model that allows the robot to adapt to environmental changes and its own movements without predetermined information. In addition, as a characteristic of neural networks, these can estimate the output even when nontraining data are given based on the experiences that the neural networks have gained from training data.

We will now describe the third step of the approach. In the early days of infants, they tend to predict tool functions by using dynamic touch [5]. Dynamic touch refers to the movement of the body to acquire the characteristics of an object by moving the object [13]. Through tool manipulation experiences, humans tend to learn to use tool with tool-body assimilation. Nishide et al. [19] used a neural network model to build a tool-body assimilation model with dynamic touch. Because the tool features were self-organized with no settings determined in advance, the model is also applicable to untrained tools. The model predicted the features of the new tool, updated the parametric bias (PB) nodes connected to the neural network, and performed object pulling tasks. In their study, dynamic touch was required to differentiate between different tools. Michaels et al.'s experimental results imply that human estimates the tool function from the shape of tool as an overlay of tool use experiences [20]. If robots can also estimate tool function from the shape of tool, it should be possible to estimate tool function from the shape of unknown tools. Unlike Nishide et al.'s approach, our approach is to perform the estimation of the tool function from the image of the tool. Moreover, if the robot estimates the tool function from target state and generates motion with final states close to the target state, it will be useful because there is no necessity to teach the process of motion. In Nishide et al.'s study it is necessary to search motion parameters to generate target motion before motion generation. This results in large calculations. We further developed this model, allowing the model to predict tool functions from tool shapes, and designed the model to generate motion by minimizing errors between the final goal state and the final state predicted by the MTRNN.

The rest of this paper is composed as follows. In Section 2, the DNNs are discussed. In Section 3, an overview of the tool-body assimilation model is provided. In Section 4, the experimental setup is presented. In Section 5, the experimental results are given. In Section 6, we present our conclusions and describe future studies.

2. Deep Neural Networks

DNNs are multilayer neural networks, usually with five to 10 layers. DNNs allow highly precise image and speech recognition, so they have attracted attention in recent years [21, 22]. It is possible to reduce the dimensionality of data and to automatically extract features without predetermined information by using an autoencoder with a sand-glass-type DNN. It is possible to extract features from a hidden layer by substituting the unknown data in a trained DNN. DNNs are used in field of image recognition and speech recognition; however, they are not often used in the field of robotics [23, 24]. This is because for training DNNs require enormous, multidimensional training data. It is time consuming to acquire training data with robots, causing robots to break owing to use for a long time. To overcome these problems, we used a robot model in a simulator to perform random movements in the form of body babbling. This will be described in detail in Section 3, describing the tool-body assimilation model.

2.1. Training of Deep Neural Networks. A recent advance in learning for deep networks is to use layer-wise unsupervised pretraining methods [25]. Applying these methods before running fine-tuning methods overcome the difficulties related to deep learning. Martens [26] proposed the approach that developed a 2nd-order optimization method based on the Hessian free approach without using pretraining. This method is easy to use, effective, and efficient in training. Therefore, we adapted the learning method proposed by Martens.

2.2. Hessian-Free Optimization. Newton's method is the canonical 2nd-order optimization. The method is iteratively updates of the parameters $\theta \in \mathbb{R}^N$ of an objective function f by computing search vectors p and updating θ as $\theta + \alpha p$ for some α. The main idea of Newton's method is that f can be locally approximated around each θ, up to the 2nd-order, calculated as

$$f(\theta + p) \approx q_\theta(p) \equiv f(\theta) + \nabla f(\theta)^\top p + \frac{1}{2} p^\top B p, \quad (1)$$

where $B = H(\theta)$ is the Hessian matrix of f at θ. However, H can be indefinite so that (1) may not have a minimum. Moreover, if the values of p are large, the approximation of f becomes inaccurate. Therefore, H uses damping parameter λ for calculation to readjust $B = H + \lambda I$ for some constant $\lambda \geq 0$.

In standard Newton's method, $q_\theta(p)$ is optimized by computing the $N \times N$ matrix B and then solving the system

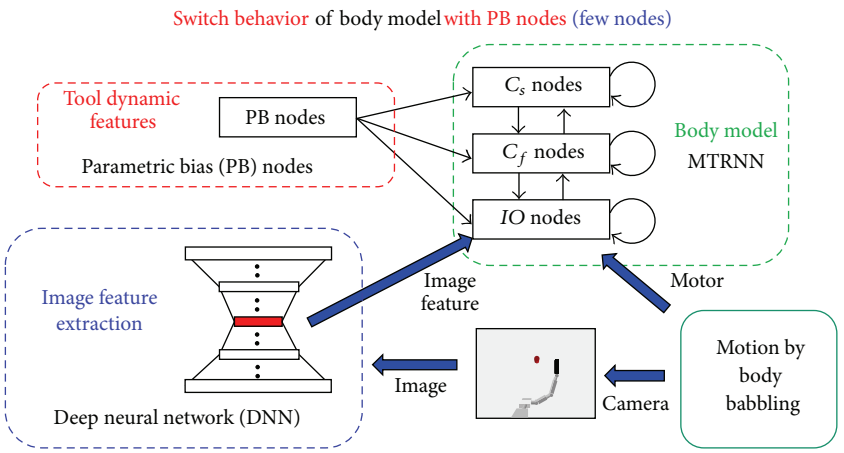


Figure 1: Overview of tool manipulation model.

$Bp = \nabla(\theta)^{\top}$. However, the calculation costs are very expensive when N is large even with modestly sized neural networks. Instead of this calculation, Martens develops the basis of the 2nd-order optimization approach with the linear conjugate gradient algorithm (CG) [26]. This optimizing quadratic objectives require only matrix vector products with B. Details about the implementation are provided in several other studies [26–28].

2.3. Competition with Self-Organizing Map. Some studies have used a self-organizing map (SOM) for the extraction of features [19, 29]. Arie et al. proved that SOM is the high compatibility with MTRNN. SOM is an unsupervised learning neural network proposed by Kohonen [30]. It is composed of input and output neurons. The neurons in the output layers are two-dimensionally arranged and possess weight vectors, w. The weight vectors are set to have the same dimensions (I) as the input vector, v. The image features are defined by the following formula:

$$p_i = \frac{\exp\left\{-\left\|w_i - v\right\|^2/\sigma\right\}}{\sum_{j\in N}\exp\left\{-\left\|w_j - v\right\|^2/\sigma\right\}}, \quad (2)$$

where N is the dimension of the SOM and $i \in I$.

It is possible to reduce the dimensionality of data by using an SOM. However, if there are many motion patterns for the robot, it is difficult to extract features using an SOM. Therefore, we used an autoencoder with the DNN for the extraction of features. This will be discussed in detail in Section 5, which describes the experimental results.

