

Achieving "Organic Compositionality" through Self-organization:
Reviews on Brain-Inspired Robotics Experiments
(To appear in Neural Networks in 2008.)

Jun Tani, Ryunosuke Nishimoto and Rainer W. Paine*

RIKEN Brain Science Institute

2-1 Hirosawa, Wako-shi, Saitama, 351-0198 Japan

Tel +81-48-467-6467, FAX +81-48-467-7248

E-mail tani@brain.riken.go.jp

Acknowledgment

This study was conducted with collaborations with Sony Corporation. 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. Requests for reprints should be sent to Jun Tani, Brain Science Institute, RIKEN. 2-1 Hirosawa, Wako-shi, Saitama, 351-0198 Japan

Running Title

Achieving Organic Compositionality

Key Words

Compositionality, self-organization, CTRNN, robot, prediction, hierarchy, parietal cortex, premotor cortex, mirror neurons, dynamical systems

*Currently at Dartmouth-Hitchcock Medical Center, USA, Dept. of Neurology

Abstract

The current paper examined how compositional structures can self-organize in given neuro-dynamical systems when robot agents are forced to learn multiple goal-directed behaviors simultaneously. Firstly, we propose a basic model accounting for the roles of parietal-premotor interactions for representing skills for goal-directed behaviors. The basic model had been implemented in a set of robotics experiments employing different neural network architectures. The comparative reviews among those experiment results address the issues of local vs distributed representations in representing behavior and the effectiveness of level structures associated with different sensory-motor articulation mechanisms. It is concluded that the compositional structures can be acquired “organically” by achieving generalization in learning and by capturing the contextual nature of skilled behaviors under specific conditions. Furthermore, the paper discusses possible feedback for empirical neuroscience studies in the future.

1 Introduction

We humans learn our behavior skills through repetitions of own behavior experiences. We learn to manipulate objects, to use tools and to navigate to desired locations. When we investigate the capabilities of human or artificial agents to generate diverse and complex skilled behaviors, mechanisms for behavior compositionality become an essential issue. The term compositionality here is adopted from “Principle of compositionality” (Evans, 1981) in linguistics which claims that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them. This principle can be translated in our problem such that skilled behaviors can be generated by combining a set of re-usable behavior primitives adaptively corresponding to required goals by following rules. For example, an attempt of drinking a cup of water might be decomposed into multiple behavior primitives such as reaching to a cup, grasping the cup and moving the cup toward ones’s mouth. Each behavior primitive can be re-utilized also as a component for other goal-directed actions e.g. reaching to a cup can be used for another goal such as clearing it away. Also it is noted that this idea of decomposition of whole behaviors into sequences of reusable primitives is parallel to the motor schemata theory by Arbib (1981).

However, the compositionality assumed in motor behaviors might have different aspects from the one considered in conventional linguistics. Firstly, when we think about the behavior primitives, they should not be regarded as concrete objects. Instead, each behavior primitive should be flexible enough to be used in various situations. For example, a behavior primitive of grasping an object should be flexible enough to be used for different object positions and shapes. This requires generalization in learning of the skills through experiencing variations of sensory-motor flow, like practicing to grasp objects with different shapes and positions. Secondly, the rules for combining behavior primitives in sequences should be “fluid” and context-dependent rather than formalistic. For example, exact motor trajectories of grasping a cup should be affected by the next motor act to follow them as well as the entire goals of whether to drink a cup of water or to clean off the cup. The whole motor behaviors should be generated fluently by capturing the context-dependent nature of intended action goals. The requirement here is that behavior generation systems should afford certain compositionality in manipulating primitives while their rules and elements are fluid and elastic enough such that they can entail the tacit knowledge of skilled behaviors. In the current paper this sort of compositionality is called “organic” compositionality.

Then, the essential question is how this sort of “organic” compositionality can be

achieved in behavior generation systems. The question is deeply related to the sensory-motor articulation problem which has been investigated by some research groups (Kuniyoshi, Inaba, & Inoue, 1994; Tani & Nolfi, 1999; Yamamoto & Kuniyoshi, 2002). The problem asks how experiences of continuous sensory-motor flow as a whole can be segmented into sequences of meaningful parts. Kuniyoshi et al. (1994) addressed this articulation problem in their robot learning experiment studies. In their experiment with an assembling robot, the robot recognizes the various task performances by decomposing them into sequences of modular representations. Subsequently, the robot is able to learn various tasks in terms of combinations of the reusable modular representations obtained. For attaining such a modular representation, the task performance was temporally segmented by means of detecting “meaningful changes” in the observed sensory flow. The problem, however, is that the definitions of these meaningful changes were predetermined by designers. Our prior study (Tani & Nolfi, 1999) showed that robots can learn to recognize meaningful changes by themselves and can perceive a continuous sensory-motor flow as segmented into reusable behavior primitives with utilizing the gated networks scheme (Wolpert & Kawato, 1998; Tani & Nolfi, 1999). The similar scheme will be introduced in one of the experiments in the current paper.

It might be inadequate to regard behavior primitives or motor schemes as just reflex-type functions that map the current sensory state to next motor state. Instead, the functions should involve anticipation by which diverse sensory-motor flows achieving various goals can be imaged (Ito, 1970; Kawato, Furukawa, & Suzuki, 1987; Jeannerod, 1994; Tani, 1996). When the sensory-motor flow of a particular act can be imaged, the same act by others can be recognized (Oztop & Arbib, 2002; Oztop, Wolpert, & Kawato, 2005; Tani & Ito, 2003; Ito & Tani, 2004), as will be explained later in terms of mirror systems (Rizzolatti, Fadiga, Galles, & Fogassi, 1996).

In the brain science perspective, we speculate that structures responsible for generating or mentally simulating skilled behaviors “organic” compositionality might be acquired in the inferior parietal lobe (IPL) through repeated sensory-motor experiences. Our ideas have been inspired by Ito (2005) who suggested that information from daily sensory-motor experiences is first consolidated in parietal cortex and then further consolidated into the cerebellum as internal models (Kawato et al., 1987) related to actions. Conventionally, parietal cortex has been viewed as a core site to associate and integrate the multi-modality of the sensory inputs. However, the neuropsychological studies investigating various apraxia cases, including ideomotor apraxia and ideational apraxia (Liepmann, 1920; Heilman, 1973), have suggested that IPL should be also an essential

site to represent a class of behavior skills, especially related to object manipulations. We speculate that this region might function especially as a predictor for future sensory inputs for particular goals of actions based on neuroscience evidence that will be detailed in later sections. Furthermore, it has been speculated that IPL might function both for generating and recognizing the goal-directed behaviors (Fogassi et al., 2005) by having dense interactions with ventral premotor (PMv) cells which are known as mirror neurons (Rizzolatti et al., 1996).

The current paper reviews our recent trials on synthetic robotics approaches (Nishimoto, Namikawa, & Tani, 2008; Tani, Nishimoto, Namikawa, & Ito, 2008; Paine & Tani, 2005) to model the PMv-IPL networks for representing goal-directed behavior skills. We present three different architectures adopted for different robot experiments including a simulation in which each experiment has a specific modeling aspect. However, all three experiments share the same research motivation that asks how “organic” compositionality can be self-organized in some forms of neuro-dynamical systems when the robots are required to learn multiple goal-directed behaviors simultaneously. If such pressures could actually lead to the self-organization of compositional structures, next questions might be how modular or level structures of the networks could enhance the self-organization of such structures. Those research interests will be examined from the dynamical systems view point (Schoner & Kelso, 1988; Kelso, 1994; Beer, 1995; Tani & Fukumura, 1994; Schaal, Ijspeert, & Billard, 2003). The repertoire of dynamical systems language, including attractor, bifurcation and initial sensitivity, would qualitatively describe the characteristics of “organic” compositionality observed in those experiments.

After obtaining these results in the synthetic robotics study, the current paper discusses how the results can be reconciled with empirical neuroscience studies. It is considered that empirical studies can show us certain evidence in real brains, but that they rarely describe their underlying neuronal mechanisms. On the other hand, the synthetic studies can handle possible mechanisms directly in computational models but without assuring that such mechanisms can exist in real brains. In this situation, it is expected that cyclic collaborative research between proposing possible models for brain mechanisms in the synthetic studies and proving the models in reality in the empirical studies could improve our understanding of brain mechanisms substantially.

Before closing this section, we briefly describe the specific goal of each modeling and robotics experiment shown in the current paper. The first experiment by Nishimoto et al. (2008) examines how the simplest implementation of a dynamic neural network in modeling the parietal-PMv interactions can achieve both generation and recognition

of a set of goal-directed behaviors of a robot in a simple object manipulation learning task. The second experiment by Tani et al. (2008) examines how the scheme of using the dynamic neural network model can be scaled by introducing some modular and level-structured architecture and incorporating a developmental learning scheme. The third experiment by Paine and Tani (2005) examines the emergence of level structures in answering the question of how levels and modules are crucial in representing compositional behavior skills.

2 Premotor and parietal cortex interactions

This section describes our interpretations for the functional roles of IPL and PMv in representing goal-directed behaviors by their interactions. Conventionally, parietal cortex has been regarded as a cortex region for associating multi-modal sensory information including visual, auditory, tactile and proprioceptive sensations (Colby, Duhamel, & Goldberg, 1993). The sensory information integrated in parietal cortex is considered to be sent to PMv for organizing the corresponding motor programs which are further sent to the primary motor area (M1) for generating more detailed programs (Sakata, Taira, Murata, & Mine, 1995).

Fagg and Arbib (Fagg & Arbib, 1998) introduced the so-called FARS model which attempts to explain how PMv and AIP (the Anterior Intra-Parietal sulcus) in parietal cortex can generate object grasping behaviors. In this model, the visual information of an object is sent from the IT area in the visual ventral pathway to AIP. AIP extracts "affordance" information of the object which is a set of important visual properties of the object to be grasped. This affordance information is sent to canonical neurons in PMv where the corresponding motor program of grasping the object is generated by means of an inverse model.

Oztop and Arbib (Oztop & Arbib, 2002) proposed a model of mirror neurons (Rizzolatti et al., 1996) found in PMv which recognizes the goal-directed behaviors of oneself and of others. They proposed that information about the positional relation between an object and the hand of an other person or animal is extracted and represented in AIP neurons. Then, this AIP neural representation is mapped to categorical activation of the mirror neurons in PMv. (They employ a simple three-layered perceptron-type neural network to learn this mapping from the sensory feature inputs to the categorical outputs.)

One common idea in these two models by Arbib's group is that parietal cortex is considered to preprocess the sensory information prior to PMv in both cases of the

behavior recognition and generation. It is assumed that parietal cortex may deal with sensory inputs merely as static patterns rather than temporally changing ones. On this account, we speculate that parietal cortex may function as forward models to anticipate coming sensory inputs.

Our interpretation originated from literature of neuropsychological studies. It is well known that patients whose IPL are impaired often suffer from deficits in the usage of tools (Liepmann, 1920; Geschwind & Kaplan, 1962). The deficit is called ideomotor apraxia. Because their basic movements of limbs are not impaired, it is plausible to consider that IPL contains tacit knowledge about the skills to use the tools. More interestingly, some of ideomotor apraxia patients have deficits only in pantomiming but not in actually using the tools (Liepmann, 1920; Geschwind & Kaplan, 1962; McDonald, Tate, & Rigby, ; Ohshima, Takeda, Bandou, & Inoue, 1998). The pantomime requires a capability of generating mental imaginary of sensory inputs associated with the actual tool usages for their self-feedback. Because the chain of sensory imaginary can be generated by means of look-ahead prediction using the forward models (Tani, 1996; Hesslow, 2002), it is speculated that the deficit in pantomime might be originated from impairment of the sensory forward prediction mechanism assumed in IPL.

Eskandar and Assad (1999) investigated the role of the posterior parietal cortex (PPC) in the visual guidance of movements in monkeys trained to use a joystick to guide a spot to a target. In the electrophysiological experiments of the monkeys, they found cells in the lateral intraparietal area (LIP) which seem to encode a predictive representation of stimulus movement. Ehrsson, Fagergren, Johansson, and Forssberg (2003) observed specific activity in posterior parietal cortex (PPC) during control of fingertip forces for grasp stability in human fMRI scan experiments. Because this force coordination requires anticipation of the grip forces that match the requirements imposed by the self-generated load forces, PPC is assumed to implement such anticipatory mechanism.

Another clue came from recent studies on the involvement of the medial parietal regions of monkeys with goal-directed navigation by Taira et al (Sato, Sakata, Tanaka, & Taira, 2006). In their experiments, monkeys were trained to navigate to reach specific goal locations in a virtual office environment. The monkeys watched the computer simulated egocentric view, which was projected into a front screen, and maneuvered the virtual workspace using a joystick controller. In the examination of the neural activities in the medial parietal region, large portions of neurons activate with specific movements (turning left or right) at specific positions. Moreover, some neurons respond in a goal-dependent manner i.e. - they respond not only depending on specific position

and movement but also on specific goals. This result suggests that the medial parietal region stores route-based navigation knowledge or, in other words, the internal models of the workspace.

In summary, the evidence suggests that the parietal cortex implements different types of sensory anticipation functions at different local regions depending on the goal of behavior. In the following, our modeling focuses on sensory anticipation mechanisms assumed especially in IPL in the parietal cortex. IPL is known to have a dense connectivity with mirror neurons in the PMv. Our modeling assumes that the anticipation mechanisms in IPL might be different from the one considered in the conventional forward model (Kawato et al., 1987; Wolpert & Kawato, 1998) hypothesized in the cerebellum. The conventional forward model predicts the next step sensory outcomes regarding the current action taken and the current sensory state given. In our model, IPL predicts the next step sensory state only with the current sensory state given. We may call this modified version of the forward model as the sensory forward model.

The sensory forward model, however, requires one more piece of information, that is goals for current actions. For example, let us consider a situation in which there is a coffee mug in front of us that is to be manipulated. Our anticipation of sensory sequences is based on the visual image of the mug and its relative position among the fingers and the arm. It is also based on the proprioception of each joint angle for the fingers and arms. The anticipation of sensory experiences might be different depending on our current goals of grasping the mug for drinking coffee or for throwing it toward somebody. Once the goal is set, the sensory sequence associated with this goal-directed behavior can be predicted. Therefore, in our formulation the sensory forward model is provided with the goal state as well as the current sensory state in its inputs and then the next step sensory state is predicted in the outputs. It is important to note that the sensory forward model cannot play an equivalent role of an internal model or a world model that can tell sensory consequences for whatever motor command sequences. The function of the sensory forward model is much limited in a sense that it can predict sensory sequences only for a set of task goals which are frequently experienced. Therefore, it can be said that the sensory forward model can work only for skilled goal-directed behaviors rather than arbitrary motor behaviors.

It is also noted that the current sensory input could be provided by feeding back its own sensory prediction without having the actual sensory input. This enables the sensory imaginary loop required for the pantomime behaviors, as described previously.

