Learning for Goal-Directed Actions Using RNNPB: Developmental Change of "What to Imitate"

Jun-Cheol Park, Dae-Shik Kim, and Yukie Nagai

***Abstract*—"What to imitate" is one of the most important and difficult issues in robot imitation learning. A possible solution from an engineering approach involves focusing on the salient properties of actions. We investigate the developmental change of what to imitate in robot action learning in this paper. Our robot is equipped with a recurrent neural network with parametric bias (RNNPB), and learned to imitate multiple goal-directed actions in two different environments (i.e., simulation and real humanoid robot). Our close analysis of the error measures and the internal representation of the RNNPB revealed that actions' most salient properties (i.e., reaching the desired end of motor trajectories) were learned first, while the less salient properties (i.e., matching the shape of motor trajectories) were learned later. Interestingly, this result was analogous to the developmental process of human infant's action imitation. We discuss the importance of our results in terms of understanding the underlying mechanisms of human development.**

***Index Terms*—Error-based learning, imitation learning, predictive learning, recurrent neural network with parametric bias (RNNPB), what to imitate.**

I. Introduction

IMITATION learning is a promising approach through which intelligent robots can learn complex and novel behaviors from humans [1]. Moreover, computational models for robot imitation learning could be used to understand how humans, particularly infants, could learn motor skills [2]. This theory has been suggested in the field of cognitive developmental robotics. The two purposes of this research area are to design an intelligent robot inspired by human development and to understand human development through designing intelligent robots [3], [4]. Regarding imitation learning, there are specific key issues (e.g., correspondence problems, what to imitate, when to imitate, whom to imitate, and so on) which should be solved [1], [5]–[7].

Manuscript received November 28, 2016; revised January 24, 2017; accepted March 1, 2017. Date of publication March 8, 2017; date of current version September 7, 2018. This work was supported in part by MEXT/JSPS KAKENHI under Research Project 24119003 and Research Project 24000012, and in part by the Brain Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planing under Grant NRF-2010-0018837.

J.-C. Park and D.-S. Kim are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: pakjce@kaist.ac.kr; daeshik@kaist.ac.kr).

Y. Nagai is with the Department of Adaptive Machine Systems, Osaka University, Osaka 565-0871, Japan (e-mail: yukie@ams.eng.osaka-u.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2017.2679765

Here, we focus on the issue of "what to imitate," which is concerned with which properties of demonstrated actions should be copied by imitators [8], [9]. For instance, a robot imitator could ignore a demonstrator's superficial behavior and reproduce only essential actions that are related to the goals of the actions. Then, what are the essential properties of actions, and what are superficial behaviors? In engineering approaches, salient features from sensory data or salient properties of actions in an action space can be selected as the essential properties of actions. For example, Mohammad and Nishdia [10] proposed a method of combining salient feature detectors and causality to enable a robot to decide which properties of demonstrated actions to imitate. Lee *et al.* [11] suggested a probabilistic approach in which reusable common action components are extracted under noisy environmental conditions (a hierarchical action learning and observation strategy) [12]. A study based on an inverse reinforcement learning framework [13] inferred goals by observing demonstrations.

How do humans solve this issue? Infant behavioral studies have shown that infants can understand the intended goals of demonstration and grasp the salient properties of actions via imitation [14]–[17]. Goal-directed behavior could have multiple subgoals, which are often organized hierarchically [18], [19]. Therefore, the target of imitation (i.e., what to imitate) would be chosen based on the hierarchy. Cognitive neuroscience studies [20], [21] suggested that mirror neurons play a major role in imitation behaviors, as they are not only activated when generating motor actions but also when observing others' movements. Moreover, Kilner *et al.* [22] suggested that the mirror neuron system uses a predictive coding scheme to solve multiple goal problems in human imitation. Nagai and Rohlfing [23] showed that caregivers tend to emphasize the most valuable properties, which are usually the goals at the top of the hierarchy, to bootstrap the learning of infants' goal-directed behaviors. Behavioral studies of infant imitation provide further insights into which aspects of actions are more or less important. Bekkering *et al.* [24] and Carpenter *et al.* [25] showed that young infants tend to ignore the means and superficial behaviors of actions, which are less salient, whereas adults and children can imitate entire actions.

The nonlinear improvement of infant abilities, as described above, is not specially observed in goal-directed actions, but rather commonly observed in infant development. The U-shaped change [26]–[28] is a phenomenon in which infant's capabilities appear to diminish at first and then improve later. It has been suggested that young infants' limited memory and perception capabilities might cause U-shaped changes [29]

and such nonlinear changes appear as a result of interactions between dynamical systems such as the neural mechanism, the body, and the environment [30]. Nevertheless, a period of decreased performance is believed necessary for infants to better organize their acquired abilities.

Other important issues in robot imitation include how to represent actions as sensorimotor information and how to reproduce demonstrators' actions. Calinon *et al.* [31] suggested a robot experiment that utilizes hidden Markov models (HMMs) [32] to imitate human actions by recognizing them and thus generating sensorimotor actions. Studies of dynamic movement primitive (DMP) [33] have adopted a dynamical system approach. They could generate adaptive actions (e.g., obstacle avoidance) because they implemented differential equations. In this scheme, Matsubara *et al.* [34] proposed a stylistic DMP for robotic learning through demonstration tasks. In addition, a predictive coding scheme with sensorymotor associations was used for robotic imitation learning tasks [35]. Yokoya *et al.* [37] utilized a recurrent neural network with parametric bias (RNNPB) [36] to learn and generate motor actions for robot imitation tasks. Commonly, computational models that are able to learn primitive motor actions as a form of time-series such as HMM, DMP, and recurrent neural network (RNN) are needed to implement robot imitation learning tasks.

RNNPB is one of the best options with which to model imitation learning tasks because it has the ability to encode multiple dynamic patterns into a static pattern of parametric bias (PB) unit activations as a biologically inspired model. An interesting feature of RNNPB models is the PB units' self-organization, through which multiple actions are coded in the RNN. Moreover, the network can generate novel actions based on previous ones due to the PB units generalization capabilities [38]. Using this generalization capability as a starting point, we utilize the RNNPB for the imitation learning tasks of robotic arms and suggest that the PB units self-organization process could solve the issue of what to imitate, which has not yet been investigated in robot imitation learning. The main contribution of this paper is the analysis of RNNPB during learning for multiple goal-directed actions. It would demonstrate how the issue of what to imitate could be explained by the developmental dynamics of PB values. We will discuss our experimental results with a focus on gaining new insights about the development of human imitation.

In the following section, we first introduce our hypothesis about goal-directed action imitation. The architecture and learning procedure of an RNNPB model are described in Section III. Next, the experimental setup and goal-directed behaviors are explained in Section IV. The results of experiments are then presented in Section V. Finally, we discuss our results, conclude this paper, and present suggestions for future work.

II. Our General Hypothesis

Our general hypothesis is that the learning process of RNNs such as RNNPB might be similar to humans' developmental changes regarding what to imitate. A back-propagation through time (BPTT) algorithm [39] is usually used to train RNNs as a supervised learning method that involves errors between the desired and generated output of the networks within the time frame. When multiple goal-directed motor actions are learned with RNNs through BPTT, the network parameters are organized to decrease the errors. The saliency of actions naturally appears as the error value from these error measures, where a larger error value indicates stronger saliency and a smaller error value indicates weaker saliency. Hence, we suggest that the salient properties of actions are extracted as a natural aspect of models' learning processes.

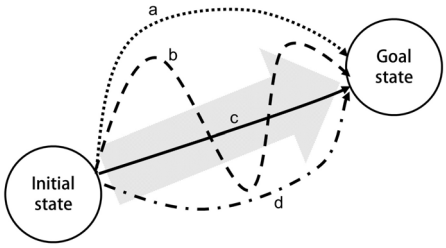


Fig. 1. There are several means to reach the goal state from the initial state in the action state space. Each action "*a*," *b*, *c*, and "*d*" has the same goal but different means.

A goal-directed action could have multiple methods through which to achieve a goal, as illustrated in Fig. 1. According to studies of human imitation [24], [25], goal-directed actions have two types of property: 1) the goal and 2) the means. "The goal" indicates the main properties of the actions intended by the demonstrator, while "the means" represents the properties of actions that are not directly related to the demonstrator's intentions, such as surface behaviors (i.e., style or specific trajectory of a movement). For example, when an agent tries to make its arm reach a desired position, the arm could directly reach for it or progress in a zigzag manner to avoid an obstacle.

Within the concept of a functional hierarchy of motor actions, a higher-level action refers to a goal-intended behavior with a long time scale, and a lower-level action refers to a local or primitive behavior with a short time scale, as seen in a study of real robot experiments with an RNN [40]. For example, actions "*b*" and "*c*" in Fig. 1 have the same goal property but different means. Additionally, action *b* moves slower than action *c*. An interesting point here is that the saliency of the goal and the means differ depending on the agents error function. The goal, which is the difference between the initial and the final state, produces a larger error if it is not yet achieved. In contrast, the means, which is the shape of the trajectory from the initial to the final state, produces a smaller error than the goal, because the intermediate states of the means are located between the initial and the final states. Therefore, investigating the developmental dynamics of errors for both goal and means permits the observation of quantitative measurements of a dynamical change of what to imitate.