3. Tool-Body Assimilation Model

In this section, we provide an overview of the tool-body assimilation process. This process consists of three phases: (1) learning of the body model with random movement based on body babbling by a humanoid robot, (2) learning of tool dynamics features, and (3) generation of motion by using only a target image. Figure 1 shows an overview of the model. The model consists of three modules:

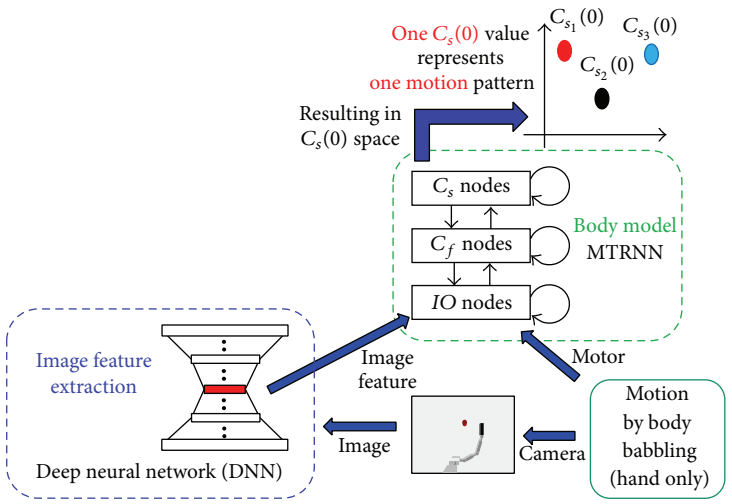
(i) body model module: MTRNN,

(ii) image feature extraction module: DNN,

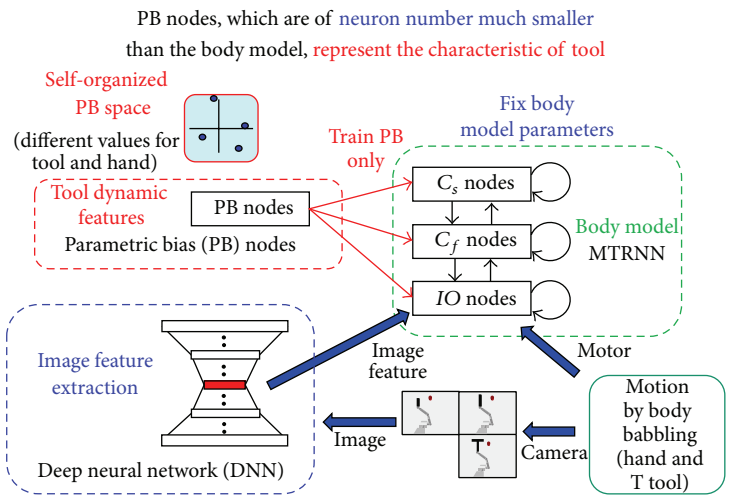
(iii) tool dynamic feature module: PB-nodes.

Figure 2 shows the learning process of tool-body assimilation. First, body babbling is performed with a bare hand, producing motor data and camera images. The image features are extracted with the DNN. Next, the MTRNN is trained by using these motor data and image features. Upon training, the MTRNN learns the body model of the robot. Next, by training only the PB nodes with different tool images, it is possible for the robot to learn the tool features without changing the body model. PB neurons are capable of learning how to modulate the body depending on the tool being used. This combination of MTRNN and PB nodes thus expresses "body model that uses tool." The number of PB nodes is much fewer than that needed for the body model (in this study, the number of PB nodes is five and body model requires 98 nodes), and the PB nodes represent the tool characteristics. Because the tool is treated as a part of the body, we expect the robot to use tools based on the experience of how to move the body. PB nodes do not represent the complicated methods of tool use itself but only show how to modulate the original body model. Therefore, even if the number of tools is increased, the number of neurons of PB nodes will still be fewer than in the body model. The time to learn the PB nodes is shorter than that needed for the body model. However, we suspect that the proposed model cannot learn all possible types of tools. This is mainly because of the difficulties in representing tools that cannot be treated as a modulation of the body. For example, tools with rotating motions such as a mixer will be a challenging tool to use.

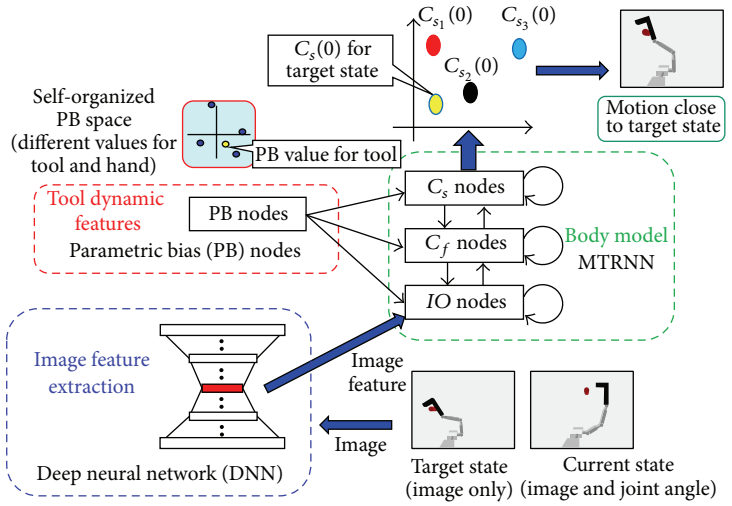
3.1. Learning of Body Model Based on Body Babbling by a Humanoid Robot. In this phase, the robot, possessing a humanoid robot model, performs body babbling with a target object in its bare hand to learn its own body model. Figure 2(a) shows the learning process of body model. Firstly, the robot performs body babbling using bare hand, and motor data and camera images are produced. Then image



(a) Learning process of body model



(b) Learning process of tool dynamics feature



(c) Process of motion generation

Figure 2: Learning process of tool-body assimilation.

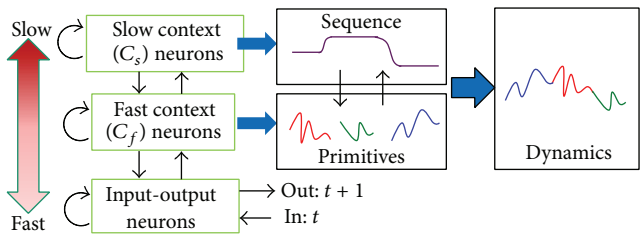


Figure 3: Dynamics representation of MTRNN.

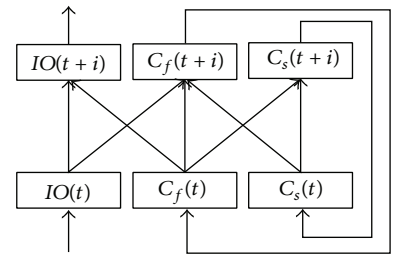


Figure 4: Composition of MTRNN.

features are extracted by the DNN. The relationship between motor and image features is learned by the MTRNN; that is, the robot learns its own body model.

Body babbling is a behavior observed in infants. Body babbling is considered as an explorative motion. Such motions that are related to human's intrinsic aspects such as motivation and preference are phenomena that are difficult to model. Hence, we modeled this as random motions in a similar way to the previous cognitive robotics studies [14–17]. By using the concept of body babbling, predetermined parameters are not required for performing motion by a robot. Body babbling requires numerous numbers of movements. Simulated experiments are effective, because it is difficult to perform many trials with real robots. During body babbling, sequential images and motor data are acquired. The features are extracted from the images by an image feature extraction function.

To adapt to unknown situations, the robot should have the ability to extract image features by itself. To achieve this, we propose the use of an autoencoder with the DNN extraction of image feature. We describe DNN in the next section. In this research, the input data to the DNN consists of the raw image pixels from the robot model's camera.

For the robot's body model, we implemented the MTRNN proposed by Yamashita and Tani [31]. The MTRNN is a kind of recurrent neural network (RNN) [32], which can predict the next state, IO($t + 1$), given the current state, IO(t). This MTRNN is capable of learning multiple sequential data. The MTRNN is composed of three types of neurons: fast context (C_f) nodes, slow context (C_s) nodes, and input-output (IO) nodes. Each type of node has a different time constant, representing the firing rate of the nodes. The faster (C_f) nodes learn the movement primitives of the data, whereas the slower (C_s) nodes learn the sequence of the data (Figure 3). The structure of the MTRNN proposed in this research is shown in Figure 4. During the learning of the MTRNN, the back propagation through time (BPTT) algorithm [33] is applied.

