

Codevelopmental learning between human and humanoid robot using a dynamic neural network model

Jun Tani, Ryunosuke Nishimoto, Jun Namikawa and Masato Ito

Abstract—The paper examines characteristics of interactive learning between human tutors and a robot having a dynamic neural network model which is inspired by human parietal cortex functions. A humanoid robot, with a recurrent neural network that has a hierarchical structure, learns to manipulate objects. Robots learn tasks in repeated self-trials with the assistance of human interaction which provides physical guidance until tasks are mastered and learning is consolidated within neural networks. Experimental results and the analyses showed that 1) codevelopmental shaping of task behaviors stems from interactions between the robot and tutor, 2) dynamic structures for articulating and sequencing of behavior primitives are self-organized in the hierarchically organized network, and 3) such structures can afford both generalization and context-dependency in generating skilled behaviors.

Index Terms—Humanoid robot, CTRNN, development learning, compositionality.

I. INTRODUCTION

The tasks required of robots are becoming increasingly more complex. As a result, robot designers must develop robotic programs that enable complex task execution. Just imagine what would be needed for a humanoid robot to grab an object it sees. Conventionally, two cameras, placed on the robot's "head" would determine an object's global position stereo-optically [1] within three dimensional space. Based on that information, the trajectories of arm and hand movements are computed so that the object can be grasped. The corresponding trajectory of the joint coordinate is then computed using inverse kinematics [2]. These computations are not trivial, because all possible, collision-free trajectories that meet optimal criteria are determined in a combinatorial manner [3].

However, it seems that in our everyday behaviors limb movements are elegantly coordinated with little conscious effort. Such coordinated movements in humans are likely to be the result of inherent constraints from the innate structures determined genetically as well as those structures that emerge through learning that follows birth. The connections between the muscular skeletal structures and basic neuronal circuitry provide the basis for all possible patterns of movement. Synaptic changes and re-wiring of neuronal circuits resulting from everyday sensory-motor experiences and activity then refine our basic movement patterns over time, with practice.

J. Tani, R. Nishimoto and J. Namikawa are with RIKEN Brain Science Inst., Wako, 351-0198 Japan e-mail: (see <http://www.bdc.brain.riken.go.jp/> tani/).

M. Ito is with Sony Corp. Tokyo, Japan

Manuscript received xxx; revised xxx, 2007.

Neuroscientific literature suggests that learning of complex skills—those which enable object and tool manipulations—is consolidated in the inferior parietal cortex where internal models for those behaviors are self-organized as a result of integrating sensory information from vision and proprioception [4], [5], [6], [7]. These findings indicated that task behaviors in robots could be acquired more efficiently through iterative learning rather than programming.

The decades of research in robot learning have established several different schools of thought. Evolutionary learning [8] and reinforcement learning [9] are approaches in which robots acquire behavior skills through self-exploration and reward signals. These are forms of evaluative learning without direct supervisions. Some researchers found that the self-exploratory behaviors of robots appear to correspond to those human development processes in which innateness and learning are structurally coupled [10], [11], [12]. Although such unsupervised learning schemes might reduce the need for human assistance to acquire skills, the robot would require numerous training trials to be able to execute complex tasks in real robot situations. Because the approach is unrealistic for robots in practical settings [13], we conducted a series of supervised learning experiments involving robots equipped with dynamic neural network models [14], [15], [16].

Supervised learning has tended to be seen as trivial because the optimal trajectories for the desired task behaviors are provided by human tutors, but this is not always the case as this paper attempts to show. First, the type of learning we are interested in is not simple rote learning whereby a robot generates a "recording" of all the teaching patterns without organizing structures. Rather, we see learning as a process through which implicit rules are extracted from patterns of experience. These rules become the foundations for generalizations in the networks that enable the robot to respond appropriately to unlearned situations. Second, it is important to note that the networks have biases which prevent exact learning of teaching patterns. These biases originate from "innate" structures predefined in the network models. These structures include various parameters such as time constant and decay coefficient for each neuron's activation dynamics, initial synaptic connectivities, and the number of neurons that the recurrent neural network (RNN) model recruits [17], [18].

For example, the time constant of the network dynamics which is determined inherently by parameters of neuron units affect the goodness of learning for particular profiles of teaching patterns. The network dynamics with a large time

constant is not good at learning quickly changing temporal patterns. On the other hand, the one with small time constant is not good at learning slowly changing ones. Also, RNNs with dense synaptic connections with all neurons, including the input/output units, tend to generate certain correlation structures among different output dimensions which do not exist in the teaching output patterns. In addition, the characteristics of generalization in RNN learning yield biases for better learning with specific teaching patterns [15]. Teaching patterns that can be generalized into a shared structure are learned more easily, while those patterns that do not share a common structure with others are more difficult to learn [15]. Just providing a set of arbitrary teaching patterns to robots will not result in successful learning outcomes. In this light, it may be possible to improve task training outcomes by carefully examining adequate teaching patterns to be used.

The current paper presents a novel robot tutoring scheme that emphasizes the codevelopment of robots and human tutors as they interact to achieve a specific task goal. More specifically, when robot learning is immature, human tutors physically interact with the robots in order to guide the robots to perform the task better. Through the physical interactions, tutors can easily learn which ways the robots tend to move and which parts of the movements should be modified. The tutor simply exerts intentional forces on to the robots to modify some parts of the trajectories which are crucial for the goal achievement in the tasks. On the other hand they do not interfere with the robot intentionally for unnecessary parts of the trajectories which do not affect the goal achievement directly. It is hypothesized that the re-training of the networks with these newly codeveloped trajectories could lead to much better learning results than cases of training the networks with arbitrary determined teaching trajectories. By this means, the robot tutoring becomes a process to jointly explore a better training path which can influence the network's internal structure to be gradually organized to an adequate one for achieving the desired task goal.

Here we describe our proposed neuronal architecture that uses a continuous time recurrent neural network (CTRNN) [18], [19] and its implementation in a humanoid robot for object manipulation tasks. The model focuses on anticipatory learning [14] of temporal sequences of vision-related signals for object and proprioception of arm position, which we assume corresponds to similar activity in the parietal cortex. Then, two classes of experiments are described in which the humanoid robot learns schemes of specific object manipulations through the human tutoring. In the discussion section, the essence of the proposed scheme is discussed with our special focus on the problem that how complex robot skills can be developed by self-organizing compositional structures with having a set of behavior primitives through the interactive learning. Also the paper will discuss possible correspondences of our experiments to neuropsychological evidences related to parietal cortex.

II. ANTICIPATORY LEARNING OF VISUO-PROPRIOCEPTIVE SENSATION BY CTRNN

The proposed network is designed to learn and to regenerate trained visuo-proprioceptive sensation sequences by means of anticipatory learning mechanisms. In the current robot setup, the visual inputs \bar{s}_t represent sensation of an object's position in the retina coordinate of the robot video camera. The proprioceptive inputs \bar{m}_t actually represents encoder reading of the arm joint positions. Essentially, this network takes sensations of different modalities and mingles them together to generate predictions of their future states.

This flow of information is analogous to those suspected in the parietal cortex, albeit abstractly, where the motion vector of visual stimulus enters the parietal cortex from occipital lobe and proprioceptive information comes from the somatic sensory cortex. Those two types of sensory information appear to be integrated at various locations in parietal cortex [20]. For example, Duhamel et al [21] found some neurons in ventral intraparietal area that are activated bimodally, with either visual or proprioception stimuli.

We assume that the anticipation of this integrated state of vision and proprioception for the next time step (s_{t+1}, m_{t+1}) could emerge through associative learning that is temporally bound. Although no strong physiological evidence has been obtained, many have said that this type of anticipation mechanism can be achieved with a simple forward model [22], [23], [24] in animal and human brains. Once the next step proprioceptive state is predicted, an inverse model can generate the necessary motor torques to achieve the predicted state. Although inverse models are assumed to be in cerebellum or motor cortex in real brains, we just use a PID controller to generate motor torques to achieve the predicted target joint positions in our robot setup. This point is summarized that the CTRNN produces next step prediction for the proprioceptive state m_{t+1} which is sent to the PID controller as the next step target joint positions of the robot.

From dynamical systems view, the anticipatory learning is considered as a process to acquire a dynamic function described in the following differential equation.

$$(\dot{s}, \dot{m}, \dot{c}) = F(s, m, c) \quad (1)$$

This equation simply means that the δ change of the visuo-proprioceptive state is function of its current state and the internal state c . The internal state c is often necessary when the system dynamics state cannot be well represented by the observable states of s and m . The function $F()$ generates the δ change of the internal state c also.

It is essential to note that the dynamical system described by the equation is an autonomous one. This means that the prediction of (s, m) can be self-generated without having the actual sensation of (\bar{s}, \bar{m}) . This enables so-called the closed-loop look-ahead prediction [14] which could account for human capability of the mental simulation of own actions without having actual sensation from the real world[23]. On the other hand, (s, m) of the self-generated can be replaced with (\bar{s}, \bar{m}) of the actual sensation for prediction of its δ change. This operation is called as the open-loop prediction

[14] in which the sensory prediction can be generated with well utilizing the current sensation. The open-loop mode is used when the robot actually acts on the environment.

The visuo-proprioceptive anticipatory learning scheme is implemented with the CTRNN. Why is the CTRNN utilized? First, the CTRNN is considered as a direct implementation of Eq.1 into a neural network model which can learn spatio-temporal structures in a continuous time and space domain. Second, the CTRNN preserves contextual flow by self-organizing appropriate internal dynamic state that corresponds to c in Eq.1. In this way, the robot implemented with the CTRNN can avoid hidden state problems [25]. Third, the CTRNN learns the temporal correlations of any time scale by properly setting the time constants for the neuron activation dynamics, as will be detailed later. (Although in the current experiment with using supervised learning scheme, time constants were set manually by trials and errors, they can be self-adapted if the scheme can be incorporated with genetic algorithms as shown in [26].) From these three reasons the CTRNN is employed in the current study. However, it is noted that the CTRNN architecture does not correspond directly to the real anatomical neuronal structures in parietal cortex. In the current study, the CTRNN is used in the connectionist level abstraction for the purpose of attaining the functions of the forward model which is assumed especially in inferior parietal lobe. Also, the usage of the error back propagation learning scheme [27] which may not represent real neuron synaptic modulation mechanisms can be accounted by the same reason.

We designed our CTRNN model, shown in Figure 1, by modifying the Jordan-type RNN—a discrete time model. Before going to detailed mathematical descriptions of the model, we explain the basic mechanism in a intuitive manner. As shown in Figure 1, the network consists of three layers, namely the bottom layer, the hidden layer and the top layer where there are reentrance loops from the top layer to the bottom layer. The bottom layer consists of groups of units in which the visible units a^X corresponds to the visuo-proprioceptive state (s, m) and the context units a^C does to the internal state c in Eq.1. The activation states of these units are propagated into the hidden units in the hidden layer through the synaptic weights w_{ij}^H and further to groups of the output units o^X and o^C in the top layer. The output units here are utilized in a different way from usual perceptron-type networks or RNNs. o^X and o^C mostly represent δ changes of a^X and a^C in the same way that the time derivative term in the left-hand side of Eq.1 does. Actually, the time development of a^X and a^C in the bottom layer are obtained by integrating o^X and o^C in time through the reentrance loop. This corresponds to the closed-loop prediction of the visuo-proprioceptive sequences.