III. RNNPB Model for Learning Agent

The RNNPB model [36] can memorize and reproduce multiple dynamics of input-output relationships using the static activations of PB units. PBs self-organize through learning

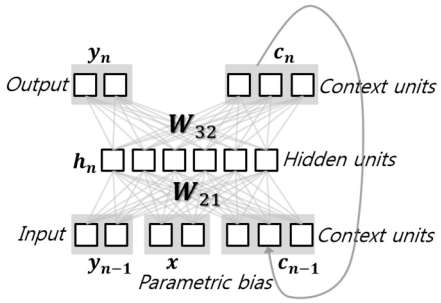


Fig. 2. Architecture of RNNPB model consisting of three layers: input layer, hidden layer, and output layer. The input layer is divided into the input units **y**, the PB units **x**, and the context units **c**. The output layer is divided into the output and context units. There is recurrent feedback from the output layer to the input layer in the context units, and the input and output layers are connected to the hidden layer with the fully connected weights ($\mathbf{W}_{21}$ and $\mathbf{W}_{32}$).

based on their experiences. Because of this advantage of the RNNPB model, we modeled a learning agent using it to build its internal memory for multiple goal-directed actions.

A. Architecture of the Model

As a modified version of Jordan-type RNNs [41], the RNNPB consists of a three-layered structure (input, hidden, and output layers) with recurrent feedback from the output layer to the input layer. Hence, it has a network parameter ψ that consists of two weight matrices and two bias vectors $\psi = \{\mathbf{W_{21}}, \mathbf{W_{32}}, \mathbf{b_1}, \mathbf{b_2}\}$. In addition, it has PB units in the input layer that allow the network to learn multiple actions (see Fig. 2). The input and output units representing time series data (e.g., motor action sequence in this paper) have n_{io} elements, and are denoted by $\mathbf{y} = [y_1, y_2, \ldots, y_{n_{io}}]^T$. The context units and the PB units encode the internal states of the time series data. The number of elements of PB units was two because it is easier to visualize and analyze them in a 2-D state space. The number of context and hidden units was empirically set to be able to represent all reference behaviors.

The primary role of the RNNPB network is generating memorized actions as a form of time-series. Therefore, the PB units have static values **x** as an input when generating actions, whereas the values of the context units change for each time step based on the recurrent connection from the context output unit of the previous time step $\mathbf{c_{n-1}}$ to the context input unit of the current time step $\mathbf{c_n}$. The hidden unit values at the nth time step $\mathbf{h}_n$ are produced with weight $\mathbf{W}_{21}$ and bias $\mathbf{b}_1$ from the input unit values at the previous time step $\mathbf{y}_{n-1}$, the PB values **x**, and the context unit values at the previous time step $\mathbf{c}_{n-1}$. The output unit values $\mathbf{y}_n$ and the context output values $\mathbf{c}_n$ at the n^{th} time step are produced with weight $\mathbf{W}_{32}$ and bias $\mathbf{b}_2$ from the hidden unit values

$$\mathbf{h}_n = \text{sigmoid}\left(\mathbf{W_{21}} \cdot \begin{bmatrix} \mathbf{y}_{n-1} \\ \mathbf{x} \\ \mathbf{c}_{n-1} \end{bmatrix} + \mathbf{b}_1\right) \tag{1}$$

$$\begin{bmatrix} \mathbf{y}_n \\ \mathbf{c}_n \end{bmatrix} = \text{sigmoid}\ (\mathbf{W_{32}} \cdot \mathbf{h}_n + \mathbf{b}_2). \tag{2}$$

B. Learning Procedure

When N desired motor actions $\mathbb{A} = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(N)}\}$ are given, the learning procedures main objective is to find an optimal network parameter ψ^* and N corresponding PB values $\mathbb{X} = \{\mathbf{x}^{(1)} \cdots, \mathbf{x}^{(N)}\}$ to make the network generate the desired action with low error [see Fig. 3(left)]

$$\mathbf{E}_n^{\text{out}} = \mathbf{y}_n^{\text{ref}} - \mathbf{y}_n^{\text{gen}} \tag{3}$$

$$\psi^*, \mathbb{X}^* = \underset{\psi, \mathbb{X}}{\text{argmin}} \sum_{a \in \mathbb{A}} \sum_{n=1}^{L} \mathbf{E}_{a,n}^{\text{out}}. \tag{4}$$

The BPTT algorithm [39] is applied to determine optimal network parameters. Similar to the back-propagation algorithm in feed-forward neural networks, the error $\mathbf{E}_n^{\text{out}}$ between a generated motor action $\mathbf{y}_n^{\text{gen}}$ and a given motor action $\mathbf{y}_n^{\text{ref}}$ at the nth time step is back-propagated from the third layer to the first layer. When the length of the desired time series is L, the recurrent connections of the context units are unfolded through time, and the unfolded network is then identical to a deep feed-forward neural network that has $3L$ layers. The network parameter ψ is updated iteratively through back-propagation as follows:

$$\Delta W_{32,ij} = \epsilon \sum_{n=1}^{L} h_{n,j} \begin{cases} \delta_{\text{out},n,i} & \text{if } i \in \text{output unit} \\ \delta_{\text{cxt},n,i} & \text{if } i \in \text{context unit} \end{cases} \tag{5}$$

$$\Delta b_{3,i} = \epsilon \sum_{n=1}^{L} \begin{cases} \delta_{\text{out},n,i} & \text{if } i \in \text{output unit} \\ \delta_{\text{cxt},n,i} & \text{if } i \in \text{context unit} \end{cases} \tag{6}$$

$$\Delta W_{21,ij} = \epsilon \sum_{n=1}^{L} \delta_{\text{hid},n,i} \begin{cases} y_{n-1,j} & \text{if } j \in \text{input unit} \\ x_{n,j} & \text{if } j \in \text{PB unit} \\ c_{n-1,j} & \text{if } j \in \text{context unit} \end{cases} \tag{7}$$

$$\Delta b_{2,i} = \epsilon \sum_{n=1}^{L} \delta_{\text{hid},n,i} \tag{8}$$

$$\Delta \mathbf{x} = k_{\text{bp}} \sum_{n=1}^{L} \delta_{\text{PB,n}}. \tag{9}$$

C. Action Recognition As Estimating PB Values

The trained RNNPB network generates different time-series based on static PB unit values. Hence, it can generate similar actions when the correct PB values are given. However, although we do not know the correct PB value in advance in most imitation tasks, we will already know the desired actions $\mathbf{y}_{\text{ref}}$. In that case, the corresponding PB values $\mathbf{x}_{\text{recog}}$ could be estimated via the BPTT process. The network parameter ψ is fixed and only the PB value is updated in this procedure, as illustrated in Fig. 3

$$\mathbf{x}_{\text{recog}} = \underset{\mathbf{x}}{\text{argmin}} \sum_{n=1}^{L} \mathbf{E}_n^{\text{out}}|_{\mathbf{y}_{\text{ref}}}. \tag{10}$$

D. Action Generation With the Given PB Values

Actions from the RNNPB model are generated through a chain of forward activity calculations. When PB unit values are given as $\mathbf{x} = [x_1, x_2]^T$, hidden unit values at the n^{th} time step

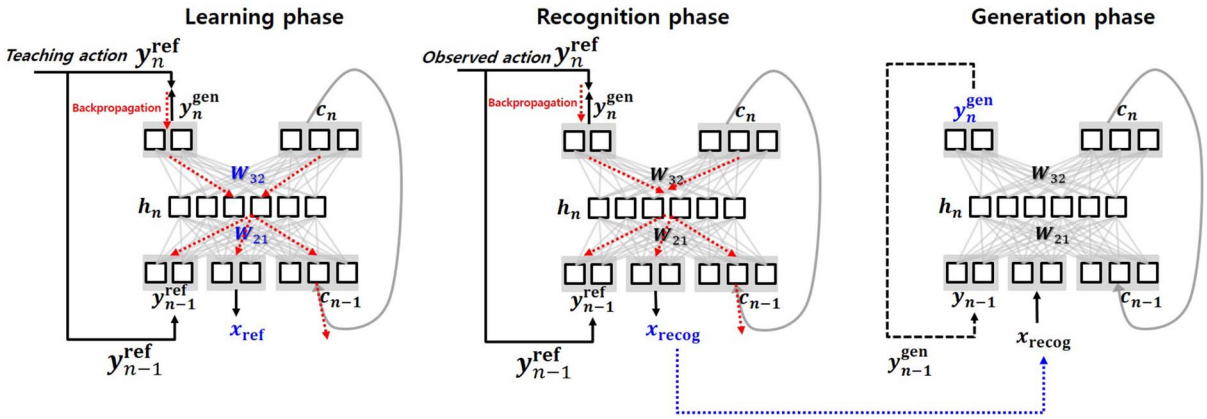


Fig. 3. Procedures for action learning, recognition, and generation. In the learning phase, differences between the desired output $\mathbf{y}_{\text{ref}}$ and generated output $\mathbf{y}_{\text{gen}}$ are back-propagated for whole time steps through the BPTT algorithm. Thus, the network parameter ψ and the corresponding PB value $\mathbf{x}_{\text{ref}}$ are calculated. In the recognition phase, the PB values $\mathbf{x}_{\text{recog}}$ are estimated from the given motor action $\mathbf{y}_{\text{ref}}$ through the BPTT algorithm, but the network parameter ψ is not updated. In the generation phase, the PB value $\mathbf{x}$ is set to a static value then the corresponding action $\mathbf{y}$ is generated.