First, the output neurons are calculated by forward calculation. The internal value of the ith neuron at step t, $u_i(t)$, is calculated as

$$u_i(t) = \left(1 - \frac{1}{\tau_i}\right) u_i(t-1) + \frac{1}{\tau_i}\left[\sum_{j \in N} \omega_{ij} x_j(t-1)\right], \quad (3)$$

where τ_i is the time constant of the ith neuron, $x_j(t)$ is the input value of the ith neuron from the jth neuron, ω_{ij} is the weight value of the ith neuron from the jth neuron, and N is the number of neurons connecting to the ith neuron. The output of the $y_i(t)$ is calculated by multiplying the internal value and the sigmoid function:

$$y_i(t) = \text{sigmoid}\left(u_i(t)\right) = \frac{1}{1 + \exp\left(-u_i(t)\right)}. \quad (4)$$

The input value is calculated as

$$x_i(t) = \begin{cases} \alpha \times y_i(t-1) + (1-\alpha) \times T_i(t) & i \in \text{IO} \\ y_i(t-1) & \text{otherwise,} \end{cases} \quad (5)$$

where $T_i(t)$ is the teacher signal and α is the feedback rate ($0 \le r \le 1$). The input value is calculated by adding the output value of the previous step to the teacher signal. This is done to avoid a diverging error during training. The inputs of the context layer are the outputs of the previous step.

The weight value is updated using the training error. The training error is calculated as

$$E = \sum_i \sum_{i \in \text{IO}} \left(y_i(t-1) - T_i(t)\right)^2. \quad (6)$$

The weight from the ith neurons to the jth neurons is updated with the training error $\partial E / \partial \omega_{ij}$:

$$w_{ij}(n+1) = w_{ij} - \alpha \frac{\partial E}{\partial \omega_{ij}}, \quad (7)$$

where α is the training coefficient and n is the number of updates.

The initial value of C_s, $C_s(0)$, is also calculated by the BPTT algorithm:

$$Cs_i(n+1) = Cs_i(0) - \alpha \frac{\partial E}{\partial Cs_i(0)}. \quad (8)$$

After training, $Cs(0)$ represents the parameters of each learned sequence. Each sequence can be recovered by substituting the $Cs(0)$ values into the MTRNN.

By applying the BPTT algorithm to $C_s(0)$ with the fixed weights of MTRNN, it can be used as a recognition unit.

3.2. Learning of Tool Dynamics Features. In addition to the C_f, C_s, and IO nodes, PB nodes are connected to the MTRNN. The time constant of PB nodes is set to infinity; therefore, the values of the PB nodes do not change during

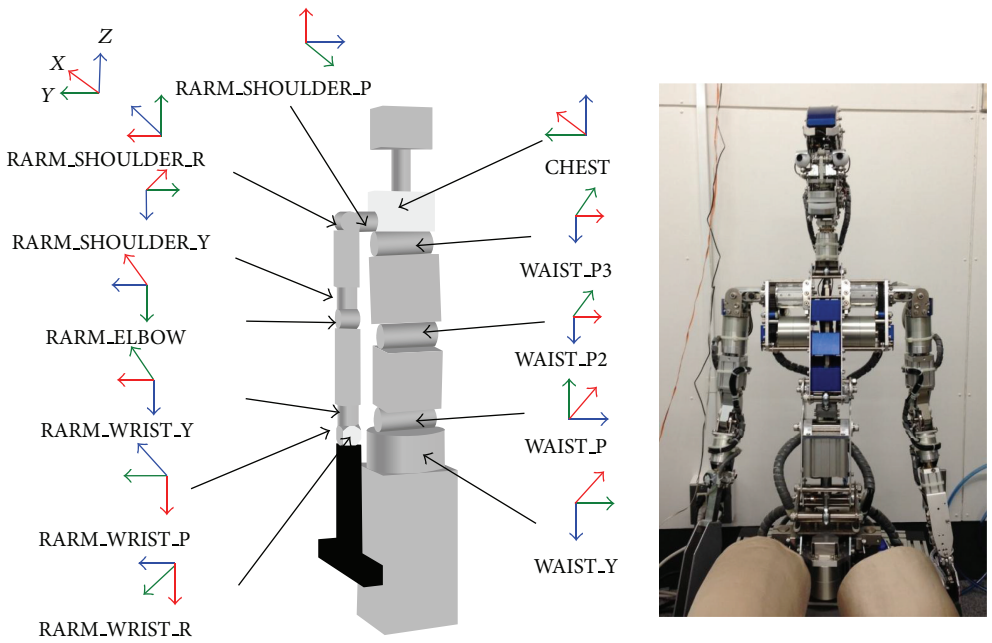


Figure 5: Robot based on ACTROID with T-shaped tool.

each sequence. During training, the weight of the MTRNN and $C_s(0)$ are fixed, and only the weights and values of the PB nodes are trained (Figure 2(b)). After training, the PB space that represents the tool dynamic features is formed. In other words, PB nodes are able to learn the visual changes resulting from differences in tool types. Then, the PB node applies bias to the body model according to the tool dynamic features that it learns, changing the body model's dynamics accordingly. This means that there is no need to retrain the robot's body model when a new tool is introduced. The value of the PB node is calculated by using the same method as for $C_s(0)$, and the weights from the PB nodes to other nodes are updated in the same manner as for other weights of the MTRNN.

3.3. Generation of Motion from Goal Image. In this phase, firstly, the initial image and joint angle are provided to the robot. With this the robot will be able to understand the environment's and the robot's own current state. Secondly, a target image is shown to the robot. During this time, the weights of the MTRNN and PB nodes are fixed. The $C_s(0)$ and PB values are calculated using the BPTT algorithm. Next, the differences between the target image and the associated image from the MTRNN are minimized. Using the PB and $C_s(0)$ values calculated with the algorithm, the MTRNN generates motor sequence data (Figure 2(c)). In Nishide et al.'s research, it was necessary to apply dynamic touch for tool type recognition. However, in this research, it is possible to recognize a tool from an image of the grasped tool.

4. Experimental Setup

4.1. Robot Model in Simulation. To evaluate the tool-body assimilation model, we built a humanoid robot model in the robotics simulator OpenHRP3 [34]. The model's size and

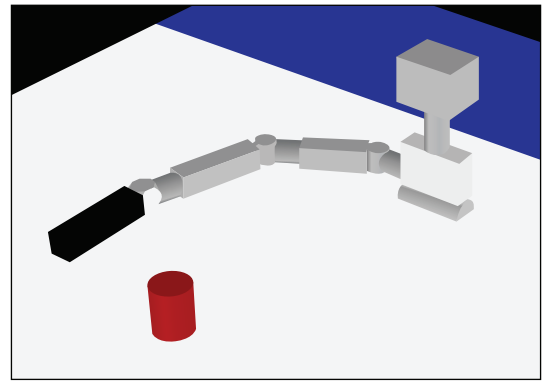


Figure 6: Experiment setting.

degrees of freedom (DOFs) were based on the humanoid robot ACTROID [35]. The range of motion of the model's joint angles was based on human [36]. To reduce calculation time, only the right arm, torso, and head of the robot were used. In addition, the left arm was removed and the legs were replaced with a box (Figure 5, Table 1).

