Modeling Tool-Body Assimilation using Second-order Recurrent Neural Network

Shun Nishide, Tatsuhiro Nakagawa, Tetsuya Ogata, Jun Tani, Toru Takahashi, and Hiroshi G. Okuno

Abstract— Tool-body assimilation is one of the intelligent human abilities. Through trial and experience, humans are capable of using tools as if they are part of their own bodies. This paper presents a method to apply a robot's active sensing experience for creating the tool-body assimilation model. The model is composed of a feature extraction module, dynamics learning module, and a tool recognition module. Self-Organizing Map (SOM) is used for the feature extraction module to extract object features from raw images. Multiple Time-scales Recurrent Neural Network (MTRNN) is used as the dynamics learning module. Parametric Bias (PB) nodes are attached to the weights of MTRNN as second-order network to modulate the behavior of MTRNN based on the tool. The generalization capability of neural networks provide the model the ability to deal with unknown tools. Experiments are performed with HRP-2 using no tool, I-shaped, T-shaped, and L-shaped tools. The distribution of PB values have shown that the model has learned that the robot's dynamic properties change when holding a tool. The results of the experiment show that the tool-body assimilation model is capable of applying to unknown objects to generate goal-oriented motions.

I. INTRODUCTION

Tool use is one of the fundamental and indispensible abilities for humans. Most human environments consist of a variety of tools, each with its peculiar functionality. Introduction of robots into such society calls for implementation of tool use into the robot's abilities.

A key to developing the robot's ability to use tools is to refer to how humans develop to learn to use tools. Infants develop and generalize their knowledge of tools through numerous execution of trial and error [1]. Most inexperienced infants tend to rely on dynamic touch for determining the functionality of tools. Dynamic touch refers to the process of perceiving properties of hand-held or hand-wielded objects through a single perceptual system [2]. Through tool manipulation experience, humans tend to learn to use tools as if they are part of their own bodies. Reports from cognitive fields have also shown that such phenomenon can be observed with monkeys [3]. This phenomenon is called "tool-body assimilation" in neurophysiology. Our goal is to implement the tool-body assimilation model into the robot through its own active sensing [4] experiences.

S. Nishide, T. Ogata, T. Takahashi, and H. G. Okuno are with the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, Japan {nishide, ogata, tall, okuno}@kuis.kyoto-u.ac.jp

T. Nakagawa is with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan tatsuhiro-n@is.naist.jp

J. Tani is with the Brain Science Institute, RIKEN, Saitama, Japan tani@brain.riken.jp

There are two fundamental issues for creating the tool-body assimilation model.

1) Tool recognition.
2) Design of tool manipulation motion.

Most existing studies on robot tool use have aimed to deal with either 1) or 2). In these studies, the features for representing tools are predefined. Therefore, these models were incapable of adapting to unknown tools which cannot be represented by the predefined features.

We dealt with the two issues through the following approaches.

1) Self-organization of tool features, and active waving for tool determination.
2) Generalization of tool features and robot motions.

Concerning the first approach, we utilize Self-Organizing Map (SOM) [5] for self-organizing tool features from camera images. During this process, the necessary features are self-organized without the need to predefine the features. For dynamic touch motion, we apply active waving for determining the dynamical properties of the tool. Concerning the second approach, we utilize neural networks for creating our model. The generalization capability of neural networks provide the model the ability to deal with unknown tools and motions.

Our model consists of three modules: dynamics learning module, tool recognition module, and feature extraction module. As described above, we utilized SOM for the feature extraction module. For the dynamics learning module, we utilize Multiple Time-scales Recurrent Neural Network (MTRNN) [6]. MTRNN is capable of learning and generalizing multiple sequential data in a single model. A second-order hierarchical neural network with Parametric Bias (PB) nodes in the input layer is used as the tool recognition module. The tool recognition module is linked to MTRNN to modulate its behavior based on the value of PB nodes, which represents the tool.

