Neural Networks, Vol.17, pp.1273-1289, 2004

Self-Organization of Distributedly Represented Multiple Behavior Schemata in a Mirror System: Reviews of Robot Experiments Using RNNPB

Jun Tani
Brain Science Institute, RIKEN
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

Masato Ito
Sony Corp.
Gotanda, Shinagawa-ku, Tokyo

Yuuya Sugita
Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama, 351-0198 Japan

(Appeared in Neural Networks, 17, 8-9, pp1273-1289, 2004)

Acknowledgment
Requests for reprints should be sent to Jun Tani, Brain Science Institute, RIKEN. 2-1 Hirosawa, Wako-shi, Saitama, 351-0198 Japan

Running Title
Self-Organization of Multiple Behavior Schemata

Key Words
self-organization, distributed representation, robot, behavior primitives, mirror neurons

Abstract

The current paper reviews a connectionist model, the recurrent neural network with parametric biases (RNNPB), in which multiple behavior schemata can be learned by the network in a distributed manner. The parametric biases in the network play an essential role in both generating and recognizing behavior patterns. They act as a mirror system by means of self-organizing adequate memory structures. Three different robot experiments are reviewed: robot and user interactions; learning and generating different types of dynamic patterns; and linguistic-behavior binding. The hallmark of this study is explaining how self-organizing internal structures can contribute to generalization in learning, and diversity in behavior generation, in the proposed distributed representation scheme.

1 Introduction

The ideas of motor and behavior schemata (Arbib, 1981; Feldman, 1980; Bizzi, Acornero, Chapple, & Hogan, 1984) are indispensable when biological or artificial systems are required to generate behavior patterns flexiblely as adapted to their environmental situations. The underlying idea is that a set of motor programs are stored by which their combinations in space and time can generate variety of behaviors. There are two essntial requirements concerning behavior schemata. One is about the compositionality. The motor systems should have certain manipulatable structures by which various combinations of behavior patterns can be generated. The other is about their grounding. It is required that actual behavior patterns should be generated from the schemata robustly in the tight coupling with the environmental dynamics. The current paper address these issues by reviewing studies conducted by the authors as well as others for recent years.

One central discussion in the current paper is how a set of behavior schemata are embedded in memories organized in neuronal network models. Is each of the behavioral schemata memorized in a corresponding local network module independently? Or are all of them memorized distributively in a network without having specific modules? The most of prior studies have investigated the localist representation scheme. Tani and Nolfi (1998) and Wolpert and Kawato (1998) proposed modular neural network schemes in which each pair of inverse and forward models for specific behavior primitives is embedded in a corresponding local expert network. One network module is selectively activated by means of winner-take-all dynamics among the modules in order to recognize or generate a specific learned behavior primitive. In the computational models of Amit and Mataric (2002) and Inamura, Nakamura, Ezaki, and Toshima (2001), a cluster of motor primitives are organized in the lower level. Then, each local hidden Markov model (HMM) in the higher level learns a specific sequence of activating those primitives. Ijspeert, Nakanishi, and Schaal (2003) envision motor primitives that consist of oscillatory movements and discrete movements. Each of these two types of primitives can be represented by a specific differential equation, where movement profiles can be modified by changing the equation parameters. In the localist scheme, a new behavior schema can be learned by just adding its template to the existing set of local modules. This type of learning is quite easy since the addition does not cause any memory interference with the current memory contents because of the independence of each module. However, the question may arise of how such learning can achieve generalization for unlearned patterns. In order to generalize the learned contents, certain underlying structures, accounting for each instance of learning, should be acquired. If the localist scheme is employed, each behavior schema is learned as an independent template in which generalization across different behavior schemata becomes difficult.

The current paper reviews the possibility of a distributed representation scheme as an alternative which the authors have studied in recent years in the context of behavior learning by robots. In the distributed representation, multiple behavior schemata are embedded in a single neuronal network. Each schema is memorized distributively over all synaptic weights, and is thus represented among the activations of all neurons within the network. In such situations, each schema memory is no longer independent but can exist only in relation to others. It is considered that learning is a process of self-organizing the global structure, accounting for the relations among the memorized schemata. The main focus of the current paper is to examine what sorts of structures emerge through the self-organization process by means of the proposed distributed representation scheme. We further investigate how such structures could contribute to generalization in learning and diversity in behavior generation. This part of the analysis will be the highlight in the current study.

Our distinction between the local and the distributed representation so far might be similar to the distinction between synchronic and diachronic modularity discussed by Ziemke (2000) in which "synchronic" referes to multiple modules existing as separated hardwares at the same time while "diachronic" referes to multiple schemata that are different instantiations of a single hardware at different occasions.

Another essential characteristics in the model proposed by the authors is that the system performs in both generation and recognition of behavior patterns as a mirror system (Pellegrino, Fadiga, Fogassi, Galless, & Rizzolatti, 1992; Rizzolatti, Fadiga, Galless, & Fogassi, 1996). Pellegrino et al. (1992) discovered that there is a "mirror system" in which those neurons active when the monkey executes a specific object handling behavior are also active when the monkey observes other monkeys or humans carrying out the same behavior. Based on this finding, Oztop and Arbib (2002), Billard and Mataric (2001) proposed biologically inspired models of mirror systems. Inamura et al. (2001), Amit and Mataric (2002) showed modeling of mirror systems in the computational level by using HMMs . On the other hand, the authors attempt to explain functions of mirror systems in the dynamical systems level where the behavior generation is regarded as a top-down process and its recognition as a bottom-up process. This bi-directional processes are considered to play an important roles also in generating its own behaviors while recognizing their consequences in the environment simultaneousely. The authors speculate that adequate dynamic interactions between these bottom-up and top-down processes might solve possible conflicts between the compositionality and the grounding of behavior schemata.

The paper first reviews our proposed scheme, the so-called recurrent neural network with parametric biases (RNNPB) (Tani, 2002, 2003; Tani & Ito, 2003), through which multiple behavior patterns can be learned for their generation and recognition as a mirror system. The scheme has been implemented for three different robotics tasks. The first experiment, utilizing a small humanoid robot, will demonstrate how the RNNPB in terms of a mirror system works in the imitative interactions between a user and the robot. This study will examine the dynamic interactions between the top-down behavior generation process and the bottom-up recognition process of others, especially focusing on synchronization between those two processes. The second experiment with an arm robot will clarify the dynamic structures self-organized in the RNNPB, especially when different types of movement patterns (end-point movements and cyclic movements) are learned simultaneously. The analysis will clarify the generalization characteristics in learning movement patterns and the diversity in their generations. The third experiment with a mobile robot will show binding of behavioral processes dealing with objects and simple linguistic processes consisting of verb and object pairs of which task setting is analogous to Arbib (2002)'s hypothesis relating the mirror neurons with language. The analysis of the self-organized structures will reveal the compositionality and the generalization in the associative learning between behaviors and language by using the RNNPB.

2 Model overview

This section presents the main ideas behind our proposed model RNNPB. For details of the modeling, please refer to our prior publications (Tani, 2002, 2003; Tani & Ito, 2003; Sugita & Tani, 2003). The main characteristic of the RNNPB is that chunks of spatio-temporal patterns of the sensory-motor flow can be represented by a vector of small dimensions. This vector plays the role of the bifurcation parameters of nonlinear dynamical systems. In other words, different vector values make the system generate different dynamic patterns. In our modeling, the nonlinear dynamical system is implemented by a Jordan-type recurrent neural network (Jordan, 1986). The parametric biases (PB) that are allocated in the input layer function as the bifurcation parameters. It is reminded that the RNNPB scheme is different from Doya and Yoshizawa (1989)'s scheme of multiple oscillatory patterns learning by an RNN since their model utilizes the multiple attractor structures rather than the parameter bifurcation for embedding different oscillatory patterns. The main advantage of utilizing the parameter bifurcation is that ideally the RNNPB can encode infinite number of dynamic patterns with modulating analog values of the PB vector while the number of patterns eoncoded by multiple attractors is limited in general.

The role of learning is to self-organize the mapping between the PB vector and behavioral spatio-temporal patterns. It is important to note that the PB vector for each learning pattern is self-determined in a non-supervised manner, without teacher signals. Another feature of the RNNPB is that the system works as both a behavior recognizer and generator as a mirror system after learning. When given a fixed PB vector, the RNNPB generates the corresponding dynamic patterns. On the other hand, when given target patterns to be recognized, the corresponding PB vectors are obtained through an iterative inverse computation.