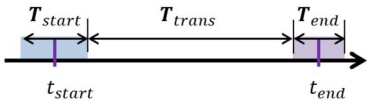


Fig. 4. Timing of motor behaviors in our tasks; all behaviors consist of three phases: 1) waiting in the initial position T_{start}; 2) transition from the initial position to the desired goal position T_{trans}; and 3) waiting in the goal position T_{end}.

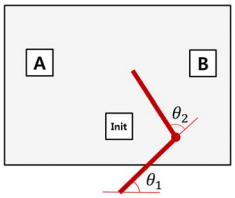


Fig. 5. Robotic arm moving from the initial position to one of two different goal positions.

$\mathbf{h}_n$ are first produced with the motor actions at the previous time step $\mathbf{y}_{n-1}$, the PB values $\mathbf{x}$, and the context unit values at the previous time step $\mathbf{c}_{n-1}$. The motor action $\mathbf{y}_n$ and context unit values $\mathbf{c}_n$ at the n^{th} time step are calculated from the hidden unit values at n^{th}. Consequently, a time series of motor actions $\mathbf{Y} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_L]$ with length L, is generated by repeating this procedure. In that case, the initial values of the context units are set to a constant value (0.5 is used in our experiments).

IV. Experimental Setup

Agents equipped with the RNNPB model can be trained with a set of goal-directed actions that consist of two different goals with different movement styles through, for example, kinesthetic teaching. When the agents experience desired motor behaviors $\mathbf{Y}_{\text{ref}}$, they first recognize the action as estimating the corresponding PB value $\mathbf{x}_{\text{recog}}$. After which they regenerate the proposed action $\mathbf{Y}_{\text{gen}}$ based on their internal knowledge. These tasks are conducted under two different experimental conditions: 1) simulation and 2) real robot environment.

A. Timing for the Motor Actions

All desired motor behaviors in this task consist of three phases as illustrated in Fig. 4. In the first phase, the robot's arm waits at the initial position for T_{start} time steps, and then moves to one of the two different goal positions for T_{trans}. Upon reaching the goal state, it waits for T_{end}. Hence, the total length of the reaching behavior is $L = T_{\text{start}} + T_{\text{trans}} + T_{\text{end}}$.

When generating actions in the RNNPB model, the context unit values are initialized as static values, so several steps are required for the agent to reach the initial position. The mid-points of $\mathbf{T}_{\text{start}}$ and $\mathbf{T}_{\text{end}}$ are, respectively, used for the measurement points t_{start} and t_{end} (see Fig. 4).

B. Virtual Two-Joint Robotic Arm

A virtual robotic arm that moves within a 2-D Euclidean space is defined for the simulation task. As illustrated in Fig. 5, the arm reaches from the initial position (marked as init) to one of the two different goal positions (A or B). Each joint ($\theta = [\theta_1, \theta_2]^T$) is able to move from 0 to 180°. Hence, the RNNPB models in this experiment have two input and output units whose activations indicate normalized joint angle values $\mathbf{y} = [y_1, y_2]$, and 60 hidden units and 40 context units.

As illustrated in Fig. 6, three different styles of movement are defined for each goal position (A and B) for two different cases. In case 1, the average of the three different types of trajectory is biased as illustrated in Fig. 6(a), whereas there is no bias for the three movement in the second case. Therefore, the robotic arm is supposed to move with three different types of trajectory like Fig. 6(b). Types 1 and 3 ($\mathbf{A_1}$, $\mathbf{A_3}$, $\mathbf{B_1}$, and $\mathbf{B_3}$) are curved trajectories, whereas type 2 ($\mathbf{A_2}$ and $\mathbf{B_2}$) are hopping movements. Hopping movements are defined by adding sinusoidal perturbations with T_{means} period and α amplitude in the transition phase. Curved trajectories have the same phases but different amplitudes in the biased case (case 1), whereas

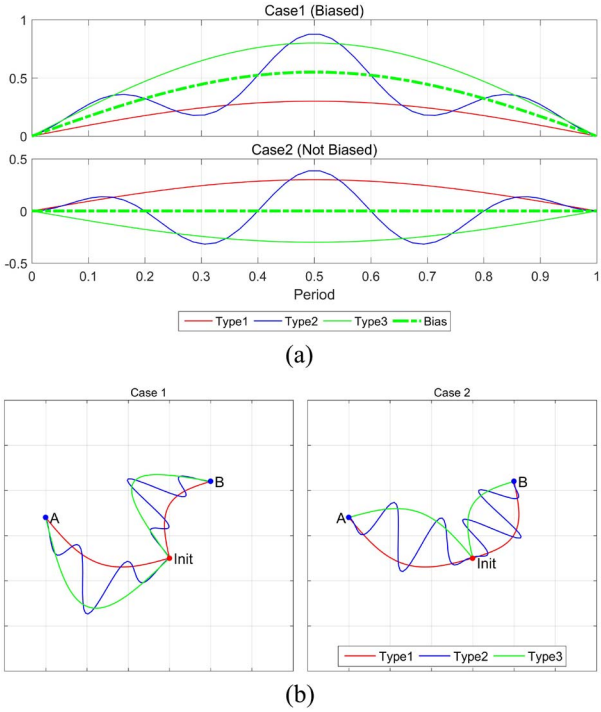


Fig. 6. (a) Three different types of trajectories from the initial position to goal positions. An average of three trajectories is biased in case 1 and is not biased in case 2. (b) Motor trajectories for the three different styles of the two different cases in the 2-D workspace of the simulation environment.

they have the opposite direction but same amplitude in the unbiased case (case 2).

In total, there are six motor behaviors for the robot to learn in each case (four straight movements and two hopping movements), denoted by $\mathbf{Y_{ref}} \in \{\mathbf{Y_{A_1}}, \mathbf{Y_{A_2}}, \mathbf{Y_{A_3}}, \mathbf{Y_{B_1}}, \mathbf{Y_{B_2}}, \mathbf{Y_{B_3}}\}$ (see Fig. 7). When the virtual agents generate their motor actions after experiencing motor behavior $\mathbf{Y_{ref}}$, the output unit values $\mathbf{y}_{n-1}$ at the $(n-1)$th time step are fed into the input unit values $\mathbf{y}_n$ at the nth time step.

C. NAO Humanoid Robot

The right arm of an NAO humanoid robot (Aldebaran Robotics, Paris, France) was used for the real robot task. Only three joints ($\theta = [\theta_1, \theta_2, \theta_3]^T$) among the five joints of the right arm were used to reduce complexity; the unused joints were fixed at specific angles. Therefore, the RNNPB networks of the agents had three input and output units as normalized values of the selected joints $\mathbf{y} = [y_1, y_2, y_3]$ with 40 hidden units and 30 context units.

Similar to the previous task, the robotic arm moves from its initial position to one of two different goal positions, respectively, denoted by A and B (see Fig. 8). The arm moves via two different means for each goal. The first is a basic movement; the robot directly reaches its arm to either goal A or B. These actions are denoted as $\mathbf{A_1}$ and $\mathbf{B_1}$, respectively. The second is a hopping movement, which has a sine wave trajectory and is denoted by either $\mathbf{A_2}$ or $\mathbf{B_2}$. Hence, the desired motor behavior $\mathbf{Y_{ref}}$ is composed of four motor patterns $\mathbf{Y_{ref}} \in \{\mathbf{Y_{A_1}}, \mathbf{Y_{A_2}}, \mathbf{Y_{B_1}}, \mathbf{Y_{B_2}}\}$.