Figure 1(a) shows the training mode of the CTRNN. The goal of the training is to minimize the error between the teaching sequences (\hat{s}_t, \hat{m}_t) and the self-generated sequences (s_t, m_t) by adjusting the synaptic weights of w_{ij}^H and w_{ij}^O . This can be achieved by applying the back-propagation through time algorithm [27] to the errors. Figure 1(b) shows the open-loop generation mode when the robot actually acts on the environment. In this mode, a_{t+1}^X is obtained through the integration of o_t^X not on a_t^X of self-generated but on \bar{a}_t^X of

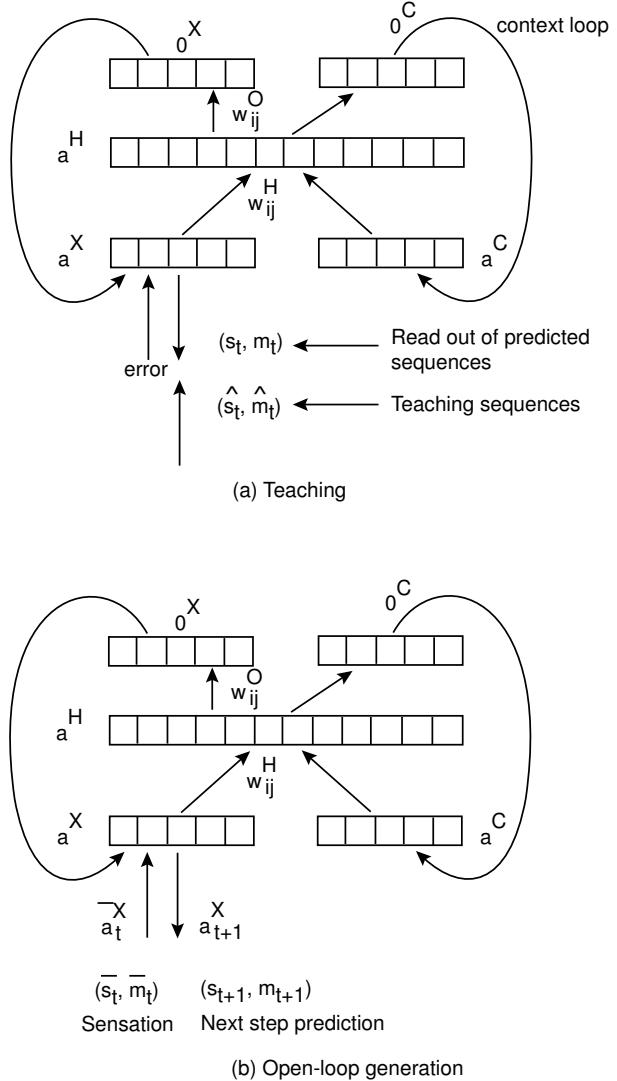


Fig. 1. The CTRNN model employed. (a) training mode and (b) behavior generation by open-loop mode.

sensed from the external.

Next, the exact computation scheme for the forward dynamics of the CTRNN is described. First, the hidden layer activation a_t^H is computed. The potential of the i th hidden unit $u_{i,t}^H$ is obtained by summing the activation propagated from the bottom layer through the synaptic weights w_{ij}^H . Then its activation value is computed by applying the sigmoid function to the potential. This is described as:

$$u_{i,t}^H = \sum w_{ij}^H a_{j,t}^{X,C} + b_i^H \quad (2)$$

$$a_{i,t}^H = \text{sigmoid}(u_{i,t}^H) \quad (3)$$

where b_i^H represents the bias value for the i th hidden unit. Next, the top layer activations of o_t^X and o_t^C are computed in the same way using hidden layer activation values.

$$o_{i,t}^{X,C} = \text{sigmoid}(\sum w_{ij}^O a_{j,t}^H + b_i^O) \quad (4)$$

Then, time development of the potential of the bottom layer units, u^X and u^C , are obtained by integrating o^X and o^C by

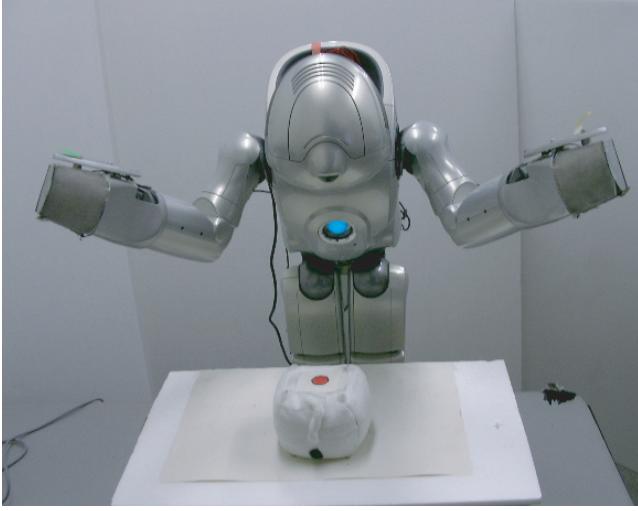


Fig. 2. A humanoid robot made by Sony Corp is at its home position in the task space. The object to be manipulated is in front of the robot.

following the first order differential equation of Eq.5.

$$\tau \dot{u}_i^{X,C} = -u_i^{X,C} + (o_i^{X,C} - 0.5)/\alpha \quad (5)$$

where τ and α denote a time constant and amplification coefficient, respectively. The actual update of these values are computed with the numerical approximation of Eq.6.

$$u_{i,t+1}^{X,C} = -u_{i,t}^{X,C}/\tau + (o_{i,t}^{X,C} - 0.5)/(\alpha \cdot \tau) + u_{i,t}^{X,C} \quad (6)$$

Finally, the next time step activation of the visible units are obtained by taking the sigmoidal output of the potential.

$$a_{i,t+1}^{X,C} = \text{sigmoid}(u_{i,t+1}^{X,C}) \quad (7)$$

Here, $a_{i,t+1}^{X,C}$ represents the closed-loop prediction for visuo-proprioceptive state at the next time step (s_{t+1}, m_{t+1}). The open-loop one-step prediction can be performed by utilizing the inputs from the external rather than the self-predicted ones as shown in Figure 1(b). In the open-loop mode $u_{i,t}^X$ in the right-hand side terms in Eq.6 is replaced with $\bar{u}_{i,t}^X$ which is the potential value representation of the actual visuo-proprioceptive state. Here, $\bar{u}_{i,t}^X$ can be obtained by taking the inverse sigmoid of $\bar{a}_{i,t}^X$.

III. THE 1ST EXPERIMENT

A. Robot platform

A small humanoid robot, shown in Figure 2, is utilized for the experiment. In the current setup only left and right arms are allowed to move where each arm has 4 DOF. Those joints have specific maximum rotation ranges from 70 degree to 110 degree depending on the joints and the rotation angles are mapped to the neuron activation values ranged from 0.0 to 1.0. All other joints except head joints are fixed. The robot has a vision system mounted on its head and the head automatically fixates on a red mark on the object to be manipulated. Eventually, 2 DOF of the head motor positions will provide a rough estimate for direction of object on the table. This relative location information for the object is treated

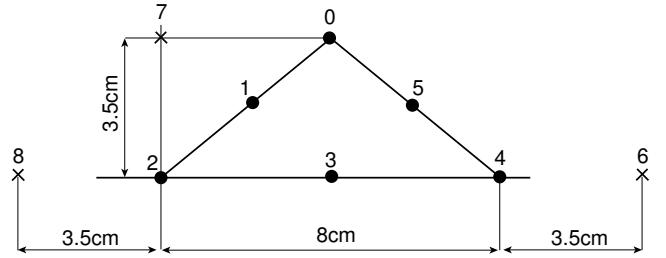


Fig. 3. The initial object positions. 0 to 5 for training and 6, 7 and 8 for generalization tests (without training).

as the visual inputs for the CTRNN implemented in the robot. The object is an elastic cube of 5 cubic centimeter which is located on the task table shown in Figure 2. A human tutor takes both arms of the robot and guides them to the object to teach the object manipulation tasks. The control gains for motors in the arm joints can be changed at each operation mode of the robot. The gains are set as 100 percentage for the self-generation, 20 percentage for the teaching after the network is trained (after the second teaching session), and 0 percentage for the teaching before any training of the network (in the first teaching session).

B. Experiment procedures

We examined the developmental tutoring processes for two separate tasks, task-A and task-B. For the purpose of examining the basic characteristics of the performance developments across the tutoring sessions with later described statistical analyses, the experiment is repeated with three different tutor subjects for each task case.

In task-A, the robot is to take the object by both hands from a fixed home position and lift it up. For task-B, the robot first touches the object only with the right hand, while the left arm remains in the home position. As the right arm returns to the home position, the left arm begins to approach a mark point at a fixed position on the table. In both tasks the object position on the table is changeable within a fixed range and therefore its manipulation requires visual feedback. In the tutoring, the robot is repeatedly guided for the object manipulations with the object located in 6 different positions as indicated by dots labeled by from 0 to 5 in Figure 3.

Each subject proceeds with three succeeding sessions each of which consists of the robot teaching phase, the CTRNN off-line training phase and the robot self-generation phase. Initially the subjects are instructed how to guide the robot in teaching for each task behavior. The subjects are reminded that only specific points during each task behavior are important, not all details need reinforcement for task behavior. In task-A it is important that both hands approach precisely to the right position for the object grasping and that the object can be lifted-up more than 5cm without dropping as the goal. Beyond these specifications, it is not important what sorts of splines are made in the arm trajectories provided that they are smooth. In task-B, the subjects should pay attention to both the position precision in touching the object with the right hand and in reaching the fixed mark on the table by the left hand. For each

task the experimenter demonstrates an example of adequate tutoring. Before the trials, the subjects practice the tutoring patterns with the robot for several times. Because variances in time courses among teaching sequences for each task may confuse the learning processes of the network, the subjects are asked to guide the robot mostly in the same time period with the example demonstrated. If the time period of a particular teaching sequence is 5 percentage longer or shorter than the example one, the sampled sequence is not used for training of the network.

The teaching in the first session and that in subsequent sessions are different. In the first teaching (before any training of the network), the robot is guided without its self-generated force because the control gain is set to Zero. Therefore, the teaching in the first session is not interactive while the second and third sessions are. The tutor guidance interacts with the self-generated movements of the robot with a small gain. The movements are based on the network dynamic structures that were self-organized in the training phase of the previous session. In this situation, the subjects are instructed to follow the movements generated by the robot as much as possible while gently forcing the robot arms so that the task can be achieved. A set of the visuo-proprioceptive sequences, in terms of the head direction and the encoder readings for the arm motor joints, is obtained from these guided behaviors for all object positions. The network is trained with these obtained visuo-proprioceptive sequences as targets in a parallel manner with modulating the synaptic weights and the biases obtained in the previous training session. The first training session starts with the synaptic weights and the biases set randomly near zero values. The BPTT training is repeated for 10000 epochs for each session with a constant learning rate.

At each self-generation session, robot behaviors are examined using three of the trained object position cases (0, 2 and 4 in Figure 3) for both task behaviors as well as for two untrained position cases only for task-A (see 6 and 7 in Figure 3). By using untrained object positions in task-A, we can examine the generalization characteristics of the proposed learning scheme. For the purpose of measuring the developments of task performances across three sessions, we introduce different measures. In task-A, the joint position errors between the trained ones (demonstrated ones by subjects) and the generated ones (replayed ones by the robot after learning) at a specific event are measured for the trained object position cases and the relative position distances between the hands and the object at a specific moment are measured for the untrained object position cases. According to Atkeson et al [28] and Calinon et al [29], there are two different constraints in generating robot motor trajectories in the imitation learning. One is an absolute position constraint and the other is a relative position constraint. It is considered that the measure of the joint position error applied in the trained object position cases corresponds to the former constraint and the one of the relative position distance applied in the untrained position cases does to the latter constraint. In addition to these, the failure rate in terms of the goal achievement (lifting-up the object more than 5cm without dropping) is measured in task-A. In task-B, only the joint position errors between the trained position and the

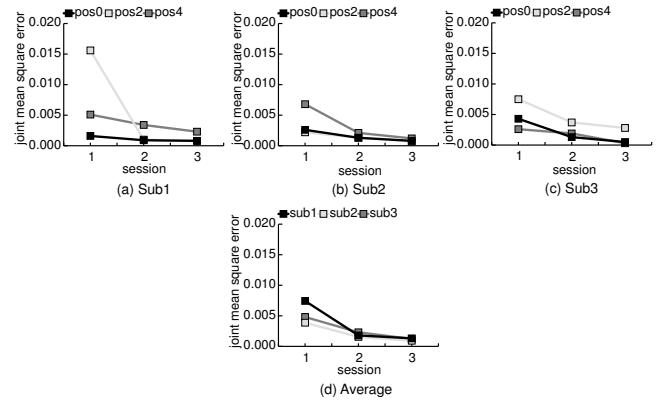


Fig. 4. The developments of the mean square joint position errors between the teaching and the generated through 3 succeeding sessions for (a) subject1, (b) subject2 and (c) subject3 in task-A where plots are shown for three object position cases, denoted by pos0, pos2 and pos4. (d) shows the average among three object position cases for each subject.

generated one are measured because no generalization tests are conducted and the task achievement cannot be measured in terms of success or failure in this task.