As a criterion for tool use, we focus on extension of reach. This is one of the four factors for which Beck proposes that animals use tools [7]. A related work by Stoytchev was conducted to ground the affordance of tools based on the robot's experience [8]. The work showed the robot's capability to adapt to unknown tools by removing the motion affordance when the target object does not move as expected. Although the method showed great effectiveness with a variety of tools, the robot motions were limited to predefined motions. The model also used color information for determining the tools, and therefore required trials of object manipulation to confirm the tool affordance (specifically for unknown or

broken tools) by observing the target object motions. Our model is capable of generalizing motion data using neural networks. Image features are also self-organized during the process to describe the tools. Therefore, our model is capable of observing the tool motions for determining the dynamic properties of the tool, without the necessity to generate motions with the target object.

Other related works on tool-body assimilation have been conducted by Nabeshima [9] and Hikita [10]. The work by Nabeshima used tools to retrieve (or pull) an unseen object towards the robot through a simulation model. Although the work proved effective with a variety of rigid tools, the features describing the tools were predefined and would not be able to adapt to general tools (e.g. soft tools) which are incapable of being defined by these features. The work by Hikita used a saliency map to express how the representation of robot arm extends based on the grasping tool. Our work presents how such representation can be used to generate robot motions with tools.

The rest of the paper is composed as follows. Section II describes the overview of the model. Section III describes the tool-body assimilation method using the model. Section IV describes the experimental setup. Section V describes the experimental results. Section VI presents some discussions considering the results. Conclusions and future works are presented in Section VII.

II. OVERVIEW OF MODEL

This section describes the overview of the technique. The model is composed of a feature extraction module, dynamics learning module and a tool recognition module. Self-Organizing Map (SOM) [5] is used for the feature extraction module to extract object features from raw images. Multiple Time-scales Recurrent Neural Network, shown in Fig. 1, is utilized for the dynamics learning module. MTRNN works as a predictor which inputs the current motor value $M(t)$ and image feature $I(t)$, and outputs the next motor value $M(t+1)$ and image feature $I(t+1)$. It also has fast context nodes C_F and slow context nodes C_S to deal with history information. A detail of the model is presented in the next subsection. The selection of the two models are due to works applying SOM and MTRNN, showing the compatibility of the two models [11]. The tool recognition module is constructed by a second-order network [12] which connects Parametric Bias (PB) nodes to the weights of MTRNN. The PB nodes possess the same role as those of Recurrent Neural Network with Parametric Bias (RNNPB) [13]. Changing the value of the PB nodes would change the behavior of MTRNN. Thus, training of the whole system would learn dynamics of the robot/tool/object in MTRNN while self-organizing the relationship between different tools and robot motion in the PB nodes and slow context nodes. An overview of the model is shown in Fig. 2.

A. MTRNN Model

MTRNN, shown in Fig. 1, is a variant of the Jordan type RNN [14] with two layers, which acts as a predictor that
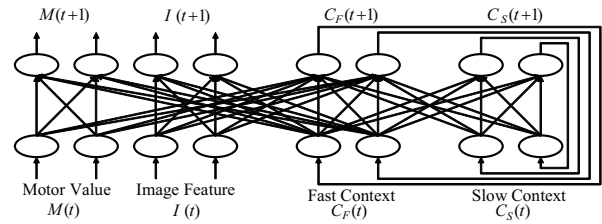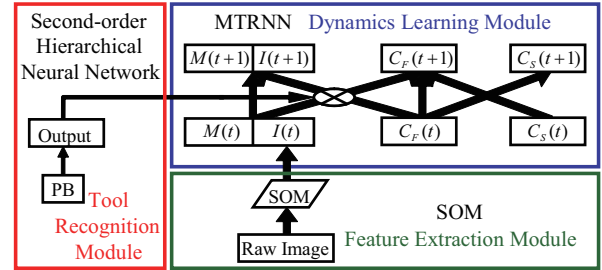