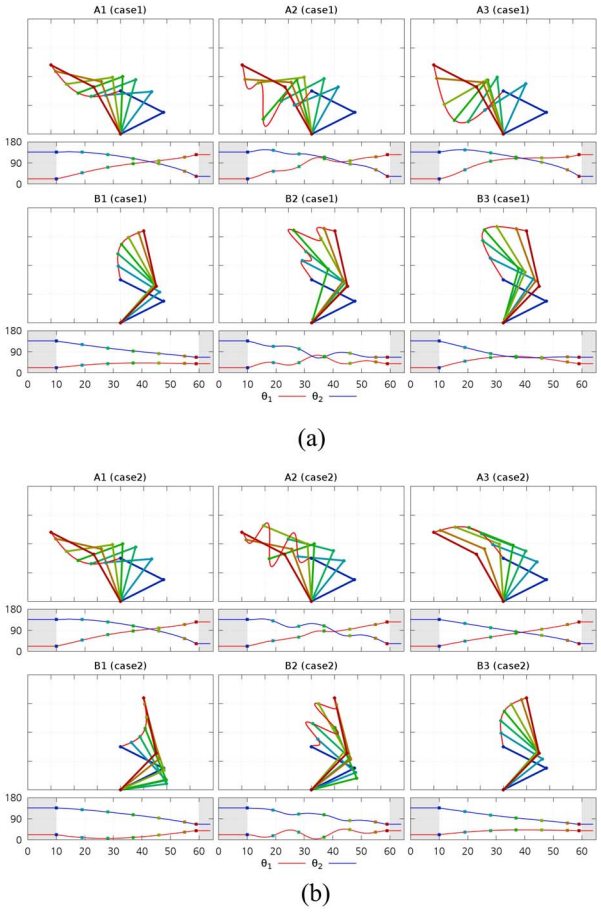


Fig. 7. Six desired movements of the robotic arm in the 2-D workspace for two goal positions in two different cases. (a) Biased movements (case 1). (b) Unbiased movements (case 2).

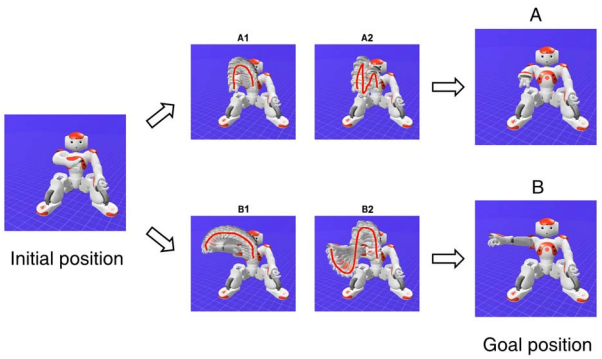


Fig. 8. Experimental setup for real robot configuration. From the initial position, the robotic arm moves to the two different goal positions with the two different means.

Unlike the simulation environment, the physical environment for a robotic experiment has unexpected factors such as sensory noise and incomplete actuator responses. Moreover, humanoid robots have more joints than the two-joint model of the simulation environment. Fortunately, the RNNPB has noise tolerance during its generation procedure. Hence, in the real robot experiment, the generation procedure is designed slightly differently to reflect unexpected factors. When the robot is generating actions from the estimated PB values $\mathbf{x}_{\text{recog}}$, the

input of the networks $\mathbf{y}_n^{(\text{net})}$ at the nth time step is determined by the weighted sum of the two values. One is produced by the network $\mathbf{y}_{n-1}^{(\text{net})}$, and the other is sensed from the state of the NAO robot $\mathbf{y}_{n-1}^{(\text{robot})}$. The weight factor a is set to 0.7 to reflect the effects of any unexpected factors more strongly

$$\mathbf{y}_n^{(\text{net})} = a \cdot \mathbf{y}_{n-1}^{(\text{robot})} + (1-a) \cdot \mathbf{y}_{n-1}^{(\text{net})}. \tag{11}$$

V. Experimental Results

The agents trained with the RNNPB model to reproduce the goal-directed actions for both virtual robotic arm and real robot are described in Section IV.

A. Simulation Result

The learning performance of the simulation agent was assessed in terms of two points. The first was whether the agent successfully reached the desired goal posture from its initial posture. The error E_{goal} was calculated by taking the average of two error values at the initial posture E_{start} and the end posture E_{end}. The two error values (E_{start} and E_{end}) were calculated as the Euclidian distance between the desired actions $\mathbf{y}_t^{\text{ref}} \in \mathbf{Y}_{\text{ref}}$ and the generated actions $\mathbf{y}_t^{\text{gen}} \in \mathbf{Y}_{\text{gen}}$ at t_{start} and t_{end}, respectively (see Fig. 4). The main reason for measuring E_{start} is that the agent should be taught how to remain in the initial position as untrained agents cannot do it

$$\begin{aligned} E_{\text{start}} &= \left\| \mathbf{y}_{t_{\text{start}}}^{\text{ref}} - \mathbf{y}_{t_{\text{start}}}^{\text{gen}} \right\| \\ E_{\text{end}} &= \left\| \mathbf{y}_{t_{\text{end}}}^{\text{ref}} - \mathbf{y}_{t_{\text{end}}}^{\text{gen}} \right\| \\ E_{\text{goal}} &= \frac{E_{\text{start}} + E_{\text{end}}}{2}. \end{aligned} \tag{12}$$

The second point is how well the agent traces the movement style (i.e., the means of action). An error in the shape of the trajectory E_{shape} was defined as the averaged error over T_{trans} time steps, where the Euclidian distance between the desired actions $\mathbf{y}_t^{\text{ref}}$ and generated actions $\mathbf{y}_t^{\text{gen}}$ was applied

$$E_{\text{shape}} = \frac{1}{T_{\text{trans}}} \sum_{t \in \mathcal{T}_{\text{trans}}} \left\| \mathbf{y}_t^{\text{ref}} - \mathbf{y}_t^{\text{gen}} \right\|. \tag{13}$$

Fig. 9 shows the transitions of E_{goal} and E_{shape} over learning for the two difference cases. They are averages of 100 RNNPBs with different initial parameters ψ^0. Overall, the error values decreased as the learning progressed. The error for the end point of the generated motor trajectories E_{goal} became smaller than the error of the trajectories' shape E_{shape} when the agent had been sufficiently trained. As the main objective of the task in our experiment was to move the arm into the correct position, 36 of the 100 networks (case 1) and 31 of 100 networks (case 2) trained with different initial network parameters were chosen as successfully trained networks for a threshold based on a goal error $E_{\text{goal}} < 0.05$. All of the 36 (case 1) and 31 (case 2) successfully trained networks showed similar characteristics that sufficiently supported our hypothesis. Due to space limitations,

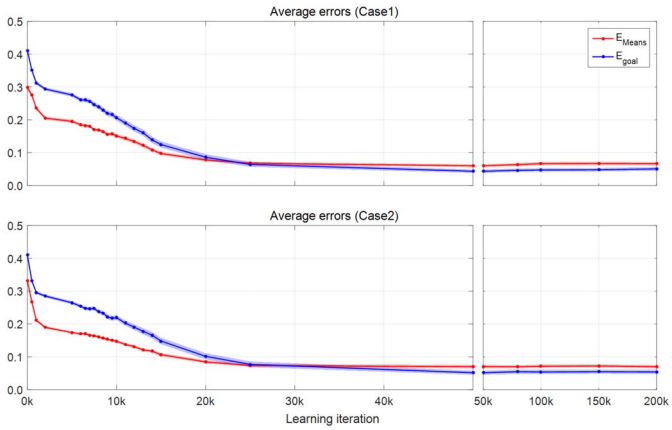


Fig. 9. Transition of errors concerning goal positions of trajectories E_{goal} and shapes of trajectories E_{means}. The two curves plot the average of 100 networks with different initial parameters for the two different cases.

one of the successfully trained networks was selected for the two cases (cases 1 and 2) for further analysis and visualization.

The errors in the shape of trajectories E_{shape} for all possible PB values were examined to investigate the PB space's developmental dynamics. Additionally, based on recognized PB values $\mathbf{x}_{\text{ref}}$ for each reference action $\mathbf{Y}_{\text{ref}}$, corresponding actions $\mathbf{Y}_{\text{gen}}$ were generated to examine the agent's action recognition abilities. Three iteration points (0, 8, and 200 k) were chosen based on the dynamical self-organization of the PB space.

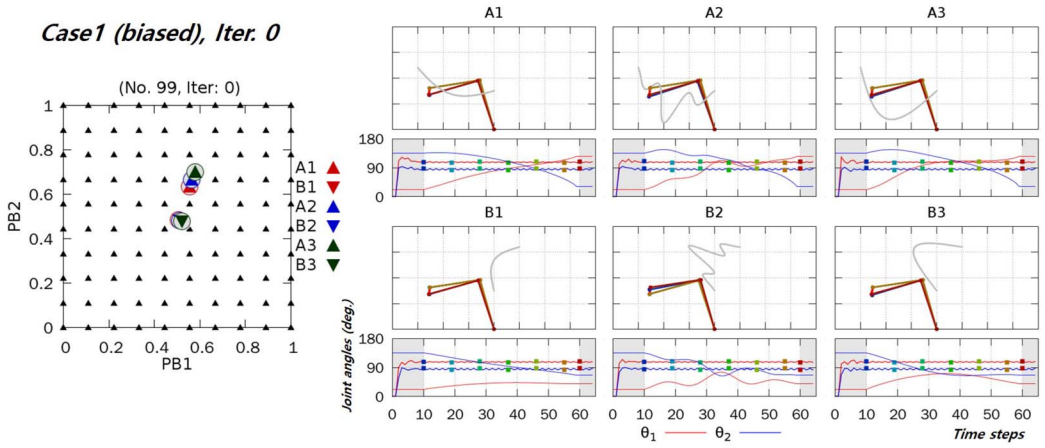
Figs. 10 and 11 represent the result of the selected network for each case. The direction and the color of the triangular markers on the left of these figures indicate which type of action $\mathbf{S}_{\mathbf{x}}$ has a minimum error value for the corresponding PB values. The size of the triangular markers is inversely proportional to the error amount E_{shape}. Hence, a larger marker implies that an agent could generate an action with a smaller error by using the corresponding PB values at the marker position

$$\mathbf{S}_{\mathbf{x}} = \underset{\mathbf{S} \in \{A_1, \ldots, B_3\}}{\operatorname{argmin}} \left\{ E_{\text{shape}} \middle| \mathbf{Y}_{\text{ref}} = \mathbf{Y}_{\mathbf{S}} \right\}. \tag{14}$$