In the current experiments, τ and α of the CTRNN parameters are set as 10.0 and 0.1, respectively. 12 context units and 20 hidden units are allocated in the CTRNN.

C. Results

Firstly, the results of task-A is described. Figure 4 shows the developments of the joint position errors between the trained and the generated at the onset of the object grasping in task-A. Here, the error is taken as the mean square error (MSE) of each joint position (mapped in the neuron activation range from 0.0 to 1.0) averaged for 8 arm joints. The plots are shown for three object position cases (position 0, position 2 and position 4) for the three subjects in Figure 4(a), (b) and (c). Figure 4(d) shows the development of the MSE averaged over three object position cases for each subject.

The results indicate that tutoring substantially reduces the joint position error between the trained values and the generated ones as the session proceeds for task-A. The randomized test analysis indicates that the reduction of the MSE is statistically significant ($p \leq 0.00195$) both in the second session and the third session measured across the all subject results with the three object position cases. The average among three different object position cases shown in Figure 4(d) indicates that amounts of the error reductions in the 3rd session are less than the ones in the second session. It is also observed that there are variances in the errors among three object position cases for each subject. It seems that the error in some position cases are already minimized even in the first session and others are not. However, those variances are minimized in the 3rd session.

In task-A, the generalization test was conducted with the cases for the object position 6 and 7. Because there are no teaching data for these object positions for the generalization test, we measured the minimum distances between the robot hand and the object in the trial of the object grasping. In

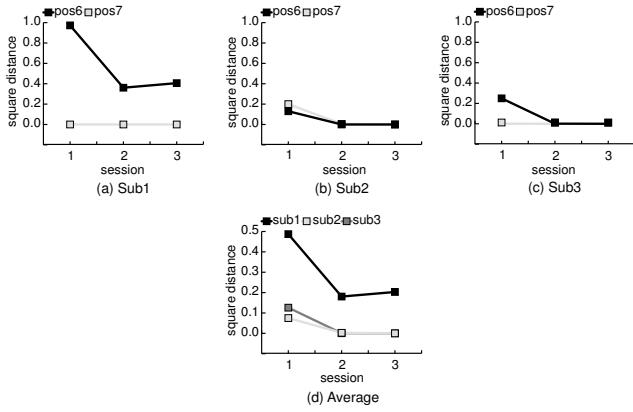


Fig. 5. The developments of the mean square hand-object distances for left and right hands with the object located in the position 6 and 7 for (a) subject1, (b) subject2 and (c) subject3 in task-A where Y axis unit is square decimeter.

the current analysis, the distance is measured in the top-view lateral direction from the inner surface of each robot hand to the object surface of the contact in decimeter. The mean square average is taken for the left-hand and right-hand distances to the object. Figure 5 shows the developments of the hand-object distances across 3 sessions for cases of the object located at position 6 and 7. It is observed that the minimum distance is reduced significantly ($p \leq 0.0156$) with randomized test) from the first session to the 2nd session but not for from the 2nd session to the 3rd session ($p \leq 0.5$). The distance is minimized in the second session mostly where the hands touch the object with zero distance in 3 cases and they do near miss with distance less than 1cm in 2 cases out of the total 6 cases. In the third session, zero distance is achieved in 5 cases out of 6 cases. It is also seen that the developments of the hand-object distance are different between two object position cases for each subject. Especially in the trials of subject 1 and subject 3, position 7 cases show near-zero distances from the first session while the position 6 case starts with larger distances. It is noted that in one case out of 6 cases, namely the case with subject 1 with position 6, the distance cannot be reduced enough to achieve the goal of this task. These results suggest that the generalization for the adopted situations can be achieved mostly but not in a uniform way.

Now we examine how the task performance in terms of the goal achievement develops across the sessions. Figure 6 (a) shows the development of failure rate of each subject sampled over 5 trials with different object positions of 0, 2, 4, 6 and 7 (If the object is lifted-up more than 5cm, it is regarded as a success and otherwise as a failure.) It is seen that the failure rate is reduced largely both in the session 2 and session 3. Figure 6 (b) shows the developments of the failure rates compared between the cases of the trained positions (0, 2 and 4) and the untrained ones (6 and 7) across three subjects. It can be seen that the failure rates in the trained position cases are slightly less than the ones of the untrained.

The above analyses with three different task performance measures indicate some interesting aspects of the development learning processes in task-A. Firstly, it can be said that there are variances in the development of the task performances

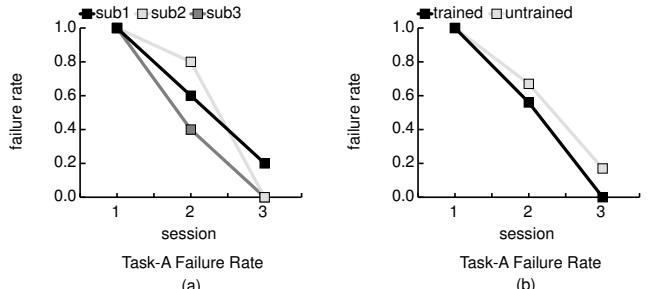


Fig. 6. The developments of the failure rates for each subject in task-A.

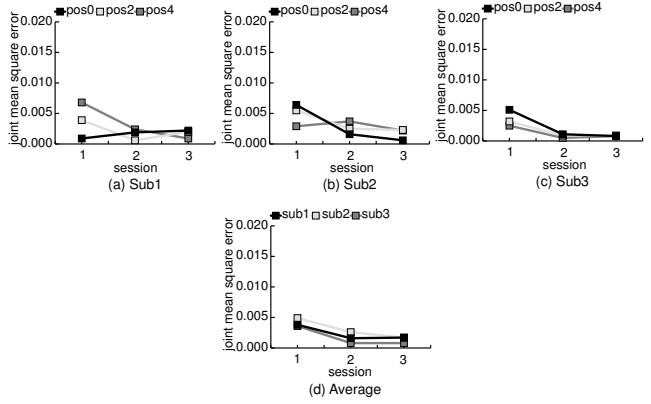


Fig. 7. The developments of the mean square joint position errors between the teaching and the generated through 3 succeeding sessions for (a) subject1, (b) subject2 and (c) subject3 in task-B where plots are shown for three different object position cases, denoted by pos0, pos2 and pos4. (d) shows the average among three object position cases for each subject.

among different situations related to object positions. The variances are large in the beginning in both measures of the joint position error for the trained cases and the hand-object distance for the untrained cases which are reduced significantly in the later sessions in most cases. The P-values obtained in our randomized test analysis indicated that the task performances in these two measures are improved significantly in the second session but less significantly in the third session. However, the failure rate is significantly improved not only from the first session to the second session but also from the second session to the third session. This implies that the achievement of the task goal (lifting up the object without dropping it) may not be always accounted only by the measures of the joint position error and the hand-object distance. Actually, skills for precise position adjustment and subtle force control by utilizing elasticity of the object and the hand surfaces at the very moment of grasping seem to develop from the second session to the third session with showing only slight improvements in these two measures.

Nextly, the results of task-B is described briefly. Figure 7 shows the developments of the joint position errors between the trained and the generated in task-B. The errors at 4 joint positions in the right arms are obtained as differences between the the guided ones at the moment of the right hand touching the object and the generated ones at the moment of the same hand approaching to the object in the closest

range. The joint position errors for the left arm are obtained in the same ways with the fixed mark on the table. The MSE is computed for all joint position errors in both arms. The randomized test analysis indicates that the reductions of the MSE are statistically significant with ($p \leq 0.0078$) in the second session but not so significant with ($p \leq 0.221$) in the third session. The average among three object position cases shown in Figure 4(d) indicates that amounts of the error reductions in the 3rd session are less than the ones in the second session. The errors show relatively large variances among different object position cases in the first session which are reduced significantly in all subject cases. It can be said that the developments of the joint position errors in task-B share similar trends with the ones in task-A.

Now, let's look at the developments of the exact motor profiles, i.e., trajectories of proprioceptive states of generated as compared with the ones trained at each session. Such observation would derive further understanding of how the interactive tutoring develops the motor schemes for the tasks through the iterative sessions. Figure 8 and Figure 9 show the developments of the teaching motor profile (trajectory of 4 representative joint positions in the arms) and the one generated for a subject case of Task-A and Task-B, respectively.

The 4 DOF motor profiles are plotted for the 0, 2 and 4 object position from the left, center and right downward for three sessions in these figures. $mL0$, $mL1$, $mL2$ and $mL3$ in Figure 8 denote the joint positions at the pitch, the roll and the yaw of the left shoulder and the pitch of the left elbow. $mR1$ and $mR2$ in Figure 9 denote the joint positions at the roll and the yaw of the right shoulder. There are significant differences between the teaching profiles and the generated profiles in the first session where teaching profiles jackknife discontinuously at some points in both task-A and task-B. (The exact moments of discontinuous changes in the profiles are indicated by small arrows on the plots.) However in the second and third sessions, the teaching profiles become much smoother and the differences between profiles for teaching and generation are reduced.

We observed some interesting phenomena during the interactive developmental processes. The subjects reported that it was difficult to generate similar or correlated guided behaviors for all the object positions in the first teaching session. However, as the session proceeded they found it became easier since the robot is basically leading the movement. Hence, the tutors needed to adjust the arm positions only at precise moments such as of grasping and touching the object.