Fig. 1.   Composition of MTRNN

Fig. 2.   Overview of the Whole Model

inputs the current state $[M(t), I(t)]$ and outputs the next state $[M(t + 1), I(t + 1)]$. $M(t)$ and $I(t)$ each represent the motor value and image feature value. Context nodes are composed of fast context nodes (C_F), which possess small time constant, and slow context nodes (C_S), which possess large time constant. Difference in time constants leads to difference in firing speed of the context nodes. Therefore, C_F nodes represent the primitives of sequential data, while C_S nodes represent the sequence of these primitives (Fig. 3). Thus, MTRNN is capable of learning longer and more complex sequential patterns compared to conventional RNN models. The time constant of input nodes $[M(t), I(t)]$ are the smallest out of the three types of nodes. Each node of MTRNN in the input layer is connected to the output layer, with an exception of $[M(t), I(t)]$ nodes to $C_S(t+1)$ nodes, $C_S(t)$ nodes to $[M(t + 1), I(t + 1)]$ nodes, $M(t)$ nodes to $I(t+1)$ nodes, and $I(t)$ nodes to $M(t+1)$ nodes. Compared to RNNPB, which has been used as the dynamics learning module in our previous work (e.g. [15]), MTRNN has the advantage in complexity and capacity of learning sequences.

Training of MTRNN is done using the Back Propagation Through Time (BPTT) algorithm [16]. The algorithm consists of forward calculation and weight update.
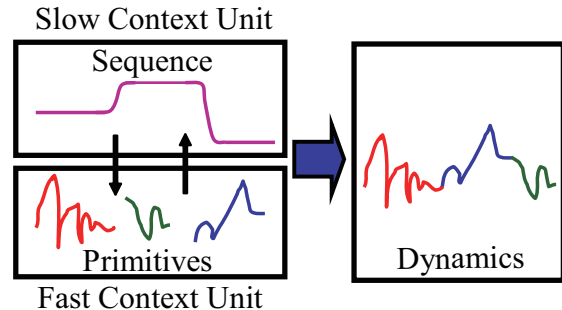

Fig. 3.   Dynamics Representation of MTRNN

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

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

where τ_i is the time constant of the ith neuron, w_{ij} is the weight value from the jth input neuron to the ith output neuron, and $x_j(t)$ is the input value. The output of the ith neuron $y_i(t)$ is calculated by applying the sigmoid function,

$$y_i(t) = sigmoid(u_i(t)), \quad (2)$$

$$sigmoid(x) = \frac{1}{1 + exp(-x))}. \quad (3)$$

The input value $x_i(t)$ is calculated by

$$x_i(t) = \begin{cases} 0.9 \times y_i(t-1) + 0.1 \times T_i(t) & i \in M, I \\ y_i(t-1) & otherwise \end{cases}, (4)$$

where $T_i(t)$ is the teacher signal for the ith neuron at step t. The input values $x_i(t)$ are calculated using the output of the previous step $y_i(t-1)$, feedbacking the teacher signals $T_i(t)$. This process prevents the training error to increase. The outputs of the context nodes of the previous step $(t-1)$, $C_F(t-1)$ and $C_S(t-1)$, are directly input into the input of the context nodes of the current step t, $C_F(t)$ and $C_S(t)$.

Using the outputs calculated in forward calculation, the weights are updated using the training error E defined as

$$E = \sum_t \sum_{n \in M, I} (y_i(t-1) - T_i(t))^2. \quad (5)$$

The weight from the jth input to the ith output is updated using the derivative of the training error $\partial E / \partial w_{ij}$ as

$$w_{ij}(n+1) = w_{ij}(n) - \alpha \frac{\partial E}{\partial w_{ij}}. \quad (6)$$

Back propagating the errors to the initial step, the initial C_S value, $C_S(0)$, is also updated along with the weight values.

Using the trained MTRNN, sequences can be recovered by the C_S value. By inputting the initial C_S value ($C_S(0)$), C_F value (which is fixed), and the initial motor/image feature values $[M(0), I(0)]$ the output of step 0 can be calculated. By recursively inputting the output back into the input, the whole sequence can be recovered. This process is called *closed loop calculation*. On the other hand, the process which inputs the observed motor/image feature values for each step is called *open loop calculation*.

B. Training of Tool Recognition Module

The tool recognition module is also trained using BPTT algorithm. We denote the output neuron of the tool recognition module, linked to the weight from the jth input to ith output of MTRNN, as O_{ij}.

Weight update of the tool recognition module is conducted using the derivative $\partial E / \partial w_{ij}$. The update equation for

weight from the kth PB node to the output O_{ij} is calculated by

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

where α is the learning constant.