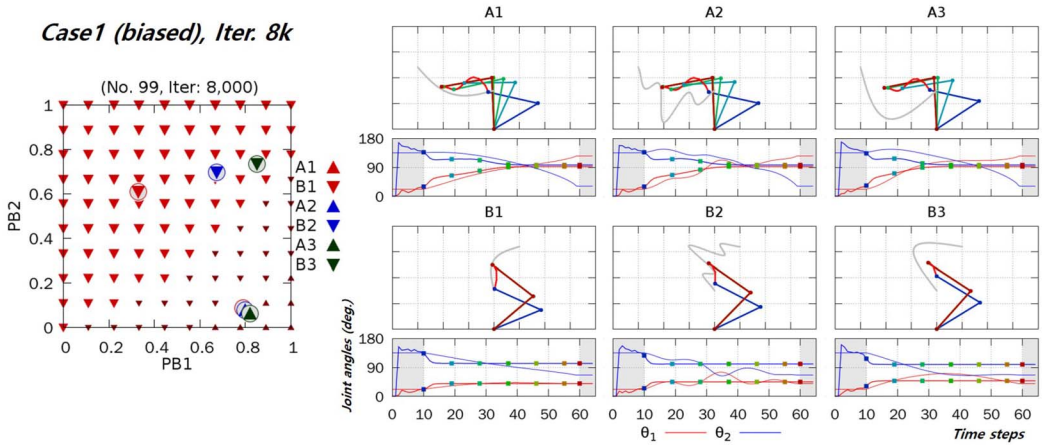
Recognized PB values for each reference behavior $\mathbf{x}_{\text{recog}}$ are depicted as circles with triangular markers. When the agent generates six desired motor behaviors based on the recognized PB values $\mathbf{x}_{\text{recog}}$, the trajectories of the robotic-arm and their joint angles in the simulation environment are visualized in Figs. 10(right) and 11(right).

The results show that the agent gradually improved its ability to reproduce the reference actions as its experience increased. Meanwhile, the PB space gradually became self-organized to represent the actions. When the agent had no experience of the reference behaviors (0 iterations), it did not produce the desired actions due to the undifferentiated PB values [see Figs. 10(a) and 11(a)]. The agent did not reach the initial position at that time.

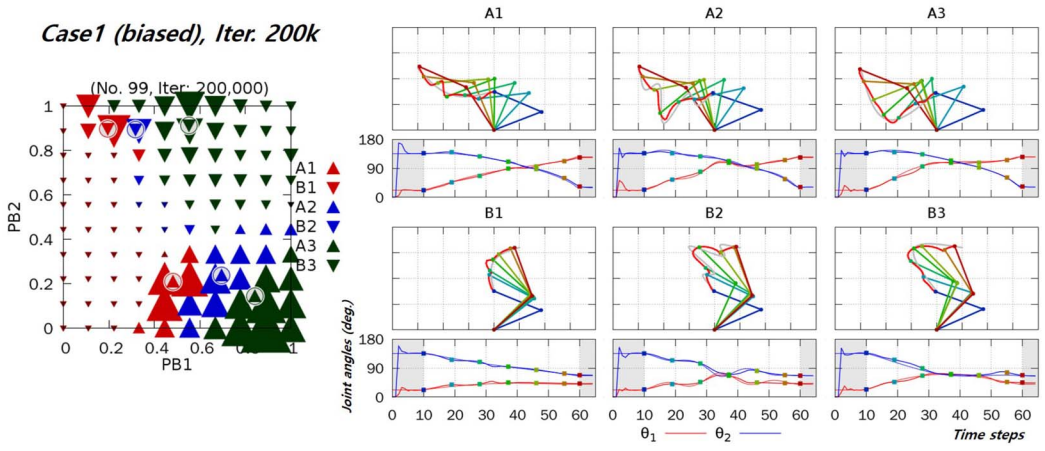
When the agent was trained with 8000 iterations, it moved toward the desired goal positions from the initial position, but errors still existed. The PB spaces



(a)



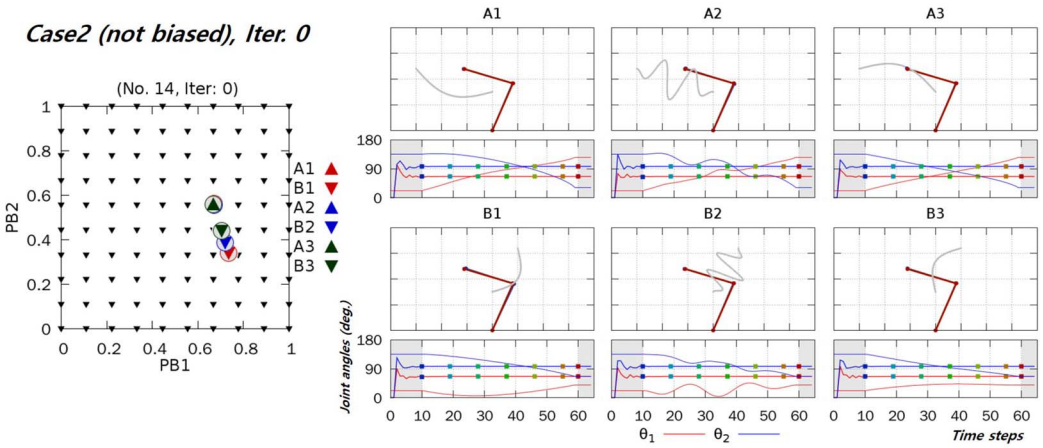
(b)



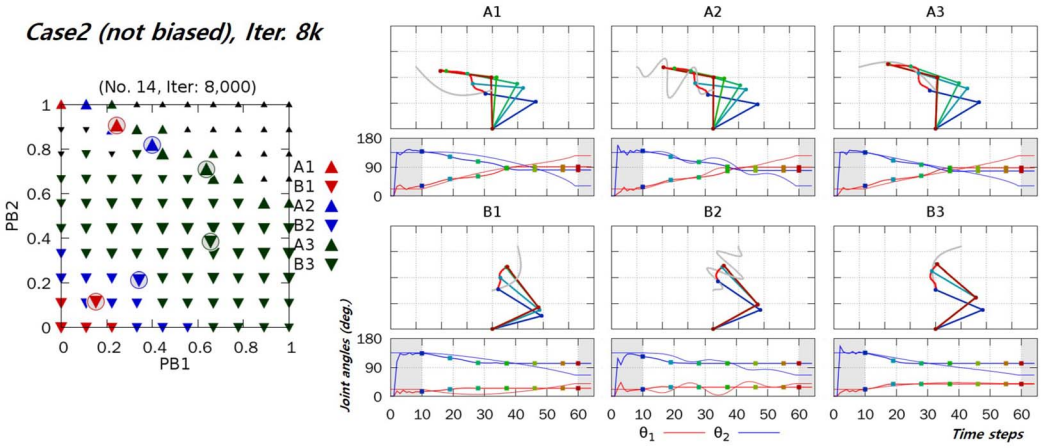
(c)

Fig. 10. (Case 1: biased) dynamics of a PB space and the results of action generation; the left side of the figure illustrates which reference actions (from A_1 to B_3) have minimal error in the PB space. The direction and the color of the triangular markers, respectively, indicate the goal and style of movement. The size of the markers is inversely proportional to the size of the error E_{shape}: the larger the marker, the smaller the error. Recognized PB values $\mathbf{x}_{\text{recog}}$ are illustrated as circles with triangular markers inside. The right side of the figure represents the actions generated by the agent and its joint angles. The figures of joint angles represent $\mathbf{Y}_{\text{gen}}$ (thick lines) for all reference actions $\mathbf{Y}_{\text{ref}}$ (thin lines) in the time domain. The red and blue lines are, respectively, the first and second joint angles.