Another interesting phenomena can be observed in the generated motor profiles for Task-B shown in Figure 9. In the 1st teaching session, 2 DOF motor profiles in the left arm are flat until 30 time step because the left arm is fixed while the right arm is moved to reach the object in the first half of this task behavior. However, in the 1st self-generation session after teaching, the left arm joint positions start to change slightly from the beginning of the task behavior. This sort of deviation of motor profiles happens for the all subject cases. These deviations may be due to the tendency of the CTRNN to preserve correlations among the profiles of all the output dimensions because they are mutually connected through the

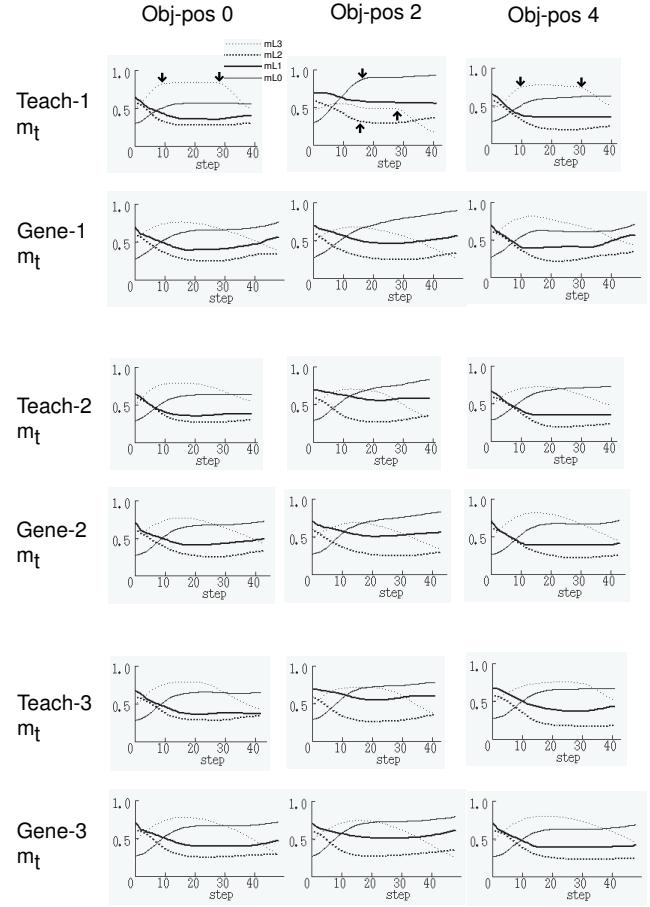


Fig. 8. Motor profiles of teaching and generation in three succeeding tutoring sessions for task-A. $mL0,..mL3$ denote the 4 DOF joint positions in the left arm. Small arrows indicate moments of discontinuous changes in the profiles.

common hidden units. Therefore, when the right arm joint positions move, the left arm ones tend to move in some degree from the intrinsic correlation hidden in the network. The same deviation in the left arm motor profiles can be observed also in the 2nd teaching session as subjects permit left arm deviations that is not against task specification.

An additional experiment was conducted and yielded more interesting results. In this experiment, two subjects were asked to fix the left arm rigidly during the first half of the task behavior in the second teaching session. This resulted in significant increases in the joint position errors (averaged among three object position cases) in the 2nd self-generation session. The error increases from 0.00262 to 0.00436 for one subject and 0.00083 to 0.00168 for the other subject when compared to the cases where this arm was able to move. These results imply that the CTRNNs are good at learning patterns with more smooth profiles of having more correlations among different dimensions. This is natural because the CTRNNs are continuous-time dynamical systems with predefined time constants and their unit activities of large dimensions are tightly coupled by means of the dense synaptic connections. When the CTRNN are forced to learn “unnatural” patterns against their intrinsic characteristics, such training could hamper the learning process severely.

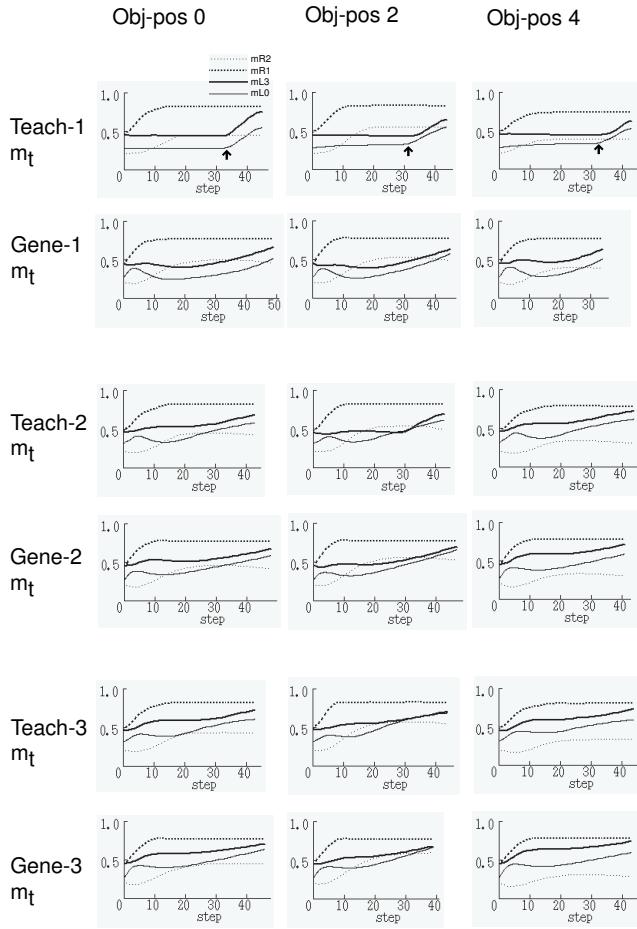


Fig. 9. Motor profiles of teaching and generation in three succeeding tutoring sessions for task-B. mL0 and mL3 denote 2 DOF joint positions in the left arm and mR1 and mR2 denote those in the right arm.

IV. THE 2ND EXPERIMENT

If a robot is to learn more diverse and complex behaviors, introducing level structures to its neuronal architectures seems inevitable. The problem is how the continuous visuo-proprioceptive flow of experience can be segmented into chunks of behavior patterns that can be compositionally used for other situations [30], [31], [32], [9], [33], [34], [26]. If we suppose a two-level structure, the lower level might organize a behavior repertoire and the higher level might select and combine behavior patterns from it. Previous studies showed that the behavior repertoire can have either local [24], [32], [35] or distributed representations [34], [26] in neuro-dynamic systems [36]. Unlike local representations, a distributed representation has globally shared structures that can be used to represent whole patterns that could be acquired in a single network once final generalizations are achieved. However, any distributed representation scheme would have difficulty handling increasingly larger numbers of patterns because catastrophic memory interferences among different patterns would occur. Therefore, the current study utilizes a local representation scheme so-called the mixture of local experts [37]. More specifically, we introduce a model of hierarchically organized mixture of local experts which has

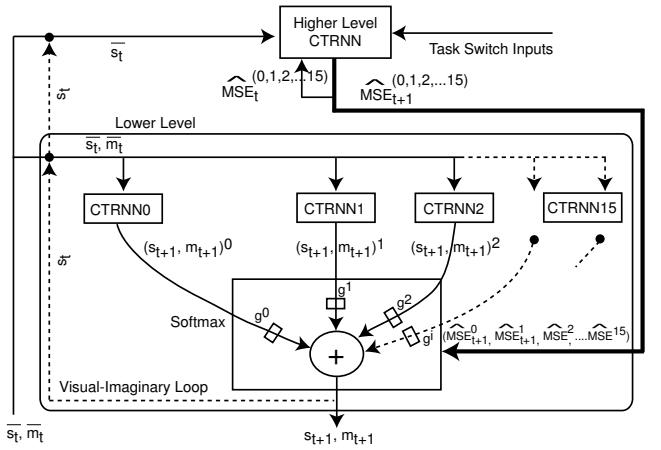


Fig. 10. Two-level structured CTRNNs. The information flow shown in this figure is particularly for the generation phase.

been developed from our previous study[32]. This newly proposed architecture is applied to a developmental tutoring scheme for acquisition of more complex task skill.

A. The two-level architecture

Figure 10 shows the two-level neural network architecture used. Before proceeding to detailed descriptions of the scheme, abstract mechanisms in its learning phase and generation one are introduced. The lower level consists of ($N = 16$) CTRNNs. Although each local CTRNN in the lower level has the same structure with the one used in the first experiment, it is associated with a gate in this new architecture. The current gate opening for a local CTRNN determines its effect on the total output in the lower level. The local CTRNN with the largest gate opening among others at each time step is called as the winner module. The output of the winner dominates the total output in the lower level. The gate opening mechanism is different between in the learning phase and in the ganeration phase.

In the learning phase, each local CTRNN in the lower level competes to become an expert for generating outputs with the minimum error compared to the target one among others. In sweeping a teaching sequence, the winner CTRNN with the maximum gate opening switches from one to another depending on the error of each CTRNN at each step in the sequence. On the other hand, a single CTRNN is located in the higher level of which time constant is set larger than the one in the lower level networks. This CTRNN learns to predict the winner switching sequence in the lower modules by memorizing how the errors of all modular CTRNNs change in the teaching sequence with rough approximation.

In the generation phase, the gate opening for each local CTRNN in the lower level is determined not by its actual error but by the one predicted by the higher level CTRNN. Consequently, the higher level determines switching of expert modules in the lower level by means of the error prediction. The module selected as the winner by the higher level generates exact visuo-proprioceptive sensation patterns at the

moment by its forward dynamics. In this way, the proposed hierarchically organized networks can learn to generate relatively long sequences through their decomposition into segments.

Now, the scheme is described in details with introducing mathematical formula. Each CTRNN in the lower level has a gate and a_t^X of the overall outputs of this level is computed as the gate weighted summation of each network output a_t^{Xi} :

$$a_t^X = \sum_{i=1}^n g_t^i \cdot a_t^{Xi} \quad (8)$$

where g_t^i is the gate opening of the i th network. Here, a_{t+1}^X corresponds to next time step prediction of visuo-proprioceptive state s_{t+1}, m_{t+1} . The temporal sequence of g_t^i is determined by the learning process of the lower level networks. Basically the gate at time step t opens larger with the network of generating less error between the target values in training (\hat{s}_t, \hat{m}_t) and the self-generated ones (s_t, m_t) at the time step. As the softmax function is applied, the gates tend to open in a winner-take-all (WTA) manner. The gate opening is computed as:

$$g_t^i = \frac{\exp(s_t^i)}{\sum_{j=1}^n \exp(s_t^j)} \quad (9)$$

$$s_t^i = \frac{-er_t^{i^2}}{(2\sigma^2)} \quad (10)$$

where $er_t^{i^2}$ denotes the time average of the mean square prediction error (MSE) of the i th network at time step t . The time average is obtained by using a fixed time steps window. σ^2 denotes the standard variance of the training errors of all local CTRNNs in the lower level. Upon computing the gate opening for each network, the error vector er_t^i generated by the i th network is weighted by its gate opening which is computed as $er_{g_t}^i$.

$$er_{g_t}^i = er_t^i \cdot g_t^i \quad (11)$$

Then, the i th network is trained by back-propagating the obtained error vector $er_{g_t}^i$. This means that a network with less MSE for certain segments of visuo-proprioceptive sequences will learn those segments better since its gate has a larger opening. In this way, local “expert” networks develop for particular segments that repeatedly appear. The visuo-proprioceptive sequence patterns experienced in the lower level are articulated into sequences of segments accompanied with gate switching sequences[32].

Meanwhile the higher level CTRNN learns to mimic the gate switching sequences observed in the lower level. The learning of the higher level occurs only after the lower level is trained at each tutoring session. In the current implementation a single CTRNN is allocated in the higher level. This CTRNN learns sequences of the MSE for each lower level CTRNN instead of the gate opening. The higher level CTRNN has mostly the same structure as shown in Figure 1. 21 visible units a^X in the bottom layer are allocated for 16 dimensions of the MSE vector, 2 dimensions of the visual inputs at the current step that are the same as those received at the lower level, and 3 dimensions of the task switching vector. Here,

MSE is the prediction of the MSE vector generated in the lower level. o^X units in the top layer are allocated only for the MSE vector because the prediction is made only for this vector. The next step prediction MSE_{t+1} is fed-back to the CTRNN as the next step inputs. The visual sensation as well as the task switcher vector are regarded just as inputs and they are not for prediction.

Then, the CTRNN makes a prediction about the MSE vector for the next step. For behavior generation, the predicted MSE vector is sent to the lower level which is converted to the gate openings with the softmax function in Eq. 9. This pathway serves as a top-down control by the higher executive level onto the lower visuo-proprioceptive level. The time constant τ of the higher level is set as four times that of the lower level. This time constant difference enables the abstraction of information flow from the lower level to the higher one [26]. The exact value of τ is 5.0 for the lower level CTRNN and 20.0 for the higher one, correspondingly. α is set as 0.1 for both of the CTRNNs. 4 context units and 10 hidden units are allocated for each of 16 CTRNNs in the lower level. 6 context units and 16 hidden units are allocated in the higher level CTRNN. These numbers are determined as mostly minimum ones required for successful learning of the networks in our preliminary experiment.