In the learning phase, a set of movement patterns are learned through the forward model of the RNNPB by self-determining both the PB vectors, which are assigned differently for each movement pattern, and a synaptic weight matrix, which is common for all the patterns. The information flow of the RNNPB in the learning phase is shown in Figure 1(a). This learning is conducted using both target sequences of motor values m_t and the sensory values s_t. When given m_t and s_t in the input layer, the network predicts their values at the next time step in the output layer as $\hat{m_{t+1}}$ and $\hat{s_{t+1}}$. The outputs are compared with their target values m_{t+1} and s_{t+1} and the error generated is back-propagated (Werbos, 1990; Rumelhart, Hinton, & Williams, 1986) for the purpose of updating both the synaptic weights and PB vectors. Note that the determined synaptic weights are common to all learning patterns, but the PB vector is differently determined for each pattern. The manner of determining the PB vectors will be detailed in later sections. c_t represents the context units where the self-feedback loop is established from c_{t+1} in the output layer to c_t in the input layer. The context unit activations represent the internal state of the network.

After the learning is completed, the sensory-motor sequences can be generated by means of the forward dynamics of the RNNPB with the PB vectors fixed as shown in Figure 1(b). The PB vectors could be given from another network, as in the behavior-language association task described later, or self-determined through the recognition process, as in the imitative interaction task with the humanoid robot. In the generation phase, the RNNPB can be operated in a closed-loop mode where the next step's sensory-motor prediction outputs are fed back to the current step as inputs, as denoted by a dotted line on the left-hand side in Figure 1(b). Thus, the RNNPB can generate imaginary sensory-motor sequences without receiving the actual sensory inputs from

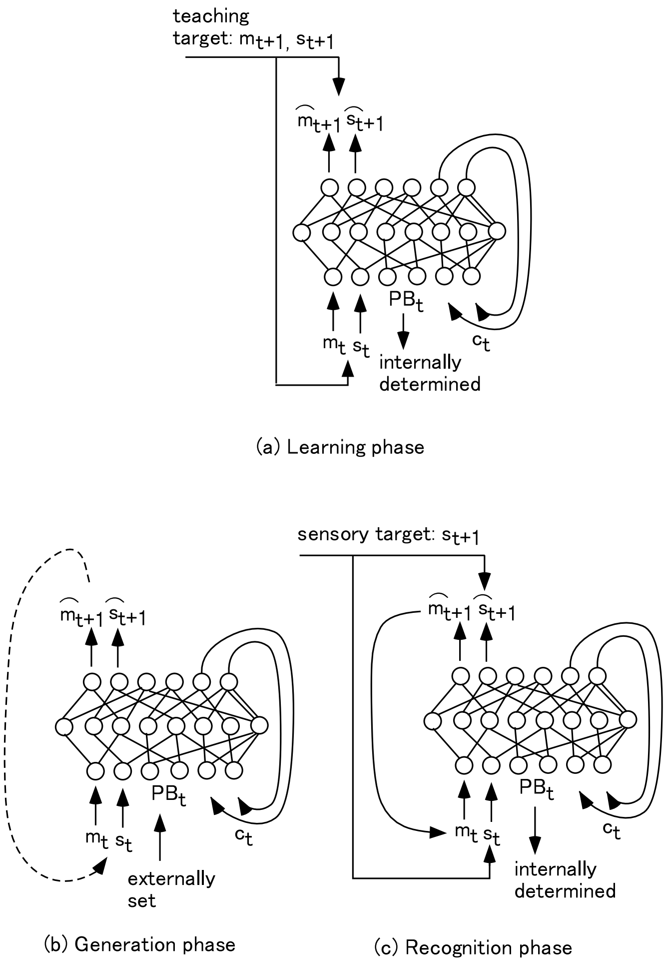

Figure 1: The system flow of RNNPB in learning phase (a) and testing phase (b).

the environment.

Figure 1(c) illustrates how the PB vectors can be inversely computed for the given target sensory sequences in the recognition phase. The RNNPB, when receiving the current sensory inputs s_t, attempts to predict their next vectors, $\hat{s_{t+1}}$, by utilizing the temporarily obtained PB vectors. The generated prediction error from the target value s_{t+1} is back-propagated to the PB units and the current PB vectors are updated in the direction of minimizing the error. The actual computation of the PB vectors is conducted by using the so-called regression window of the immediate past steps, by which the PB vectors can be modulated smoothly through the steps. (This mechanism will be detailed in the next section.) If pre-learned sensory sequence patterns are perceived, the PB vectors tend to converge to the values that were determined in the learning phase.

It is noted that the role of the PB vector is similar to the integration units introduced in layered networks by Yamauchi, Ohta, and Ishii (1999) where the integration units encode multiple static patterns through the self-supervised learning. The RNNPB could be regarded as the developments of their model in order to deal with dynamic patterns.

3 Computing the PB values

The PB vectors are determined through regression of the past sequence pattern. In the recognition phase, the regression is applied for the immediate past window steps L, by which the temporal profile of the PB, p_t from L steps before to the current step ct, is updated. The window for the regression shifts as time goes by while p_t is updated through the iterations. In the learning phase the regression is conducted for all steps of the training sequence patterns. (This means that the window contains the whole sequence and it does not shift.)

The temporal profile of p_t in the sequence is computed via the back-propagation through time (BPTT) algorithm (Werbos, 1990; Rumelhart et al., 1986). In this computation ρ_t, the internal value of the parametric bias, is obtained first.

The internal value ρ_t changes due to the update computed by means of the error back-propagated to this parametric bias unit, which is integrated for a specific step length in the sequence. Then the parametric bias, p_t, is obtained by a sigmoid function of the output of the internal value. The utilization of the sigmoid function is just a way of computationally bounding the value of the parametric bias to a range of 0.0 to 1.0. In this way, the parametric bias is updated to minimize the error between the target and the output sequence.

For each iteration in the regression of the window, L steps of look-ahead prediction, starting from the onset step of the window, are computed by the forward dynamics of the RNN. Once the L steps of the prediction sequence are generated, the errors between the targets and the prediction outputs are computed and then back-propagated through time. The error back-propagation updates both the values of the parametric bias at each step and the synaptic weights. The update equations for the ith unit of the parametric bias at time t in the sequence are:

$$\begin{aligned} \delta{\rho_t}^i &= k_{bp} \cdot \sum_{step=t-l/2}^{t+l/2} \delta_t^{bp^i} + k_{nb}(\rho_{t+1}^i - 2\rho_t^i + \rho_{t-1}^i) && (1) \\ \triangle\rho_{t(s+1)}^i &= \epsilon \cdot \delta{\rho_t}^i + \eta \cdot \triangle\rho_{t(s)} && (2) \\ p_t^i &= sigmoid(\rho_t) && (3) \end{aligned}$$

In Eq. (1), $\delta\rho_t$, the delta component of the internal value of the parametric bias unit, is obtained from the summation of two terms. The first term represents the summation of the delta error, $\delta_t^{bp^i}$, in the parametric bias units for a fixed time duration l. $\delta_t^{bp^i}$, which is the error back-propagated from the output units to the ith parametric bias unit, is summed over the period from $t-l/2$ to $t+l/2$ time steps. By summing the delta error, the local fluctuations of the output errors will not affect the temporal profile of the parametric bias significantly. The parametric bias should vary only with structural changes in the target sequence. Otherwise it should become flat, or constant, over time.

The second term plays the role of a low pass filter through which frequent rapid changes of the parametric bias are inhibited. k_{nb} is the coefficient for this filtering effect. ρ_t is updated based on $\delta\rho_t$ obtained in Eq. (1). The actual update $\triangle\rho_{t(s+1)}$ at $s+1$ learning step from that at s learning step is computed by utilizing a momentum term to accelerate convergence as shown in Eq. (2). Then, the current parametric bias p_t is obtained by means of the sigmoidal outputs of the internal values ρ_t in Eq. (3).