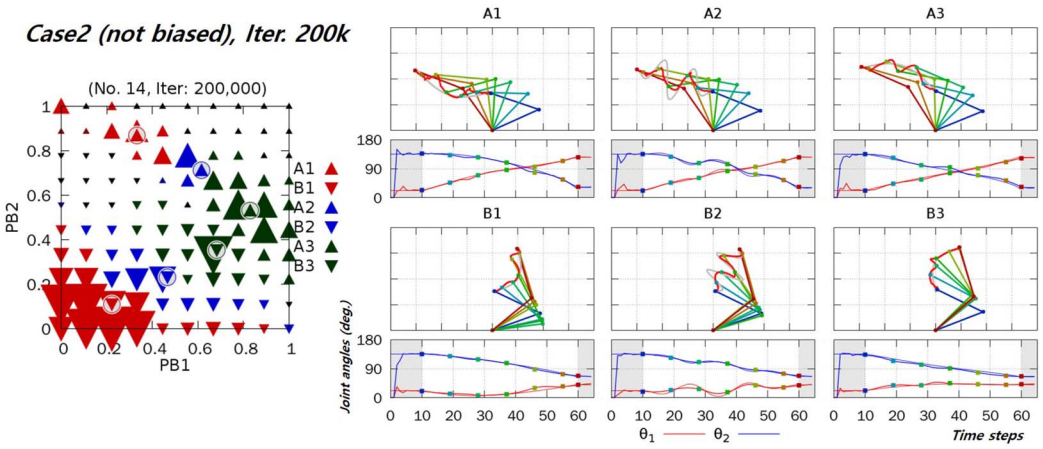
were well separated by the two different goal positions (A and B), but not for the shapes of the trajectories. Thus, the shapes of the generated trajectories with same goal positions were similar [see Figs. 10(b) and 11(b)]. When the agent was fully trained, it successfully generated six actions that fit both end points and shapes of



(a)



(b)



(c)

Fig. 11. (Case 2: no bias) the dynamics of the PB space and results of action generation; the left side of the figure illustrates which reference actions (from A_1 to B_3) have a minimal error in the PB space. The direction and the color of the triangular markers, respectively, indicate the goal and the style of movement. The size of the markers is inversely proportional to the amount of error E_{shape}: the larger the marker, the smaller the error. Recognized PB values $\mathbf{x}_{\text{recog}}$ are illustrated as circles with triangular markers inside. The right side of the figure represents the actions generated by the agent and its joint angles. The figures of joint angles represent $\mathbf{Y}_{\text{gen}}$ (thick lines) for all reference actions $\mathbf{Y}_{\text{ref}}$ (thin lines) in the time domain. The red and blue lines are, respectively, the first and second joint angles.

the trajectories. The well-organized PB values enabled the agent to discriminate actions [see Figs. 10(c) and 11(c)]. Consequently, a tendency toward phased learning (i.e., first learning the goal of the actions and then the means) was found through development with the results illustrated in Figs. 10 and 11.

B. Result of the NAO Humanoid Robot

$\mathbf{E}_{\text{goal}}$ and $\mathbf{E}_{\text{shape}}$ were analyzed in a similar manner to the simulation environment for the 100 networks trained with the different initial conditions on the robotic environment. Three iteration points (0, 3.5, and 200 k) were chosen to show the developmental changes of action generations. Based on the same error threshold $\mathbf{E}_{\text{goal}} < 0.05$, six of 100 networks with different initial conditions were selected as successfully trained networks. All six successfully trained networks also showed similar properties. A real NAO humanoid robot was used to generate actions in real time for the three iteration points of one of the six converged networks. Fig. 12 illustrates the results of the NAO robot experiment with the three iteration points. Fig. 12(left) indicates the results of PB space analysis for one of the five successfully trained networks. Unlike the previous task in the simulation environment, the color of triangular markers indicates the goal of the motor behaviors. Fig. 12(right) illustrates the trajectories of the NAO humanoid robot's right hand.

When the agent was not yet trained (i.e., 0 iterations), the robot did not generate all four of the actions [see Fig. 12(a)]. Furthermore, it failed to move its arm into the initial position, and the PB space was not yet organized. When the agent was trained for 3500 iterations [see Fig. 12(b)], the PB space was separated for the two different goals *A* and *B*. Even though we see a separation in the shape of the trajectories in the PB space, it was too small to differentiate the output. Hence, the robot successfully reached its arm to the goal position, even though its trajectories showed similar shapes. When the agent was fully trained, the PB space [see Fig. 12(c)] was well organized, and both goals and shape of trajectories were separated. Hence, the robot successfully reproduced all actions with respect to both goals and shape of trajectories as expected. Thus, a staged development phenomenon, in which the goal of actions is achieved before the means of actions was found, despite the influence of environmental noise and the movement complexity.

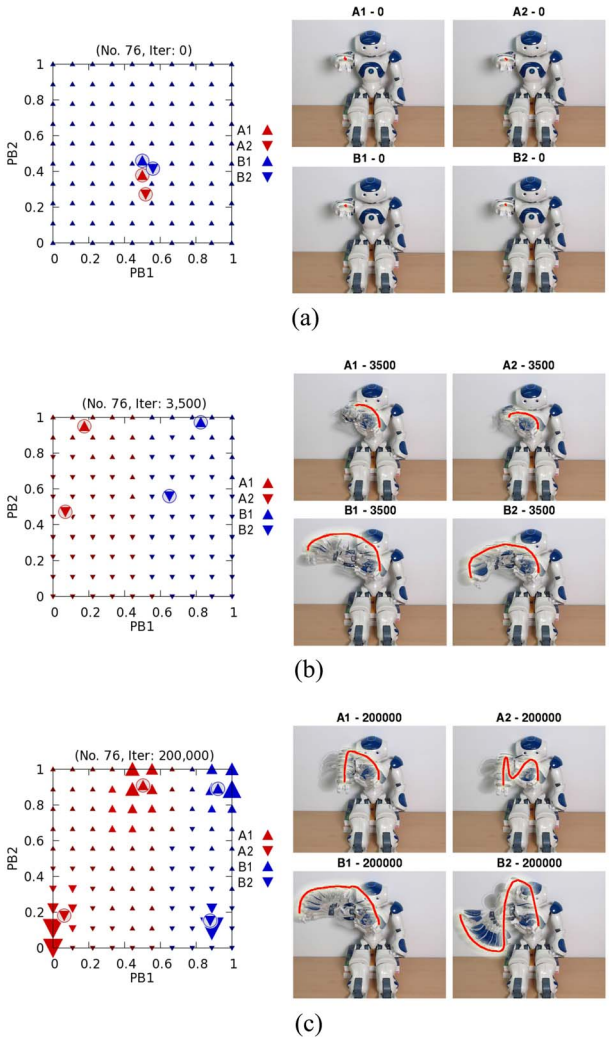


Fig. 12. Dynamics of the PB space and the results of action generation for real robot tasks. The left side of the figure illustrates which reference actions (from A_1 to B_2) have minimal error in the PB space. Unlike the previous figure, the color of triangular markers indicates the goal, and the direction implies the movement style. The size of the markers is inversely proportional to the amount of error E_{shape}; the larger the marker, the smaller the error. Recognized PB values $\mathbf{x}_{\text{recog}}$ are illustrated as circles with triangular markers inside. The right side of the figure represents the generated motor actions of the NAO humanoid robot. Actions of the agent trained for (a) 0 iterations, (b) 3500 iterations, and (c) 200 000 iterations.

VI. Discussion

We analyzed the developmental dynamics of the RNNPB model with two robot imitation tasks of goal-directed motor behaviors in this paper. The results illustrated in Fig. 9 indicate that the overall training error decreased during training, which means that the RNNPB models were trained well. However, when a tight convergence condition ($E_{\text{goal}} < 0.05$) was applied, only 36 of the 100 networks in case 1 and 31 of the 100 networks in case 2 (in the case of the real robot task, 6 of the 100 networks) were selected. When we investigated the networks that failed to converge in the simulation task, most generated desired motor behaviors well except for one or two actions. This phenomenon is related to an optimization issue in the learning procedure. Local minima and overfitting issues based on the initial condition might have affected the learning procedure, because we used a naive BPTT method with restricted network sizes, especially with regard to the number of the PB units, to visualize the PB space on a 2-D state space. Hence, the number of converged networks could be increased if an advanced optimization technique was used or if the number of the PB units was increased. Despite the optimization issue, the results from the converged networks are sufficient to support our hypothesis, because all converged networks showed similar characteristics. Furthermore, almost all of the trained networks that contained unconverged networks showed a staged separation of the PB space in the case of the simulation task.

According to [38] on the PB units of the RNNPB model, two behaviors with similar properties tended to appear closer in the PB space. Moreover, the RNNPB model can generate and recognize slightly different behaviors as well as trained behaviors. Therefore, it could be expected that trained behaviors would appear as a grouped area of the PB space. Using a novel visualization technique based on the generation error

of trained behaviors, this continuity of the PB space could be found in our result. The results with visualized PB spaces as illustrated on the left side of Figs. 10–12 provide interesting insights concerning the staged development of the PB space. It was found that the error distribution and recognized PB values in the PB space were first separated for the goal property of actions then separated for the means in both virtual robotic arm and NAO robot task. Interestingly, this separation appears abruptly during the learning iteration similar to the nonlinear improvement of human learning. The main reason for this phenomenon would be related to the error minimization process.