An obvious question is then where the current goal information comes from. We assume that it comes from the mirror neurons in PMv. This assumption accords with

the main arguments by the Rizzolatti group (Rizzolatti et al., 1996) that the mirror neurons do not encode exact movement profiles but encode the underlying goals of movements. Now, the sensory sequence is predicted in IPL with the goal information provided from PMv. Here, we show another assumption which could account for how actual motor behaviors can be generated from the predicted sensory sequences. The basic idea is that the predicted sensory sequences for given goals contain enough information about the corresponding motor sequences. We assume the predicted proprioception is sent to primary sensory cortex (S1) and further sent to primary motor cortex (M1) for generation of actual motor command. The analogy in robot joint control is that the position encoder value at the next time step is predicted (IPL) and set as the desired position (S1). Then, the motor controller (M1) computes necessary motor torque command. This part of the assumption is inspired by observations by Fetz and his colleagues (Fetz, Finocchio, Baker, & Soso, 1980; Soso & Fetz, 1980) about S1 involvement in motor generation. They found that some monkey S1 cells are activated immediately before the actual movements of limbs, as if preparing for the movements. Figure 1(a) summarizes of the behavior generation pathway which is assumed in the current paper.

The goals of others' behaviors can be recognized by going inversely through the generation pathway described above. In the case of recognizing a person manipulating an object, the sensory forward model can predict how the visual image, for example, the extracted features about positional relations between the object and the hand (Oztop & Arbib, 2002), evolves in time from one's own experiences if the goal of the person is correctly set. If the goal state is set incorrectly in the sensory forward model, the prediction for the coming visual sensory image generates error. In this situation, the prediction error can be minimized by searching for the optimal goal state. If the goal state is eventually found such that the error becomes negligibly small, it can be said that the goal of the behavior by this person is recognized.

Previously, we implemented this idea into a robot that can both generate and recognize a set of arm dancing patterns by itself and by others by using a scheme called the recurrent neural network with parametric biases (RNNPB) (Tani & Ito, 2003; Ito & Tani, 2004). In this scheme, the parametric bias (PB) units allocated in the input units of a jordan-type recurrent neural network (RNN) (Jordan, 1986) encodes the goal state where the RNN functions as a sensory forward model. By setting the goal state in the PB units, the corresponding behavior can be generated. On the other hand, the goal of observed behaviors by others can be recognized by optimizing the PB values for minimizing the sensory prediction error in the RNNPB. Therefore, it can be said the

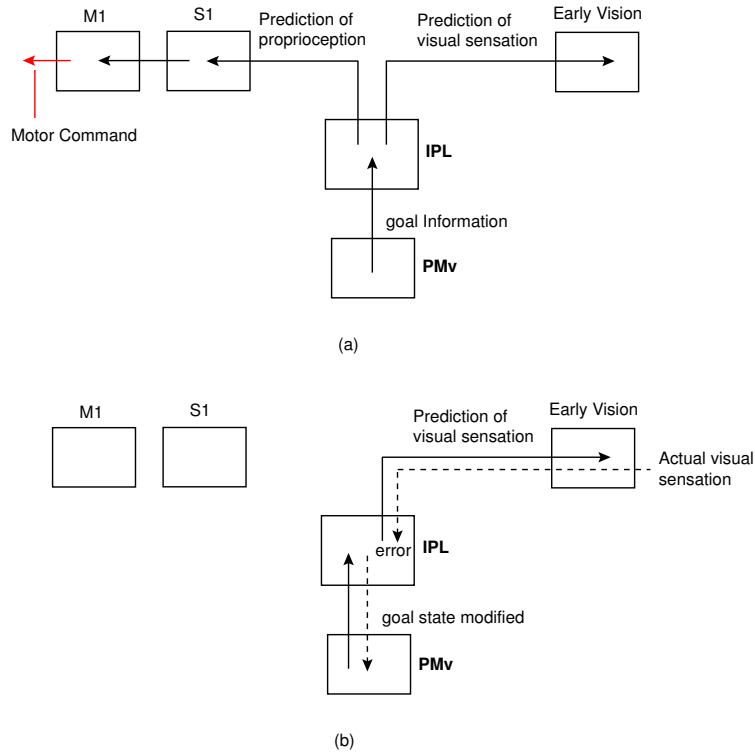


Figure 1: (a)The behavior generation pathway. Mirror neurons in PMv send the goal information to the sensory forward model in IPL and the sensory forward model generates the predicted sensory sequence. The predicted proprioceptive state is sent to S1 and further to M1 where the corresponding motor command is generated. (b) The behavior recognition pathway. The error between the expected visual image and the real outcome is propagated back to PMv where the mirror neuron activation patterns are modulated in the direction of minimizing the error.

generation and the recognition processes are mirrored by means of the PB mechanism in an abstract sense. The scheme of the PB encoding of the goal state is only one possible implementation that follows the assumptions described above. The current paper will introduce the variations of this scheme. Figure 1(b) illustrates the summary of the behavior recognition pathway which is assumed in the current paper.

Oztop et al (Oztop et al., 2005) proposed another model of mirror neurons, in addition to their former model (Oztop & Arbib, 2002), by concatenating an inverse model and a forward model. In their model, the inverse model generates the imaginary motor sequences with the inputs of the estimated goal state (the mental state in their word) of others. Then the forward model generates the prediction of the coming sensory

sequences with the inputs of the imaginary motor sequences. The estimate of the goal state in the inverse model is optimized such that the prediction error with the observed sensory sequences is minimized. In this manner, the goal of the observed behaviors can be recognized. Their model is different from ours in that their internal model is implemented by the concatenation of the inverse model and the forward model and is assumed to be located in PMv rather than in IPL. Our current model, as well as our prior model of the RNNPB, (Tani & Ito, 2003; Ito & Tani, 2004) is considered to be simpler in the sense that the goal state can be estimated directly from the sensory forward model.

There are several computational models that aim to achieve mirror systems in robots (Billard & Mataric, 2001; Inamura, Nakamura, Ezaki, & Toshima, 2001; Demiris & Hayes, 2002; Schaal et al., 2003; Inamura, Toshima, Tanie, & Nakamura, 2004). Billard and Mataric (2001) use their invented connectionist-type network called DRAMA for imitation learning. Inamura et al. (2001) utilize a hidden markov model for representing behavior primitives and their sequential combinations. Demiris and Hayes (2002) utilize a set of gated-modular networks for representing behavior primitives whose architecture is similar to MOSAIC (Wolpert & Kawato, 1998) and the mixture of RNN experts (Tani & Nolfi, 1998). Schaal et al. (2003) proposed to describe discrete and cyclic movement primitives in terms of two parameterized canonical equations of fixed point dynamics and limit cycling dynamics, respectively. Inamura et al. (2004) proposed the so-called mimesis space for representing behavior patterns in a certain metric space using artificially designed manifolds. The representation can be utilized both for generation and recognition. Although each of those models has a specific computational mechanism of achieving the mirror system for robots, its biological correspondence to networks in PMv and IPL are not discussed well.

In the current study, a continuous-time recurrent network (CTRNN) (Williams & Zipser, 1989; Doya & Yoshizawa, 1989; Beer, 1995) is used for the purpose of modeling the sensory forward model assumed in IPL in a connectionist level abstraction. The usage of the error back propagation learning scheme (Rumelhart, Hinton, & Williams, 1986) which may not represent real neuron synaptic modulation mechanisms, can be accounted for by the same reason.

The CTRNN has the advantage of having rich dynamics defined in a continuous-time and space domain which allows the network to have dense interactions with analog sensory-motor flow. An interesting characteristic of the CTRNN is that the time constant parameter of each unit can regulate the time scale of its activation dynamics. It is, however, noted that activation of a single unit in a CTRNN does not correspond

to the firing dynamics of a single real neuron. Instead, activation of a single unit in a CTRNN may correspond to an average firing rate of populations of real neurons. In this situation, the time constant parameter of a unit may represent the time constant of macroscopic dynamics consisting of population of neuronal activities. This time constant at the macroscopic level should depend on the connection paths among local neurons and their synchronal characteristics. For example, a group of cells affected by distant neurons with large variations of spike timings, which are often observed in prefrontal cortex (Sakurai & Takahashi, 2006), might be modeled by a unit with slow time constant dynamics. On the other hand, closely neighboring neurons having sharply correlated firings, such as those observed in the primary motor cortex (Hatsopoulos, Ojakangas, Paninski, & Donoghue, 1998), might be modeled by a unit with a fast time constant. Our hypothesis here is that such a difference of time constants in macroscopic neuronal dynamics might lead to the emergence of levels of behavior generation.

One of the main motivations of the current paper is to seek brain-inspired mechanisms to achieve a compositional representation of skilled behaviors. Although it is hardly confirmed that IPL and premotor cortex are actually involved with generation of such compositional structures, some observations of ideational apraxia patients may suggest its plausibility. Heilman (Heilman, 1973) defined ideational apraxia as an inability to put acts into sequences. Ohshima et al (Ohshima et al., 1998) found that patients with lesions in left parieto-occipital lobe cannot execute or pantomime skilled acts with objects in sequence. However, these patients can successfully execute each element of these acts. The observed dissociation between the elements and their combined sequences suggest that there might exist a level structure in which only the higher level for combining the elements into the sequences might be impaired in those patients. Our modeling studies accompanied with robot experiments will examine possible effects of level structures in achieving the compositional representation of skilled behaviors.

3 Learning with a simple CTRNN: experiment-1

In our 1st robot experiment (Nishimoto et al., 2008), a humanoid robot is tutored by supervisors to perform multiple goal-directed behaviors concerning simple object manipulations. Although the intended goal-directed behaviors might be seen as simple, they are actually structured to share some behavior primitives.

A single CTRNN model (Williams & Zipser, 1989; Doya & Yoshizawa, 1989; Beer,

1995) is utilized in this experiment. Then, a question is how it can learn to encode multiple goal-directed behaviors through the sensory motor experiences. The current study investigates the possibility of encoding the task goal information by utilizing the initial sensitivity characteristics of the CTRNN dynamics. More specifically, the dynamics of a CTRNN is trained to produce trajectories reaching to the multiple goal state, depending on the initial internal state of the CTRNN. The mapping of the goal states of actions to the initial internal state is acquired through the training process. In the experiments, we introduce some variations in the adopted environment by changing the position of the object to be manipulated at each trial. By doing this, we examine how robustness in performing goal-directed behaviors can be achieved under variable situations and how this sort of robustness is related to generalization in learning. The next subsection introduces the CTRNN model employed in this experiment.

3.1 A simple CTRNN model

Overview

The proposed network model is designed to learn to re-generate sensory sequence patterns represented as (s_t, m_t) . Here, s_t represents vision-based sensory information at time step t , representing an object’s position attended by the robot video camera. The other component of the sensory information m_t represents proprioception of the arms of the robot by reading encoder values in the arm joints. Essentially, this network takes input as different modalities of sensation and mingles those inputs together to generate predictions of their time developments in the future.

We designed our CTRNN model, shown in Figure 2, by modifying the Jordan-type RNN – a discrete time model. In this model, the current sensory state and the context state, representing the current internal state of the network, are represented as a^{Bx} and a^{Bc} , correspondingly in the bottom layer units. The activations in the hidden layer a^H are computed by means of standard sigmoidal activation of the input sum obtained from the bottom layer activation multiplied through the synaptic connections w_{ij}^H . And then, a^{Ux} and a^{Uc} in the upper layer are computed in the same way by using the synaptic connections w_{ij}^U . The activations of a^{Ux} and a^{Uc} are used only internally for computing the direction of changes in time predicted for the sensory state a^{Bx} and the context state a^{Bc} , correspondingly. The sensory feedback loop and the context feedback loop shown in Figure 2 instantiate this integration. Therefore, it is noted that the units of a^{Bx} in the bottom layer function both for receiving the sensory inputs and predicting their future as the outputs.

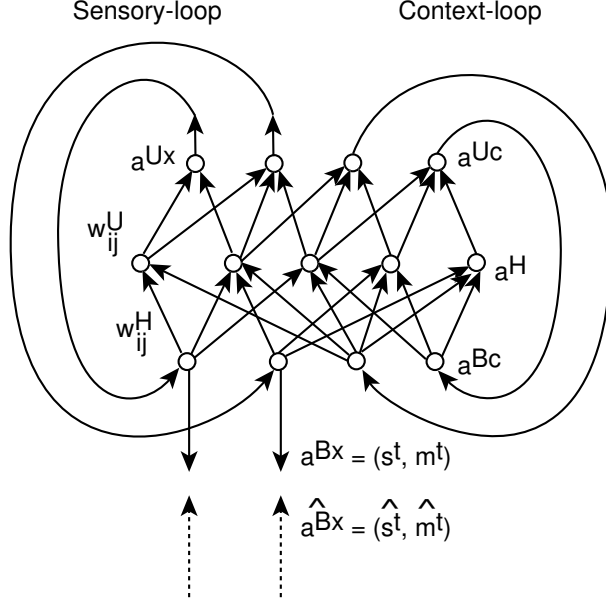


Figure 2: The CTRNN model employed.

Forward dynamics

This part of the time integration is detailed in the following. The next time step activations of those units a_{t+1}^{Bx} and a_{t+1}^{Bc} are obtained by integrating the first order differential equation of Eq.1 in which the upper layer outputs a_i^U work as force inputs to drive the internal potential of these units u_i^B . The actual updates of these values are computed with the numerical approximation of Eq.2 associated with the sigmoidal transformation in Eq.3.

$$\tau \dot{u}_i^B = -u_i^B + a_i^U / \alpha \quad (1)$$

$$u_{i,t+1}^B = -u_{i,t}^B / \tau + (a_{i,t}^U - 0.5) / (\alpha \cdot \tau) + u_{i,t}^B \quad (2)$$

$$a_{i,t+1}^B = \text{sigmoid}(u_{i,t+1}^B) \quad (3)$$

where τ and α denote the time constant and the input force coefficient, respectively. It is noted that the time constant τ determines the dynamic characteristics of the network substantially. If τ is set larger, the network easily adapts to slower dynamics of the sensory flow. Otherwise it adapts to faster dynamics. Although in the current and the next experiments τ is set manually by experimenters, the 3rd experiment will show how τ can be determined autonomously. The prediction of the proprioception m_t is utilized to generate the actual motor command of the robot by a position feedback controller at each joint. On the other hand, the prediction of the vision state s_t is used

for recognition of the observed behavior, as will be described later.

Training

The goal of training the network is to minimize the error between the teaching sequence pattern \hat{a}_t^{Bx} given from the outside and the predicted sequence pattern a_t^{Bx} generated by itself. The network learning method uses the general Back Propagation Through Time (BPTT) algorithm (Rumelhart et al., 1986). Using the BPTT algorithm, the network obtains shared weight connectivities for all the given teaching sequence patterns. At the same time, in the proposed model, the initial values of the context units a_0^{Bc} for each trained behavior are iteratively computed such that the error between the re-generated sequence starting from the initial values obtained and the training one can be minimized. The aim is for each desired goal-directed behavior to be generated by setting its corresponding initial values into the context units after the learning process has converged.

In the actual learning process, an update of a connective weight from the i -th unit to the j -th unit at the n th learning iteration step is obtained by using the delta error δ_i back-propagated to the i -th unit in the following equation.

$$\Delta w_{ij}(n+1) = \eta \delta_i a_j + \alpha \Delta w_{ij}(n) \quad (4)$$

In addition, an update of the initial context unit values corresponding to each training sequence pattern is obtained simultaneously by utilizing the delta error δ_i back-propagated through time steps to the context unit at the initial step.

$$\Delta c_{0,i}(n+1) = \eta \delta_{0,i} + \alpha \Delta c_{0,i}(n) \quad (5)$$

In the actual experiment, only two dimensions of the initial context state are adapted and all other dimensions are set to 0.5.