4.2. Experimental Evaluation. In this experiment, an object pulling task with the robot's bare hand, an I-shaped tool, a T-shaped tool, an L-shaped tool, a J-shaped tool, "⊢" shaped tool, a C-shaped tool was used to evaluate the model (Figures 6 and 7). The size of the object was 0.08 [m] in diameter. The L-shaped tool was treated as an unknown tool, which is highly similar to the learned tools, and a J-shaped, "⊢" shaped, and C-shaped tool, which have high dissimilarities with the learned tools, are only used for evaluation and not for training. This task is commonly used in the study of robotic

Table 1: DOF and link length of the robot.

Link name (Arm)	a [mm]	α [deg.]	d [mm]	θ [deg.]	$q^{\max *}$ [deg.]	$q^{\min *}$ [deg.]
RARM_SHOULDER_P	0	90	−171	90	—	—
RARM_SHOULDER_R	0	90	0	90	120	−50
RARM_SHOULDER_Y	0	90	273	90	—	—
RARM_ELEBOW	−9	90	0	0	145	0
RARM_WRIST_Y	0	−90	240	90	—	—
RARM_WRIST_P	0	−90	0	−90	15	−55
RARM_WRIST_R	0	−90	0	0	—	—
(Tool)	Tool size	0	0	0	—	—

*Range of motion of the joint angle.

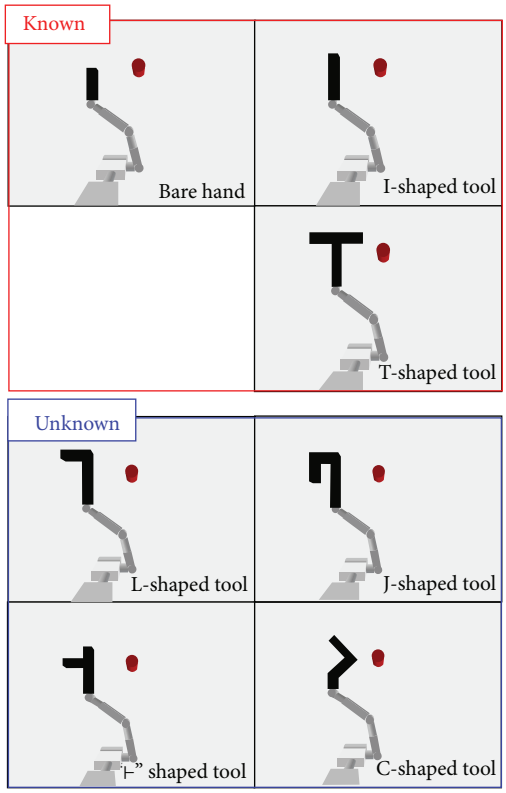


Figure 7: Tools used in experiment.

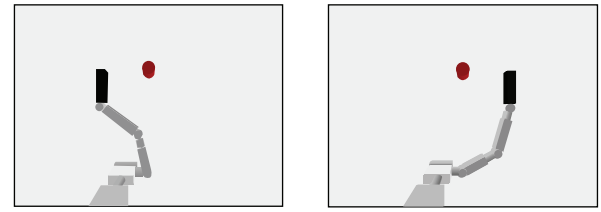


Figure 8: Hand posture used in motion pattern.

Table 2: Construction of DNN.

Dimensions of input-output nodes	768
Number of hidden layers	9
Dimensions of hidden nodes	500-250-100-50-15-50-100-250-500
Number of teaching data	13500

tool use and tool-body assimilation [3, 4, 9, 11, 19]. The robot performed body babbling in the presence of a target object (a cylinder) on a table for 6 [s] using its hand and a tool.

4.3. Motion by Body Babbling. To evaluate the effectiveness of this approach, the movement of the robot was confined to the plane of the desk (two-dimensional movements). In doing this, out of the seven DOFs of the robot's arm, only three DOFs were used. The robot's arm had two initial positions: to the left and to the right of the target object (Figure 8). For each initial position, the robot executed 75 sets of body babbling. The motions were generated by connecting the initial pose, the second pose, and the third pose. The second pose and third pose are selected randomly. The poses are connected by calculating the required acceleration with fifth-order linear interpolation. The accelerations of the beginning and end of the movement are calculated to be 0. Therefore, the movements become smooth and it is possible to control the motions according to the target motions. Although two initial positions are used for the teaching data in this study, parts of the motion paths in the training data are coded during the training of the RNN. This is synonymous to the learning of the varieties of trajectories. During body babbling, the robot obtained the teaching data that were used during training (Figure 9). The acquired data consisted of motor and image sequential data. The motor data of the three movable DOFs were recorded for 30 steps during the 6 [s] of random motion, that is, 7.5 [steps/s]. Image data constituted a gray-scale image of 32 × 24 pixels captured by a visual sensor on the robot. Twenty-five dimensions of the image features extracted from the image data by using an autoencoder with DNN, and three dimensions of the joint angles were used for the input data to train the MTRNN. The image data and joint angle were then normalized to [0.1, 0.9] and [0.0, 1.0] for use as the data for teaching the MTRNN. Table 2 shows the construction of the DNN. Table 3 shows the construction of the MTRNN.

Table 3: Construction of MTRNN.

Node name	Number of nodes	Time constant
Motor input nodes	3	2
Image feature input nodes	15	2
Fast context nodes	60	5
Slow context nodes	20	70
PB nodes	5	∞

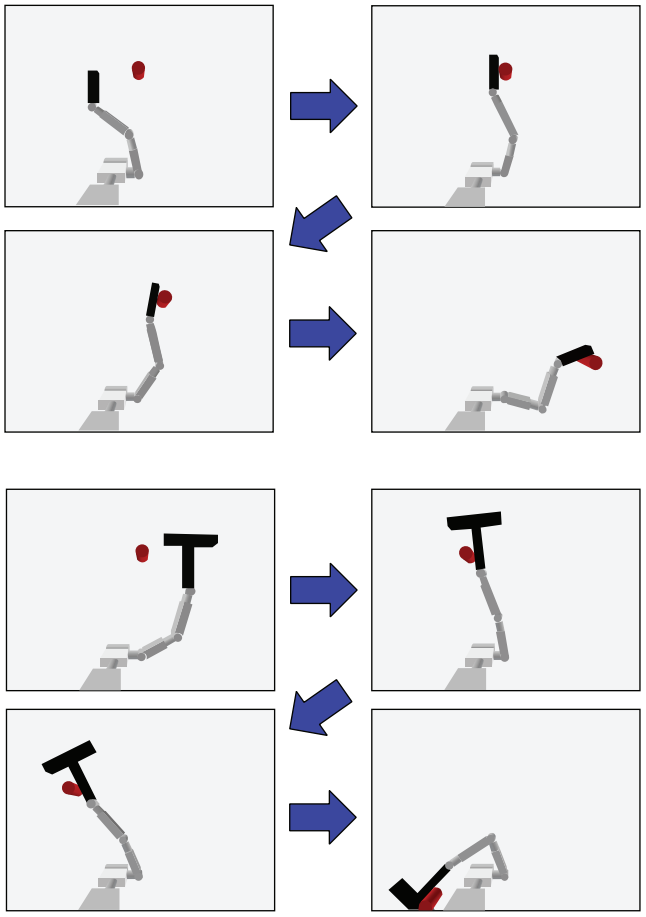


Figure 9: Body babbling with hand and T-shaped tool.