4 Imitative interactions

In this experiment (see the details in (Ito & Tani, 2003b)), we examined how the robot can recognize the user‘s hand movement patterns in the sensory inputs and generate corresponding imitative movement patterns of its own by retrieving from the learned memory as a result. In this task setting, the movement patterns of the robot have to be generated synchronously with the sensation of the user hand movement patterns.

Figure 2: A user is interacting with the Sony humanoid robot QRIO SDR-4XII.

4.1 Task setting

The Sony humanoid robot QRIO SDR-4XII (Fujita, Kuroki, Ishida, & Doi, 2003) is used as the experimental platform in this experiment (see Figure 2). In this experiment, only movement patterns of both arms are considered. Other movements are frozen.

The robot task consists of learning and interaction phases. In the learning phase, a set of robot cyclic movement patterns with different periods is learned and associated with the corresponding user's visually perceived hand movement patterns. The target trajectories of the robot movement patterns are obtained by mapping the user's arm position to the robot joint angles. This mapping is done through engineering using optical measuring as described in (Ito & Tani, 2003b). As summarized in Figure 3(a), the learning process utilizes the paired trajectories of the robot joint angles (8DOF), obtained by the mapping, and the user's hand positions (4DOF), as visually perceived by the robot. The training of the employed neural network model (RNNPB) is conducted by using a set of training patterns, corresponding to multiple robot and user movement patterns.

In the interaction phase, the robot attempts to follow synchronously the user's hand movement patterns. This is done by utilizing a mirror system characteristics of the RNNPB in which the recognition of other's hands and the generation of its own are carried out simultaneosuely. As shown in Figure3(b), while the robot perceives the user's hand movement patterns visually, the PB vectors are iteratively adapted in real

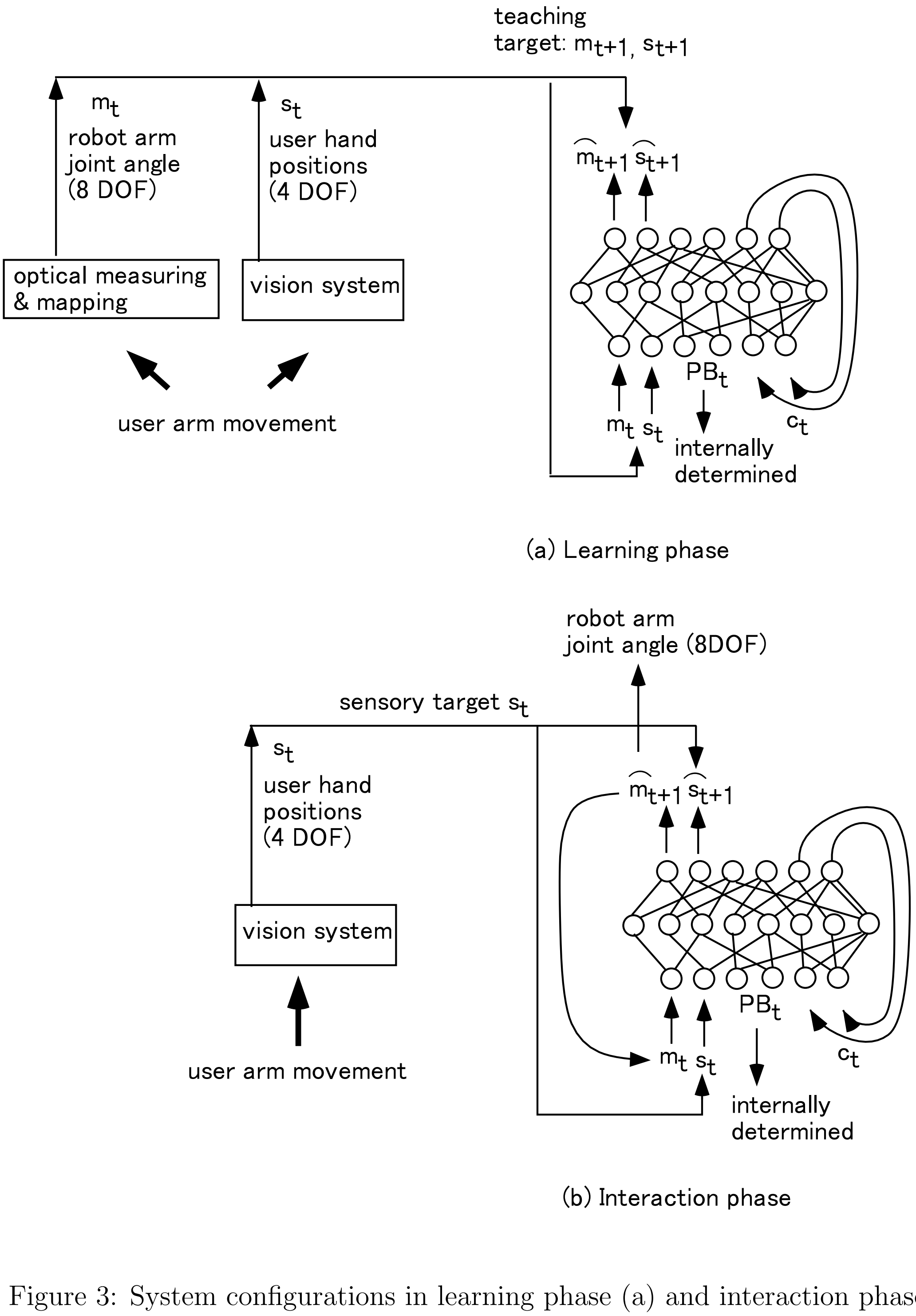

Figure 3: System configurations in learning phase (a) and interaction phase (b).

time in order to minimize the sensory prediction errors; the robot movement patterns in joint angle are generated simultaneously by means of the forward dynamics of the RNNPB with the PB vector of the current updates. The robot's ability to follow the user depends on the degree to which the user patterns are familiar to the robot, based on prior learning.

The RNNPB used in these experiments has 12 input nodes and 12 prediction output nodes for learning the forward dynamics of movement patterns. The patterns are composed of 4 vectors, representing the positions of user hands, and 8 vectors, representing the joint angles of the robot arms. It also has 4 parametric nodes, 40 hidden nodes, and 30 context nodes.

Demiris and Hayes (2002) introduced multiple pairs of forward and inverse models for imitation tasks in which a currently perceived companion's behavior pattern is recognized by one of the forward models by means of its prediction of the patterns and then the paired inverse model generates the corresponding own movement patterns. Although their ideas of the imitation by means of prediction is similar to ours, their architecture is different from ours in the current paper since they employ the localist representation which is similar to the models shown by Tani and Nolfi (1998) and Wolpert and Kawato (1998).

4.2 Experiments and analysis

In this experiment, the robot is trained with 3 different cyclic movement patterns associated with the corresponding user hand movement patterns in the learning phase. Then, in the interaction phase we examine how the robot can follow target patterns while the user switches to demonstrate among various learned patterns. More specifically, the user demonstrates patterns 1, 2, and 3 sequentially for about 20 seconds each.

The results of the experiment, including the time course of the interaction and the PB of the RNNPB, are plotted in Figure 4. In Figure 4, the plot in the top row shows the target position of user hands. The plot in the second row shows the user hands's positions predicted by the RNNPB. The plot in the third row shows the robot joint angles generated by the RNNPB (Only 6 DOF are plotted out of a total of 8 DOF). The plot at the bottom shows the parametric bias of the RNNPB. It is observed that when the user hand movement pattern is switched from one pattern to another, the patterns in the sensory prediction and the motor outputs are also switched correspondingly by accompanying substantial shifts in the PB vectors. Although the synchronization