Generation and recognition

After the network is trained, the network can be operated both in the generation mode and the recognition mode. In the generation mode, once the initial context values are set, the network generates the corresponding sensory prediction sequences either in the open-loop mode by receiving the sensory inputs or in the closed-loop mode by having the self-feedback of the sensory prediction sent into the inputs. In the recognition mode, some dimensions of the sensory sequences are given to the network as a target sequence. Then, the optimal initial context state is searched by following Eq.5 such that the error between the sequence pattern generated from this initial context state and the target sequence pattern can be minimized. Consequently, the goal for the observed sequence is recognized by obtaining the optimal initial context state.

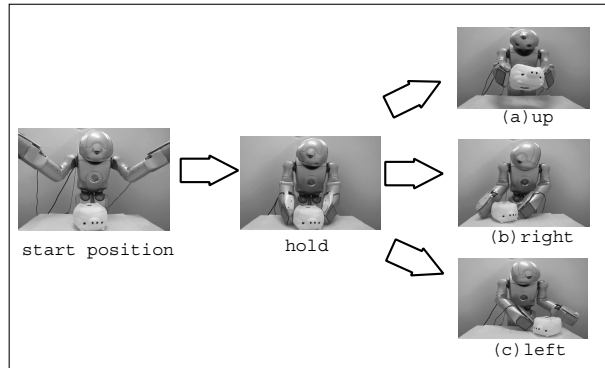


Figure 3: Three type of the goal-directed behaviors aimed in the experiment.

3.2 Robot task in experiment-1

The experiments are performed using a small-scale humanoid robot produced by SONY. It is aimed at autonomously generating three types of operational actions on objects, as shown in Figure 3, based on learning. In those goal-directed behaviors, the robot starts to move from the same start position to approach and to hold an object placed in front of the robot. The object is then either (a) held up, (b) moved to right, or (c) moved to left. The object is marked with a red color point which the robot camera mounted on its head attempts to target all the time during the operation. The direction of the robot head in terms of encoder values of two head joints represents the object position relative to the robot. This 2-dimensional vector corresponds to the visual input denoted as s_t in the previous section.

The robot has two arms each of which has 4 degrees of freedom without fingers. The encoder values of these arm joints are received as the current proprioceptive inputs m_t and its next step prediction m_{t+1} is sent to the joint controller to generate their actual movements. The CTRNN used for this experiment is allocated with 20 hidden units, 20 context units and 10 sensory inputs.

In order to examine the generalization capability of the object holding action with regard to variance in object position, the robot is trained with the object located among three different positions, “right”, “left” and “center” in front of the robot. In the training, the experimenters guide the robot arm movements by grasping the arms in order to generate the desired trajectories. The encoder values as well as the visual inputs (2-dimensional head direction) are recorded during the guidance for their later use for the off-line training of the CTRNN. The arm trajectories are guided for three types of the goal-directed behaviors with the three different object positions for each.

Therefore, there are 9 categories of the teaching sensory sequences where each category sequence is repeated 3 times. In total, 27 sensory sequences are sampled during the supervised teaching phase. The CTRNN is trained with these teaching sequences in a parallel manner for 50000 epochs.

3.3 Results of experiment-1

The repeated training examinations with randomly set different initial weights revealed that the mean square error (MSE: average square error per sensory unit per step over all teaching sequences) converges to around 0.0003. By taking one case of the trained weights with its MSE converging to 0.00028, the robot performance of generating the goal-directed behaviors was examined.

The robot behavior was generated by setting the initial context state with the value that had been obtained for each of the three goal-directed behaviors. The object was located in the center in the learning process. The experiment results showed that the robot achieved these three goal-directed behaviors successfully when the object was positioned in the center. Figure 4 shows the time-developments of the head direction representing the relative position of the object and the context state for each behavior. It is shown in the left-hand side of Figure 4 that the head direction starts to change largely after around 30 steps with developing a different profile depending on each goal. These trajectories are regenerated with mostly the same profiles with the ones in the training phase. The right-hand side in this figure shows that some dimensions of the context state develop differently for each goal. Especially, two bold lines represent the activation of the context units, of which the initial values are adapted. The units develop in different ways, starting from different initial values depending on the goals. It is noted that these two context units activation states develop differently even while the same movement pattern of approaching the object is generated. It is speculated that these states work as dynamic memories to initiate each different goal-directed movement that follows the initial same movement of approaching the object.

We further examined the generalization capability of the learned network. For this purpose, the behavior generation was tested by varying the object position from left to right within the same range used in the teaching phase but with setting the same initial context state values which had been used in the previous experiment for each goal. It was found that the same initial context state can generate the same goal-directed behavior successfully regardless of the object position.

We also examined how much this generalization capability of the robot can tolerate

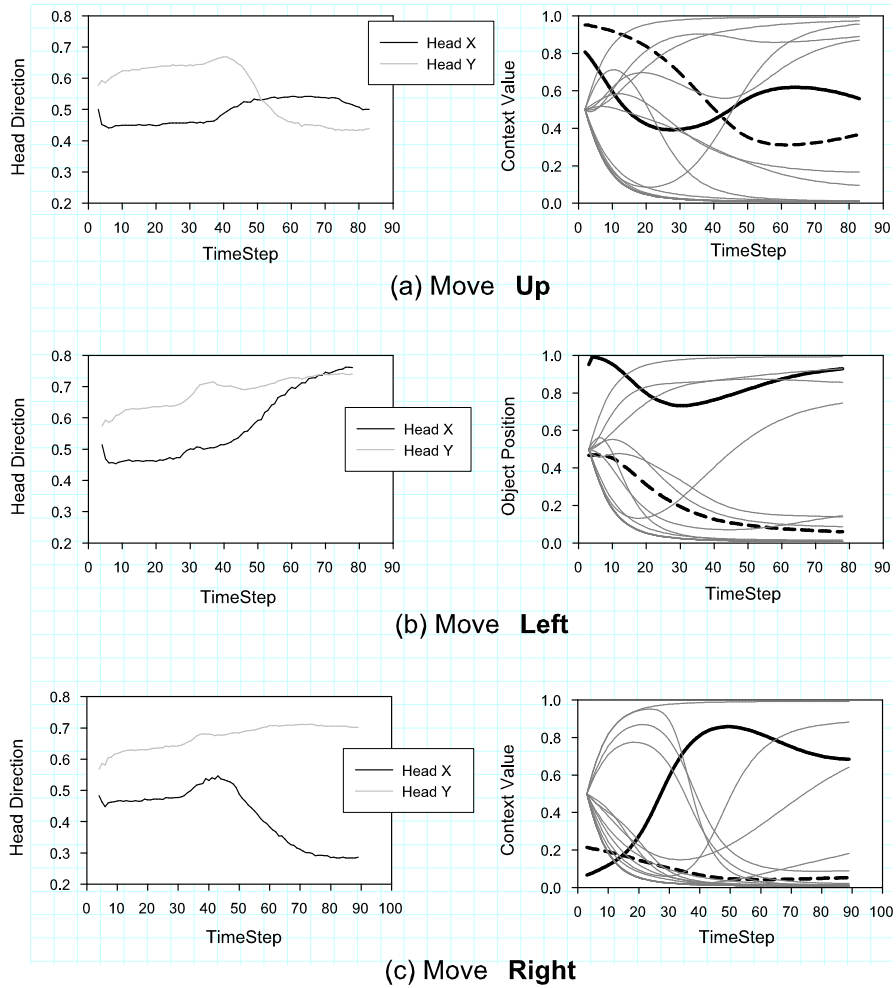


Figure 4: The generation of (a) “move up”, (b) “move left” and (c) “move right” behaviors with the object allocated in the center. On the left-hand side, head direction represents relative position of the object to be manipulated. On the right-hand side, a bold curve and a dotted one represent two context unit activation values of which the initial states are adapted.

variation of object position beyond the range trained. For this purpose, the “lift up” action was tested by varying the object position from the left position trained to 5cm further left by 1cm intervals. The initial context state was set with the representative value for the “lift up” action as it was done in the previous experiment. The success rate for each position with 5 trials is shown in Figure 5. It is shown that the trials are 100% the object is positioned within 2cm farther left from the trained left position. The robot starts to fail with the object positioned at 3cm farther left. With more than 4cm farther left the robot fails completely. In these failure cases the robot hands cannot approach the exact positions required to grasp the object. These results show some generalization capability for object position variance beyond the trained range.

We further examined how the multiple trained goals are internally represented by constructing the initial context state map. Figure 6 plots the 2-dimensional initial context state adapted for each training sequence. In Figure 6(a) the plots are made using category labels representing different goals. In Figure 6(b) different object positions sharing the same data aim to show the possible effects of position variance on the internal representation of the goals. From these two plots it is seen that the initial context state is categorized well for each goal but not for each position of the object. This result indicates that the initial context state map is self-organized to be sensitive to goals but not to positions of the object. This implies that although the motor behavior of approaching the object is adapted to the object position variance using the sensory information of the object position, the goal information is well preserved in the context state dynamics without being affected by the position variance.

The above experimental results show that the network has achieved both adaptability to environmental variances and robustness in achieving the goals. It is shown that the network can successfully acquire the skill of manipulating the object with its position variances by generalizing the interpolation as well as the extrapolation of a finite number of teaching data in learning. It seems that the robot becomes able to generate multiple goal-directed behaviors in a contextual manner, which could be expressed by Luria (1973)’s metaphor of “kinetic melody”.

However, it is noted that this successful condition holds good only at a certain range of the time constant parameter τ . Our further experiments revealed that if τ is reduced to less than half of the current value, the robot cannot achieve the goals properly. The goal information kept in the context units, dynamics is lost during the approach to the object because of its position variances. It was, however, found that the adaptability to the object position variances can be much improved in this case. On the other hand, if τ is increased to more than twice the normal value, although the

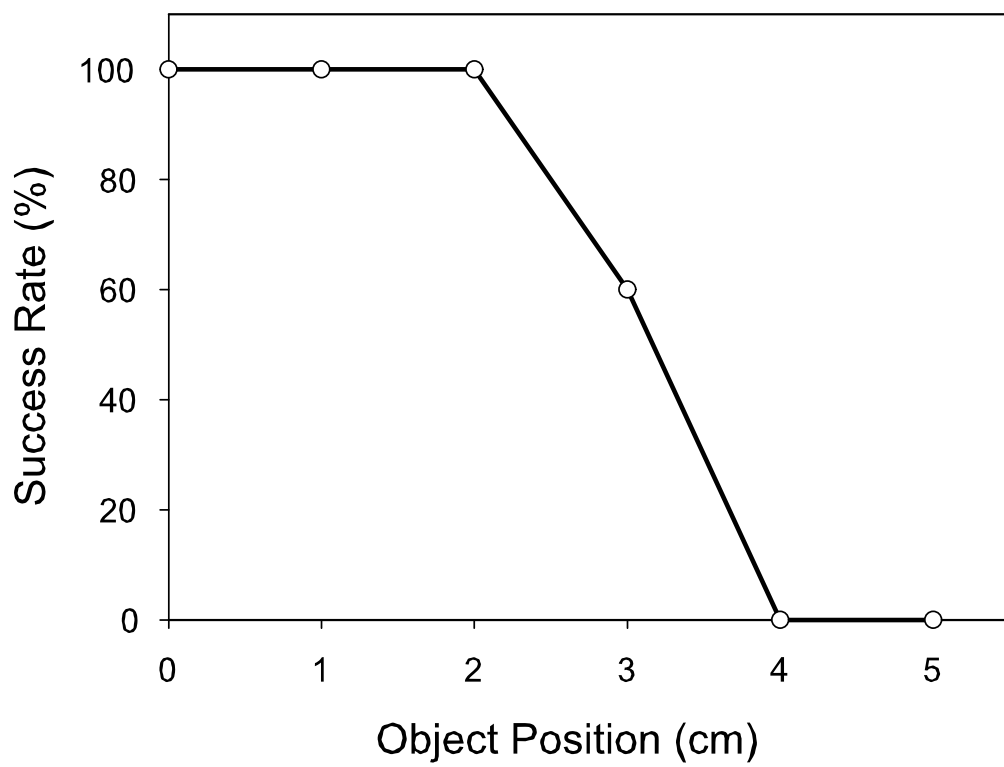


Figure 5: The success rate of the lift up action with varying the object position to further left of the trained range.

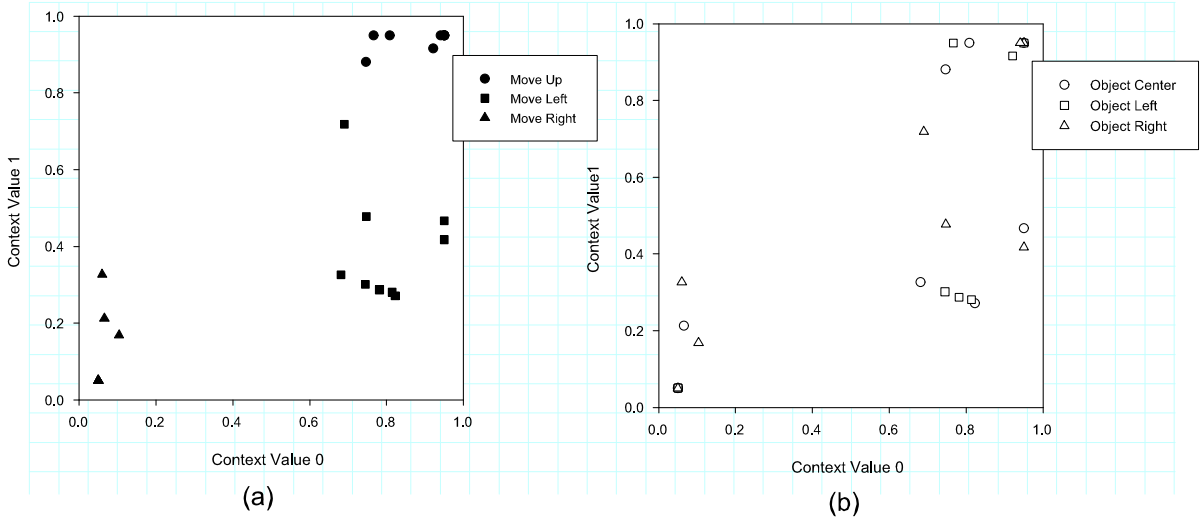


Figure 6: The initial context state map with category labels denoted for (a) different goals and (b) for different object positions.

goal information is preserved well during the movements, adaptability to the object position variances is lost completely. This implies that there is a tradeoff between the adaptability to the current sensation and the ability to achieve the specified goals. The parameter τ that represents the time constant of the memory dynamics can change the balance between the two.