5. Experimental Results and Discussion

5.1. Extraction of Image Features by Self-Organizing Map. In previous studies, SOM has commonly been used as an image feature extraction method [19, 29, 37, 38]. To compare image features extraction by DNN, image features were also extracted by SOM. Figure 10 shows a visualization of the reference vectors of the SOM. Reference vectors are the visualization of image features. Reference vectors represent the patterns that are extracted from the input data. The characteristics of reference vectors are that the units that are mapped close to each other will have close resemblances to each other. In addition, each input data is classified to the locations of the reference vectors that are similar to the data. The dimensions of this SOM were 5 × 5. The results show

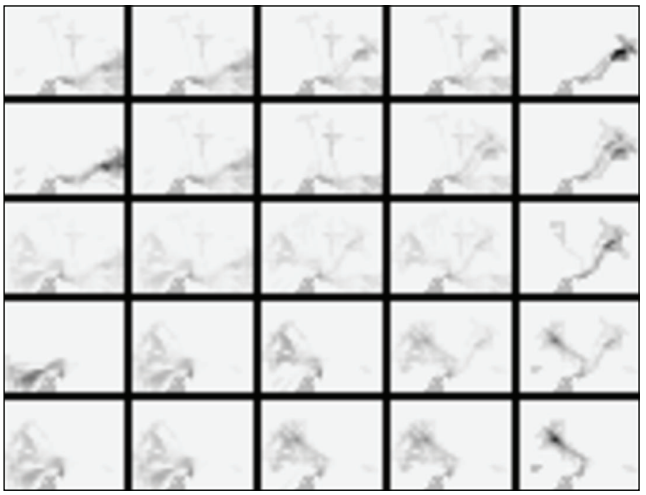


Figure 10: Reference vector of SOM.

Figure 11: Reference vector of SOM (bare hand and T-shaped tool only).

that the difference between the bare hand and tools is not learned accurately and that the motion patterns are not learned accurately. We changed the dimensions of SOM to 10 × 10. However, the motion patterns were not learned accurately after increasing the SOM dimension. Even if feature extraction is done well by increasing the dimension, it is difficult to learn by RNN because of the greater dimension. When more tools are introduced, the various tool conditions were included in each vector, causing the tool feature classification to fail. When only a few tools are used, it is possible to learn the image features accurately with SOM. This is shown in Figure 11, where image features of the bare hand and T-shaped tool were extracted by SOM.

5.2. Extraction of Image Features by DNN. Original images were recovered by substituting the image features extracted by DNN (Figure 12). The image of the bare hand and tools were accurately recovered. In addition, the position of the target object was recovered. Moreover, even with unknown

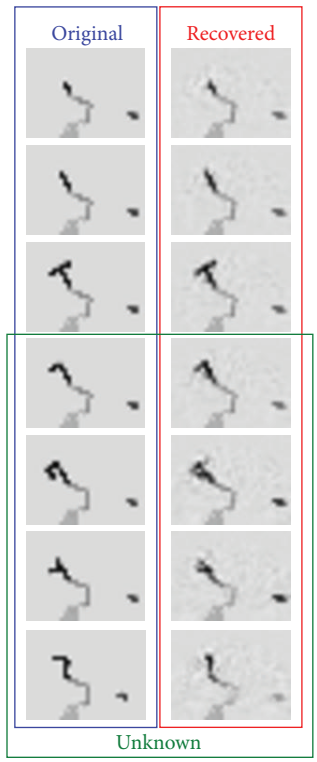


Figure 12: Original images recovered from DNN.

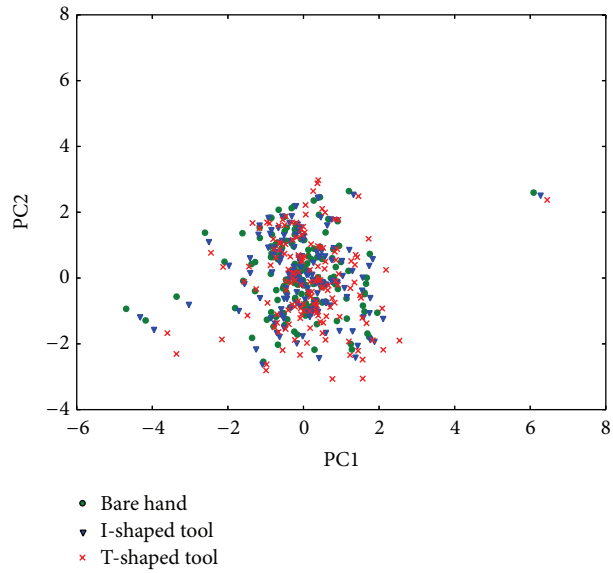


Figure 13: PCA PB space trained from the target image.

tools it was possible to recover the shapes of the unknown tools.

In the case of the SOM, extracted features of the target object were unclear; therefore, it was difficult to recover the position of the object accurately. DNN does not use classifications but instead makes use of autoencoders that are trained to produce the same output as the input data. With this, it is possible to reduce the dimensionality of large numbers of high-dimensional data followed by high reproduction performance. In addition, there is no need to fine tune the parameter settings with DNN. Thus, if there are large amounts of training data, the DNN is superior to the SOM in the extraction of image features.

5.3. Self-Organized Tool Function from PB Values. The principal component analysis (PCA) results of the PB values for the tool-body assimilation model are shown in Figure 13. The figure shows that self-organization failed for the features of each tool. Next, after body babbling was performed, as an additional condition we chose the motion patterns in which the bare hand and tools of the robot contacted the target object when moving between the two positions. PCA results of the PB values for the tool-body assimilation model are shown in Figure 14. The PB values are clustered based on the tool used during motion generation, that is, bare hand, I-shaped tool, and T-shaped tool. By these different conditions, we can see that the robot often moves without touching the target object when the additional conditions are not implemented. To generate motions there is no absolute need to gain tool functions. However, in this case the robot's adaptability to novel tools will decrease. To gain tool functions, we believe that it is important to actually have a sense of purpose to use the tool. The learning result implies that it is difficult to acquire the functions of the tool to just move the arm without purpose. We hypothesize that it is necessary to move with a purpose for acquiring tool functions. This idea is also described in a previous study [39]. When infants manipulate a grasped tool, it seems to be a random movement; however, we suspect that the intended movements are not performed correctly because their sensorimotor is not precisely formed. This problem will be solved by implementing explorative movements, and not random ones, for efficient learning.

Parts of the clusters are overlapping in Figure 14. In these overlapping regions, the robot manipulated the target object with part of its hand or a common part of tool (e.g., part of the "I" is common between the T-shaped and I-shaped tools) when the robot performed body babbling with a tool.