Both of the higher level CTRNN and the lower level CTRNN can be operated either in the open-loop mode and in the closed-loop one. In the open-loop mode, the actual visual sensation s_t is input to the lower level CTRNN and the higher level CTRNN, correspondingly. In the closed-loop mode, the visual prediction s_{t+1} from the softmax output is fed-back to the both networks through the visual-imaginary loop which is shown by a dotted line in Figure 10.

One of the interesting questions in this experiment is that how visuo-proprioceptive flow of experience could be segmented through the developmental learning processes. If segment patterns learned by local “expert” networks in the lower level appear as reusable elements that can reconstruct the whole experienced flow by their sequential combinations, they are called as behavior primitives. (Michael Arbib called them as motor schemata in his motor schemata theory [30].) In this sense, our focus in the robot learning is not for learning by rote but for achieving certain generalized structures with the compositionality by adequately organizing a set of behavior primitives mentioned in the above.

B. Setup

In this experiment, the behavior tasks are more complex than those in the previous examples. A robot with this two-level structured CTRNN receives tutoring training for a set of task behaviors. In task-1 the robot must approach the object with both hands coming down from the home position, grasp the object, let it go, and return to the home position. In task-2, the robot must approach the object, grasp it, bring the object up and down three times, release the object, and return to home position. And in task-3 robot must approach the object with its left hand from the home position, alternately touch the object with its left and right hands eight times before returning to

the home position. We see that these actions required in each task consist of a set of behavior primitives. Those are **ApB**: approaching to the object by both hands, **Gr**: grasping the object by both hands, **Hm**: going back to home position, **Li**: lifting up and down the object repeatedly, **ApL**: approaching to the object by left hand and **Tch**: touching the object by left and right hands in turn. However, it is important to note that the robot is not taught for each behavior primitive explicitly. The robot is trained with continuous visuo-proprioceptive flow for each task behavior without seeing any segmentation cues. One of the central motivations in experiment-2 is to examine whether these behavior primitives appear in the segmented flow as the results of the self-organization. If so, we will examine further how they are internally organized.

A tutor teaches these three task behaviors to the robot in 3 tutoring sessions. (Note that all three different tasks are taught to the robot at each training session, unlike the first experiments in which tasks A and B are trained in different sessions.) Tutoring procedures are similar to the first set of experiments. Tutors are given instructions for each task. For task-1, the subject is asked to ensure that both hands hit precisely the right position for object touching. In task-2, the tutor needs to make sure the object is held and brought up and down three times without dropping it. In task-3 the robot should touch the object by each hand in turn for the correct number of times. In each tutoring session, the task behaviors are guided and the initial position of the object changes 6 times for each task (Figure 3).

In the first session, the network training is iterated for 50000 epochs for the lower level and 5000 epochs for the higher level networks, respectively. In the second and third sessions, 10000 epochs and 5000 epochs for the lower and the higher level networks. The experiment was conducted with one subject.

C. Results

The experiment results showed that the robot learns to perform all the three task behaviors nearly perfectly by the 3rd session, even for untrained positions. Another important observation was that a set of behavior primitives appear in the visuo-proprioceptive sequence patterns as segmented by the gate openings.

In order to visualize how the segmentations proceed in the developmental learning processes, the developments of the gate opening are plotted for each task. Figure 11, Figure 12 and Figure 13 show the visuo-proprioceptive profile and the gate opening across three sessions of self-generation for Task-1, Task-2 and Task-3, correspondingly. The profiles of the task-1 are taken for the case of the object position 0, task-2 for the position 2 and task-3 for the object position 2. In these figures, the time developments for the openings of 16 gates, 4 joint positions in left arm and 2 dimensional camera head direction corresponding to the vision information are shown in the upper, the middle and the bottom levels. Each gate opening is represented by grey level color of a horizontal bar from white to black as corresponding to its value from 0.0 to 1.0. Those bars are aligned from the top to the bottom levels as corresponding to from the 0th to the 15th gate opening. A set

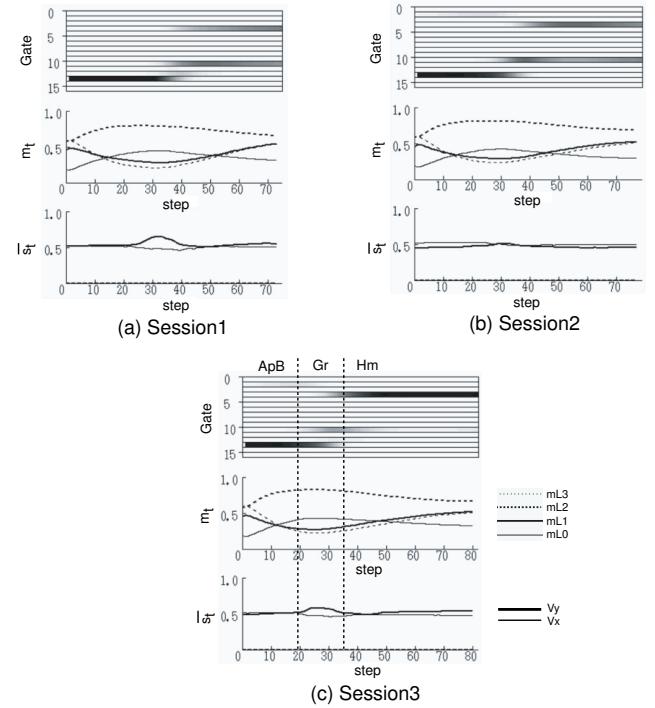


Fig. 11. Gate opening, visual inputs and joint positions profiles in (a) the 1st session, (b) the 2nd session and (c) the 3rd session for Task-1.

TABLE I
SUMMARY OF BEHAVIOR PERFORMANCES ON TASK-1.

session	pos. 0	pos. 2	pos. 4	pos. 6	pos. 7
1st.	Both	Both	Both	Both	Both
2nd.	Both	Left	Both	Right	Left
3rd.	Both	Both	Both	Right	Both

of labels attached on the gate opening profiles denote behavior primitives which are identified in the 3rd session for each task.

The general observation for all task developments is that basic structures for segmentation of the visuo-proprioceptive flow is organized in the session 1. However, the motor profiles continue to develop until the session 3 except the task-1 case in which the motor profile development almost converges in the first session. The details of this point is described later.

Table I, Table II and Table III summarizes the developments of the task performances for the task-1, the task-2 and the task-3, correspondingly. For each task, performance at different initial positions of the object for each session is described using abbreviations. For tasks 1 and 2, self-generation is evaluated for cases with the trained initial object positions of 0, 2 and 4 and untrained ones of 6 and 7. For task-1, “Both”,

TABLE II
SUMMARY OF BEHAVIOR PERFORMANCES ON TASK-2.

session	pos. 0	pos. 2	pos. 4	pos. 6	pos. 7
1st.	No HoldUp	HoldUpLost	No HoldUp	No HoldUp	HoldUpLost
2nd.	No HoldUp	Updown 1	Updown 1	No HoldUp	Updown 1
3rd.	Updown 3	Updown 3	Updown 2	Updown 2	Updown 3

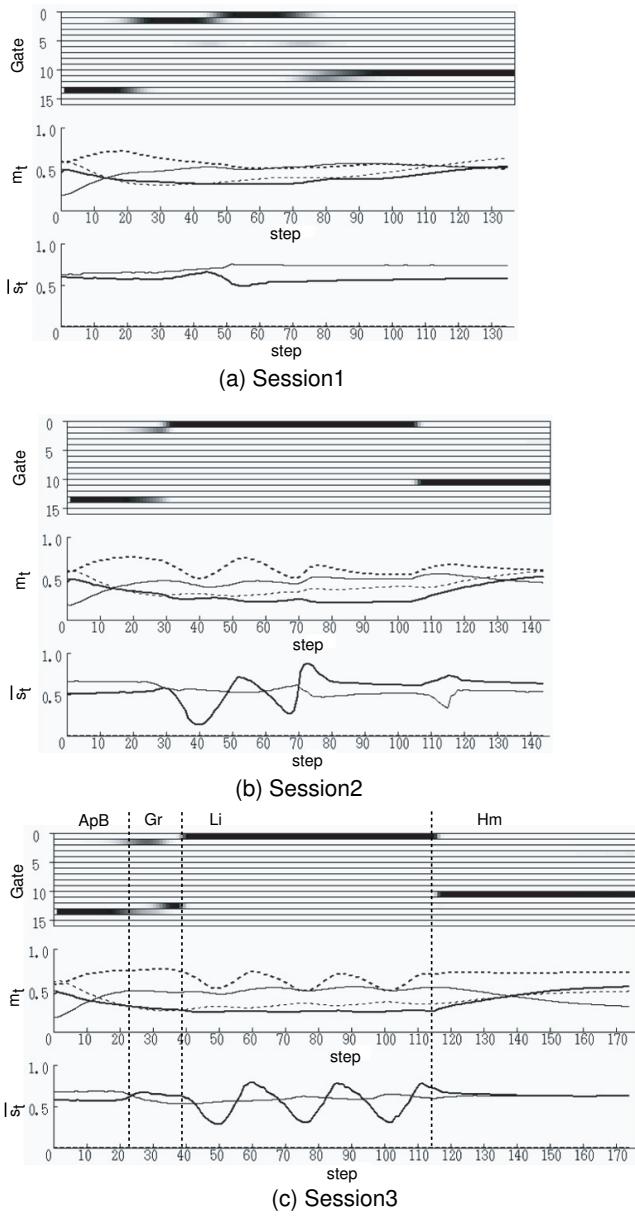


Fig. 12. Gate opening, visual inputs and joint positions profiles in (a) the 1st session, (b) the 2nd session and (c) the 3rd session for Task-2.

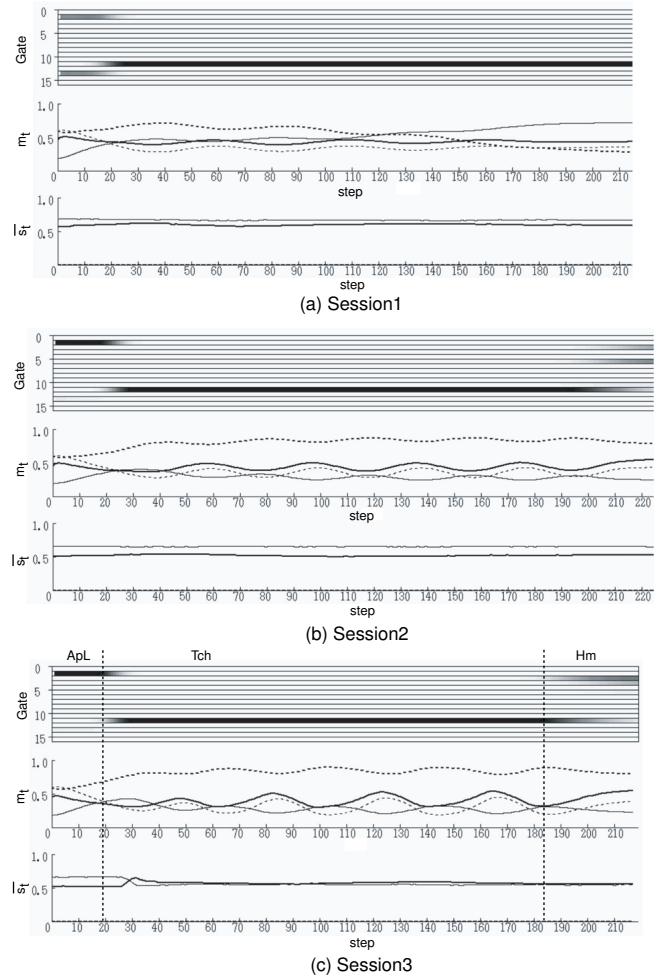


Fig. 13. Gate opening, visual inputs and joint positions profiles in (a) the 1st session, (b) the 2nd session and (c) the 3rd session for Task-3.