The results concerning the generated actions from the virtual robotic arm [see Figs. 10(right) and 11(right)] support this finding. In the middle of learning, the trajectories generated by agents were distinguishable in terms of the goal positions but not the shape, and the generated trajectories were distinguishable for both goal and shape when the agent was sufficiently trained. The results of the NAO robot [Fig. 12(right)] show the robustness of our computational model. The numbers of input and output units were also increased, whereas the number of PB units was fixed at 2. These increased complexities and unexpected factors (e.g., sensory noise and incomplete actuator responses) could have affected the task. Nevertheless, the result showed a staged development similar to that of the simulation task. Therefore, as we predicted in our hypothesis, the results of our experiments demonstrate human-like developmental processes for goal-directed behaviors.

Then, how did this human-like developmental process appear in our computational model? We measured the error of the generated trajectories' goal positions E_{goal} from a computational perspective as averaged errors at the beginning and end of transitions, and errors in the shape of the trajectories E_{shape} as errors during transitions. The RNNPB model was learned by updating the network parameter ψ while minimizing errors by (3). Our results showed that (3) contained the properties of E_{goal} and E_{shape} even though E_{goal} and E_{shape} were not directly described in (3). The major action in the given task is the transition of the body posture, which accompanies a relatively large change in joint angle compared to the movement style. The behaviors generated from immature agents are not suitable for either the initial posture or the goal posture in the early stages of the learning process. In this case, E_{goal} is relatively larger than E_{shape}. E_{goal} is already sufficiently reduced in the middle of learning, and the network is updated to allow the distinguishing of the different movement styles by reducing E_{shape} afterwards. Thus, this relative aspect during the error minimization process has been able to create a hierarchy of goal-directed motor behaviors with multiple subgoal attributes when the amount of motion required by the primary goal is always greater than that of the secondary goal. In addition, this error minimization process could be used to explain cognitive learning as a concept of predictive learning [42].

This hierarchy of goal-directed behaviors has also appeared in empirical studies on infant development [24], [25], in which younger infants tended to ignore less significant goals and imitate a primary goal well. Meanwhile, adults and children could achieve both primary goals and subgoals. This means that the goals of an action are represented hierarchically in infants, and infants selectively imitate them from one with a higher priority to the other with a lower priority. Although the internal mechanisms of such nonlinear staged development called a U-shape change remain unknown, computational modeling studies have been conducted to explain them [29], [30]. Moreover, we found that more goals are hierarchically represented even though we only defined two goals, and the time scale seems to be a key feature in determining the hierarchy. The two goals we defined as follows.

1) The end state of an action (higher priority).
2) The paths to reach the end state (lower priority). This goal can be divided into two.
    a) The average path required to achieve the end state (large time scale).
    b) The individual paths required to achieve the end state (short time scale).

Thus, developmental learning was observed for 1) → a) → b). We found that a) had a larger time scale than b), and its time scale seems as large as that of 1). Additionally, this means that if a demonstrator shows biased actions many times to infants, the infants would be interested in both bias and actions' end state. Our robotic experiments provide additional insights into this unknown internal mechanism based on an error-based developmental process for goal-directed imitation.

However, our experiments were conducted under simple experimental conditions due to optimization problems and network capacity issues. Additionally, the number of PB units was fixed to visualize developmental dynamics of the PB space in 2-D space. In this status, the timing condition of keeping the robot at its initial and goal positions helped the networks learn six desired goal-directed behaviors despite their limitations. Moreover, this timing condition itself contained the hierarchical goal property. Remaining at the end position encouraged the networks to learn the primary goal: reaching the arm to the goal position. The trajectory shapes make the networks learn the secondary goal: matching the trajectory shapes. This goal property of the actions with timing condition restricts the environment of our experiment, and that could be criticized. Unfortunately, this issue could not be clearly solved due to our model's limitations. Nevertheless, our experiments in the current conditions are still meaningful as they provide better insights into infant development for goal-directed action imitations.

Additionally, could the staged-developmental results in which the actions' goals were learned prior to the means appear in more complex experimental conditions? For instance, if there were an obstacle that the robot should avoid, the trajectories of all of reaching behaviors would become more complex and nonlinear. A new task might then have three goals.

1) Reaching the arm to the correct target position.
2) Avoiding the obstacle.
3) Matching the trajectory shapes.

When a network is not sufficiently trained, an error on an averaged line from the initial posture to the target posture will dominantly appear, as in our simulation results for the biased case (case 1). Hence, the agent may ignore an obstacle but

successfully reach its arm toward the goal position. After the primary goal is learned, an error in the trajectory shape will dominantly appear. However, an error in avoiding an obstacle may not be distinguishable from the error on the trajectory shape, because the error is only measured using proprioceptive information. Thus, a sufficiently trained agent will be able to generate actions to achieve all three goals. However, the agent is only controlled by joint posture without any sensory feedback. Hence, the agent cannot adapt when obstacle conditions change between demonstrations. An agent equipped with a dynamic system that uses differential equations such as DMP [33], [34] could resolve this issue more effectively.

However, the robotic agent in this paper only learns motor actions and omits visual properties while assuming that the correspondence problem [20], [21] is solved without the agent. Hence, the target actions were demonstrated by learning by doing, such as by grabbing and moving the agent's arm. Even though the correspondence problems of visuomotor coordination have not been dealt with in this paper, it is possible to explain developmental changes of what to imitate of the learning process of RNNPB based on our finding that the remarkable aspects of motor actions were learned first. Similarly, Castañeda *et al.* [43] and Rozo *et al.* [44] proposed a robot imitation learning approach as solving the issue of what to imitate with force-based manipulation without the visual data.

A gap still exists between our experimental data and the findings from developmental studies, because of our restricted experimental conditions. The agent with the RNNPB in this paper was trained with BPTT algorithm for 0 to 200k iterations, whereas humans could imitate actions after fewer trials of observation. However, the number of learning iterations in computational modeling is related to learning methods and hyperparameters such as the learning rate and optimization algorithm. Therefore, the iteration number itself is not directly mapped to human data, but the tendency or shape of the curve might be related to human data. Additionally, biological velocity profiles are an important factor when humans imitate actions [45]. For example, biological motion changes velocity smoothly when a peak velocity appears near in the midpoint of a demonstration, whereas nonbiological motion moves at constant velocity. However, the robot was only controlled by posture without considering velocity in this paper, and thus the velocity profile was not modeled. Another type of computational model such as a continuous time RNN or DMP that can model dynamics would be proper for modeling biological velocity profiles in robot imitation learning tasks.

VII. Conclusion

Robot experiments for imitating goal-directed motor behaviors were conducted in this paper. The developmental dynamics of the RNNPB model were thoroughly analyzed with a visualization technique for the PB space and the robots' actions were generated via simulation and in a real-world environment. The experimental results showed that the primary goal (i.e., reaching the arm to the correct position) was learned in the middle of the learning procedure, and the means (i.e., matching the trajectory shape) was achieved once the agent was sufficiently trained. The main contribution of this paper is that it provides novel insights into the relationship between the nature of neural network models, which involves a shift of what to imitate from a major to a minor property of motor actions and the human cognitive developmental process.

Similar to the experimental setup of [14], additional sensory information such as visual or auditory signals could be considered to design new error measures. Such a case could allow for the better representation of more complex behaviors with multiple subgoals such as obstacle avoidance. Moreover, additional interesting phenomena discovered in developmental studies such as the transition of the primary goal property of goal-directed actions based on sensory signals could be modeled in further research.

References

[1] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends Cogn. Sci.*, vol. 6, no. 11, pp. 481–487, 2002.

[2] D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan, "Perspectives and problems in motor learning," *Trends Cogn. Sci.*, vol. 5, no. 11, pp. 487–494, 2001.

[3] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robot. Auton. Syst.*, vol. 37, nos. 2–3, pp. 185–193, 2001.

[4] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey," *Connection Sci.*, vol. 15, no. 4, pp. 151–190, 2003.

[5] P. Bakker and Y. Kuniyoshi, "Robot see, robot do: An overview of robot imitation," in *Proc. Workshop Learn. Robots Animals AISB*, Brighton, U.K., 1996, pp. 3–11.

[6] C. Breazeal and B. Scassellati, "Challenges in building robots that imitate people," in *Imitation in Animals and Artifacts*. Cambridge, MA, USA: MIT Press, 2002, pp. 363–390.

[7] A. Billard, Y. Epars, S. Calinon, S. Schaal, and G. Cheng, "Discovering optimal imitation strategies," *Robot. Auton. Syst.*, vol. 47, nos. 2–3, pp. 69–77, 2004.

[8] E. I. Barakova and D. Vanderelst, "From spreading of behavior to dyadic interaction—A robot learns what to imitate," *Int. J. Intell. Syst.*, vol. 26, no. 3, pp. 228–245, 2011.

[9] M. Carpenter and J. Call, *The Question of 'What to Imitate': Inferring Goals and Intentions From Demonstrations*. New York, NY, USA: Cambridge Univ. Press, 2007, pp. 135–151.

[10] Y. Mohammad and T. Nishdia, "Self-initiated imitation learning. Discovering what to imitate," in *Proc. IEEE 12th Int. Conf. Control Autom. Syst. (ICCAS)*, 2012, pp. 726–732.