In the end of the experiment-1, we examined the behavior recognition processes of the robot when the object is moved by the experimenter in the same way as the robot does. As it has been described in the previous section, recognition is a process of searching for an optimal 2-dimensional vector of the initial context state that best fits with the observed behavior. Figure 7 shows how the value of the initial context state is searched from its neutral value (0.5, 0.5) during the recognition process of the visual input sequence pattern s_t for each observed goal-directed behavior. It is seen that each value of the initial context state (denoted by a trajectory reaching to a black category label) converges to a neighbor of the appropriate goal cluster (represented by clusters of white category labels) which reappeared from Figure 6. It can be said that the network recognizes the observed goal-directed behaviors correctly using the proposed scheme.

Now, the results of experiment-1 are summarized. Our experiments showed that the robot can learn to generate multiple goal-directed behaviors using the initial sensitivity

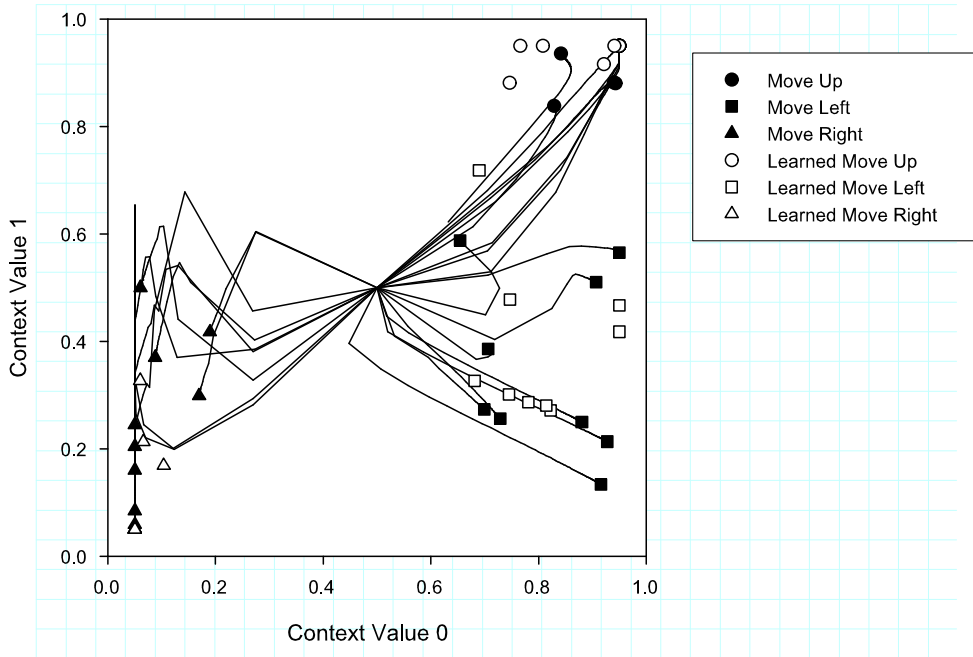


Figure 7: The trajectories of searching the optimal value of the initial context state in recognizing the observed different goal-directed behaviors. The black category labels denote the converged values and the white category labels denote the initial context values adapted in the learning.

characteristics of the CTRNN dynamics. Each goal-directed behavior can be learned with generalization for the object position variances if the time constant of the CTRNN dynamics is set appropriately. It was also shown that the robot can recognize the learned goal-directed behaviors by means of the inverse computation of the initial context state values.

4 Learning with modules and levels: experiment-2

The previous experiments showed that even a simple CTRNN can learn skills for multiple goal-directed behaviors whose representation seems to possess minimal compositionality. However, if a robot is to learn more diverse and complex behaviors, introducing level structures to its neuronal architectures seems inevitable. If we suppose a two-level structure, the lower level might organize a set of behavior primitives on

which the higher level might operate for the sequential compositions. Previous studies showed that behavior primitives can have either local (Wolpert & Kawato, 1998; Tani & Nolfi, 1998; Demiris & Hayes, 2002) or distributed representations (Tani, 2003) in neuro-dynamic systems. Unlike local representations, a distributed representation has globally shared structures that can be used to represent whole primitives 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 primitives because catastrophic memory interferences among different primitives would occur. This problem actually happened when we attempted to scale the single CTRNN scheme shown in experiment-1 with a larger number of behavior patterns. Therefore, the current study employs a level-structured architecture with an enhanced version of our local representation scheme (Tani & Nolfi, 1998) which showed originally how the sensory-motor flow can be hierarchically segmented.

This newly proposed architecture is examined with a developmental tutoring scheme in order to achieve more complex behavior skill acquisitions. The developmental tutoring scheme means that the robot skill is developed incrementally through dense interactions with tutors. In this scheme, the task learning is facilitated through the robot’s self-trials, with human guidance, that are repeated in cycles until required tasks are mastered thereby enabling consolidation learning of the neural networks. The actual interaction between the robot and the tutor takes place with force when the tutor guides the robot movements by grasping its arms while the robot attempts to move as driven by the CTRNNs trained in the last teaching session. Through this type of interactions, the trajectories by the tutor guidance can be also modulated along with the ones by the robot and thereby they could co-develop to satisfy the task goals.

4.1 The CTRNNS with levels and modularity

Before describing the details of the architecture, an intuitive account is given. Figure 8 shows the two-level neural network architecture used. The architecture is divided into the lower level consisting of multiple CTRNN modules and a single CTRNN in the higher level. When a set of teaching sensory sequences is given to the system, the lower level CTRNN modules compete to be an expert of predicting the sensory sequence. It is important to understand that a single CTRNN does not win to become an expert for the whole sequence but it does for parts of the sequence. Other CTRNNs become experts for other parts of the sequence. After the training of the networks is iterated for a set of teaching sequences, the sequences become segmented by means of

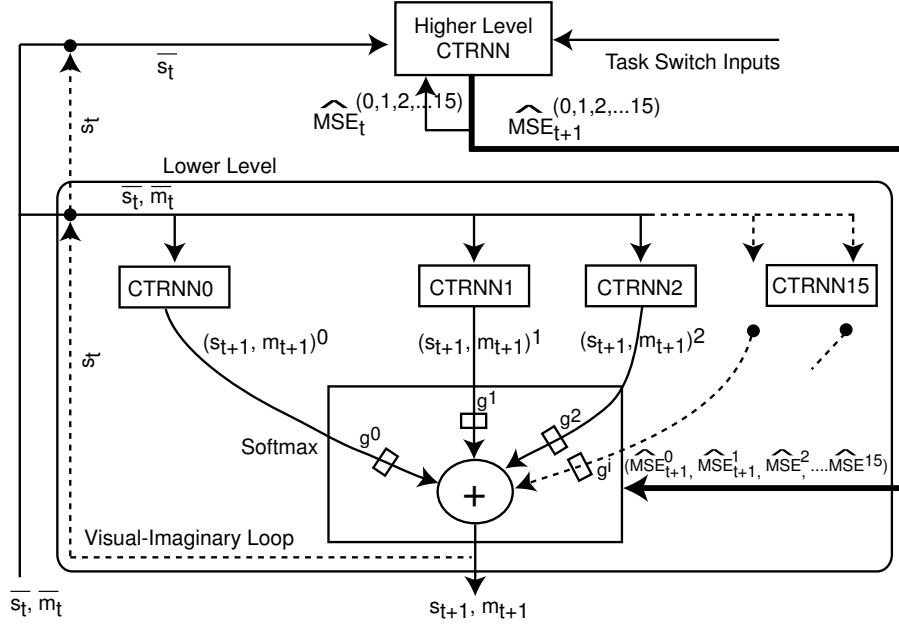


Figure 8: Two-level structured CTRNNs.

autonomous switching of the winner CTRNN associated with its gate opening. On the one hand, the higher level CTRNN learns to predict which gate opens in the sequences. As the result, when the learning of the networks converges, the sensory sequences can be regenerated in a way such that the higher level CTRNN selects the next opening of the gate and the corresponding lower level CTRNN generates the sensory prediction pattern.

Now, the details are explained. The lower level consists of ($N = 16$) CTRNNs each of which has a gate and y_t of the overall outputs of this level is computed as the gate weighted summation of each network output y_t^i :

$$y_t = \sum_{i=1}^n g_t^i \cdot y_t^i \quad (6)$$

where g_t^i is the gate opening of the i th network. The output y_t is the prediction of the sensory inputs that contain 2 dimensional vision-related information (the head direction) s_t and 8 dimensional proprioceptive information m_t whose setting is the same as in experiment-1. The temporal sequence of g_t^i is determined by the learning process of the lower level networks. Basically the gate opens more for the network with less prediction error in the current step. As the soft max function is applied, the gates

tend to open in a winner-take-all (WTA) manner. The gate opening is computed as:

$$g_t^i = \frac{e^{s_t^i}}{\sum_{j=1}^n e^{s_t^j}} \quad (7)$$

$$s_t^i = \frac{-er_t^{i2}}{(2\sigma^2)} \quad (8)$$

where er_t^{i2} denotes the time average of the mean square error (MSE) of the i th network at time step t . The time average is obtained by using a fixed time step window. σ^2 denotes the standard deviation of the errors of all the networks. Upon computing the gate opening for each network, the error of each network used for the BPTT computation is weighted by its gate opening. This means that a network with less error for certain segments of sensory sequences will learn the sequences better since its gate has a larger opening. In this way, local “expert” networks develop for segments that repeatedly appear and sensory sequence patterns develop. These segments in the sensory sequences are considered as behavior primitives. The sensory sequence patterns experienced in the lower level are segmented into a sequence of the behavior primitives that are self-organized and their sequences are represented as gate switching sequences in the higher level (Tani & Nolfi, 1998).

Meanwhile the higher level CTRNN learns to predict the gating sequences in the lower level CTRNN modules. 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 module instead of its gate opening because temporal patterns of gate opening vector frequently change much more discontinuously than temporal patterns of the prediction error due to the WTA characteristics of the softmax function.

The CTRNN generates the next prediction of the MSE for 16 lower networks denoted as $M\hat{S}E_{t+1}$ by receiving the self-feedback of its own prediction at one-step before $M\hat{S}E_t$, 2 dimensions of the visual input at the current step that are the same as those received at the lower level, and 3 dimensions of vector inputs that specify the task goal currently engaged. The visual sensation as well as the task switcher vector are regarded just as inputs and they are not for prediction.

For behavior generation, the predicted error vector is sent to the lower level which is converted to the gate openings with the softmax function in Eq. 7. This pathway serves as a top-down control by the higher executive level onto the lower sensory prediction level. The prediction of the proprioceptive state provides the target encoder value at each joint at the next time step in the same way as in experiment-1. The time constant

τ of the higher level is set four times larger than the one in the lower level. This time constant difference enables the abstraction of information flow from the lower level to the higher one (Paine & Tani, 2005). The exact value of τ is 5.0 for the lower level CTRNN and 20.0 for the higher one, correspondingly.

It is noted that the scheme of the initial context state adaptation shown in Eq. 5 is not utilized in both the lower and the higher level CTRNNs. Instead, the higher level CTRNN is given with specific constant inputs for each task goal in both learning and generation phases. All the initial context states are set with a constant neutral value.

4.2 Setup in experiment-2

The same robot, as shown in experiment-1, was utilized again. The robot with this two-level structured CTRNN receives tutored training for a set of goal-directed behavior tasks. In task-1 the robot must approach the object with both hands coming down from the home position, grab the object, let it go, and return to the home position. In task-2, the robot must approach the object, grab it, bring the object up and down three times, release the object, and return to home position. And in task-3 the robot must alternately touch the object with its right and left hands eight times before returning to the home position.

A tutor taught these three task behaviors to the robot in 3 tutoring sessions. (Note that all three different tasks were taught to the robot at each training session.) The tutor was given instructions for each task. For task-1, he was asked to ensure that both hands hit precisely the right position for object touching. In task-2, he needed to make sure the object was held and brought up and down three times without dropping it and that the robot touch the object with each hand as they alternate for the correct number of sets in task-3. In each tutoring session, the task behaviors are guided and the initial position of the object changes 6 times for each task. In Figure 9 six object positions from 0 to 5 are for the training and 6 and 7 are for the test of position generalization.

In the first session, the robot guidance is conducted by disabling active movements of the robot by setting the motor control gain to zero because the networks are not yet effective with the randomly set initial synaptic weight values. In this session, the network training was iterated for 50000 epochs for the lower level and 5000 epochs for the higher level networks, respectively. In the second and third sessions, the tutoring is conducted interactively by enabling active movements of the robot with the control gain set to 50 percent of its normal operation value. The network was trained off-line by using all tutoring sequence data obtained at each session. The network training was

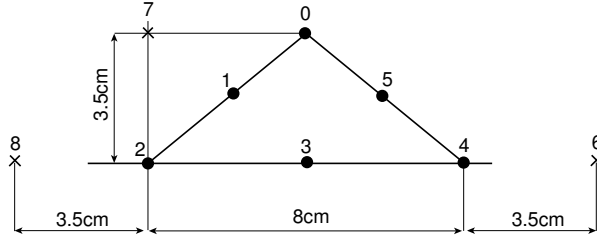


Figure 9: The initial object positions. 0 to 5 for training and 6, 7 and 8 for generalization tests.

iterated for 10000 epochs and 5000 epochs for the lower and the higher level networks, starting with the synaptic weights obtained from the previous session.

4.3 Results of experiment-2

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 of the object. We briefly describe how the task performance developed as follows. For more detailed descriptions of the observed development processes, the reader should refer to (Tani et al., 2008).

Figure 10 shows how the robot self-generated the described three task behaviors in the 3rd session. For each figure the profiles of task-1 (object position 0), task-2 (object position 2) and task-3 (object position 2) are shown in (a), (b) and (c), respectively. The time developments for the openings of 16 gates, m_t 4 left arm encoder values and s_t 2 dimensional head direction inputs for vision information are shown in the upper, the middle and the bottom levels. In this figure each gate opening is represented by the grey level color of a horizontal bar from white to black, corresponding to its value from 0.0 to 1.0. Those bars are aligned from the top to the bottom levels as corresponding to the 0th to the 15th gate openings.