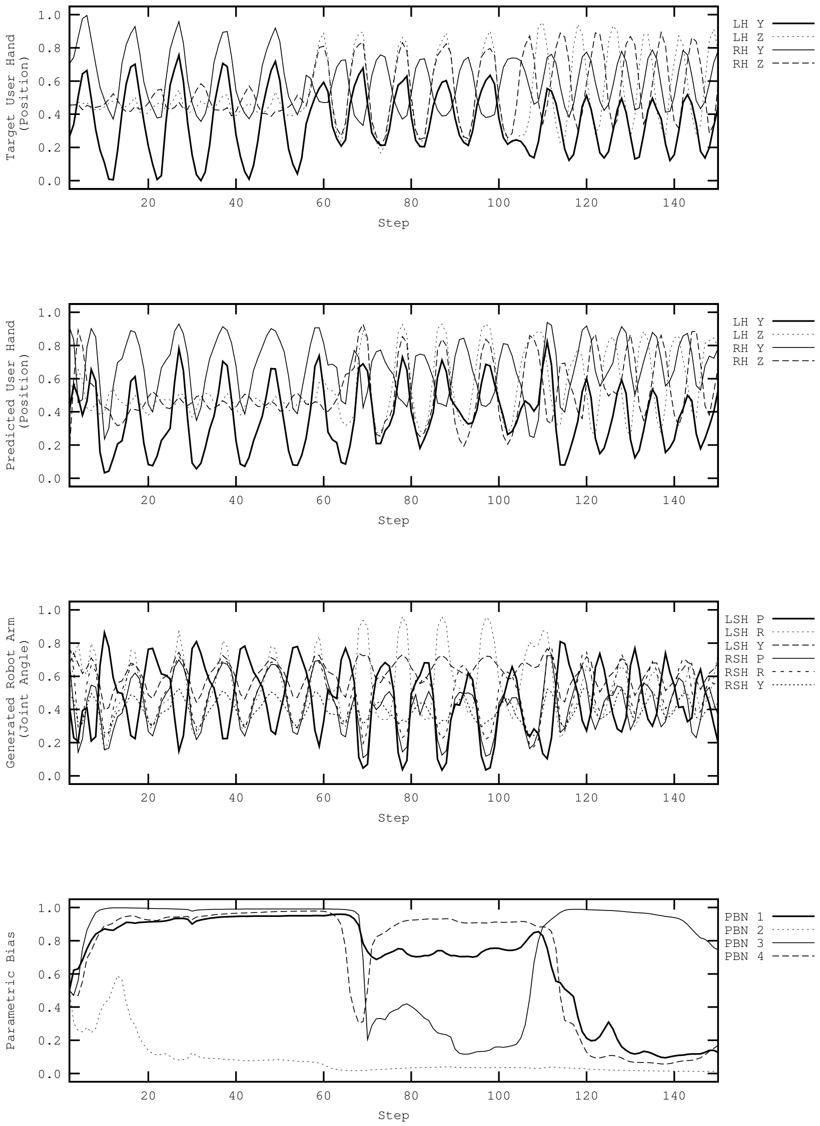

Figure 4: Switching of the robot movement pattern among three learned patterns as initiated by switching of user hand movement.

between the user hand movement pattern and the robot movement pattern is lost once during the transitions, the robot movement pattern is re-synchronized to the user hand movement pattern within several steps. During the experiments, it was also observed that the patterns once synchronized between the two were preserved robustly against certain perturbations in the repetions of the user's hand movements.

Phase space analyses were conducted to analyze the dynamical structure self-organized in the RNNPB through learning. In order to examine the attractor corresponding to each memory pattern in the RNNPB, the RNNPB is set to a closed-loop mode (the outputs of the current step predictions are fed back to the inputs in the next time step) and its forward dynamics is computed for 1000 steps for the PB vector determined for each learned pattern. Its trajectory, in terms of the values of two arbitrarily selected context units, is then plotted in the two dimensional state space. The initial transient trajectory is excluded. We also conducted phase space analyses for the open-loop dynamics of the RNNPB during the user interactions. The forward computation and the regression for updating the PB are simultaneously conducted while one of the learned user movement patterns is fed to the RNNPB repeatedly as the sensory target. Then, the trajectories of the same two context units used in the closed-loop mode are plotted. In Figure 5 (a) shows that three different shapes of attractors appeared in the closed-loop dynamics of the RNNPB as corresponding to each of the learned patterns. They turn out to be limit cycling attractors. In Figure 5 (b) shows the attractor obtained during the user interactions. It is observed that their shapes in the closed-loop and open-loop dynamics are mostly the same for each learned pattern. These analyses confirm that each learned pattern is embedded in the RNNPB as a distinct limit cycling attractor. This intrinsic dynamics of the RNNPB is preserved when it is coupled with the corresponding user movement pattern. This sort of the coherence is achieved by the entrainment of the intrinsic dynamics of the RNNPB by the external dynamics of the user hand movement. This reminds us of entrainments of walking patterns by environmental sensory feedback shown by Beer (1995), Taga (1996), Miyake (2002).

Based on these observations and the analysis, one may conclude that the attractor dynamics system with its bifurcation mechanism by the PB makes the behavior system to be manipulatable by the users as well as robust enough against possible perturbations.

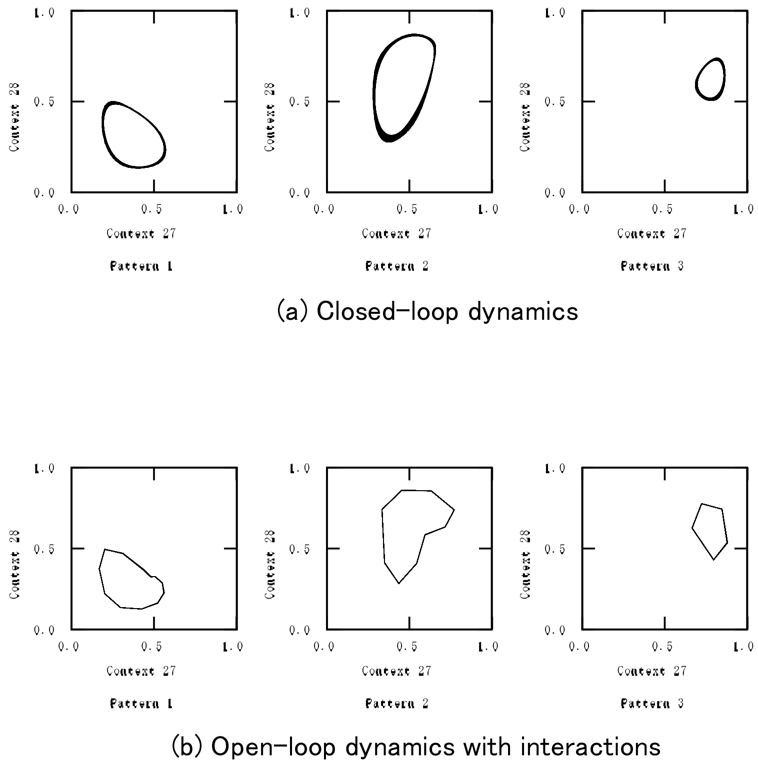

Figure 5: (a) three attractors appeared in the closed-loop dynamics of the RNNPB and (b) their corresponding attractors in the open-loop dynamics with the user interactions.

5 Learning both end-point and cyclic movements

In the second experiment using an arm robot, we demonstrate that the robot can learn two different types of movement patterns, fixed end-point movements and cyclic movements, simultaneously in the RNNPB. End-point movement means that the robot reaches a target position and stops there. In cyclic movements, the robot repeats a periodic pattern. The focus of this experiment is to examine how the mapping from the PB vectors to movement patterns is generated for embedding different types of attractor dynamics.

5.1 Task setting

The robot used in the experiments has 4 degrees of freedom in its arm rotational joints. A hand attached to the arm can sweep over the task table horizontally as shown in Figure 6. The hand has a color mark and its position in X-Y coordinates on the table can be recognized by the vision camera mounted on the robot by using a color filtering scheme. A handle is attached to the hand so that a trainer can teach behavior to the arm manually.

The RNNPB deals with 4 DOF motor outputs and 2 DOF sensory inputs in terms of the visually perceived hand position. It has 20 hidden units and 8 context units. It also has 4 parametric bias units in the input layer. The robot was simultaneously trained for 3 different end-point movement patterns and 2 different cyclic movement patterns through manual guidance of the arm trajectories. Those 5 trajectories are shown in Figure 7. The BPTT learning for all the training sequences was iterated 20,000 times starting from randomly set initial synaptic weights.