[11] K. Lee, Y. Su, T.-K. Kim, and Y. Demiris, "A syntactic approach to robot imitation learning using probabilistic activity grammars," *Robot. Auton. Syst.*, vol. 61, no. 12, pp. 1323–1334, 2013.

[12] D. Ognibene, Y. Wu, K. Lee, and Y. Demiris, "Hierarchies for embodied action perception," in *Computational and Robotic Models of the Hierarchical Organization of Behavior*, G. Baldassarre and M. Mirolli, Eds. Heidelberg, Germany: Springer, 2013, pp. 81–98.

[13] B. Michini, M. Cutler, and J. P. How, "Scalable reward learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, 2013, pp. 303–308.

[14] M. Carpenter, N. Akhtar, and M. Tomasello, "Fourteen-through 18-month-old infants differentially imitate intentional and accidental actions," *Infant Behav. Develop.*, vol. 21, no. 2, pp. 315–330, 1998.

[15] B. Elsner, "Infants' imitation of goal-directed actions: The role of movements and action effects," *Acta Psychol.*, vol. 124, no. 1, pp. 44–59, 2007.

[16] G. Gergely, "What should a robot learn from an infant? Mechanisms of action interpretation and observational learning in infancy," *Connection Sci.*, vol. 15, no. 4, pp. 191–209, 2003.

[17] A. N. Meltzoff, "Understanding the intentions of others: Re-enactment of intended acts by 18-month-old children," *Develop. Psychol.*, vol. 31, no. 5, pp. 838–850, 1995.

[18] A. Wohlschläger, M. Gattis, and H. Bekkering, "Action generation and action perception in imitation: An instance of the ideomotor principle," *Philosoph. Trans. Roy. Soc. London B Biol. Sci.*, vol. 358, no. 1431, pp. 501–515, Mar. 2003. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1693138&tool=pmcentrez&rendertype=abstract

[19] J. Loucks and A. N. Meltzoff, "Goals influence memory and imitation for dynamic human action in 36-month-old children," *Scandinavian J. Psychol.*, vol. 54, no. 1, pp. 41–50, 2013.

[20] M. Brass and C. Heyes, "Imitation: Is cognitive neuroscience solving the correspondence problem?" *Trends Cogn. Sci.*, vol. 9, no. 10, pp. 489–495, 2005.

[21] C. Heyes, "Evolution, development and intentional control of imitation," *Philosoph. Trans. Roy. Soc. B Biol. Sci.*, vol. 364, no. 1528, pp. 2293–2298, 2009. [Online]. Available: http://rstb.royalsocietypublishing.org/content/364/1528/2293

[22] J. M. Kilner, K. J. Friston, and C. D. Frith, "The mirror-neuron system: A Bayesian perspective," *Neuroreport*, vol. 18, no. 6, pp. 619–623, 2007.

[23] Y. Nagai and K. J. Rohlfing, "Parental action modification highlighting the goal versus the means," in *Proc. 7th IEEE Int. Conf. Develop. Learn. (ICDL)*, Monterey, CA, USA, Aug. 2008, pp. 1–6.

[24] H. Bekkering, A. Wohlschläger, and M. Gattis, "Imitation of gestures in children is goal-directed," *Quart. J. Exp. Psychol. A*, vol. 53, no. 1, pp. 153–164, 2000.

[25] M. Carpenter, J. Call, and M. Tomasello, "Twelve-and 18-month-olds copy actions in terms of goals," *Develop. Sci.*, vol. 8, no. 1, pp. F13–F20, 2005.

[26] M. Bowerman, "Starting to talk worse: Clues to language acquisition from children's late speech errors," in *U Shaped Behavioral Growth*. New York, NY, USA: Academic Press, 1982, pp. 101–145.

[27] E. W. Bushnell, "The decline of visually guided reaching during infancy," *Infant Behav. Develop.*, vol. 8, no. 2, pp. 139–155, 1985.

[28] N. A. Taatgen and J. R. Anderson, "Why do children learn to say 'broke'? A model of learning the past tense without feedback," *Cognition*, vol. 86, no. 2, pp. 123–155, 2002.

[29] L. Carlucci, J. Case, S. Jain, and F. Stephan, "Results on memory-limited u-shaped learning," *Inf. Comput.*, vol. 205, no. 10, pp. 1551–1573, 2007.

[30] L. Gershkoff-Stowe and E. Thelen, "U-shaped changes in behavior: A dynamic systems perspective," *J. Cogn. Develop.*, vol. 5, no. 1, pp. 11–36, 2004.

[31] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 299–304.

[32] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[33] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Kobe, Japan, 2009, pp. 2587–2592.

[34] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning stylistic dynamic movement primitives from multiple demonstrations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Taipei, Taiwan, 2010, pp. 1277–1283.

[35] J.-C. Park, J. H. Lim, H. Choi, and D.-S. Kim, "Predictive coding strategies for developmental neurorobotics," *Front. Psychol.*, vol. 3, no. 134, 2012. [Online]. Available: https://dx.doi.org/10.3389%2Ffpsyg.2012.00134

[36] J. Tani, M. Ito, and Y. Sugita, "Self-organization of distributedly represented multiple behavior schemata in a mirror system: Reviews of robot experiments using RNNPB," *Neural Netw.*, vol. 17, nos. 8–9, pp. 1273–1289, 2004.

[37] R. Yokoya, T. Ogata, J. Tani, K. Komatani, and H. G. Okuno, "Experience based imitation using RNNPB," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 2006, pp. 3669–3674.

[38] M. Ito and J. Tani, "Generalization in learning multiple temporal patterns using RNNPB," in *Proc. Neural Inf. Process. 11th Int. Conf. (ICONIP)*, Calcutta, India, 2004, pp. 592–598.

[39] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[40] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: A humanoid robot experiment," *PLoS Comput. Biol.*, vol. 4, no. 11, 2008, Art. no. e1000220.

[41] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential network," in *Proc. 8th Annu. Conf. Cogn. Sci. Soc.*, 1986, pp. 531–546.

[42] Y. Nagai and M. Asada, "Predictive learning of sensorimotor information as a key for cognitive development," in *Proc. IROS Workshop Sensorimotor Contingencies Robot.*, Hamburg, Germany, Oct. 2015.

[43] L. R. Castañeda, P. J. Schlegl, and C. Torras, "Sharpening haptic inputs for teaching a manipulation skill to a robot," in *Proc. 1st IEEE Int. Conf. Appl. Bionics Biomech.*, Venice, Italy, 2010, pp. 331–340.

[44] 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, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11370-012-0128-9

[45] J. Kilner, A. F. D. C. Hamilton, and S.-J. Blakemore, "Interference effect of observed human movement on action is due to velocity profile of biological motion," *Soc. Neurosci.*, vol. 2, nos. 3–4, pp. 158–166, 2007.

Jun-Cheol Park is currently pursuing the Ph.D. degree with the Brain Reverse Engineering and Imaging Laboratory, Korea Advanced Institute of Science and Technology, Daejeon, South Korea.

His current research interests include motor action learning with artificial neural networks and human action recognition with deep learning.

Dae-Shik Kim received the Ph.D. degree from the Max-Planck-Institute for Brain Research, Frankfurt, Germany, in 1992.

He is a tenured Full Professor with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he heads the Brain Reverse Engineering and Imaging Laboratory. He was a Post-Doctoral Research Fellow with the Massachusetts Institute of Technology, Cambridge, MA, USA, and a Frontier Researcher with RIKEN, Tokyo, Japan, for two years. He was an Assistant Professor with the University of Minnesota, Minneapolis, MN, USA, from 1999 to 2003. In 2003, he was appointed as an Associate Professor and the Director of the Center for Biomedical Imaging, Boston University, Boston, MA, USA. His current research interests include systems, developmental, and computational neurosciences, functional and connectivity mapping of the human brain, developmental robotics, and diffusion tensor imaging.

Yukie Nagai received the master's degree from Aoyama Gakuin University, Tokyo, Japan, in 1999, and the Ph.D. degree from Osaka University, Osaka, Japan, in 2004, both in engineering.

She was a Post-Doctoral Researcher with the National Institute of Information and Communications Technology, Kyoto, Japan, from 2004 to 2006, and Bielefeld University, Bielefeld, Germany, from 2006 to 2009, where she was also with the Research Institute for Cognition and Robotics. She has then been a specially appointed Associate Professor with Osaka University since 2009, and a Visiting Professor with Bielefeld University since 2017. Since 2012, she has been the Project Leader of MEXT Grant-in-Aid for Scientific Research on Innovative Areas entitled *Computational Modeling of Social Cognitive Development and Design of Assistance Systems for Developmental Disorders*. Since 2016, she has also been the Project Leader of JST CREST entitled *Cognitive Mirroring: Assisting People With Developmental Disorders by Means of Self-Understanding and Social Sharing of Cognitive Processes*. Her current research interests include computational modeling of human cognitive functions such as self-other cognition, imitation, and joint attention, and design of assistant systems for developmental disorders.