Figure 11 shows the developmental process of task-2 over three sessions (The plot is shown only for task-2 due to space limitations.) The task-1 behaviors were generated perfectly, even with the untrained object positions (6 and 7 in Figure 9) from the 1st session. This might be because the task behavior in task-1 is simpler than those in the others. We also observed that s_t and m_t profiles and gate openings for task-1 do not change drastically across three sessions. The approach to an object is encoded by the 15th gate opening. Return arm movements are encoded by the joint opening of the

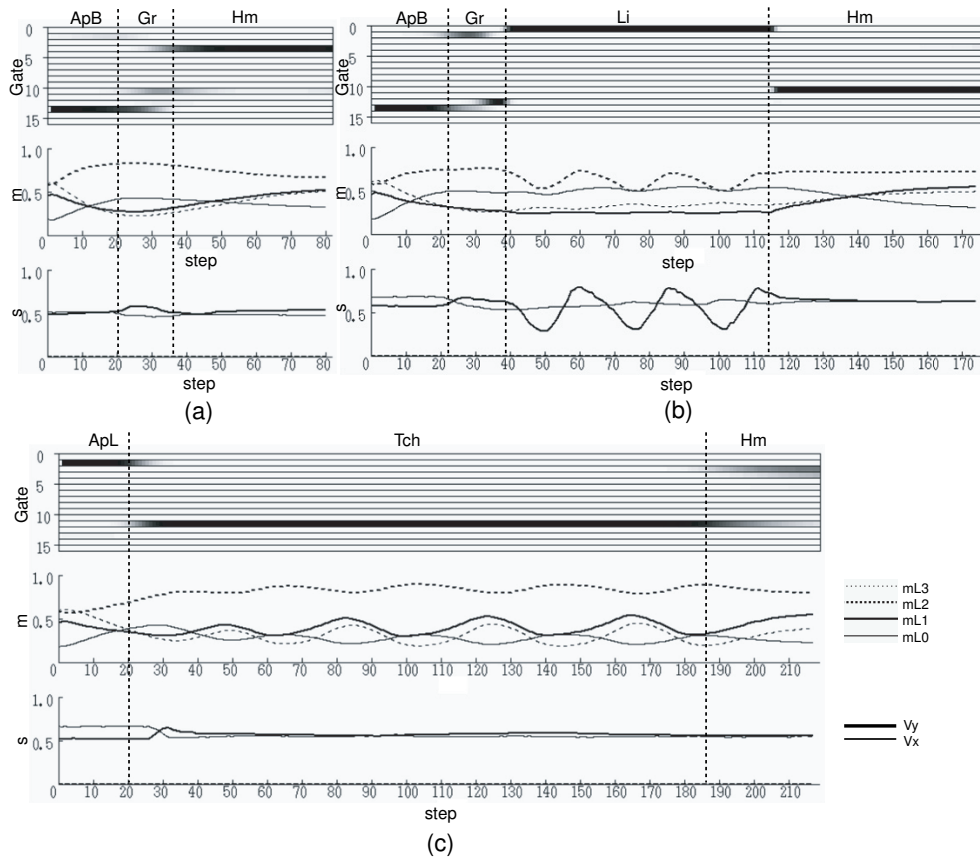


Figure 10: Gate opening, m_t prediction and s_t inputs sequences for (a)Task-1, (b)Task-2 and (c)Task-3 in the 3rd self-generation session. See the text for the bold labels that denote type of behavior primitives.

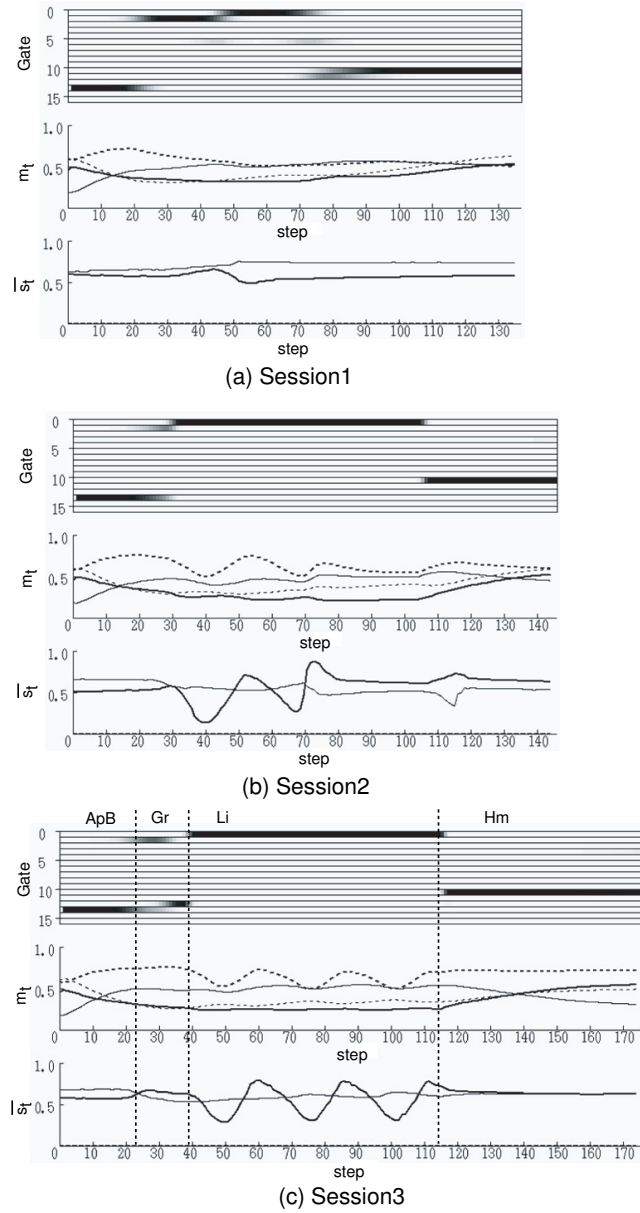


Figure 11: Gate opening, visual inputs and joint positions profiles in (a) the 1st session, (b) the 2nd session and (c) the 3rd session for Task-2.

4th and 10th gates, as shown in Figure 10(a).

In task-2, the object could not be lifted in three of the five attempts in the first session. While the approach trajectories for both hands were almost perfect, the robot could not squeeze the object suitably with both its hands in every position. However, in the following teaching session, the tutor felt that the robot could self-generate most of the desired movements 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. However by the third session the robot executes the task nearly perfectly (see Figure 11 for this developmental process).

In the first session of task-3, the robot could not touch the object most of the time. In fact, oscillatory motor patterns for repeated touching crashed over time. In the second session, the robot started to touch the object at least once and initiated oscillations in motor patterns. By the third session the robot became able to touch the object repeatedly in a stable manner as it generated more explicit cyclic patterns in the motor outputs (see Figure 10(c).)

Now we examine what sorts of behavior primitives are acquired and how they are combined to generate the desired task behaviors. By looking at the gating sequences generated in the third session, we can see how continuous behavior flow is segmented into a self-organized set of behavior primitives that emerges in our proposed neural network architecture. For example, in Figure 10, we see that the approach to the object by both hands (**ApB**) is encoded by the 13th gate opening for tasks 1 and 2. The cyclic movement of lifting the object up and down (**Li**) occurs by the 0th gate opening. The return home (**Hm**) occurs by the 3rd gate opening, with slight opening of the 10th one for task-1 and the same for task-2 does by the 10th one only. Also, the approach to the object by the left hand from the home position (**ApL**) in Task-3 is encoded by the 1st gate opening. The cyclic touching of the object by alternating left and right hands (**Tch**) is encoded by the 11th gate opening and the return home is encoded by the 3rd gate opening. The gating for grasping the object by both hands (**Gr**) seems different between Task-1 and Task-2.

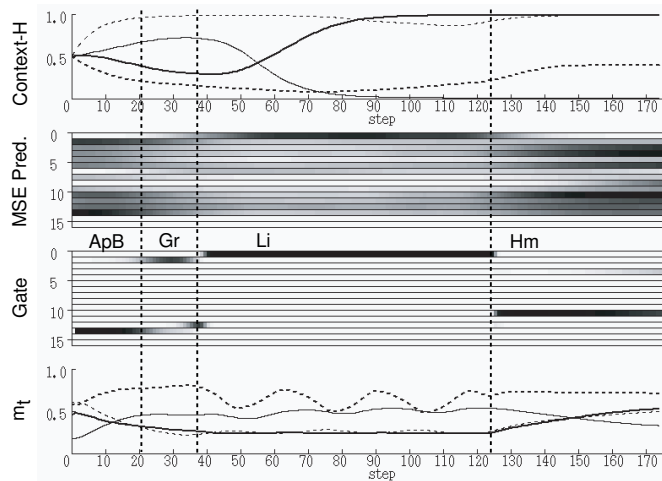
Here, two questions might come to mind. (1) Why are similar behavior patterns sometimes encoded by different gate openings? (2) Are the encodings of these behavior primitives sensitive to the object’s position or are they sufficiently generalized to be insensitive to the object’s position? To examine these questions the activation profiles in the higher level CTRNN are compared for task-2 with different initial object position cases, namely positions 7, 8 and 6 shown in Figure 12(a), (b) and (c), respectively.

(Position 8 is 3cm 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 the grey level color of each horizontal bar where the darker color denotes the smaller MSE prediction.

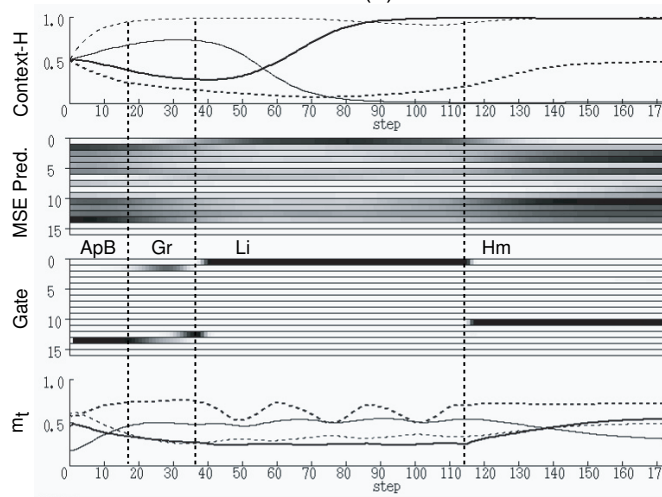
First, we see that gate opening profiles and context activation profiles are mostly the same overall for all three cases regardless of the object position difference. However, there are certain differences in the gate opening immediately before the 0th gate opens between the position 8 and 6 cases. The 1st gate opens in the position 8 case in Figure 12(b) and the 12th gate does in the position 6 case Figure 12(c). It is, however, important 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 system because both local nets are good at generating motor sequences with small learning errors. Here, most of the local CTRNNs actually participate in competition for the gate opening at subthreshold levels although only some of them are observed as explicit winners. This explains how redundant encoding of behavior primitives is self-organized and why behaviors can be generated stably regardless of such redundancy. In our preliminary experiments, it was frequently observed that reduction of number of local nets introduce instability in learning because allocation of a set of behavior primitives into limited number of local modules becomes tighter.

Furthermore it can be said that the gate openings are performed in a context-dependent way such that each behavior primitive can be smoothly continued from its predecessor. Thus, the gate opening for “return home” was different for each goal task, as was shown in Figure 10, because the previous primitives were different.

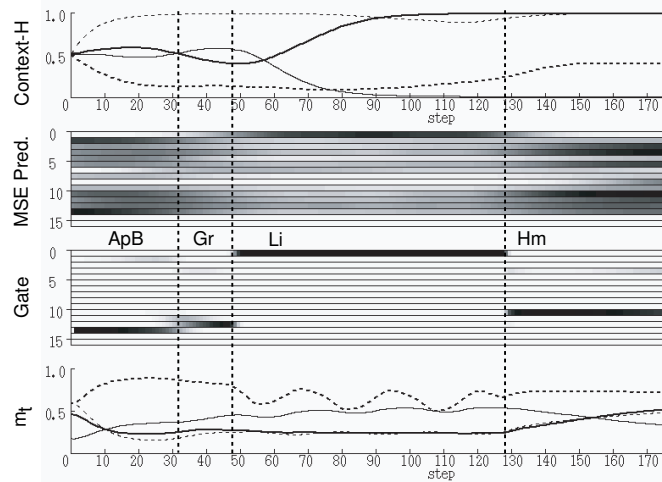
Another interesting observation in the current developmental tutoring scheme is that the human tutors as well as the robots codevelop as they interact to achieve a specific task goal. More specifically, when robot learning is immature, human tutors physically interact with the robot in order to guide the robot to perform the task better. Through the physical interactions, tutors can easily learn which ways the robot tends



(a)



(b)



(c)

..... mL3
 mL2
 mL1
 mL0

Figure 12: The context units activations and the mean square error prediction vector of 16 dimensions 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 move and which parts of the movements should be modified. The tutor simply exerts intentional forces on the robots only at some parts of the trajectories to be modified and lets other parts go rather freely. 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 arbitrarily determined teaching trajectories. By this means, the robot tutoring becomes a process of jointly exploring better training paths which can influence the network’s internal structure. It is gradually organized to become an adequate one for achieving the desired task goal.

Figure 13(a), (b) and (c) shows an observation related to the codevelopment. Each graph in this figure shows velocity profiles for the development of teaching trajectories on the left hand side and those generated by the robot on the right hand side for task-1 after each session’s learning. It is observed that some parts of the teaching velocity profile in the first session are relatively steep, especially when both arms approach the object (from 5 to 20 steps) and when they leave the object (from 35 to 45 steps). However, in the successive robot generation phase in the first session it is seen that the steepness in the velocity profile becomes much milder. Then in the second teaching session the teaching velocity profile becomes mild and is also affected by the robot’s own movement. It seems that the codevelopment of the velocity profile mostly converges since the profiles for teaching and generation after this session does not change much.

The result of this development tutoring scheme contrasts significantly with our preliminary experimental results with one-time batch training without the codevelopment scheme. In one-time batch training, task-2 and task-3 were not accomplished even with more learning iteration steps than those in the developmental learning scheme. This comparative result implies the effectiveness of the codevelopmental learning scheme.

The readers may ask why the trained CTRNN can perform well even outside of the training region for positions 6 and 7. We consider that the required movement for position 6 can be determined simply by extrapolating from the movements for positions 2 and 4. Similarly, the movement for position 7 can be determined by extrapolating from the movements for positions 0, 2, and 3. This type of generalization capability of the CTRNN has been demonstrated also in experiment-1. It was shown that the generalization can cover not only for interpolation but also for extrapolation of trained data sets.