5.2 Experiments and analysis

We tested the robot's ability to successfully regenerate each trained movement pattern by setting the corresponding PB vector. In this behavior regeneration test, the PB vectors are sequentially switched from those obtained for one cyclic movement pattern to those for another cyclic movement pattern, and then to those for an end-point movement. This sequential switching of the PB is done manually in the current experiment. Figure 8 shows motor pattern generation in the open-loop mode over time and the corresponding PB vectors in the top and bottom rows, respectively. Observe that the trained behavior patterns appear one by one, corresponding to the switching of the PB vectors. The results, indicate that different types of dynamic patterns, corresponding

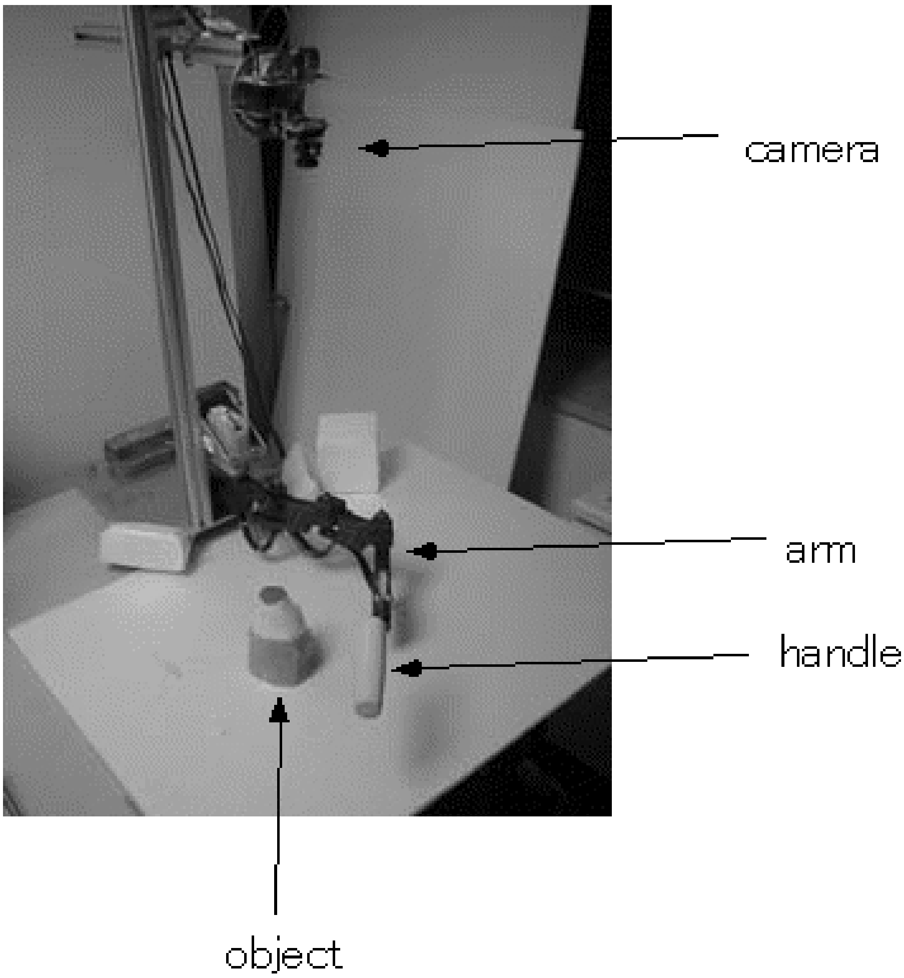

Figure 6: The arm robot with a vision system.

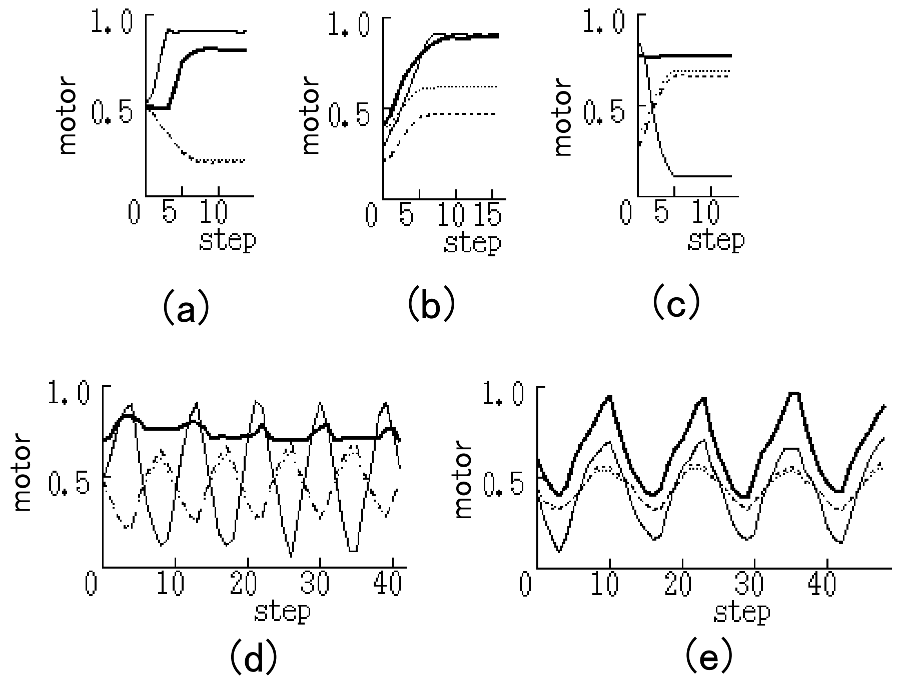

Figure 7: 4 DOF trajectories of 3 different end-point movements in (a), (b) and (c) and those of 2 different cyclic movements used in training.

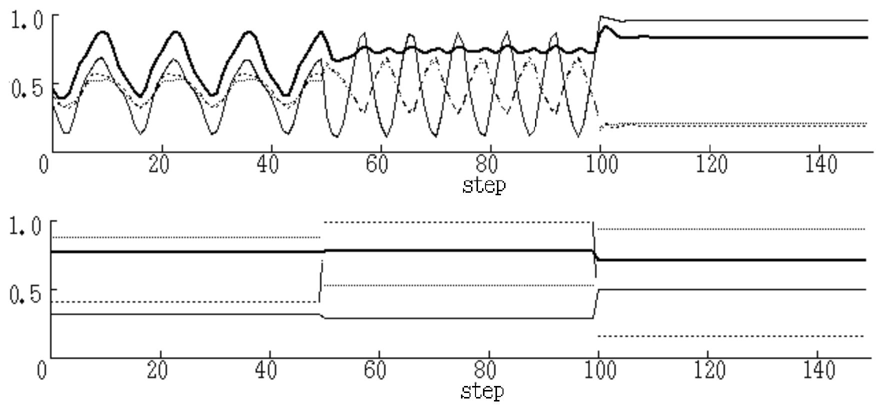

Figure 8: The results of generating two oscillatory movements followed by one end-point movement. The change over time of the motor outputs and the parametric biases are shown in the top and bottom rows, respectively. Time steps are shown in the abscissa.

to end-point and cyclic movements, can be learned simultaneously in a single RNN by changing the PB vectors.

In addition to the regeneration experiments for learned movement patterns, we examined how the movement patterns are modulated when the PB vectors are changed from the ones determined in the learning phase. Figure 9 shows successive modulations of movement patterns as one value of the PB vector is varied from 0.0 to 1.0. Observe that the movement patterns can be modulated significantly even with small changes of the parametric bias, although they are less sensitive to change in different ranges of parametric bias.

In order to clarify the mapping structure between the PB vectors and the resultant movement patterns, phase analyses of the PB vectors were conducted. Figure 10 shows how amplitude and period of one motor output in the generated movement patterns were modulated upon changing two values of the PB vector (the other two values were fixed). In Figure 10 (a), the degree of tile whiteness is directly proportional to the amplitudes of the movement patterns. The black tiles denote the regions of the end-point movement. The degree of tile whiteness is directly proportional to the period in Figure 10 (b). Again, the black tiles denote the regions of the end-point movement. When aperiodic movement patterns are generated, their amplitudes are measured by the difference between the maximum and minimum values in the sampling period. Their periods are regarded as infinite. These two plots show that the PB space is partitioned into regions of fixed-point dynamics, corresponding to end-point movements, and regions of limit cycling dynamics with various periods and amplitudes, corresponding to cycling movement patterns. An important observation is that the characteristic landscape is quite rugged in the region of the cyclic movement patterns. However, further analysis showed that the characteristics in the region of the end-point movement patterns are different. Figure 11 shows the variations of the end-point positions reached in the region of the fixed point dynamics in the 2 dimensional PB space. The end-point positions, in terms of the 1st and the 2nd joint angles of the arm, are represented by graded tile colors. It is observed that the end-point position angles fluctuate rather smoothly in the PB space. This observation suggests that the mapping between the parametric bias and the generated behaviors is quite nonlinear, in that the mappings in some regions fluctuate greatly while others are relatively smooth.