Figure 15 shows the clustering of the PB values after training of the model. As shown, each cluster is formed at a different area of the graph. Some parts of the clusters are overlapping. This is because when the robot generates the motion close to the target state by using parts of the tool that are similar to other tools, it distinguishes these as having the same tool function and chooses similar PB values. When the robot uses a different tool function, it chooses different PB values. PB values earned through recognition have large overlapping areas compared to the PB values earned during training. This is because the robot can generate the motion that uses the same tool function even if tool shape is different. It is considered that the robot generates motion with final states close to the target state which the robot has many experience with. The plots of PB values of unknown tools (L-shaped tool, J-shaped tool, "⊢" shaped tool, and C-shaped tool) overlap with each other and are hard to observe. This is because PB reflects not only the effect of tool shapes but also the tool functions used during each motion. For clarity's sake, we

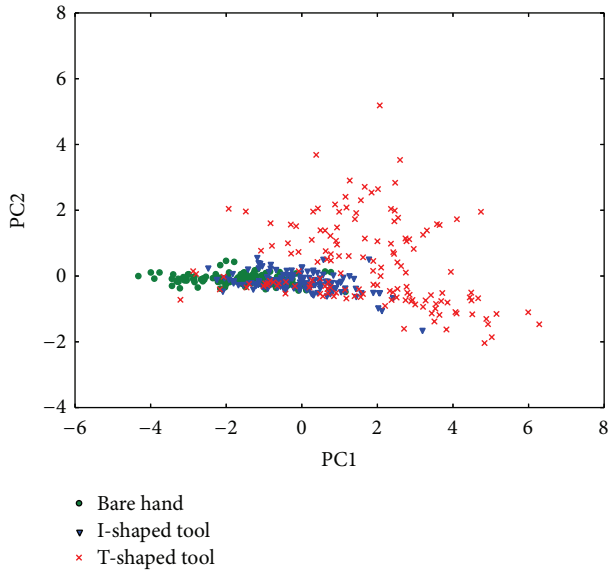


Figure 14: PCA PB space trained from the target image with additional conditions.

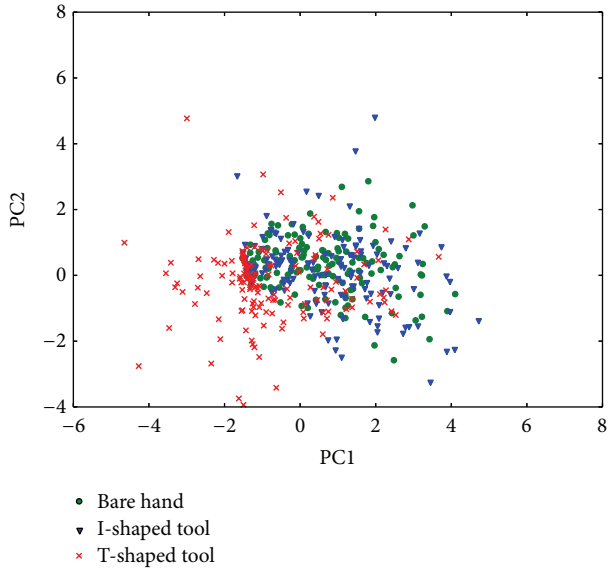


Figure 15: PCA PB space recognized (trained tools).

plotted the center of gravity of the PB of each tool (Figure 16). Variances of the PB values of the bare hand, I-, T-, L-, J-, "⊢", and C-shaped tool are 1.41, 3.17, 4.00, 3.87, 5.43, 6.89, and 3.32, respectively. Figure 16 shows that the PB values of the bare hand and I-shaped tools are similar. This is because the similar shapes of the bare hand and I-shaped tools lead to almost equivalent tool functions. With this observation, it can be said that similar shapes lead to similar usable tool functions during motions. It is observed that L-shaped tool has the intermediate tool functions of I-shaped and T-shaped tool. This is because L-shaped tool is similar to I-shaped and T-shaped tools. In the case of the J-shaped and C-shaped tools, the PC1 values are similar to the PC1 values of the

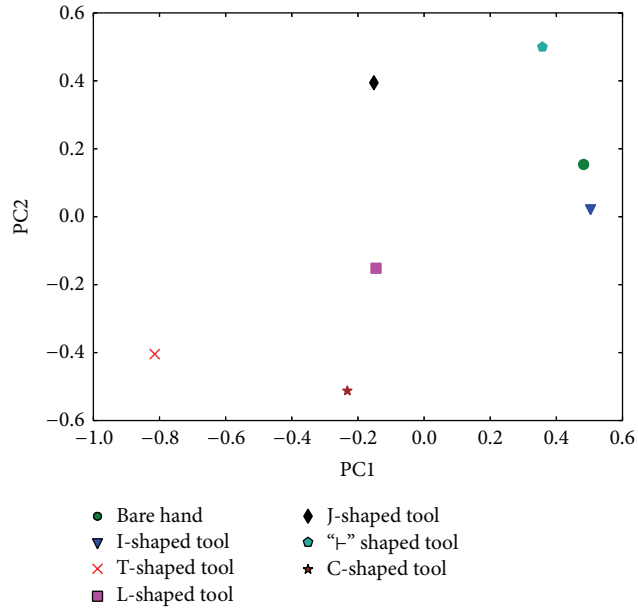


Figure 16: PCA PB space recognized (all tools).

L-shaped tool. Here, the difference of PC2 values of the L-shaped and C-shaped tools is smaller than the difference of PC2 values of the L-shaped and J-shaped. Thus, it can be said that the function of C-shaped tool is more similar to L-shaped tool than J-shaped tool. It is observed that the PB values of the "⊢" shaped tool are differ from other tools except the PC1 values of bare hand and I-shaped tool. Thus, it can be said that "⊢" shaped tool has part of the functions of bare hand and I-shaped tool and different functions when comparing other tools. With these observations, it can be said that the robot recognizes that the tools each have different tool functions.

5.4. Generated Motion. We evaluated the performance of our system by counting the number of the tasks which successfully moved the object to a position within the radius of R pixels from the goal position in the visible area (32×24 pixels). Figure 17 shows the relationships between the success rate and R. The success rate within $R = 2$ was about 20 to 35 percent; however within $R = 5$ it was more than 50 percent even if the robot used the untrained L-shaped tool, J-shaped tool, "⊢" shaped tool, and C-shaped tool. As one of the ways to improve the success rate, it may be better to learn gradually from coarse to fine images as shown by the experimental results of Kawai et al. [40].

Examples of motion generated by the robot when given a target image are shown in Figures 18, 19, 20, and 21. As shown, motions close to the target state were generated even if the robot uses an untrained tool. In Figure 17, it can be observed that it is possible to manipulate the object with the same accuracy as the learned tools even when using unknown tools. In addition, Figure 22 shows an example of motion that starts from an untrained initial pose and object position. As shown, motions close to the target state were generated even if the robot starts from an untrained initial state. As a matter of interest, in Figure 19 the robot generated the pulling

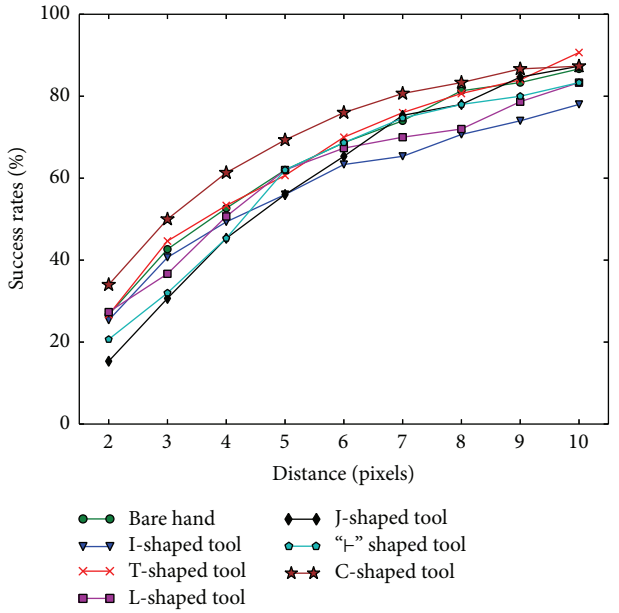


Figure 17: Relationships between the success rate and the position error.