Additionally, we examined if the robot can pantomime object manipulation behaviors without using an actual object. In this experiment, the lower level CTRNN operates in the closed-loop imaginary mode with the actual vision information inputs s_t shut off after the initial step. (The vision information inputs at the initial step

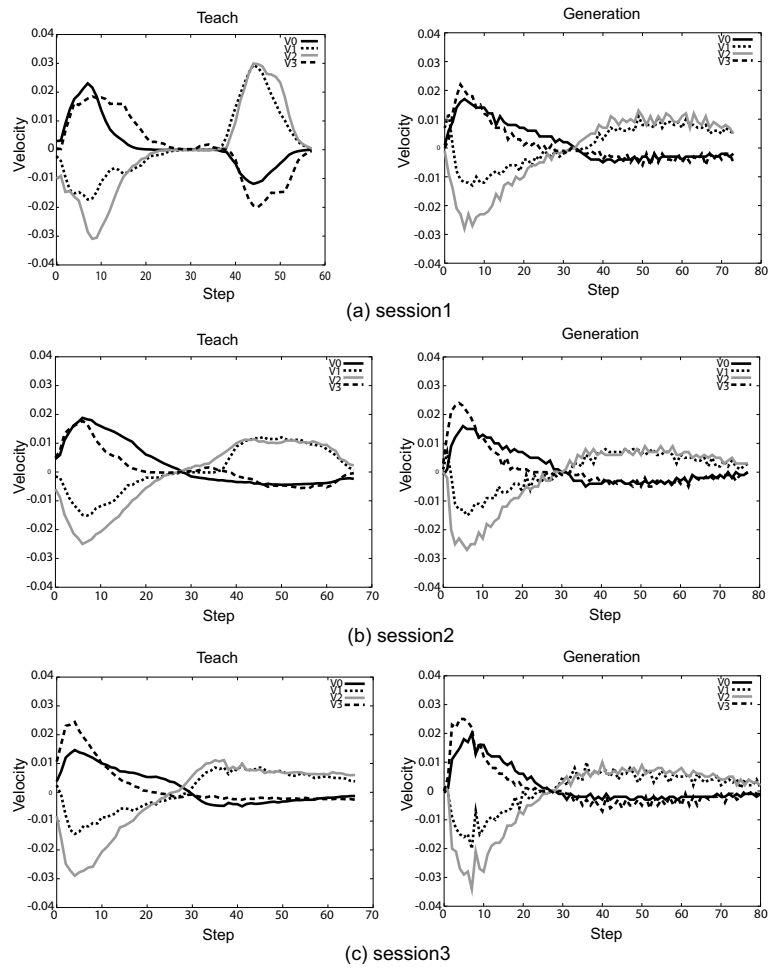


Figure 13: Development of velocity profiles guided by tutors shown on the left hand side are contrasted with the ones generated by robot shown on the right hand side for (a) session 1, (b) session 1 and (c) session 3 for task-1.

are necessary in order to integrate them through future time steps.) The higher level CTRNN also receives the imaginary s_t generated in the closed-loop operation in the lower level CTRNN rather than the actual one. We ran this experiment for task-2 because the visual inputs change dynamically in this task only when the robot brings up and down the object. The results show that the sequential behavior of approaching the object, the cyclic movement of bringing up and down the object, and homing are well imitated. It was also seen that the imaginary sensations are reconstructed well (see more details in (Tani et al., 2008).) A further interesting observation was that the same behavior could not be imitated at all when the imaginary sensor loop was cut off and the inputs were fixed as if the object were constantly positioned in the center. In this case, it was observed that the motor movement was frozen after the object was grasped even though the gating signal from the higher level was completely the same as the one in the case with the imaginary sensor loop. It is understood that the inadequately fixed sensory inputs inhibit the CTRNN dynamics of the periodic movement. This explains why the vision-related and the proprioceptive signals are treated in an inseparable manner in the lower level CTRNN. This finding is analogous to the neuroscience observation that vision and proprioception are integrated well in parietal cortex (Iacoboni, 2006).

5 Emergence of level-structured functions: experiment-3

The experiments in the previous section showed that certain complex behaviors can be learned if level-structured neuronal functions are given explicitly to the systems. The interactions between the levels are achieved by introducing the gating mechanism associated with local modular representations in experiment-2. Our previous study (Tani, 2003) showed that level interactions can be achieved also by means of a distributed representation scheme, the so-called RNN with parametric biases (RNNPB).

This section reviews a novel trial by Paine and Tani (2005) to investigate whether an appropriate mechanism of level interactions can be self-organized without explicitly introducing apriori schemes. More specifically, our new experiments investigate how the level-structured functions can emerge by means of evolutionary computation if level decomposition tasks are imposed on the system.

5.1 Set up of experiment-3

The following exploratory navigation task is considered as an experimental platform for the problem. A simulated mobile robot equipped with 8 proximity sensors and 2 motor-driven wheels explores a maze environment shown in Figure 14. The task of the robot is to find navigation paths reaching to as many different goals as possible from a start position. This navigation task can be decomposed into two levels of system functions. The first level should deal with collision-free maneuvering, going straight along a corridor and turning left or right at corners. The second level should deal with sequencing the turning at corners in order to reach the goals. The goal of the study is to understand how such two levels of functions can be self-organized in neural networks from scratch without showing explicit cues.

The robot is implemented with a fully connected CTRNN which is evolved by a genetic algorithm (GA). The activation dynamics of each neuronal unit is given by:

$$\tau \dot{u}_i = -u_i + \sum w_{ij} a_j \quad (9)$$

$$a_i = \textit{sigmoid}(u_i + b_i) \quad (10)$$

We tested two types of CTRNNs as shown in Figure 15. Figure 15 (a) is called a “bottle-neck” network since the information flow between the top and bottom levels is narrowed in its bottle-neck. (The neural activations can propagate to the other level only through the bottle-neck neurons (BN).) The bottleneck CTRNN has 5 neurons in the lower part, 2 BNs, and 4 neurons in the upper part. There are 2 so-called task neurons (TN) in the upper part whose functions will be explained later. All neurons in the lower part receive 8 proximity sensor inputs and output to two motor neurons, driving left and right wheels, through synaptic connections. Figure 15 (b) is a standard CTRNN consisting of 9 neurons including two TNs. All neurons receive 8 sensory inputs and output to 2 motor neurons.

We employed the ideas of initial sensitivity, described in the previous section, to generate combinatorial action sequences in the current task. The idea in the current setting is that the robot can reach different goals depending on the initial state values set in the TNs shown in Figure 15. In the evolutionary process, a set of the initial state values in the task goal neurons evolves, along with the synaptic weights and the biases. The time constant τ for each neuronal unit is also evolved. The fitness function is designed to increase the number of different goals reached with the set of evolved initial state values. We repeated the evolution runs 20 times for both types of networks for statistical comparisons of their performances. Each evolution run is conducted for 200 generations with an 80 robot population per generation.

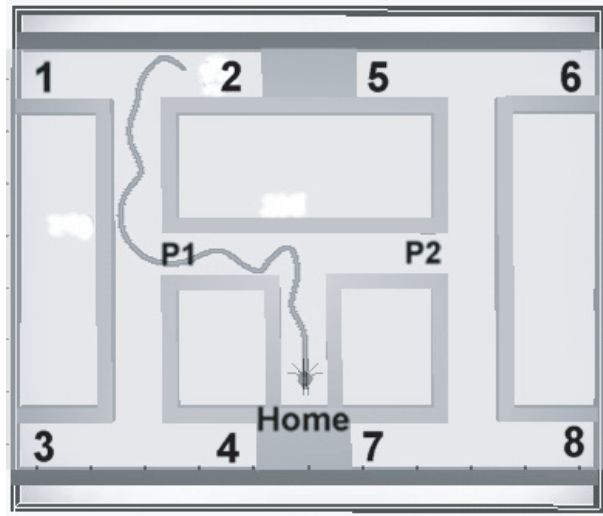


Figure 14: A simulated mobile robot learns ways to reach 8 different goals starting from the home position.

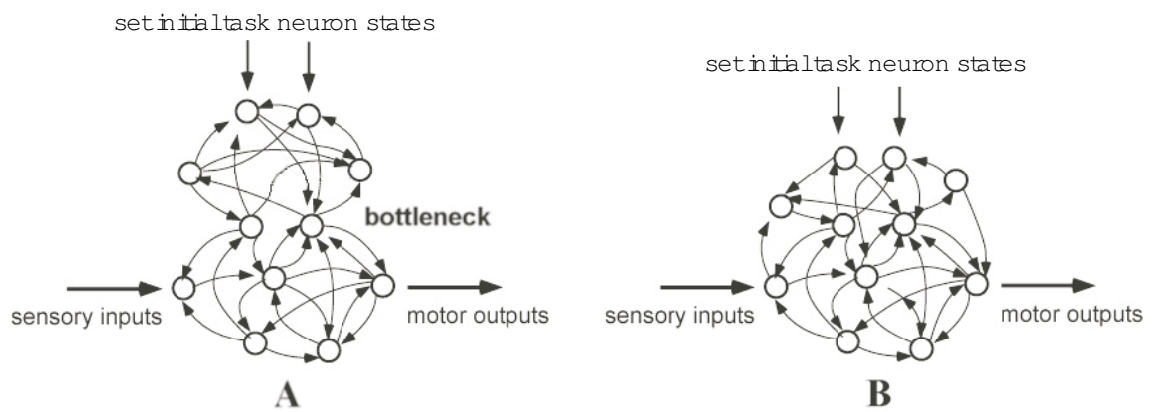


Figure 15: CTRNN with a bottle-neck in (a) and standard CTRNN in (b)

5.2 Results of experiment-3

Our experiment results showed that the best performance is obtained in the bottleneck network. In 20 evolutionary runs, the average number of different goals reached was 5.1 for the bottleneck CTRNN, and 2.3 for the standard CTRNN. The bottleneck CTRNN found five or more goals on 14 of 20 runs. The standard CTRNN found them on only six runs.

The temporal neuronal activation profiles for an evolved bottleneck network, which found 6 different goals, are shown in Figure 16. The profiles correspond to a Right-Left-Right turn sequence, starting from the home position, that reaches goal 6 of Figure 14.

The top row shows the activation profiles of two TNs and two bottleneck neurons (see Figure 15(a)). The bottom row shows the profiles of the two motor output neurons in the lower part of the network. Observe that the motor outputs show much faster dynamics than those of the TNs and BNs. Actually, we found that the time constants τ for the motor neurons evolve to be much faster than those of the TNs and BNs in all successful evolution runs. The activation profiles of the BNs correlate with right and left turns, denoted by labels in the top figure. For the right turn, both BNs have high activation values, while BN-2 takes a low value and BN-1 slightly decreases for the left turn. TN-2 shows a similar type of encoding as the BNs, while the dynamic profile of TN-1 seems uncorrelated with the turn sequence. These profiles suggest that certain level structures are self-organized in the bottleneck network. The following analysis examines such structures.

First, functions of the BNs were examined. We constructed a phase space analysis for the BNs, focusing on the cornering behavior at the T branch. Figure 17 indicates how the cornering behavior varies when the activations of two BNs are clamped externally to various values. It is observed that the BNs activation space is divided into three regions, grey, white and black, which correspond to left turns, right turns and collisions with the walls, respectively. It is considered that the BNs activation states encode the behavior primitives of turning left or right in branches.

Next, we constructed a phase space analysis for the task neurons, initial states, focusing on their possible encoding for the turning sequences. The results can be seen in Figure 18, where the regions in the initial state space that reach different goals are labeled by the corresponding turn sequence, e.g., LRR for a Left-Right-Right turn sequence. The turn sequence is denoted by number in the plot (see the legend on the right).

It is observed that the sequence patterns are arranged in clusters in the TN initial

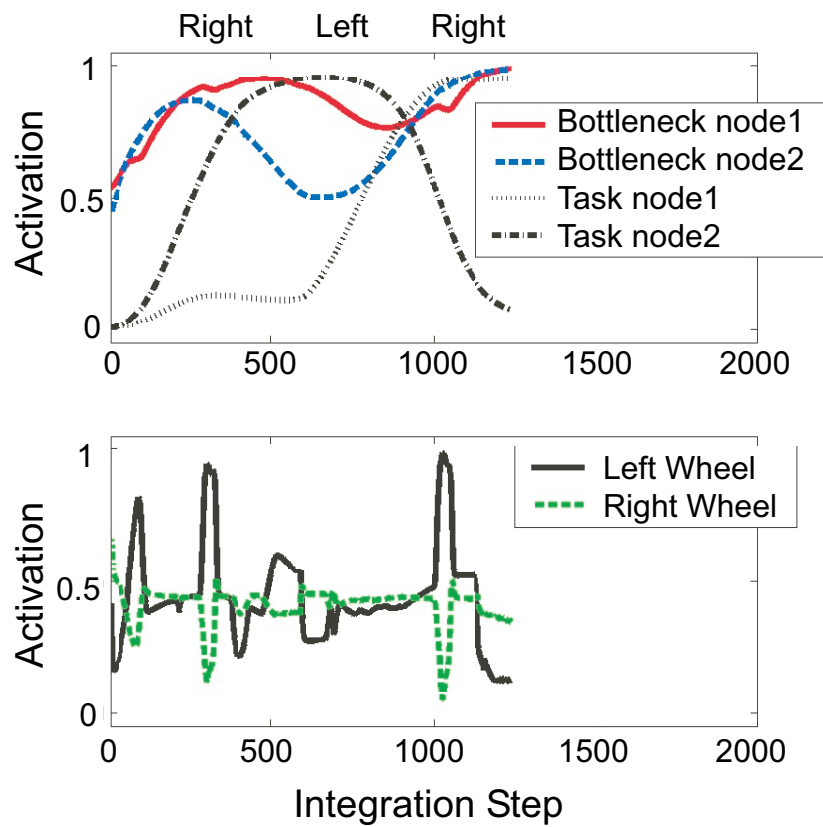


Figure 16: Neuronal activity for a Right-Left-Right turn sequence in the bottleneck network. Top: Neuronal activity of bottleneck and task neurons, respectively; Bottom: Activities of motor output nodes.

state space. First, the space is grossly clustered based on the first turn direction, left or right, of the movement sequence, as shown by a thick solid line in Figure 18. Each of these two clusters is then further divided into topologically ordered sub-clusters, depending on the second turn direction of the movement sequence, as shown by a solid line. These sub-clusters are still further divided into smaller clusters, depending on the third turn as shown by the dashed lines. These smallest clusters neighbor each other and share the first two turns of their sequences in common. In other words, the turn sequences are hierarchically ordered into progressively smaller regions of the initial TN activity space as additional turns are added. As the complexity of the movement sequence increases, so too does the initial sensitivity to the TN activities.

In order to clarify the functional roles of the upper level of the bottleneck network, we observed the activities of the upper level neurons while they were decoupled from the lower level ones i.e., disconnecting all the synaptic connections from the lower level neurons to the BNs. It turned out that the activities over time of the TNs and BNs are mostly the same as the original ones provided that the same initial states are set in TNs. Compare the disconnected case shown in Figure 19 with the original one shown in Figure 16 for reaching goal 6.

The results imply that the whole network was evolved such that the upper level generates top-down internal images or plans for achieving the goals without accessing the sensory inputs, and that the lower level deals with actual maneuvering control of the robot based on the plans. More specifically, the upper level generates the top-down anticipation of how the BNs states should develop based on the goal information encoded in the initial states of the TNs while the BNs, states activate the behavior primitives of turning left or right in sequences in the lower level.

Finally, the readers might ask why the standard CTRNN case cannot evolve successfully as compared to the bottleneck case. It is assumed that evolving different dynamic functions with different time constants is difficult within a single fully-connected network because it would cause too much interference among them. In the bottleneck case, fast and slow dynamics can be evolved more easily by having less interference with each other since they are segregated by the bottleneck of the network. Our experiments showed that a class of level-structured functions can be evolved provided that adequate topological constraints such as bottlenecks or hub-like connectivities, are given in the network.

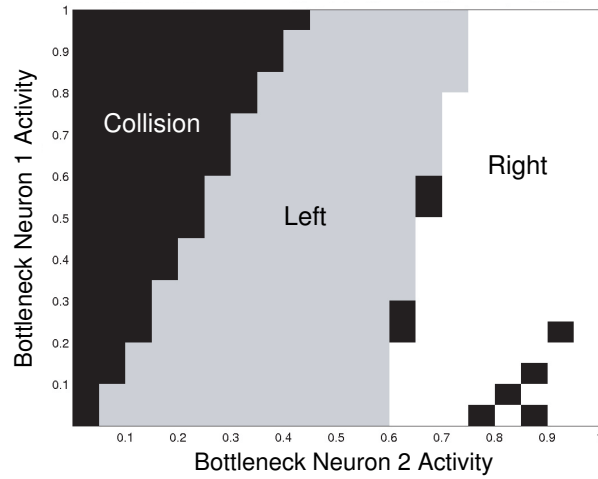


Figure 17: Phase space analysis for two bottleneck neurons.