For updating the PB values, we differentiate the output error E with the internal value of the kth PB value p_k as

$$\frac{\partial E}{\partial p_k} = \sum_{i,j} \frac{\partial E}{\partial o_{ij}} \frac{\partial o_{ij}}{\partial p_k}, \quad (8)$$

where o_{ij} is the output value for the output neuron O_{ij}. The update value for the kth PB node Δp_k can be calculated as

$$\Delta p_k = -\varepsilon \sum_{i,j} \frac{\partial E}{\partial o_{ij}} \frac{\partial o_{ij}}{\partial p_k}, \quad (9)$$

where ε is the training constant.

C. Feature Extraction Module

In this paper, the authors do not predefine object features for adaptation to unknown tools and objects. For feature extraction, the authors use Self-Organizing Map (SOM) to self-organize the necessary features required for training.

SOM is composed of two layers: input and output layers. Neurons in the output layer are arranged two-dimensionally, possessing weight vectors w with the same dimension as the input vector v. The firing rate of each output neuron is determined from the similarity of the weight vector and input vector.

In this paper, the input vector is composed of raw image, with the neurons corresponding to each image pixel. The image features are composed of the normalized firing rate to [0.1, 0.9]. For calculating the similarity between the weight vector and input vector, we use the Euclidean distance $||w - v||$.

III. TOOL-BODY ASSIMILATION PROCESS

In this section, we describe the tool-body assimilation process. Tool-body assimilation is conducted by the following four phases.

1) Train self model.
2) Train tool recognition module.
3) Determination of grasping tool based on active waving motion of the tool.
4) Motion generation using tool.

The first two phases are done during training, and the other two are done when generating motion.

A. Training Self Model

As a preliminary phase, the robot conducts active sensing motions with objects using various tools and robot motions. Features of image data obtained during this preliminary phase are acquired using SOM. Using the acquired motion and image features during active sensing, the dynamics learning module MTRNN is trained. This phase links the robot motion with the visual change in the image.

B. Training Tool Recognition Module

During active sensing, the robot uses a variety of tools. The tool recognition module (hierarchical neural network) is trained by fixing the weights of MTRNN. The training self-organizes the similarities of tools in the PB space based on their dynamical properties.

C. Active Waving of Tools

To determine the dynamical property of the tools, the robot generates active waving motion while grasping the tool. The PB value which minimizes the error of the predicted image feature and observed image feature is calculated through the BPTT algorithm only updating the PB values. The PB derived during this process represents the tool the robot is grasping.

D. Motion Generation with Tool

Using the PB value determined in the third phase, the robot generates robot motion by determining the C_S value. Given the target image, the BPTT algorithm is used to update only the C_S value. The PB value calculated in the third phase and the obtained C_S value are input into the model to calculate the motion sequence through *closed loop calculation*. The robot traces through the motor value sequence to achieve the target image state.

IV. EXPERIMENTAL SETUP

The authors used the humanoid robot HRP-2 shown in Fig. 4 for evaluation of the method. Tools (L-shaped, T-shaped, and I-shaped shown in Fig. 5) are fixed onto the left arm of HRP-2 while generating robot motions. The robot used the T-shaped, I-shaped tools for acquiring training data, in addition to generating motions without tools. The L-shaped tool is used as an unknown tool for evaluation of the model.

A. Data Acquisition

The authors conducted an experiment with two red objects: one within the robot's reaching distance and the other capable of reaching only when using a tool. During the experiment, the robot used 7 DOF (shoulder roll, shoulder yaw, shoulder pitch, elbow pitch, wrist roll, wrist yaw, and wrist pitch) of the left arm for generating robot motions. Nine types of motions shown in Fig. 6 were generated for the T-shaped and I-shaped tools (and bare handed), acquiring a

total of 27 motion patterns. Each motion is composed of movement to two different postures from the initial. Image data were acquired using the central camera with the image resolution reduced to 16×12 pixels each holding the R, G, B values. The output of SOM, which represents object features, is composed of 9 dimensions. Each data (acquired at 15 steps/sec for 90 steps) is normalized to [0.1, 0.9] based on the minimum and maximum value of each data.

B. Construction of Model

The construction of the model is shown in Table I. The time constant for the input nodes, fast context nodes, slow context nodes were set to 2, 5, and 70, respectively.

V. EXPERIMENTAL RESULTS

In this section, we present the results of the experiment.