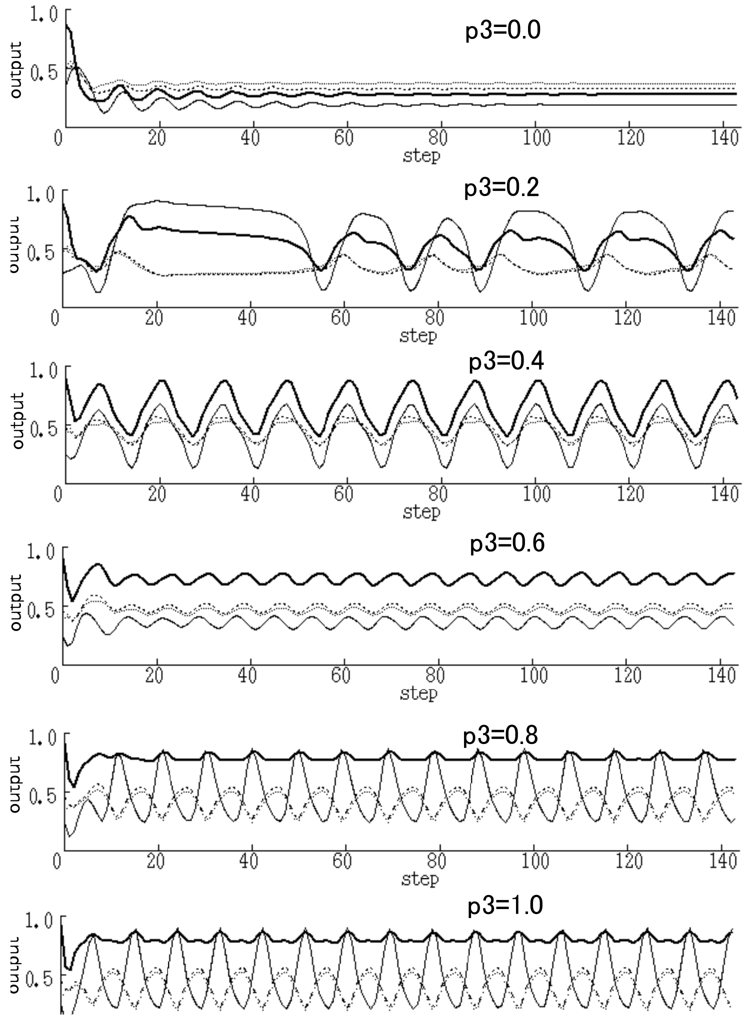

Figure 9: 6 motor activity patterns are plotted with a PB value incrementally increased from top to bottom. Ordinate: Motor Output; Abscissa: Time Step.

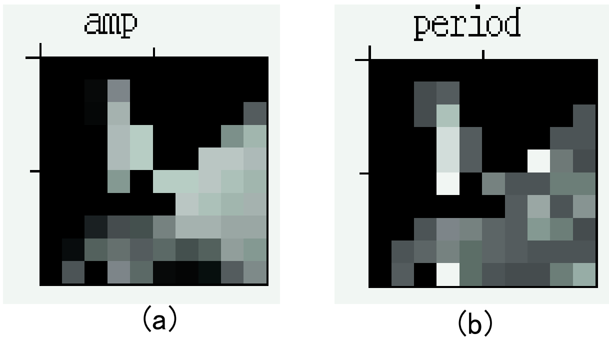

Figure 10: The phase plots for (a) the amplitude and (b) the period for one of the motor outputs using 2 values of the parametric biases.

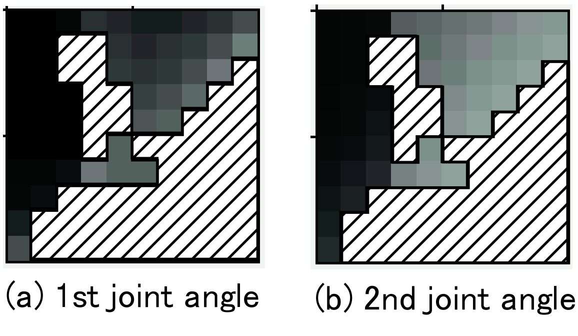

Figure 11: The phase plots for the end-point position in 2 dimensional PB space represented in terms of the first joint angle (a) and the second joint angle (b).

6 Linguistic-behavior binding

Language learning has been conducted in various contexts using RNNs, where it has been shown that certain linguistic structures can be captured from examples. Elman (1990) demonstrated that similarity in the semantics of words can be represented internally through prediction learning of word sequences using normal sentences. Pollack (1991) analyzed attractor structures self-organized in RNNs in learning some classes of artificial grammars. Miikkulainen (1999) showed that semantic slots for learned sentences can be automatically assigned by using error backpropagation to the input space, which is similar to the current scheme of determining the PB vector. The goal of this section is to learn not only word sequences, but to learn them as associated with sensory-motor sequences of the corresponding behaviors.

We demonstrate that linguistic and behavioral processes can be bound using the RNNPB scheme. Our study has been inspired by Arbib (2002)'s hypothesis that the abilities of mirror neurons for conceptualizing objects manipulation behaviors might lead to the origin of language, initially consisting of related verbs and objects. After the binding learning, a mobile robot becomes able to understand meanings of given, simple word sequences consisting of verbs and objects and then to generate corresponding object related behaviors. The meaning of a given sentence is recognized by means of the recognition mechanism of the RNNPB in the linguistic module. The determined PB vectors are passed to the RNNPB in the behavior module, by which the corresponding behaviors are generated. Note, however, that the learning in this scheme is not merely generating a mapping from each sentence to a behavior pattern. Instead, we investigate how meaningful structures can be self-organized in the mutual, interactive learning between the linguistic and behavior modules. We will discuss how such structures can enhance learning generalization, and also account for compositionality, which is especially required in the linguistic processes.

6.1 Modeling and task setting

The mobile robot utilized in the experiment has a vision camera, two motor wheels, a 1 DOF arm, and torque sensors in the arm and wheels. Figure 12 (a) illustrates the RNNPB scheme used in the co-learning of the word sequences and their corresponding behavior patterns. The linguistic module on the left-hand side receives word sequences, beginning with a "start symbol" for each sequence. Each word is locally encoded in a corresponding unit in the inputs and the outputs. There are 10 output units, 6 PB units, 50 hidden units, and 4 context units in this module. The behavior module on