“Right” and “Left” denote that the object is touched by both hands, by only the right hand and by only the left hand, correspondingly. For task-2, “No HoldUp” means that the robot cannot hold-up the object at all. “HoldUpLost” means that the object is held up once and it is dropped soon after. “UpDown X” denotes that the object is successfully held for X times. Task-3 performance was evaluated only for trained object positions 0, 2 and 4. (This was because object was frequently moved out from the workspace when touched if the object is initially located at the untrained positions of 6 and 7.) “Touch X” means that the object is touched X times by either of the right or the left hand. The correct number of touching is 8.

Now, the details of each task development are examined. From Table I, Table II and Table III, we can see that, with the exception of task-1 which were perfectly generated in the first session, task performances develop gradually in the tutoring sessions. The task-1 behaviors were generated perfectly even with untrained object positions. The performances, however, do falter slightly in the second session but, by the third session, they are near perfect again. This might be because the task behavior in task-1 is simpler than those in the others. We also noted that visuo-proprioceptive profiles and gate openings for

TABLE III
SUMMARY OF BEHAVIOR PERFORMANCES ON TASK-3.

session	pos. 0	pos. 2	pos. 4
1st.	Touch 1	Touch 0	Touch 0
2nd.	Touch 1	Touch 1	Touch 1
3rd.	Touch 8	Touch 5	Touch 7

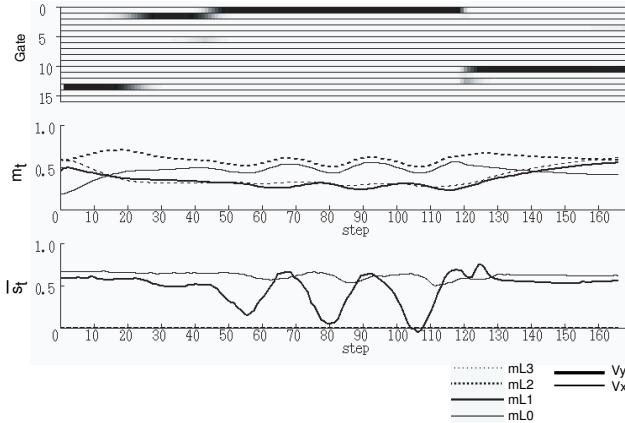


Fig. 14. Gate opening, visual inputs and joint positions sequences for Task-2 in the first self-generation session with the object moved manually by an experimenter.

task-1 do not change drastically across three sessions as shown in Figure 11(a), (b) and (c). We also saw that the behavior primitive **ApB** is encoded by the 13th gate opening and **Hm** by the joint opening of the 3rd and 10th gates for all the three sessions.

In the task-2, the object could not be lifted in three of the five attempts in the first session. While we noted that the approach trajectories for both hands were almost perfect, the robot could not squeeze the object suitably with both its hands in every position. The visuo-proprioceptive profile in Figure 12(a) shows that the object is first touched around the 20th step, then the arms go up slightly without holding up the object. They are frozen like that until the 100th step when the arms start to go back to the home position.

So we conducted an another experiment to examine how much task-2 behaviors are actually learned in this stage. The robot repeated the self-generation of the task behaviors while “fake” visual feedback was made. When the robot fails to lift the object, the tutor takes the object and moves it up and down. The robot sees this action as its own. During the experiment, it is seen that the hands of the robot follow movement of the object that goes up and down (see Figure 14).

This result indicates that some correlations on the visual inputs and the proprioceptive inputs are acquired at this stage. The neural dynamic structure generated by learning initiate the up and down arm movement patterns via entrainment from “fake” visual feedback. The proprioception profile in Figure 14 shows coherent oscillation between the proprioception and the visual inputs. It is possible that the failure in the previous trial was due to a lack of proper visual feedback when the object was not grabbed. In fact in the following teaching session, the tutor felt that the robot can self-generate the desired movements mostly and only needed a little guidance to squeeze the object slightly harder. With this guidance, the robot’s performance improved. In the second self-generation session, the robot manages to lift the object up and down once before dropping it, Figure 12(b) but by the third session the robot executes the task nearly perfectly, see Figure 12(c).

In the first session, the robot cannot touch the object in

task-3 most of the time. In fact, oscillatory motor patterns for repeated touching decay over time (Figure 13(a)). In the second session, the robot starts to touch the object at the least once and initiating oscillations in motor patterns are seen, see Figure 13(b). By the third session the robot can repeatedly touch the object in a stable manner as it generates more explicit cyclic patterns in the motor profile (see Figure 13(c).) In fact, our further analysis on the networks showed that this cyclic pattern of **Tch** is generated by limit cycling attractor self-organized in the 11th network of which gate is exclusively open during generation of this pattern. In the analysis it was observed that this network alone generates this cyclic pattern with a constant periodicity after its convergence to the steady oscillation. Similarly, it was found that another cyclic pattern of **Li** is generated by limit cycling attractor self-organized in the 0th network.

By looking at the gating sequences generated in the third session, we can see how continuous visuo-proprioceptive sequences are articulated through a self-organized set of behavior primitives that emerges in our proposed neural network architecture. For example, in Figure 11(c), Figure 12(c) and Figure 13(c), we see that the behavior primitive **ApB** is encoded by the 13th gate opening for task-1 and task-2, **Li** by the 0th gate opening, and **Hm** by the 3rd gate opening for the task-1 and the same for the task-2 does by the 10th gate opening. There are differences in the gate opening for representing **Gr** between in task-1 and in task-2. The 12th gate opening and the 10th one are mixed in the former case and the 1st, the 12th and the 13th ones are mixed in the latter case. In task-3 **ApL** is encoded by the 1st gate opening, **Tch** does by the 11th gate opening and **Hm** by the 2nd gate opening with slight opening of the 3rd one and the 11th one.

Here, some questions might come to mind. (1) Why is the same behavior primitive encoded by different gate openings in some cases?, (2) Are the encodings of behavior primitives sensitive to the object’s position or are they sufficiently generalized to be insensitive to the object’s position?, and (3) How can the timing of switching from one behavior primitive to another in terms of the gate opening be made?

To examine these questions the activation profiles in the higher level CTRNN are compared for task-2 with different initial object position cases, namely the position 7, 8 and 6 shown in Figure 15(a), (b) and (c), correspondingly. (Position 8 is 3.5cm left of position 2 and is newly introduced in this experiment in order to amplify the effects of position differences.) The profiles consist of context unit activation and the MSE prediction vector of the 16 dimensions in the higher level CTRNN and the gate openings and the motor profiles in the lower level CTRNNs. The value of the MSE prediction is represented by grey level color of each horizontal bar where the darker color denotes the smaller MSE prediction. First, we see that gate opening profiles and context activation profiles are mostly same overall for all three cases regardless of the object position difference. However, there are certain differences in the gate opening immediately before the 0th gate opens between the position 8 and 6 cases. The 1st gate opens in the position 8 case in Figure 15(b) and the 12th gate does in the position 6 case Figure 15(c). It is, however, important

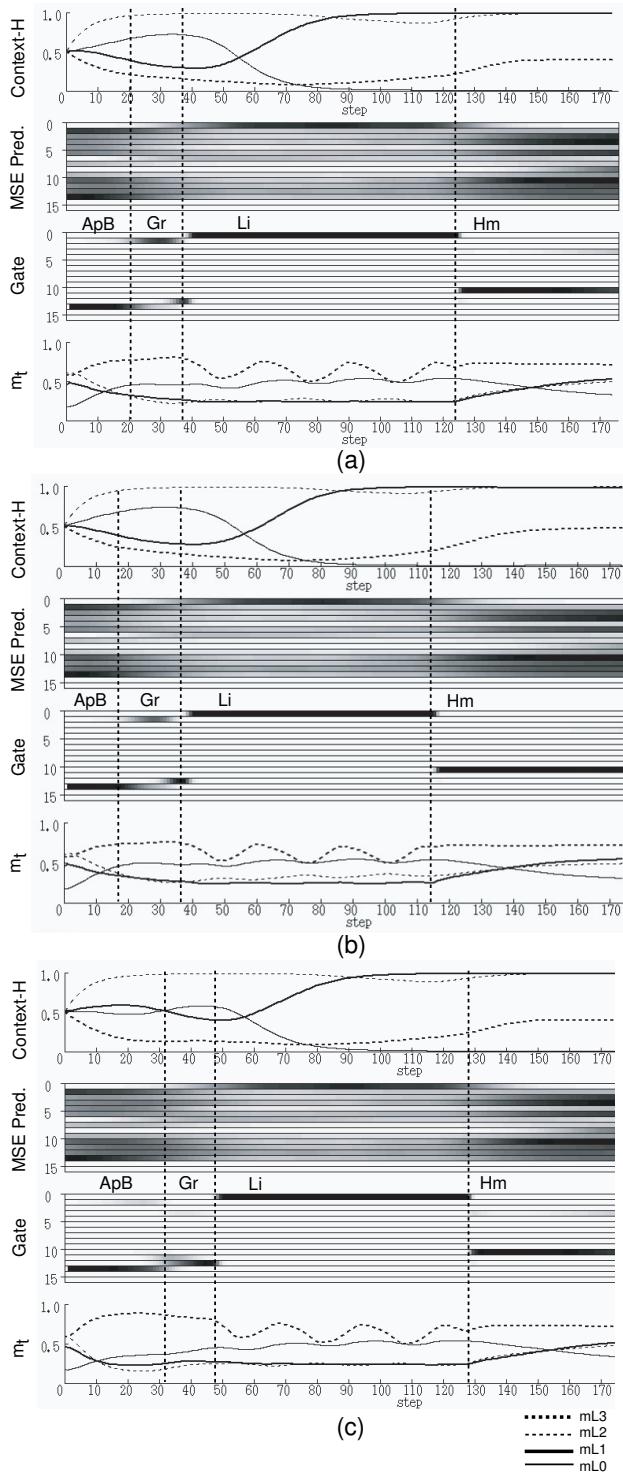


Fig. 15. The context unit activation and the mean square error prediction vector of 16 dimension in the higher level CTRNN and the gate opening and the motor profile in the lower level CTRNNs in three different initial object position cases in (a) with position 7, (b) with position 8 and (c) position 6.

to see that the MSE prediction vector profiles in these time windows are similar, as are the context activation profiles between the two even though the gate openings are explicitly different in this period. This means that the 1st and 12th local CTRNNs in the lower level are equally good at predicting the visuo-proprioceptive sequences in this segment and therefore show only slight differences in their errors. Although this may introduce certain fluctuations in the gate opening between the two because of the winner-take-all characteristics by means of the employed softmax function, it does not matter for the behavior generation because both local nets are good at generating motor patterns only with small errors. This explains how redundant encoding of behavior primitives are self-organized and why behaviors can be generated stably regardless of such redundancy.

Finally, the question about the timing of switching for behavior primitives is addressed. The exact question is that, after grasping the object and stopping there for a moment, what sorts of cues can initiate next gate opening for lifting up the object? Also, after lifting up and down the object for three times, how can the next behavior primitive **Hm** be triggered? The underlying mechanisms can be inferred by looking at the context activation profiles in Figure 15. It is seen that the context activation states gradually change during each period of behavior primitive. This means that the internal dynamics by means of the context unit activations “count” for the learned period of waiting before lifting up the object and the learned repetition times of cyclic movements, lifting up and down of the object or touching the object by alternating left and right hands. Our preliminary experiments showed that the learning error in the higher level CTRNN cannot be minimized if synaptic modulation by means of the error back-propagation is restrained for the weights connected from/to the context units (from a^C to a^H and from a^H to a^C). In such cases, the context activation profiles become mostly flat. It can be said that adequate contextual flow which can orchestrate sequencing of behavior primitives in proper timing can be self-organized through the learning of the network by utilizing the context dynamics. This part of arguments correspond to an idea of “kinetic melody” by Luria [38] who discussed importance of organizing contextual flow in unifying behavior primitives into skillful smooth actions.