A. Determination of Tools

The third phase of tool-body assimilation consists of active waving of the robot arm to determine the PB value which represents the tool type. For each tool, the robot generated eight sets of motion, determining the PB for each set. Each set of motion is composed of four motion patterns shown in Table II. The motion pattern number in Table II corresponds to the motion number in Fig. 6.

Figure 7 shows the PB distribution for the determined PB values. The blue rhombi represent PB values of robot motions without tools. The red squares, green triangles, and orange circles represent PB values of robot motions with I-shaped tool, T-shaped tool, and unknown L-shaped tool, respectively. These PB values were calculated after the model has been trained, during the third phase of tool-body assimilation process.

There are two notable characteristics in the PB space. First, the blue rhombi are distributed far from PB values with tools. This result shows the effectivity of the tool-body assimilation model, as the robot has learned that it's dynamical properties

Fig. 4. HRP-2

Fig. 5. Tools used in Experiment

Number	Motion Type
1	Initial ⟶ (lf, d00) ⟶ (lf, d45)
2	Initial ⟶ (lf, d00) ⟶ (ln, d00)
3	Initial ⟶ (lf, d00) ⟶ (ln, d45)
4	Initial ⟶ (lf, d00) ⟶ (rf, d00)
5	Initial ⟶ (lf, d00) ⟶ (rn, d00)
6	Initial ⟶ (lf, d45) ⟶ (lf, d00)
7	Initial ⟶ (lf, d45) ⟶ (ln, d00)
8	Initial ⟶ (rn, d00) ⟶ (lf, d00)
9	Initial ⟶ (rn, d45) ⟶ (lf, d00)
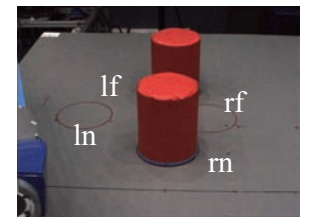
d00 : wrist facing vertical
d45 : wrist facing diagonal (45 degrees)

Fig. 6. Motion Patterns used in Experiment

TABLE I

CONSTRUCTION OF MODEL

No. of Motor Input Nodes	7
No. of Image Feature Input Nodes	9
No. of Fast Context Nodes	45
No. of Slow Context Nodes	15
No. of Parametric Bias Nodes	2
No. of Input Nodes for SOM	576

TABLE II

MOTION SET FOR PB DETERMINATION

Motion Set	Motion Patterns
A	2, 3, 6, 8
B	1, 4, 5, 6
C	1, 5, 6, 7
D	4, 5, 7, 9
E	4, 6, 8, 9
F	1, 6, 8, 9
G	1, 6, 7, 9
H	5, 6, 7, 9
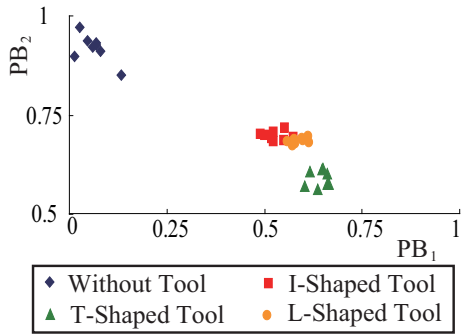
Fig. 7.   Self-Organized PB Space of Tools
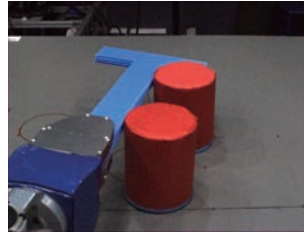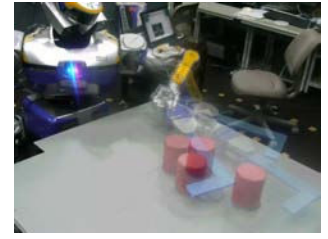

Fig. 8.   Goal State Image

Fig. 9.   Motion Generation with L-Shaped Tool

Fig. 10.   Motion Generation with T-Shaped Tool

Fig. 11.   Motion Generation with I-Shaped Tool

would change drastically when holding a tool. Second, the PB values of unknown L-shaped tool are located between the PB values of I-shaped tool and T-shaped tool. This result implies that the robot has learned that the unknown L-shaped tool possesses dynamical properties medial to that of I-shaped tool and T-shaped tool.