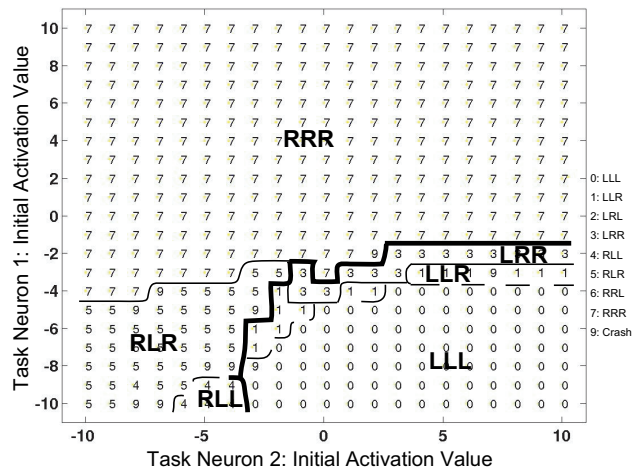


Figure 18: Phase space analysis for task neurons initial states. Plotted numbers correspond to turn sequences as in the legend on the right.

6 Discussion

6.1 "Organic" compositionality

The current paper has reviewed different schemes to represent behavior skills of robots using dynamic neural networks. However, all approaches share the same research motivation to see what sorts of "organic" compositionality can be self-organized in given neuro-dynamical systems when the robots are required to learn multiple goal-directed behaviors through diverse sensory-motor experiences.

In the beginning of the current paper, we discussed that acquisition of reusable behavior primitives is indispensable in achieving the compositionality in action representation. In our studies, behavior primitives are represented implicitly or explicitly depending on each of the proposed CTRNN architectures. Let's review them again briefly.

In experiment-1 multiple behavior primitives are embedded in the dynamics of a single CTRNN. The behavior starts from "approach to object" and then it branches to either of "hold up", "move to left" or "move to right" based on the initial state of the context units. This mechanism of branching can be explained in terms of a pitchfork bifurcation (Wiggins, 1990) as schematically shown in Figure 20.

The intuition is that there is only one attractor (a stable fixed point in the direction of x) encoding behavior-A before the branching point and it bifurcates to two attractors of behavior-B and behavior-C after the branching. It is essential that each behavior primitive is encoded by attractor dynamics because its convergent vector flow provides inherent robustness in generating behaviors. On the other hand, pitchfork bifurcation provides the composition mechanism in terms of dynamical switching from one behavior primitive to another. The way of the switching depends on the goal information embedded in the initial state of the context units and the dynamic memory organized in these units generates motor-act sequences fluently and contextually toward the goal.

Furthermore, our prior study with simple simulations (Nishimoto & Tani, 2004) showed that a conventional RNN of Jordan-type (Jordan, 1986) can also learn recursively branching structures (tree-like graph structures) but with limited depth using the same initial context state adaptation scheme. In this case, fractal structures are self-organized in the mapping between the initial state and the category of infinite number of branching sequences. It can be said that the combination of attractor dynamics and bifurcation mechanisms self-organized in nonlinear adaptive dynamical systems can provide both robust and compositional characteristics to the systems.

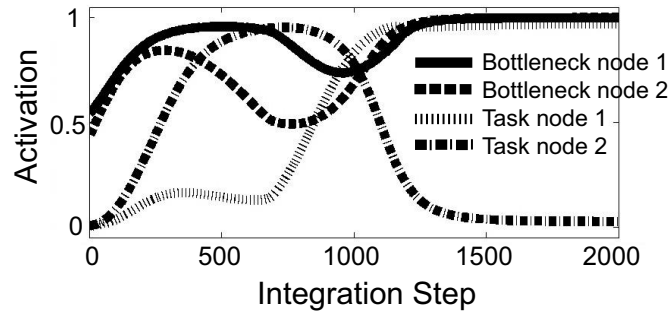


Figure 19: The profiles of BNs and TNs activities in upper level disconnected case (reaching to goal 6).

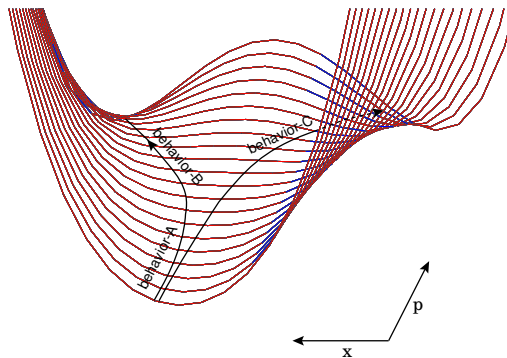


Figure 20: Behavior-A is bifurcated into behavior-B and behavior-C by means of Pitchfork bifurcation. Initially there is only one stable fixed point in the direction of x . As P develops in time, the fixed point bifurcates into two stable fixed points.

The idea of switching behavior by pitchfork bifurcation is analogous to the observations of human rising motions by Yamamoto and Kuniyoshi (2002). This study showed that diverse trajectories of human rising motions can be qualitatively represented by topological graphs in which each arc denotes a transitional motor act and each node denotes an unstable branching point among possible behaviors. In their studies, the branching structures are mostly generated based on the physical body kinematics. A similar idea had been seen in a robot navigation scheme (Tani & Fukumura, 1994) in which the branching, in mobile robots maneuvering in obstacle environments, is made by means of a pitchfork bifurcation in virtually generated potential fields. On the other hand, in the current case, the branching structures are generated based on the neuro-dynamic structure self-organized through learning. It is highly likely that both effects of adaptive neuro-dynamics and fixed body kinematics contribute to generations of the branching structures in skilled human behaviors.

The second experiment showed a trial to scale the CTRNN scheme. Our preliminary studies showed that a single CTRNN cannot store more than 3 or 4 behavior primitives in a stable manner. Surprisingly, it was true that increasing the number of neurons in the network does not help this situation progressively. The reason for this is that the activation of each neural unit becomes too correlated with others because of tight coupling among them. This induces the so-called catastrophic interference problems (McCloskey & Cohen, 1989) in which acquisition of one attractor severely interferes with other ones, and thus limits the memory capacity of the network. The gating networks scheme is employed to avoid this problem by allowing local representation for encoding each distinct behavior primitive (Wolpert & Kawato, 1998; Tani & Nolfi, 1999).

One novel idea shown in experiment-2 is to introduce different time constant dynamics in the lower and the higher levels. The lower level networks learn to anticipate how sensory flow develops in time with their fast time constant dynamics. The higher level network learns to anticipate how the prediction error is generated at each local module in the lower level. The gate opening at each module is controlled by its slow time constant dynamics. The system can be scaled much more by decomposing the complex behavior dynamics into multiple levels of different time constant dynamics, incorporated into local modular representation schemes.

The interactive developmental tutoring is another interesting trial examined in experiment-2. From the experiment results, we learned that the CTRNNs do not learn well all of the arbitrary patterns given, but they have preferences for the patterns to be learned (see details in (Tani et al., 2008)). The network has its own inherent

dynamic characteristics pre-determined by various parameters of the network, which determines such preferences. For example, a CTRNN with a slow time constant cannot catch up to rapid changes in the given teaching set of sensory-motor profiles. In addition, the tendency of generalization in learning at each network also generates the preferences. Each network learns each behavior primitive with variations in teaching trajectories depending on the object position. If the changes of teaching trajectories versus the object position are smooth, such relations can be learned as generalized through repeated learning processes. However, if there are some patterns which cannot be generalized among others, such patterns cannot be learned well. Tutors may not always be conscious about generating such particular teaching patterns by themselves.

The problem here is that it is difficult for tutors to know any networks, learning preferences prior to their actual trials and also to predict how the generalization would proceed in repeated learning. Therefore, tutors need to explore the learning preferences by interacting with the robot driven by 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 because the teaching patterns could not be generalized well. The tutors feel the “intentionality” of the robot driven by the network’s dynamics 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 directions of generalization by the network and the ways of tutoring reconcile each other gradually through the repeated interactive trials. Consequently the motor trajectories satisfy the constraints imposed by the task specifications. Such intensive interactions during the tutoring processes allow the compositionality achieved in the networks to be more “organic” in the sense that inherently natural dynamics gradually develop in the networks.

Experiment-3 looked at another aspect of organizing level structures. When we design more complex network architectures with modules and levels, parameter tuning of the networks becomes more difficult. Actually, the success in experiment-2 required an adequate tuning of both time constants in the lower level network and the higher one. It is also true that the parameters for the gate dynamics such as σ^2 have to be carefully hand-tuned depending on patterns to be learned. Therefore, it is natural to ask if the scheme of the gating modular networks is the only way to scale. Experiment-3 employed the simulated evolutionary exploration scheme in order to examine this problem. It was shown that a level-structured function with different time constant dynamics evolves in the network. The dynamic function with the slower time constant

predicts abstract plans for generating goal-directed behaviors while the one with the faster time constant generates actual motor acts based on the plans.

An interesting finding is that the evolutionary exploration found a mechanism similar to the RNNPB (Tani, 2003), namely a set of behavior primitives are represented distributively in a single network with bifurcation parameters represented by the BNs activations. Although we have discussed that the distributed representation scheme has difficulty encoding a large number of behavior primitives because of interference problems, the scheme has its own merits in other aspects. Firstly, the structure of the network is simple, consisting of a single network associated with a small vector for bifurcation parameters. More importantly, the distributed representation allows sharing of common functions among different behavior primitives within a single network. For example in experiment-3, two behavior primitives that evolved in the network share a common behavior function of collision avoidance against walls. The left turning and the right turning behaviors are just achieved on the top of this collision avoidance mechanism. It can be said that a generalized representation is more likely to self-organize in a distributed representation than in a local one because a set of memory patterns is memorized while preserving relational structures among them in a shared network (Sugita & Tani, 2005). This has been theorized by the PDP group (Hinton, McClelland, & Rumelhart, 1986) for more than two decades. The results of experiment-3 showed us the most implicit way to achieve behavior compositionality where we do not see any explicit “homunculus” to manipulate the representations internally. All that exist are just evolved dynamical structures and their natural time developments in which we, as external observers, might see the emergence of compositionality in their phase space. This aspect should account for the essence of “organic” compositionality.

The discussions in this subsection are summarized as follows: It can be said that the compositionality in representing skills for goal-directed behaviors can be achieved to certain extents in various CTRNN architectures proposed by the authors and that it can be enhanced especially by introducing adequate level structures into the networks. It can be also said that the compositional mechanisms can be “organic” provided that the CTRNNs achieve generalization through learning various sensory-motor experiences and that they capture the contextual nature of the skilled behaviors. It is still a debatable problem whether a distributed representation or a local one should be employed for representing behavior primitives. Although the local representation scheme could gain compositionality more easily than the distributed one because each primitive can be manipulated as an independent object as encoded in a separate local network, it might have less generalization capability among a set of primitives. On this

account, one important future research direction is to explore an intermediate representation scheme between the two extremes of a distributed representation and a local one. It would be beneficial if the degree of distribution in the representation could be optimized in the learning processes. This problem is open for future.

6.2 Interpretations of parietal-premotor interactions

Now, we discuss how the results of our synthetic robotics studies can be reconciled with empirical neuroscience studies. The central hypothesis in the current paper is that the IPL-PMv network might acquire a compositional representation for generating as well as recognizing goal-directed behaviors through iterative learning processes. Under this central hypothesis, one essential assumption is that IPL might play an important role in predicting future sensory images and/or abstract plan level images regarding actions rather than just associating different sensory modalities. The Rizzolatti group had similar speculation in their recent paper (Fogassi et al., 2005) that IPL might form predictive chains regarding sequences of motor acts. They stated that “the organization of IPL receptive fields also favors a model that postulates a chain between neurons coding subsequent motor acts” (Fogassi et al., 2005). Their idea came from the finding of IPL neurons in monkeys which facilitate the mouth opening after they respond to touching or grasping an object (Yokochi, Tanaka, Kumashiro, & Iriki, 2003).

Furthermore, some of our robotics experiment results reviewed in the current paper can be directly related to the recent finding of the goal encoding IPL neurons by Fogassi et al. (2005). In their experiments, monkeys were trained for two different goal-directed behaviors, where one is to grasp a piece of food to eat and another is to grasp a solid object (the same size as the food object) to place it into a cylinder. Their interesting finding is that activation patterns of many IPL neurons during grasping are different depending on the subsequent goals, namely to eat or to place, even though the kinematics during grasping are the same. Their supplemental experiments confirmed that the activation preferences during grasping do not originate from the visual stimulus difference between food and solid objects, but from the goal differences. This finding may correspond to the results obtained from experiment-1. In experiment-1 we showed that the activation temporal patterns of some context neurons while approaching an object are specific to the subsequent goals even though the sensory-motor profiles during this period are the same. It turned out that goal information is preserved in this temporal activation pattern of the context neurons in terms of memory dynamics by utilizing the initial sensitivity characteristics. The same mechanism was observed

in experiment-3 in which navigation skills reaching to different goals are successfully evolved. In this case, the goal information is embedded in the initial state of the TNs. This might also correspond to the finding by Sato et al. (2006) that some neurons in medial parietal regions are involved in goal-directed navigation in monkeys.

It is noted that there is another way to represent goal information. The goal can be represented also by static activation patterns of neurons as have been shown in experiment-2 as well as in our prior studies using the PB vector (Tani & Ito, 2003; Ito & Tani, 2004). In experiment-2 the gate opening sequences are encoded by static input patterns in the higher level CTRNN. However, our preliminary simulation experiments showed that the static vector scheme cannot be extended to learning of tree-like branching action sequences. Such complex sequences might be learned much better with the scheme utilizing the initial sensitivity characteristics of neuro-dynamic systems. Our previous study (Nishimoto & Tani, 2004) as well as the experiment-3 results showed that action branching sequences can be learned by self-organizing fractal structures in the mapping between the initial state and the target sequences. However, a disadvantage utilizing the initial sensitivity is that information about long distance goals preserved in the dynamic memory might be volatile. On the other hand, the goal information encoded as a static input vector is stable because the information is always there. It is expected that future neurophysiological experiments will clarify whether the goal information is encoded in static activation patterns of neurons or is preserved in the dynamic memory of neural activations utilizing the initial sensitivity characteristics.

Another question is whether the goal information is encoded in PMv or in IPL. Originally, mirror neurons that encode goals of actions were found in PMv (Rizzolatti et al., 1996). Fogassi et al. (2005) speculate that the same chained organizations might exist both in PMv and IPL firstly because they are interconnected tightly and secondly because they share similar receptive fields from sensory inputs. We, however, may construct a more persuasive proposition from the modeling view. Our hypothesis is that PMv sets goals of behaviors by sending the initial state vector to the internal neurons in IPL before actual movements initiate. By receiving the initial state, the IPL network initiates its dynamics for generating the corresponding goal-directed behavior. PMv can also recognize goals of observed behaviors by inversely searching the optimal initial states that account for the observations. The central assumption here is that PMv might be involved mainly with manipulation of the initial state either for setting goals or for recognizing goals. On the other hand, the networks in IPL might embody the actual ongoing dynamics of generating the contextual flow of a “kinetic melody”

(Luria, 1973) for achieving the intended goals. It is expected that future joint research by modeling and empirical researchers will examine these arguments.