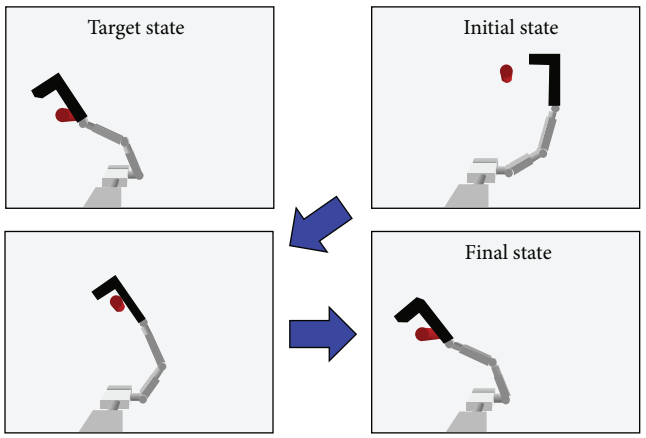


Figure 18: Generated motion by L-shaped tool from right side to object.

motion with a J-shaped tool after first avoiding contact of the protruding part of the tool and the object. If the robot did not avoid the contact at first, the object will be repelled and the robot would not be able to manipulate the object correctly.

Among the generated motions that have a final state close to a given target state, some have different movement courses compared to the teaching data (Figure 23). Because motions other than the learned ones are generated, it can be said that the model does not overfit. Figure 14 shows that the PB values formed different clusters for each tool. Figure 16 shows that robot recognizes different PB values for each tool. In such cases, it can be said that the robot have acquired the tool function (affordance) and generated the motion by using the tool function.

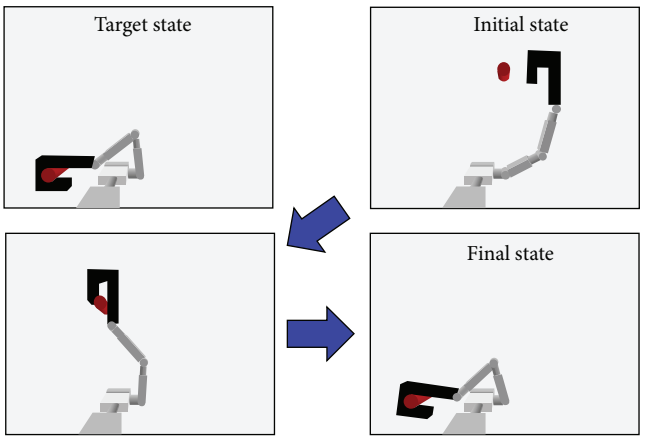


Figure 19: Generated motion by J-shaped tool from right side to object.

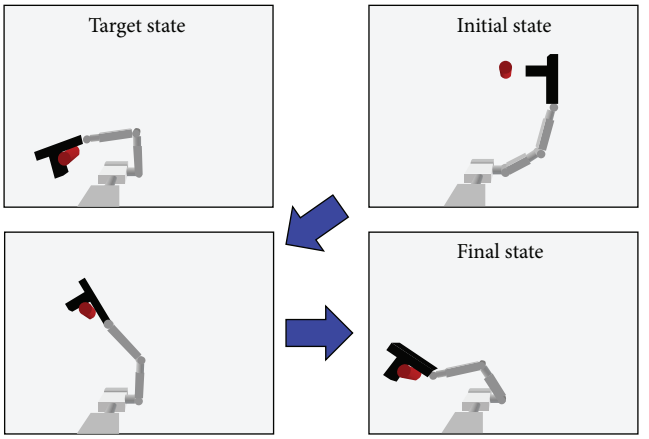


Figure 20: Generated motion by "⊢" shaped tool from right side to object.

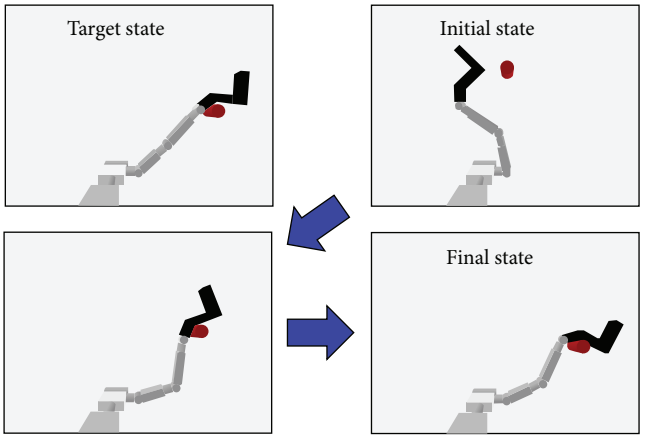


Figure 21: Generated motion by C-shaped tool from left side to object.

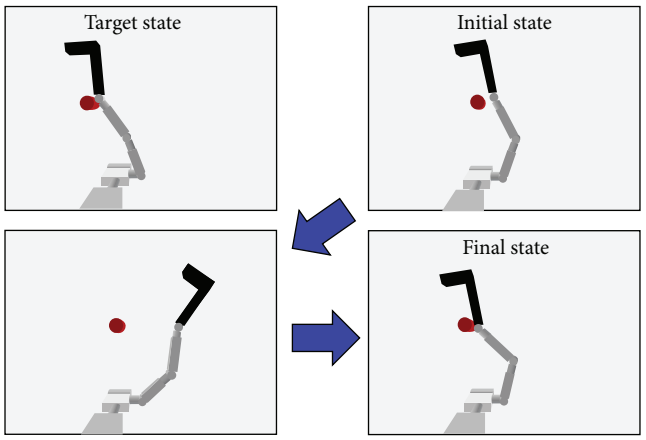


Figure 22: Generated motion by L-shaped tool from random initial state.

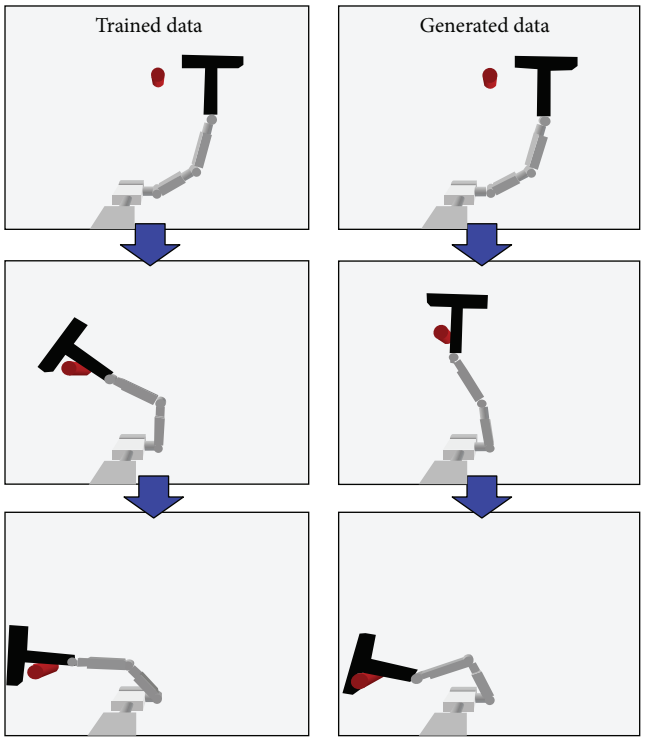


Figure 23: Generated motion by T-shaped tool on the way does not match.