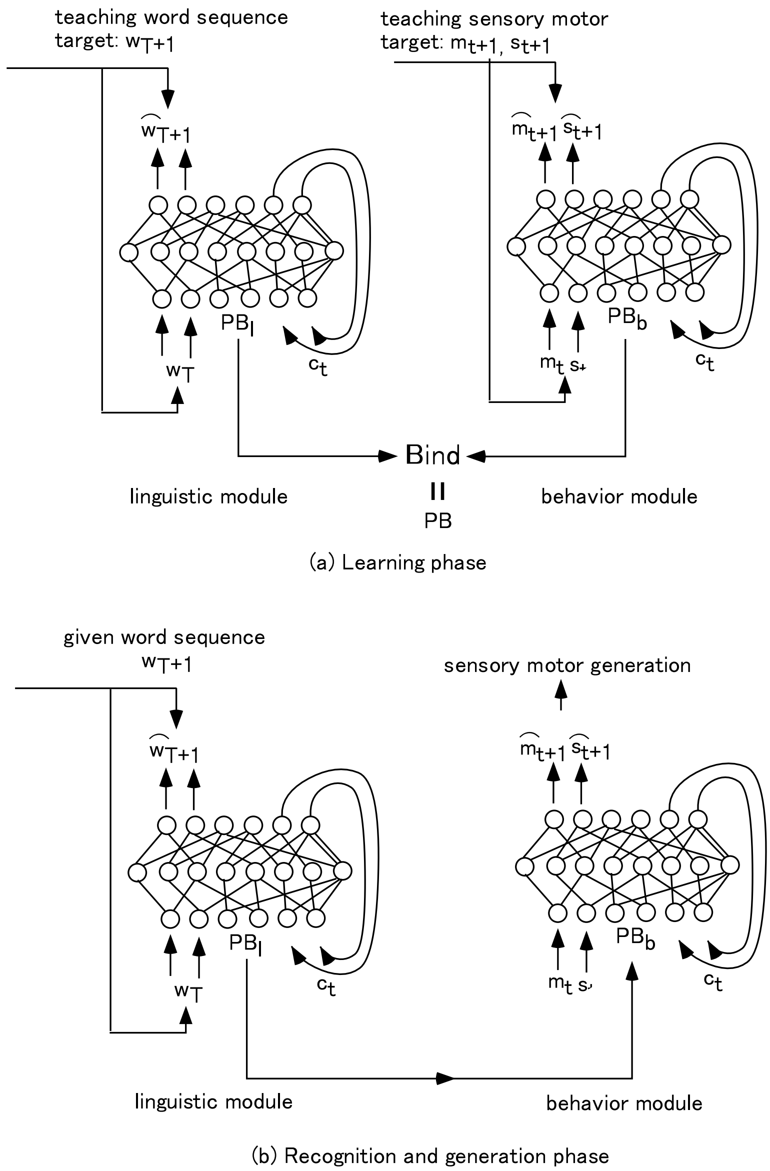

Figure 12: (a) Model for co-learning of word sequences and corresponding behaviors, (b) model for recognizing word sequences and generating corresponding behaviors.

the right-hand side receives sensory-motor sequences consisting of 3 motor values (for 2 wheels and the arm) and 23 sensory values (2 torque values and 21 values encoding the visual image). It has 26 input/output units, 6 PB units, 70 hidden units and 4 context units. During co-learning, word sequences are bound to the corresponding behavior sequences. More specifically, PB_l in the linguistic module and PB_b in the behavior module are simultaneously updated, under the constraint that the difference between these two vectors be minimized for each bound sequence. In the ideal situation, PB_l and PB_b become equal at the end of co-learning for each sequence. Note that the time steps in the word and sensory-motor sequences are not synchronized. The word sequence contains up to 3 word steps, including the starting symbol, and the sensory-motor sequence contains up to about 50 sensory-motor steps. The learning is conducted as off-line by using all pairs of the training sequences once stored in a computer. In this off-line learning process, the PB vector for each pair is updated every after the BPTT computation is conducted for the pair of sequences. The synaptic weights are updated after all pairs are swept for the BPTT computation.

Figure 12 (b) illustrates the RNNPB scheme utilized in the recognition and generation phases. The PB_l in the linguistic module is determined by recognizing a given word sequence. Its vector is set to PB_b in the behavior module for generating the corresponding behavior.

The robot experiment is conducted in the environment shown in Figure 13, where red, blue, and green objects are located in the left, center, and right positions respectively in front of a white rear wall. The robot learns to "POINT" with its arm, "PUSH" with its body, and "HIT" with its arm these three objects. Each sentence consists of two words, a verb followed by a noun. The verbs used are point, push, hit, and the nouns are red, blue, green, left, center, right. There can be 9 different combinations of behavior categories and 18 different sentences in this setting. Note that "red", "blue" and "green" turn out to be equivalent to "left", "center" and "right", respectively, in this task context.

In order to investigate the generalization capability, especially in the linguistic learning, only 14 sentences out of 18 possible sentences are trained. All of the 9 behavior categories are trained for the behavior module in the co-learning. This means that 14 sentences are bound to their corresponding 7 behavior categories in the co-learning process, while the 2 remaining behavior categories are learned without any binding constraints in determining their PB_b vector.

For each sentence, its corresponding teaching sensory-motor sequence is sampled five times by manually guiding the robot along the desired trajectory. The object

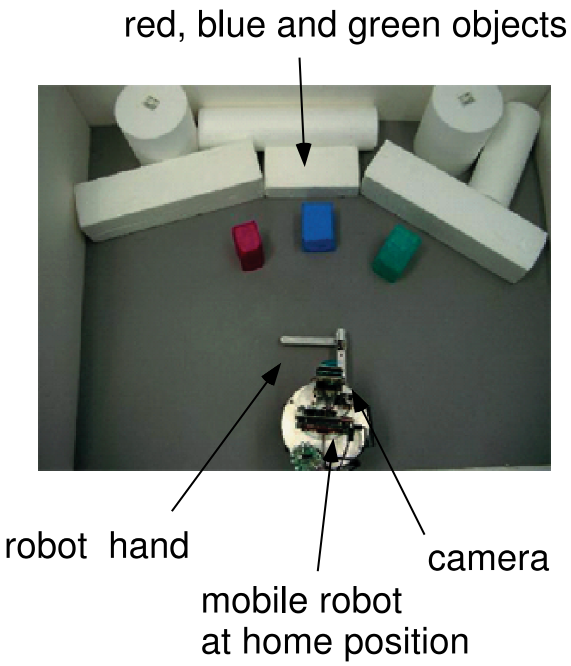

Figure 13: The task environment consists of red, blue and green objects placed in left, center, and right positions, respectively. The mobile robot is at the starting position.

positions and starting position of the robot are perturbed within 20 percent of the robot travel distance for each sampling, in order to make each sensory-motor sequence slightly different. This was necessary to make the robot generate the trained behaviors robustly. In summary, 70 (14 x 5) pairs of linguistic and sensory-motor sequences are learned and bound to each other. Further, 20 (4 x 5) sensory-motor sequences are learned without binding. The learning is iterated for 50,000 steps. The mean square errors converged to 0.0091 and 0.025 for the linguistic and the behavior modules, respectively.

6.2 Results and analysis

Recognition and generation tests were conducted after learning was completed. The appropriate corresponding behaviors were generated for all 18 word sequences, including the 4 untrained ones. In order to analyze the internal structures self-organized in the co-learning process, a phase space analysis was conducted for PB_l and PB_b. In this analysis, the original 6-dimensional PB space is projected onto the 2-dimensional surface determined by principal components analysis. In Figure 14 (a) the PB_l vectors, corresponding to all possible 18 word sequences, are plotted in the 2-dimensional space. The PB_l vector is inversely computed during the recognition of each word sequence in the linguistic module. The PB_l vectors for 4 unlearned word sequences are surrounded by dashed circles. Figure 14 (b) shows the PB_b vectors that are determined for 90 behavior sequences in the co-learning phase. Figure 14 (c) shows the averaged PB_b vector for each of 9 behavior categories.

There are some interesting findings in these figures. First in Figure 14 (a), two congruent sub-structures can be observed among the PB points corresponding to word sequences. There are 6 word sequences, each of which has the same verb followed by one of 6 nouns. All 3 of the hexagons, made up of the 6 PB points for each verb, seem to be congruent. Similarly, 6 congruent triangles can be seen for the 3 verbs preceded by the same noun. This doubly congruent structure is crucial for representing the compositionality hidden in the learned sentences i.e.– each verb can be followed by one noun in the same noun set. The combinatorial relationship between the verbs and the nouns is well represented in the multiplication of these two congruent structures. An amazing fact is that this structure was self-organized without using all possible combinations of word sequences during learning. However, 4 PB points, corresponding to unlearned word sequences, are actually found to come to the right positions in the structure when they are inversely computed in the recognition processes (thus correct behaviors can be successfully generated for them). This sort of generalization becomes possible because each word sequence is learned not as an independent instance, but rather in the form of relational structures among others, which is the compositionality of nouns and verbs in the current case.