D. Pantomime via the visual imaginary loop

In the last an experiment on pantomime is described. The robot can pantomime object manipulation behaviors without accessing an actual object but with accessing its visual imaginary. In this experiment, both of the lower level CTRNN and the higher level one operate in the closed-loop mode with the actual visual inputs shut off. These results are compared with the case in which the visual imaginary cannot be accessed (the robot is operated in the open-loop mode in which the visual inputs are fixed artificially at constant values of seeing the object in position 0 in this case.)

Figure 16 shows the comparison of the two cases for performing Task-2 by pantomime with and without accessing the visual imaginary. In Figure 16(a) of the case with the visual

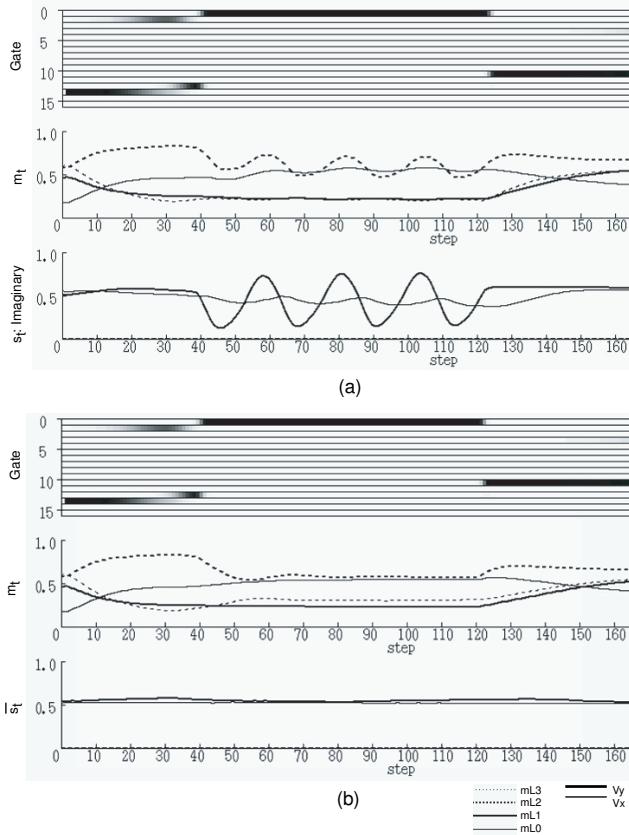


Fig. 16. Comparison of pantomime performances of Task-2 (a) with and (b) without the visual imaginary loop.

imaginary loop, it is observed that the sequential behavior of approaching the object, the cyclic movement of holding up and down the object and homing are well imitated. It is also seen the visual imaginary is reconstructed well. On the other hand, in the case without the visual imaginary loop shown in Figure 16(b), the motor movement is frozen while lifting the object up and down even though the gating signal from the higher level is completely the same as the one in the case with the visual imaginary loop. It is understood that inadequate visual inputs with fixed values disturb the CTRNN dynamic of the periodic movement. This explains that the visual inputs and the proprioceptive inputs are treated in an inseparable manner in the lower level CTRNN which is analogous to the neuroscience observation that visual stimulus and proprioception are integrated well in parietal cortex [20]. The higher level CTRNN, on the other hand, is totally insensitive to the external perturbation. The reason comes from the fact that the higher level CTRNN does not learn to anticipate the visual inputs sequences—it just receives the visual input. No anticipation of those inputs makes the CTRNN dynamics less sensitive to them. Therefore the robot can pantomime as well as execute actual behavior by utilizing the visual imaginary loop once the two levels of CTRNNs has learned the processes.

This result might explain underlying mechanisms for some apraxia cases related to parietal cortex impairment of human brains. Left hemisphere damage can result in various types of apraxia [39], ideomotor apraxia is of particular interest in light

of our results as this form of apraxia shows deficits in skilled tool use caused by lesions in the inferior parietal cortex [40]. Patients with ideomotor apraxia typically perform worse when attempting to pantomime gestures than when using actual objects [39], [40], [41], [42]. The question here is why damage in inferior parietal cortex would impair pantomimed acts more than actual activities. As we have described, pantomime requires the visual-imaginary which can compensate for the visual input received when actually handling objects. Our experiments showed that the robot can pantomime when there is closed-loop feedback of the visual imaginary but not in the absence of that feedback. Based on this result one may speculate that the loss of the closed-loop circuit for generating the visual imaginary inhibits pantomime for the patients with lesions in the inferior parietal cortex.

V. DISCUSSION

Our experiments suggest that the innateness of neural network characteristics cannot be ignored in the learning processes. The innateness in terms of the network parameters, the initial connectivity as well as number of neuronal units and synapses determines the preferences for profiles of patterns to be learned in the network. In addition, the generalization characteristics inherent to neural networks of distributed representation [43] determine the preferences for the set of patterns to be learned in a single network. A set of patterns that share some structures with other profiles are easier to learn than those with no shared profiles [15].

However, it is difficult for tutors to know what any networks learning preferences could be prior to their actual trials. Therefore, the tutors need to explore them by interacting with the network dynamics. In the first session, the teaching trajectories in our experiments were arbitrarily provided to the networks via the tutor's attempts to satisfy the task specifications. Such arbitrariness resulted in generating large errors in the self-generated motor patterns. The tutors feel the “intentionality” of the robot as a force that represents the gap between how the robot intends to move and how the tutors intend to guide. However, these errors were minimized in the subsequent tutoring sessions because the tutor starts to follow the self-generated movements of the robot as long as the task specifications are satisfied. Consequently the joint position trajectories satisfy the constraints imposed by the task specifications by accommodating the innate characteristics of the network.

Although the idea of codevelopment between trainers and trainees has been well discussed in the context of developmental psychology or education, they have not been applied to issues in robot development. The study done by Nagai et al [44] might be one exception. In that study, the robot was tutored to direct the camera head in the direction that the human tutor is visually attending to. It was shown that the learning control by the tutor from easy tasks to difficult ones by monitoring the performances of the learner makes the learning process easier.

Some in the field [11], [10] of the developmental robotics [45] have shown that sensory-motor skills can be developed efficiently if functions of innate reflex behaviors are

appropriately coupled with sensory-motor mapping learned later. Further study might combine the idea of coupling the innate behaviors and the later learning with our proposed codevelopment tutoring scheme.

Wolpert and Kawato [24] and Demiris and Hayes [46] have also proposed mixture of local experts type architectures which are similar to ours described in the experiment-2. However, our motivation to utilize this type of architecture has been different from these studies. Our main motivation has been to investigate brain-inspired mechanisms for the sensory-motor flow articulation with utilizing this type of architectures. For this purpose, Tani and Nolfi [32] proposed that the mixture of RNN experts with multiple levels can learn to perceive the sensory-motor flow as hierarchically articulated. Tani [47] further discussed that the observed mechanism of the articulation could account for the psychological mechanism for event perception and the phenomenological questions of subjective time perception by Edmund Husserl [48]. The current study showed that complex sensory-motor flow experienced through the tutoring process can be learned by decomposing them into reusable segments by the gating mechanism.

One interesting discussion here might be that whether these sensory-motor segments can be regarded as behavior primitives or not. According to the motor schemata theory by Arbib [30], behavior primitives should generalize their functions in various situations. It is considered that in our experiment the sensory-motor functions acquired as experts in modular networks have achieved generalization but with certain limitations. They can generalize position differences of the object by utilizing the differences in the visual inputs. But they can do neither for variations in speed of all movement patterns nor in amplitude in generating cyclic patterns (e.g., size of swing-up in **Li** and that of lateral swing in **Tch**). A single expert network can generate the movement pattern only with the same speed and the same displacement. In order to generate these variations in movements, additional adaptive parameters which can modulate the sensory-motor function might be required in each local network. This could be achieved by introducing the parametric biases (PB) scheme [15] to the current gating networks model. It is expected that the PB vector implemented in the input layer of each local network would play the roles of the adaptive parameters in order to modulate movement speed or amplitude. Aside from this, we can recognize other type of generalization in encoding of cyclic patterns. It is reminded that the expert networks that acquire cyclic patterns do not encode number of cycle times demonstrated by tutors. The number of cycle times are encoded by the higher level network in terms of timing of gate switching. As have been discussed, the lower level expert network can generate cyclic patterns with infinite cycle times as the patterns are embedded in limit cycling attractors. This mechanism of generating arbitrary number of cycle times by means of self-organized attractor dynamics incorporated with the gating mechanism can provide another type of generalization in generating cyclic movement patterns. In addition, it was shown that those segments of learned sensory-motor patterns can be utilized in a compositional way for the purpose of reconstructing complex task behaviors. They generalize in

the ways of their appearances in different sequential contexts. These discussions summarize that a set of sensory-motor flow segments acquired in the gating networks can be regarded as behavior primitives but with some limitations in their generalization competency.

Calinon et al [29] introduced a novel imitation learning scheme by combining Principal Component Analysis and Gaussian Mixture Model for extracting correlation structures among different modalities. The proposed scheme can extract different types of behavior constraints such as the relative position constraint and the absolute position constraint from the data acquired in demonstrations by experimenters. The relative position constraint corresponds to the positional relationship between robot hands and an object to be grasped at variable positions. The absolute position constraint does to absolute values of arm joint positions when reaching to a specific position. An interesting result in their study is that these constraints are time dependent during each task and their scheme can extract such time dependent correlation structures. On this point, it is considered that the CTRNN scheme of ours can deal with the same problem. Actually, the current paper demonstrated that the robot can learn to touch the object located at variable positions and then to reach to a specific point. The main difference between two approaches is that our scheme utilizes attractor characteristics of deterministic dynamical systems and the scheme by Calinon and his colleagues does with a probabilistic framework. Future studies would explore advantages and disadvantages between the two.

One interesting future study is about incremental learning of new behavior primitives. Our preliminary experiments on this topics using the hierarchical networks showed that an additional learning of a new task containing a novel behavior primitive often interferes with memories of pre-learned behavior primitives. It seems necessary that the pre-learned behavior primitives should be retrained by rehearsing the old teaching data while a new behavior primitive is trained, which is related to the memory consolidation learning scheme discussed by one of the authors [49].

In the current paper, the results of experiment-2 were described for one subject case with one parameter setting. Future work should conduct some parametric studies on the development learning processes with having more repeated experiments. It is especially interesting to examine how number of local modules and time constant value at each level affect the processes of self-organization and segmentation for behavior primitives. Our preliminary study [50] suggested that number of behavior primitives memorized cannot scale simply with number of local networks prepared in the system. The gating becomes unstable with increasing number of patterns to be memorized and that of modules. There remains a space for further investigations for more stable gating mechanisms in achieving the scaling of memory capacity.

Finally, the issue of the compositionality is revisited. The current paper discussed that each task behavior is decomposed into sequences of behavior primitives which are reusable in different task contexts. It was also discussed that the behavior primitives should not be regarded as concrete objects

of encoding fixed patterns. Instead, each behavior primitive should be acquired as “elastic” enough such that its profile can be flexibly modulated as to adapt to various situations. It is important to understand that behavior primitives and local networks may not have one-to-one type mappings. They can be organized with having redundancy as we have described in the results of experiment-2. It was observed that some behavior primitives are encoded not by a single corresponding local network but by multiple ones each of which has a slightly different internal representation. Such redundancy is beneficial for representing subtle differences in pattern profiles of behavior primitives depending on their ways to be connected to next ones with fluency. Actually, in the result of experiment-2, the gate openings for a behavior primitive **Gr** were slightly different depending on next primitive (**Li** or **Hm**) to follow. The representation for skilled actions seem to require two apparently contradictory factors, namely the decompositionality to elements and the wholeness without seams. Our proposed model seems to be successful in having both factors with less conflicts by achieving, what we may call as, the “organic” compositionality through its self-organization processes.