6.3 Toward understanding brain mechanisms for cognitive behaviors

Finally, we briefly discuss how the robotics modeling approach can contribute to understanding brain mechanisms for cognitive behaviors. It is fair to say that the brain mechanisms of acquiring compositional representations for skilled behaviors are far from completely understood. It is true that recent electrophysiological studies provide us a number of interesting findings on correlations between behaviors and neuron activation patterns in various conditions. Although we can draw plausible explanations on the mechanisms of target cognitive behaviors from those observed data, they are still just hypotheses. On the other hand, although the robotics researchers can show the exact mechanisms of their own robots for generating target cognitive behaviors, they can never prove that those mechanisms actually take place in real brains. What can be shown by either empirical studies or robotics modeling studies might be just a plausibility at the most. Nevertheless, it should be true that the constraints from both sides can elucidate more comprehensive and realistic accounts for the target mechanisms in real brains.

There seem to be two extreme approaches in modeling studies. One general approved approach is to build models of target biological systems as precisely as possible. The objective is to reconstruct the biological behavior of the target systems in digital computers by paying close attention to the exact properties of the target organisms, such as neuronal populations, synaptic connectivities, and the cell firing characteristics. This approach, however, faces potential difficulties when the properties of the target organisms are not well known. And this actually happens in many situations in brain science. For example, we don't yet know the details of neuronal network structures or the exact synaptic connectivities inside parietal cortex. Therefore, we consider an alternative. The alternative approach is concerned only with abstract models of the targets. The objective in this case is not to mimic the reality but to elucidate general mechanisms by comparing among possible models that assume various conditions in the target biological systems, as we have shown in the current paper. The current paper examined how two extremes of local and distributed representations can affect learning and generating behaviors. We also examined how the connectivity constraints in networks, such as the bottleneck, could affect the self-organization of the level-structured

functions. The attempt here is not to look at a specific point in a parameter space even though it might exactly correspond to the biological reality. Rather it is to elucidate the general mechanisms or principles of the target biological systems by comparatively studying various possible points.

Robotics provides us a good opportunity to embody brain models under real world settings by having sensory-motor interactions with the physical world. Studies focusing on such couplings between internal neuronal dynamics and the outer physical world have often resulted in obtaining unique insights for understanding cognitive behaviors. At the same time, we often notice that there is a large diversity in neural network models embodied in robotics experiments. Of course, it is nice to know that each model has a specific advantage in enhancing cognitive behaviors of robots. However, the more important thing is to extract more plausible mechanisms by comparing among various robotics experiments conducted by different research groups. We again emphasize the importance of comparative studies among the various possible assumptions of models. The knowledge accumulated through such trials could inspire empirical neuroscientists to conduct biological experiments in real brains. The cycle of proposing plausible brain mechanisms through robotics modeling and examining the reality of biological brains by empirical research has a good chance of describing the principles of the brain and accounting for various cognitive behaviors.

In order to kick off such trials, a list of possible experiments is proposed that might interest neuroscientists.

1. One question from the modeling side is that if PMv cells actually encode goal information for a particular action, is it by their initial activation state or by their static activation state which lasts during the whole action period? This could be answered by conducting unit recording for populations of PMv cells. Animals would perform for different trained goal tasks which are similar to the monkey tasks shown in (Fogassi et al., 2005).
2. Another question is whether the encoding of behavior primitives assumed in IPL takes a local representation or a distributed one. This question can be examined again by unit recording of populations of IPL cells to see if each behavior primitive is represented by particular activation patterns of the population. After animals are trained to be able to generate a set of behavior primitives with minimum error, measuring overlap between the population of firing for different behavior primitives would show the distribution in their neuronal encoding. If some locality is found, a further question might be if we can find mutually inhibitory

mechanisms among such local regions, corresponding to a gating mechanism.

3. However, if behavior primitives are in fact represented by the correlation of cell firing in a population, it might be necessary to use simultaneous recording with multiple electrodes.
4. The last experiment might be a bit difficult. In the previous section, it was speculated that some population of cells which have a similar but broad response to the same stimulus could be modeled as a dynamic unit with a slow time constant. On the other hand those cells with a sharp response with synchronization could be by the ones with a fast time constant. If we can identify such populations characterized by different time constants in their collective activities, the question is whether the distribution of the time constants shows certain locality, for example in IPL. Although measuring such locality by using simultaneous recording would be very difficult, finding such locality in IPL could lead to the finding of level structures as well as possible mechanisms for the organic compositionality in this cortical region.

7 Summary

The current paper introduced a synthetic approach using robots to understand brain mechanisms for achieving “organic” compositionality in representing skills to achieve multiple goal-directed behaviors. Inspired by neuroscience studies on the roles of IPL-PMv interactions, we assumed that IPL networks may predict motor-related sensory flow by means of forward dynamics based on the goal information input from PMv. Motor flow may be generated in association with the forward prediction of future sensory flow. We also assumed that the goals of observed motor behaviors can be recognized by means of an inverse computation through the same forward prediction network in IPL. Based on these assumptions, a set of different dynamic neural network models has been proposed and they have been implemented in different robotics experiments.

The first experiment showed that multiple goal-directed behaviors can be learned with certain generalization in a simple CTRNN architecture utilizing its initial sensitivity characteristics. Our analysis showed that the motor behaviors are fluently articulated by means of a pitchfork bifurcation appearing in the network dynamics. The second experiment showed that the CTRNN scheme can be scaled by introducing some modular and level structures in the network architectures. We also showed that the interactive and incremental tutoring scheme can enhance the acquisitions of

the skilled behaviors. The third experiment showed that the CTRNN can evolve to possess fast and slow time constant dynamics internally by limiting their mutual interactions by having the bottleneck. It was observed that by coupling these two sets of dynamics, level-structured functions emerge by which multiple goal-directed behaviors can be generated. It is concluded that the self-organization process in each of the proposed dynamic neural network models can achieve “organic” compositionality that affords generalization, robustness and context-dependency in generating multiple goal-directed behaviors under specific conditions for each.

It is expected that future iterative conversations between robotics modellers and empirical neuroscientists will elucidate more plausible mechanisms of brain function with compositionality in generating cognitive behaviors.

References

- McDonald, S., Tate, R., & Rigby, J. Error types in ideomotor apraxia: a qualitative analysis. *Brain and Cognition*, 25(2).
- Arbib, M. (1981). Perceptual structures and distributed motor control. In *Handbook of Physiology: The Nervous System, II. Motor Control* (pp. 1448–1480). Cambridge, MA: MIT Press.
- Beer, R. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1), 173–215.
- Billard, A., & Mataric, M. (2001). Learning human arm movements by imitation: Evaluation of a biologically-inspired connectionist architecture. *Robotics and Autonomous Systems*, 941, 1–16.
- Colby, C., Duhamel, J., & Goldberg, M. (1993). Ventral intraparietal area of the macaque: anatomic location and visual response properties. *Journal of Neurophysiology*, 69, 902–914.
- Demiris, J., & Hayes, G. (2002). Imitation as a dual-route process featuring predictive and learning components: a biologically plausible computational model. 327–361.
- Doya, K., & Yoshizawa, S. (1989). Memorizing oscillatory patterns in the analog neuron network. In *Proc. of 1989 int. joint conf. on neural networks, washington, d.c.* (pp. I:27–32).

- Ehrsson, H., Fagergren, A., Johansson, R., & Forssberg, H. (2003). Evidence for the involvement of the posterior parietal cortex in coordination of fingertip forces for grasp stability in manipulation. *Journal of Neurophysiology*, *90*, 2978–2986.
- Eskandar, E., & Assad, J. (1999). Dissociation of visual, motor and predictive signals in parietal cortex during visual guidance. *Nature Neuroscience*, *2*, 88–93.
- Evans, G. (1981). Semantic Theory and Tacit Knowledge. In S. Holzman & C. Leich (Eds.), *Wittgenstein: To Follow a Rule* (pp. 118–137). London: Routledge and Kegan Paul.
- Fagg, A. H., & Arbib, M. A. (1998). Modeling parietal-premotor interactions in primate control of grasping. *Neural Networks*, *11*, 1277–1303.
- Fetz, E., Finocchio, D., Baker, M., & Soso, M. (1980). Sensory and motor responses of precentral cortex cells during comparable passive and active joint movements. *Journal of Neurophysiology*, *43*, 1070–1089.
- Fogassi, L., Ferrari, P., Gesierich, B., Rozzi, S., Chersi, F., & Rizzolatti, G. (2005). Parietal lobe: from action organization to intention understanding. *Science*, *308*, 662–667.
- Geschwind, N., & Kaplan, E. (1962). Human cerebral disconnection syndromes. *Neurology*, *12*, 675–685.
- Hatsopoulos, N., Ojakangas, C., Paninski, L., & Donoghue, J. (1998). Information about movement direction obtained from synchronous activity of motor cortical neurons. *Proc. Natl. Acad. Sci. USA*, *95*, 15706–15711.
- Heilman, K. (1973). Ideational apraxia - a re-definition. *Brain*, *96*, 861–864.
- Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cog. Sci.*, *6*(6), 242-247.
- Hinton, G., McClelland, J., & Rumelhart, D. (1986). Distributed representation. In D. Rumelhart & J. McClelland (Eds.), *Parallel distributed processing* (pp. 77–109). Cambridge, MA: MIT Press.
- Iacoboni, M. (2006). Visuo-motor integration and control in the human posterior parietal cortex: Evidence from tms and fmri. *Neuropsychologia*.

- Inamura, T., Nakamura, N., Ezaki, H., & Toshima, I. (2001). Imitation and primitive symbol acquisition of humanoids by the integrated mimesis loop. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 4208–4213).
- Inamura, T., Toshima, I., Tanie, H., & Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *International Journal of Robotics Research*, *23*(44), 363–377.
- Ito, M. (1970). Neurophysiological basis of the cerebellar motor control system. *Int. J. Neurol*, *7*, 162–176.
- Ito, M. (2005). Bases and implications of learning in the cerebellum - adaptive control and internal model mechanism. *Progress in Brain Research*, *148*, 95–109.
- Ito, M., & Tani, J. (2004). On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, *12*(2), 93–114.
- Jeannerod, M. (1994). The representing brain: neural correlates of motor imitation and imaginary. *Behavioral and Brain Science*, *17*, 187–245.
- Jordan, M. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proc. of eighth annual conference of cognitive science society* (pp. 531–546). Hillsdale, NJ: Erlbaum.
- Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural network model for the control and learning of voluntary movement. *Biological Cybernetics*, *57*, 169–185.
- Kelso, J. (1994). Elementary coordination dynamics. In S. Swinnen (Ed.), *Interlimb coordination: neural, dynamical, and cognitive constraints*. New York, Academic Press.
- Kuniyoshi, Y., Inaba, M., & Inoue, H. (1994). Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance. *IEEE. Trans. on Robotics and Automation*, *10*(6), 799–822.
- Liepmann, H. (1920). Apraxie. *Erg ges Med*, *1*, 516–543.
- Luria, A. (1973). *The working brain*. Penguin Books Ltd.

- McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist network. *Psych. Learning Motivat.*, *24*, 109–165.
- Nishimoto, R., Namikawa, J., & Tani, J. (2008). Learning Multiple Goal-Directed Actions through Self-Organization of a Dynamic Neural Network Model: A Humanoid Robot Experiment. *Adaptive Behavior*. (in press)
- Nishimoto, R., & Tani, J. (2004). Learning to Generate Combinatorial Action Sequences Utilizing the Initial Sensitivity of Deterministic Dynamical Systems. *Neural Networks*, *17*, 925–933.
- Ohshima, F., Takeda, K., Bandou, M., & Inoue, K. (1998). A case of ideational apraxia -an impairment in the sequence of acts-. *Journal of Japanese Neuropsychology*, *14*, 42–48.
- Oztop, E., & Arbib, M. A. (2002). Schema Design and Implementation of the Grasp-Related Mirror Neuron System. *Biological Cybernetics*, *87*, 116–140.
- Oztop, E., Wolpert, D., & Kawato, M. (2005). Mental state inference using visual control parameters. *Cognitive Brain Research*, *22*, 129–151.
- Paine, R., & Tani, J. (2005). How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior*, *13*(3), 211–225.
- Rizzolatti, G., Fadiga, L., Galless, V., & Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, *3*, 131–141.
- Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. Rumelhart & J. McClelland (Eds.), *Parallel distributed processing* (pp. 318–362). Cambridge, MA: MIT Press.
- Sakata, H., Taira, M., Murata, A., & Mine, S. (1995). Neural mechanisms of visual guidance of hand action in the parietal cortex of the monkey. *Cereb Cortex*, *5*, 429–438.
- Sakurai, Y., & Takahashi, S. (2006). Dynamic synchrony of firing in the monkey prefrontal cortex during working-memory tasks. *The Journal of Neuroscience*, *26*(40), 10141–10153.
- Sato, N., Sakata, H., Tanaka, Y., & Taira, M. (2006). Navigation-associated medial parietal neurons in monkeys. *PNAS*, *103*, 17001–17006.

- Schaal, S., Ijspeert, A., & Billard, A. (2003). Computational approaches to motor learning by imitation. *philosophical transaction of the royal society of london: series b, biological sciences*(1431), 537–547.
- Schoner, S., & Kelso, S. (1988). Dynamic Pattern Generation in Behavioral and Neural Systems. *Science*, *239*, 1513-1519.
- Soso, M., & Fetz, E. (1980). Responses of identified cells in postcentral cortex of awake monkeys during comparable active and passive joint movements. *Journal of Neurophysiology*, *43*, 1090–1110.
- Sugita, Y., & Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior*, *13*(3), 33–51.
- Tani, J. (1996). Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective. *IEEE Trans. on SMC (B)*, *26*(3), 421–436.
- Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction process. *Neural Networks*, *16*, 11–23.
- Tani, J., & Fukumura, N. (1994). Learning Goal-directed Sensory-based Navigation of a Mobile Robot. *Neural Networks*, *7*(3).
- Tani, J., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment. *IEEE Trans. on Sys. Man and Cybern. Part A*, *33*(4), 481–488.
- Tani, J., Nishimoto, R., Namikawa, J., & Ito, M. (2008). Codevelopmental learning between human and humanoid robot using a dynamic neural network model. *IEEE Trans on System, Man and Cybern*, *38*(1), 43–59.
- Tani, J., & Nolfi, S. (1998). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. In R. Pfeifer, B. Blumberg, J. Meyer, & S. Wilson (Eds.), *From animals to animats 5*. Cambridge, MA: MIT Press. (later published in *Neural Networks*, vol12, pp1131–1141, 1999)
- Tani, J., & Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, *12*, 1131–1141.

- Wiggins, S. (1990). *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. New York: Springer-Verlag.
- Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*, 270–280.
- Wolpert, D., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.
- Yamamoto, T., & Kuniyoshi, Y. (2002). Stability and controllability in a rising motion: a global dynamics approach. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems 2002* (pp. 2467–2472).
- Yokochi, H., Tanaka, M., Kumashiro, M., & Iriki, A. (2003). Inferior parietal somatosensory neurons coding face-hand coordination in Japanese macaques. *Somatosensory and Motor Research*, *20*, 115–125.