B. Goal-Oriented Motion Generation

For motion generation experiment, we considered a condition that the robot is required to pull the target objects toward the robot. The robot first determined the tool by calculating the PB value of the tool recognition module using waving motion in the third phase of the tool-body assimilation process. Then the robot was provided a goal state image to calculate the C_S value in the fourth phase of the tool-body assimilation process. The goal state image was generated with the T-shaped tool in the image as shown in Fig. 8.

The generated motions for the L-shaped tool, T-shaped tool, and I-shaped tool are shown in Fig. 9, Fig. 10, and Fig. 11, respectively. While, the robot succeeded in generating the motion that would pull the objects toward the robot for L-shaped tool and T-shaped tool, the robot just touched the object when holding the I-shaped tool. The motions in Fig. 9 and Fig. 10 are similar in the sense that the objects are pushed a little to the right from the robot, before being pulled. In contrast, the robot generated a different motion when holding the I-shaped object as the robot did not even push the objects to the right. As the robot cannot pull the objects, the robot did not generate a pulling motion as it did with the T-shaped and L-shaped tools. These results show that the robot has learned the affordance of the hooked shape of the tool, to pull an object towards the robot.

VI. DISCUSSIONS

In this section, we present discussions considering the experiments.

A. Tool-Body Assimilation Model

The ability to assimilate tools to the body is a product of the capability to perceive affordance through experience. Mark has conducted experiments using 10-cm platform shoes to sit and climb stairs [17]. The results have shown that perception of affordance boundaries become accurate over trials. The ability to assimilate these platform shoes (tools) has developed through experience.

In our experiment, we have shown that through the robot's active sensing experience with various tools, the robot has learned to alter its dynamical properties by changing PB values. The distribution of PB values between a tool holding and without tool conditions differed greatly compared to the distributions between different tool types. This result shows that the dynamical properties of the robot's arm change greatly when holding tools, compared to changing tools.

The result of the motion generation experiment denotes the model's capability to deal with unknown tools. A difficulty in training the model, is in the determination of tool features and object features. By self-organizing image features using SOM and applying the generalization capability of neural networks, the robot was successful at generalizing training data to adapt to an unknown L-shaped tool. The success of motion generation with the L-shaped tool and T-shaped tool, in contrast to the failure with I-shaped tool, implies the model's capability to perceive and generalize tool affordance.

B. Motion Generation with General Objects

In this paper, we conducted an experiment which uses a single type of object and three types of tools. The experiment showed the model's generalization capability to handle an unknown tool (L-shaped tool), by generalizing the experience with the T-shaped tool and I-shaped tool. In this subsection,

we discuss the model's capability to deal with general target objects.

A difficulty in dealing with various objects is in the variety of motions the object can take based on its shape. In the experiment presented in this paper, we used an upright cylinder which would slide when pushed. However, the cylinder may roll when laid, or may fall over if the cylinder was taller and pushed on the top. These object motions all have different characteristics, and the model should be capable of adapting to different object motions when generating goal-oriented motions.

In our previous work, we created a model that predicts object motion based on the visual image and robot motion [15]. The work showed the model's capability to predict four types of motion (slide, roll, fall over, and bounce) using general objects. Although RNNPB was used as the dynamics learning model in the paper, the model could be easily refined to use MTRNN for the dynamics learning model. By integrating the two methods, we believe that the model could be improved to deal with a larger variety of objects.

C. Tool-Body Assimilation from Cognitive Fields Perspective

Experiments by Michaels have shown that visual information dominates over information acquired by dynamic touch when humans perceive affordance of tools [18]. On the other hand, studies with infants have shown that dynamic touch is the main property for perceiving objects [1]. The two works imply that inexperienced infants tend to rely on dynamic touch for perceiving tool affordance, and change to rely on visual information as they gain experience.

In our work, we used the waving motion as dynamic touch motion to determine the dynamical properties (affordance) of the tool. The result has shown that the model is capable of perceiving the dynamical properties of unknown objects. Based on the human development model, our next step would be to transit to the stage where visual information would be introduced to predict the dynamical property. The model would require a prediction model to link the static image to tool dynamics, which we presented in our previous work [15]. We plan to link the two works to improve our model to the next stage of infant development.