6. Conclusion

This study's objective is to achieve robot tool use without the need for predetermined features and models by having the robot self-organize the required features inspired by human development and cognitive mechanisms. By using the concept of tool-body assimilation, it is possible to treat a tool as an extension of the body. Therefore, it is possible to represent the body and tool use models in one single model. Previous models implemented additional models per additional tool to be used in addition to the body model. However, our proposed model made it possible to represent all this with only one model. Related studies have required preset motions, preset tool and motion features, and full searches of all possible motions during motion generation. To overcome these issues, we propose the following approach: (1) body babbling with a humanoid model that does not require preset motions, (2) learning algorithm that does not require preset sensorimotor integration and tool features, with the concept of tool-body assimilation by using MTRNN and image feature extraction by an autoencoder with DNN, and (3) recognition of motion from the goal state. The evaluation experiment is an object manipulation task conducted with OpenHRP3, a robotics simulator. As a result, when given an image of a final state, the robot is able to generate a motion similar to the final state.

As next steps, we plan to extend the study to a seven-degrees-of-freedom model, design research settings that consider more of the human body, and set a more specific task for quantitative assessments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work has been supported by JST PRESTO "Information Environment and Humans," MEXT Grant-in-Aid for Scientific Research on Innovative Areas "Constructive Developmental Science" (24119003), JSPS Grant-in-Aid for Scientific Research (S)(2522005), "Fundamental Study for Intelligent Machine to Coexist with Nature" Research Institute for Science and Engineering, Waseda University, and Grants for Excellent Graduate Schools, MEXT, Japan. The authors would like to thank H. Arie, K. Noda, and S. Murata for their help in conducting the experiments.

References

[1] M. Asada, K. Hosoda, Y. Kuniyoshi et al., "Cognitive developmental robotics: a survey," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.

[2] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, vol. 8, no. 2, pp. 79–86, 2004.

[3] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Towards a model for tool-body assimilation and adaptive tool-use," in *Proceedings of the IEEE 6th International Conference on Development and Learning (ICDL '07)*, pp. 288–293, July 2007.

[4] M. Hikita, S. Fuke, M. Ogino, and M. Asada, "Cross-modal body representation based on visual attention by saliency," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 2041–2046, September 2008.

[5] J. J. Gibson, *The Senses Considered as Perceptual Systems*, Houghton-Mifflin Company, Boston, Mass, USA, 1966.

[6] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: from sensory—motor coordination to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[7] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[8] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *Proceeding of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 1579–1585, Taipei, Taiwan, October 2010.

[9] A. Stoytchev, "Behavior-grounded representation of tool affordances," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 3060–3065, April 2005.

[10] R. Detry, E. Başeski, M. Popović et al., "Learning object-specific grasp affordance densities," in *Proceedings of the IEEE 8th International Conference on Development and Learning (ICDL '09)*, pp. 1–7, June 2009.

[11] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta, "Exploring affordances and tool use on the iCub," in *Proceedings of the IEEE/RAS International Conference of Humanoids Robotics*, 2013.

[12] R. Pfeifer and J. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence*, MIT Press, Cambridge, Mass, USA, 2007.

[13] A. Streri and J. Féron, "The development of haptic abilities in very young infants: from perception to cognition," *Infant Behavior and Development*, vol. 28, no. 3, pp. 290–304, 2005.

[14] R. Saegusa, G. Metta, G. Sandini, and S. Sakka, "Active motor babbling for sensorimotor learning," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '08)*, pp. 794–799, 2009.

[15] K. Mochizuki, S. Nishide, H. G. Okuno, and T. Ogata, "Developmental human-robot imitation learning of drawing with a neuro dynamical system," in *Proceedings of the International Conference on Systems, Man, and Cybernetics*, pp. 2337–2341, 2013.

[16] D. Caligiore, T. Ferrauto, D. Parisi, N. Accornero, M. Capozza, and G. Baldassarre, "Using motor babbling and hebb rules for modeling the development of reaching with obstacles and grasping," in *Proceedings of the International Conference on Cognitive Systems (COGSYS '08)*, 2008.

[17] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 3328–3333, Pasadena, Calif, USA, May 2008.

[18] T. Sasaoka, H. Mizuhara, and T. Inui, "The interaction between the parietal and motor areas in dynamic imagery manipulation: an fMRI study ," in *Advances in Cognitive Neurodynamics (II)*, pp. 345–349, 2011.

[19] S. Nishide, J. Tani, T. Takahashi, H. G. Okuno, and T. Ogata, "Tool-body assimilation of humanoid robot using a neurodynamical system," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 2, pp. 139–149, 2012.

[20] C. F. Michaels, Z. Weier, and S. J. Harrison, "Using vision and dynamic touch to perceive the affordances of tools," *Perception*, vol. 36, no. 5, pp. 750–772, 2007.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS Proceedings*, vol. 1, no. 2, 2012.

[22] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[23] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Intersensory causality modeling using deep neural networks," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, Manchester, UK, October 2013.

[24] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of object manipulation behaviors using deep neural networks," in *Proceedings of IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, IEEE/RSJ, Tokyo, Japan, 2013.

[25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[26] J. Martens, "Deep learning via Hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 735–742, June 2010.

[27] B. Pearlmutter, "Fast exact multiplication by the Hessian," *Neural Computation*, vol. 6, no. 1, pp. 147–160, 1994.

[28] N. N. Schraudolph, "Fast curvature matrix-vector products for second-order gradient descent," *Neural Computation*, vol. 14, no. 7, pp. 1723–1738, 2002.

[29] H. Arie, T. Endo, T. Arakaki, S. Sugano, and J. Tani, "Creating novelgoal-directed actions at criticality: a neuro-robotic experiment," *New Mathematics and Natural Computation*, vol. 5, pp. 307–334, 2009.

[30] T. Kohonen, *Self-Organization and Associative Memory*, vol. 8, Springer, New York, NY, USA, 2nd edition, 1988.

[31] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment," *PLoS Computational Biology*, vol. 4, no. 11, Article ID e1000220, 2008.

[32] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pp. 513–546, Erlbaum, Hillsdale, NJ, USA, 1986.

[33] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McLelland, Eds., MIT Press, Cambridge, Mass, USA, 1986.

[34] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: open architecture humanoid robotics platform," *International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[35] "Kokoro: custom-made robot: ACTROID," 2014, http://www.kokoro-dreams.co.jp/rt_tokutyu/actroid.html.

[36] I. A. Kapandji, "Physiologie Articulaire," MALOINE S.A. EDITEUR, 1980.

[37] K. Takahashi, T. Ogata, H. Tjandra, Y. Yamaguchi, Y. Suga, and S. Sugano, "Tool—body assimilation model using a neurodynamical system for acquiring representation of tool function and motion," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '14)*, Besancon, France, July 2014.

[38] K. Takahashi, T. Ogata, H. Tjandra, S. Murata, H. Arie, and S. Sugano, "Tool-body assimilation model based on body babbling and a neuro-dynamical system for motion generation," in

Proceedings of the 24th International Conference on Artificial Neural Networks (ICANN '14), Lecture Notes in Computer Science, Hamburg, Germany, September 2014.

[39] J. Namikawa, R. Nishimoto, and J. Tani, "A neurodynamic account of spontaneous behaviour," *PLoS Computational Biology*, vol. 7, no. 10, Article ID e1002221, 2011.

[40] Y. Kawai, Y. Nagai, and M. Asada, "Perceptual development triggered by its self-organization in cognitive learning," in *Proceeding of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 5159–5164, Vilamoura, Portugal, October 2012.