VI. SUMMARY

Our experiments show that interactive developmental tutoring can teach complex task behaviors to robots. In addition, the hierarchically organized CTRNNs was able to successfully form generalizations of learned experiences by extracting the compositional structures of behavior primitives in a context-dependent manner. Our model suggests possible mechanisms in the inferior parietal cortex for generating pantomimed and actual behavior for object manipulation. The uniqueness of the proposed scheme stems from the physical interactions between the human and robot, the interaction of their intentions, of the tutoring process. More interestingly, this interaction provides a novel communication channel between humans and robots at the phenomenological level through their direct experiences.

ACKNOWLEDGMENT

The study has been partially supported by a Grant-in-Aid for Scientific Research on Priority Areas “Emergence of Adaptive Motor Function through Interaction between Body, Brain and Environment” from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

REFERENCES

- [1] B. K. Horn, *Robot Vision*, MIT Press, Cambridge, MA, USA, 1986.
- [2] R. P. Paul, *Robot Manipulators*, MIT Press, Cambridge, MA, USA, 1982.
- [3] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots”, *The Int. J. of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [4] K.M. Heilman, “Ideational apraxia - a re-definition”, *Brain*, vol. 96, pp. 861–864, 1973.
- [5] H. Sakata, M. Taira, A. Murata, and S. Mine, “Neural mechanisms of visual guidance of hand action in the parietal cortex of the monkey.”, *Cereb Cortex*, vol. 5, pp. 429–438, 1995.
- [6] L. Fogassi, P.F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti, “Parietal lobe: from action organization to intention understanding.”, *Science*, vol. 308, pp. 662–667, 2005.
- [7] H. Yokochi, M. Tanaka, M. Kumashiro, and A. Iriki, “Inferior parietal somatosensory neurons coding face-hand coordination in Japanese macaques”, *Somatosensory and Motor Research*, vol. 20, pp. 115–125, 2003.
- [8] R. Bianco and S. Nolfi, “Evolving the neural controller for a robotic arm able to grasp objects on the basis of tactile sensors”, *Adaptive Behavior*, vol. 12, no. 1, pp. 37–45, 2004.
- [9] K. Doya, K. Samejima, K. Katagiri, and M. Kawato, “Multiple model-based reinforcement learning.”, *Neural Computation*, vol. 14, pp. 1347–1369, 2002.
- [10] G. Metta, G. Sandini, and J. Konczak, “A developmental approach to visually-guided reaching in artificial systems.”, *Neural Networks*, vol. 12, no. 10, pp. 1413–1427, 1999.
- [11] L. Berthouze and Y. Kuniyoshi, “Emergence and categorization of coordinated visual behavior through embodied interaction”, *Machine Learning*, vol. 31, pp. 187–200, 1998.
- [12] F. Kaplan and P.-Y. Oudeyer, “Neuromodulation and open-ended development.”, in *Proceedings of the Third International Conference on Development and Learning (ICDL'04): Developing Social Brains*, 2004, pp. 2–11.
- [13] Long-Ji Lin, *Reinforcement Learning for Robots Using Neural Networks*, PhD thesis, Carnegie Mellon University, 1989.
- [14] J. Tani, “Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective”, *IEEE Trans. on SMC (B)*, vol. 26, no. 3, pp. 421–436, 1996.
- [15] J. Tani and M. Ito, “Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment”, *IEEE Trans. on Sys. Man and Cybern. Part A*, vol. 33, no. 4, pp. 481–488, 2003.
- [16] M. Ito and J. Tani, “On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system.”, *Adaptive Behavior*, vol. 12, no. 2, pp. 93–114, 2004.
- [17] M.I. Jordan, “Attractor dynamics and parallelism in a connectionist sequential machine”, in *Proc. of Eighth Annual Conference of Cognitive Science Society*. 1986, pp. 531–546, Hillsdale, NJ: Erlbaum.
- [18] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks”, *Neural Computation*, vol. 1, pp. 270–280, 1989.
- [19] K. Doya and S. Yoshizawa, “Memorizing oscillatory patterns in the analog neuron network”, in *Proc. of 1989 Int. Joint Conf. on Neural Networks, Washington, D.C.*, 1989, pp. I:27–32.
- [20] M. Iacoboni, “Visuo-motor integration and control in the human posterior parietal cortex: Evidence from tms and fmri”, *Neuropsychologia*, 2006.
- [21] CL. Colby JR. Duhamel and ME. Goldberg, “Ventral intraparietal area of the macaque: congruent visual and somatic response properties”, *J. Neurophysiol*, vol. 79, pp. 126–136, 1998.
- [22] M. Kawato, K. Furukawa, and R. Suzuki, “A hierarchical neural network model for the control and learning of voluntary movement”, *Biological Cybernetics*, vol. 57, pp. 169–185, 1987.
- [23] M. Jeannerod, “The representing brain: neural correlates of motor imitation and imaginary”, *Behavioral and Brain Science*, vol. 17, pp. 187–245, 1994.
- [24] D. Wolpert and M. Kawato, “Multiple paired forward and inverse models for motor control”, *Neural Networks*, vol. 11, pp. 1317–1329, 1998.
- [25] L.-J. Lin and T.M. Mitchell, “Reinforcement learning with hidden states”, in *Proc. of the Second Int. Conf. on Simulation of Adaptive Behavior*, 1992, pp. 271–280.
- [26] R. Paine and J. Tani, “How hierarchical control self-organizes in artificial adaptive systems”, *Adaptive Behavior*, vol. 13, no. 3, pp. 211–225, 2005.
- [27] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, “Learning internal representations by error propagation”, in *Parallel Distributed Processing*, D.E. Rumelhart and J.L. McClelland, Eds., pp. 318–362. Cambridge, MA: MIT Press, 1986.
- [28] C.G. Atkeson and S. Schaal, “Robot Learning From Demonstration”, in *Machine Learning: Proceedings of the Fourteenth International Conference (ICML '97)*, 1997, pp. 12–20.
- [29] S. Calinon, F. Guenter, and A. Billard, “On learning, representing and generalizing a task in a humanoid robot”.
- [30] M.A. Arbib, “Perceptual structures and distributed motor control”, in *Handbook of Physiology: The Nervous System, II. Motor Control*, pp. 1448–1480. Cambridge, MA: MIT Press, 1981.
- [31] Y. Kuniyoshi, M. Inaba, and H. Inoue, “Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance”, *IEEE Trans. on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.

- [32] J. Tani and S. Nolfi, "Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems", in *From animals to animats 5*, R. Pfeifer, B. Blumberg, J. Meyer, and S. Wilson, Eds. Cambridge, MA: MIT Press., 1998, later published in *Neural Networks*, vol12, pp1131–1141, 1999.
- [33] T. Inamura, N. Nakamura, H. Ezaki, and I. Toshima, "Imitation and primitive symbol acquisition of humanoid by the integrated mimesis loop", in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4208–4213. 2001.
- [34] J. Tani, "Learning to generate articulated behavior through the bottom-up and the top-down interaction process", *Neural Networks*, vol. 16, pp. 11–23, 2003.
- [35] Y. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model", in *Imitation in animals and artefacts*, K. Dautenhahn and C. L. Nehaniv, Eds. pp. 1448–1480. Cambridge: MIT Press, 2002.
- [36] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives.", in *Advances in Neural Information Processing Systems 17*, pp. 1547–1554. Cambridge, MA: MIT Press, 2003.
- [37] R.A. Jacobs and M.I. Jordan, "Adaptive mixtures of local experts", *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.
- [38] A.R. Luria, *The Working Brain*, Penguin Books Ltd., 1973.
- [39] H. Liepmann, "Apraxie", *Erg ges Med*, vol. 1, pp. 516–543, 1920.
- [40] N. Geschwind and E.A. Kaplan, "Human cerebral disconnection syndromes", *Neurology*, vol. 12, pp. 675–685, 1962.
- [41] S. McDonald, R. Tate, and J. Rigby, "Error types in ideomotor apraxia: a qualitative analysis", *Brain and Cognition*, vol. 25, no. 2.
- [42] F. Ohshima, K. Takeda, M. Bandou, and K. Inoue, "A case of ideational apraxia -an impairment in the sequence of acts-", *Journal of Japanese Neuropsychology*, vol. 14, pp. 42–48, 1998.
- [43] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, Cambridge, MA: MIT Press, 1986.
- [44] Y.Nagai, K.Hosoda, A.Morita, and M.Asada, "A constructive model for the development of joint attention.", *Connection Science*, vol. 15, no. 4, pp. 211–229, 2003.
- [45] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey", *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.
- [46] John Demiris and Gillian Hayes, "Imitation as a dual-route process featuring predictive and learning components: a biologically plausible computational model", pp. 327–361, 2002.
- [47] J. Tani, "The Dynamical Systems Accounts for Phenomenology of Immanent Time: An Interpretation by Revisiting a Robotics Synthetic Study", *Journal of Consciousness Studies*, vol. 11, no. 9, pp. 5–24, 2004.
- [48] E. Husserl, *The phenomenology of internal time consciousness, trans. J.S. Churchill*, Indiana University Press, Bloomington, IN, 1964.
- [49] J. Tani, "An interpretation of the "self" from the dynamical systems perspective: a constructivist approach", *Journal of Consciousness Studies*, vol. 5, no. 5-6, pp. 516–42, 1998.
- [50] J. Namikawa and J. Tani, "Learning to segment temporal sequences by mixture of rnn experts with adaptive variance model", 2007, arXiv:0706.1317v1, <http://aps.arxiv.org/abs/0706.1317v1>.

PLACE
PHOTO
HERE

Jun Tani Jun Tani received the B.S. degree in mechanical engineering from Waseda University, dual M.S. degree in electrical engineering and mechanical engineering from University of Michigan and Dr. Eng. from Sophia University. He started his research career in Sony Computer Science Laboratory in 1990. He has been appointed as a team leader in Lab. for Behavior and Dynamic Cognition, Brain Science Institute, RIKEN in Tokyo since 2000. He was also appointed as a visiting associate professor in Univ. of Tokyo from 1997 to 2002. He has studied the problems of robot learning with theoretical modeling of complex adaptive systems and neuronal networks for more than 15 years. He has been also interested in phenomenological problems of "self-consciousness" and his studies have addressed those problems from the view of embodied cognition. Five years ago, he started neuroscience studies on behavior learning processes in real brains utilizing both scheme of human brain imaging and animal electrophysiology. His envision is to establish the brain-inspired robotics by integrating these approaches.

PLACE
PHOTO
HERE

Ryu Nishimoto Ryunosuke Nishimoto received his Ph.D. in the Department of General Systems Studies of the University of Tokyo, Japan. Currently, he works as a Researcher at RIKEN Brain Science Institute, Japan. His research interests include generalization in learning, skil learning theory and embodied cognition.

PLACE
PHOTO
HERE

Jun Namikawa Jun Namikawa took M.S. and Ph.D. degrees in 2005 at the Japan Advanced Institute of Science and Technology (JAIST). From April to September in 2005, he was a Technical Staff at the Brain Science Institute, RIKEN. Since October 2005, he has been a Researcher at the Brain Science Institute, RIKEN. He is interested in understanding intelligent systems and evolutional systems based on the theory of dynamical systems and computation.

PLACE
PHOTO
HERE

Masato Ito Masato Ito was born in 1974 in Osaka, Japan. He received the B.S. degree in electrical engineering and the M.S. degree in computer science from the Keio University, Tokyo, Japan, in 1997 and 1999, respectively. His main research interests concern open-ended development mechanisms for intelligent systems.