VII. Conclusions

In this paper, we presented a tool-body assimilation model composed of dynamics learning module, tool recognition module, and a feature extraction module. MTRNN is utilized for the dynamics learning module to learn multiple sequential data of robot motor and image feature values. PB nodes are linked to the weights of MTRNN as the tool recognition module to alter the behavior of MTRNN based on the dynamical properties of tools. SOM is utilized as the feature extraction module for self-organizing image features. Experiments were conducted with the humanoid robot HRP-2 using I-shaped, T-shaped, and L-shaped tools. The active waving motion experiment to determine the grasping tool has shown that the model has learned that the robot's dynamic model alters greatly when holding a tool. The PB distribution has

also shown that the PB value of the unknown L-shaped tool is calculated appropriately in the middle of the PB distributions of I-shaped tool and T-shaped tool. The motion generation experiment of pulling an object toward the robot has shown that the robot is capable of appropriately generating pulling motions with the unknown L-shaped tool.

As our next step, we plan to develop our model through integration with our past works. This would lead to the next stage of infant development where visual information would be mainly used for perceiving the dynamical properties of tools. Further on, we plan to apply our model to more complex motions and a larger variety of tools and objects.

References

[1] A. Streri and J. Féron, "The development of haptic abilities in very young infants: From perception to cognition," *Infant Behavior and Development*, Vol. 28, pp. 290-304, 2005.

[2] J. J. Gibson, "The Senses Considered as Perceptual Systems," *Boston, MA, Houghton Mifflin*, 1966.

[3] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, Vol. 8, No. 2, pp. 79-86, 2004.

[4] R. Bajcsy, "Active Perception," in *IEEE Proc., Special issue on Computer Vision*, Vol. 76, No. 8, pp. 996-1005, 1988.

[5] T. Kohonen, "Self-Organization and Associative Memory," *2nd Ed., New York, Springer*, 1988.

[6] Y. Yamashita and J. Tani, "Emergence of Functional Hierarchy in a Multiple Timescales Recurrent Neural Network Model: A Humanoid Robot Experiment," *PLoS Computational Biology*, Vol. 4, No. 11, e1000220, 2008.

[7] B. B. Beck, "Animal tool behavior: The use and manufacture of tools by animals," *New York: Garland STMP Press*, 1980.

[8] A. Stoytchev, "Behavior-Grounded Representation of Tool Affordances," in *Proc. ICRA*, pp. 3060-3065, 2005.

[9] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Towards a Model for Tool-Body Assimilation and Adaptive Tool-Use," in *Proc. ICDL*, pp. 288-293, 2007.

[10] M. Hikita, S. Fuke, M. Ogino, and M. Asada, "Cross-modal body representation based on visual attention by saliency," in *Proc. IROS*, pp. 2041-2046, 2008.

[11] H. Arie, et al., "Creating novel goal-directed actions at criticality: A neuro-robotic experiment," *New Mathematics and Natural Computation*, Vol. 5, pp. 307-334, 2009.

[12] J. B. Pollack, "The Induction of Dynamical Recognizers," *Machine Learning*, Vol. 7, pp. 227-252, 1991.

[13] J. Tani and M. Ito, "Self-Organization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment," *IEEE Trans. on SMC Part A*, Vol. 33, No. 4, pp. 481-488, 2003.

[14] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *Eighth Annual Conf. of the Cognitive Science Society* (Erlbaum, Hillsdale, NJ), pp. 513-546, 1986.

[15] S. Nishide, et al., "Predicting Object Dynamics from Visual Images through Active Sensing Experiences," *Advanced Robotics*, Vol. 22, No. 5, pp. 527-546, 2008.

[16] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation," in *D. E. Rumelhart and J. L. McLelland, editors Parallel Distributed Processing (Cambridge, MA: MIT Press)*, 1986.

[17] L. S. Mark, "Eyeheight-Scaled Information about Affordances: A Study of Sitting and Climbing," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 10, pp. 683-703, 1987.

[18] C. F. Michaels, Z. Weier, and S. J. Harrison, "Using vision and dynamic touch to perceive the affordances of tools," *Perception*, Vol. 36, pp. 750-772, 2007.