Second, a cluster structure can be seen in the PB_b vectors in the behavior module, as shown in Figure 14 (b). Although there are certain distributions in each cluster due to the perturbations in the sensory-motor sequences in the learning set, the layout of the averaged center of those clusters seems to have the same congruent structures as the linguistic module, as shown in Figure 14 (c). It is interesting to note that this sort of congruent structure cannot self-organize when the behavior module is trained without binding with the linguistic module (Sugita & Tani, 2003). The linguistic structure affects the behavior module, allowing generation of the observed congruent structure. On the other hand, the behavior constraints can also affect the structure self-organized in the linguistic module. In Figure 14 (a), the PB points for pairs of sentences ending with “red” and “left”, “blue” and “center”, and “green” and “right”, are quite close in the space. This is due to the fact that those pairs of nouns have the same meaning in the behavioral context in the current task. It is also noted that the robot achieved each

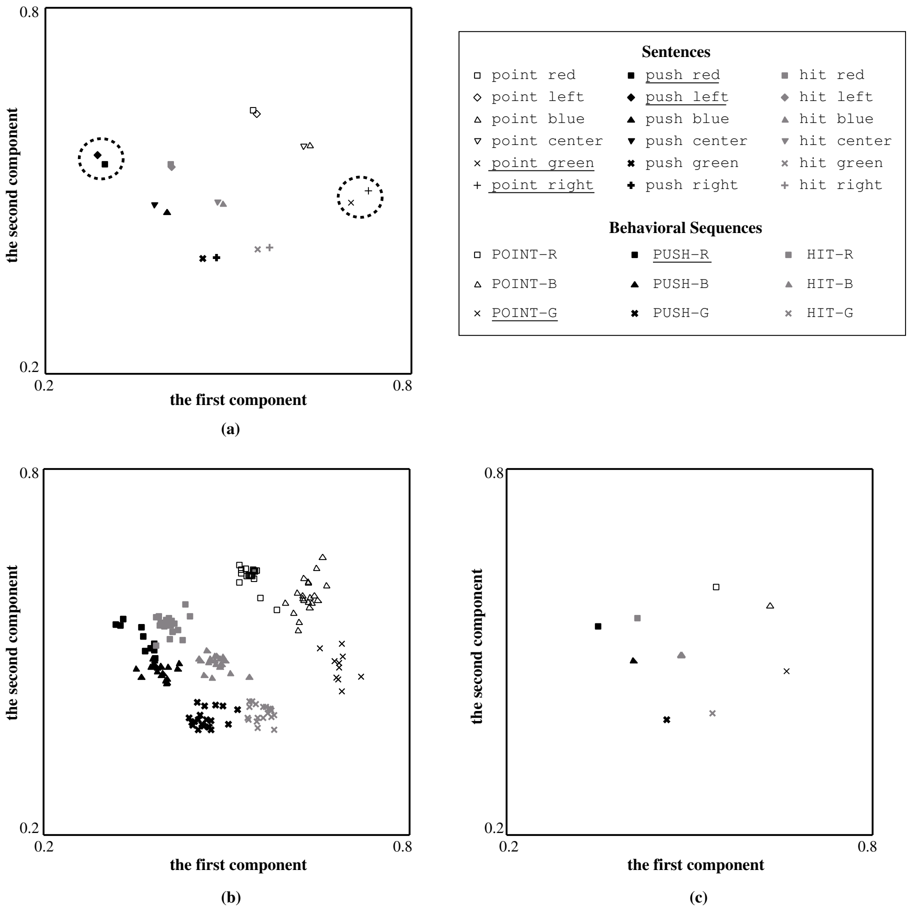

Figure 14: In each plot, the PB vectors for recognized sentences in the bound linguistic module (a), the PB vectors for training behavioral sequences in the bound behavioral module (b), and the averaged PB vectors of (b) over each behavioral category (c) are plotted. All the plots are projections of the PB spaces onto the same surface determined by the PCA method.

goal-directed behavior quite robustly against perturbations. For example, even when the object was moved slightly toward the left or right during the "PUSH" behavior, the robot could follow the directional changes of the target as long as the target was within its vision sight. This robustness is achieved because the training of each behavior, using bundles of sensory-motor sequences, results in generalization in the acquired sensory-motor mapping. (Note that the same robustness could not be achieved when fewer sensory-motor sequences were utilized in the training.)

Based on these observations, one may conclude that certain generalizations are achieved in recognizing sentences and generating behaviors by self-organizing adequate structures in the PB mapping, utilizing both linguistic and behavioral constraints.

7 Discussion and summary

The current paper reviewed the RNNPB, which can learn multiple behavior schemata distributively encoded in a single network. The scheme is characterized by the PB vector, which plays essential roles both in generating and recognizing patterns as a mirror system by self-organizing adequate structures internally. The model was implemented in three different robot platforms. Imitative interactions, learning to generate different types of dynamic movement patterns, and linguistic-behavior binding were demonstrated. In the experiment on imitative interaction with the humanoid robot, it was shown that multiple cyclic movement patterns can be learned as limit cycling attractors with different PB values self-determined. Each of learned patterns was robustly regenerated by means of the entrainment by the user's hand movements. In the experiment with the arm robot, it was shown that limit cycling and fixed point attractor dynamics can be simultaneously embedded in the PB phase space of a single network. Our dynamical systems analysis clarified the nonlinear characteristics of the PB mapping. In the last experiment of linguistic-behavior binding, it was shown that the robot becomes able to generate corresponding goal-directed behaviors by recognizing given, two-word sequences through supervised learning processes. It was shown that a compositional structure of combining verbs and objcets, as related to the object manipulation behaviors, were self-organized in the PB mapping. This resulted in the generalization by which unlearned word sequences could be recognized by analogy with learned ones.

The hallmark of the current study was explaining how internal memory structures self-organized, and how such structures could account for the generalization and behavioral diversity observed in each experiment. The proposed scheme differs significantly from the localist scheme in this aspect. In the localist scheme, no structures exist for memory organization since each behavioral schema is memorized as an independent template in a corresponding local module. On the other hand, in the proposed distributed representation scheme, learning is considered as not just memorizing each template of behavior patterns, but as reconstructing them by extracting the structural relationships among them. If there are tractable relationships among a set of learning patterns, those relationships should appear in the memory structures. Ito and Tani (2003a) have shown such an example in simulations of learning a set of sinusoidal patterns, each of which has a different amplitude and frequency. After learning it was found that the same shapes of sinusoidal patterns were regenerated by modulating their amplitudes and frequencies in 2-dimensional PB space, as when they were generated by interpolating the learning set. Similar observations were made in the fixed point dynamics in the PB phase space, as shown in Figure 10, where the end-point arm configurations change smoothly in the PB phase space. This generalization was made since all training trajectories of end-point behaviors share similar profiles but have different end-point configurations. In the linguistic-behavior binding experiment, it was observed that the relationships among verb and noun compositions were captured by self-organizing congruent structures in the PB space. On the other hand, fluctuations in the PB mapping are generated when no structural relationship can be found among learned patterns. Such an example was seen in the limit cycling dynamics region of Figure 10, where even small changes in the PB vector could induce sudden bifurcations in pattern generation. Note that two cyclic movement patterns in the learning set have no tractable relations in their trajectory profiles. On the other hand, the diversity in pattern generation was dramatically enhanced in the fluctuating PB space.

Haruno, Wolpert, and Kawato (2001) proposed that an explosion in the number of local modules needed for arbitrary movements could be avoided through a linearly weighted combination of a given set of modular outputs. However, one question with their model is how generalization can be achieved simply through linear interpolation among the arbitrarily obtained modules. It is assumed that certain kernel modules have to be self-organized through their mutually interactive computations for the purpose of attaining the generalized internal representation.

The scheme presented here may provide an abstract account for the requirement of memory consolidation in the organization of long-term memory (Squire, Cohen, & Nadel, 1984) in cortex, as opposed to short-term memory in the hippocampus. It is known that memory consolidation processes take relatively long time from days to years, which is assumed to be necessary for organizing generalized structures in the cortical distributed representation while interleaving various behavioral experiences. Behavioral experiences, initially stored independently in short-term memory, begin to interfere with each other, either cooperatively or destructively, as the relational structures among them are gradually shaped in the cortex during consolidation.

We assume that memory consolidation plays important roles in generating diversity of ideas and behaviors in humans. The hypothesis derived from our experimental results is that conflicts among memory episodes may cause fluctuations in the self-organized memory structure, where diverse false memories are generated. Although this hypothesis has little empirical support so far, its characteristics can be applied to various adaptive entertainment agents, including humanoid robots. Entertainment agents should exhibit diverse behavior in user interactions in order to avoid boring the user.

In the imitative interaction experiments, the humanoid robot often generated various emergent behaviors when the users attempted to demonstrate novel hand movement patterns in front of the robot (Ito & Tani, 2003b) [1]. Although one may be aware that this behavior is simply due to the nonlinear dynamic characteristics of the RNNPB as described in this article, it is difficult to avoid feeling as if "live cognition" emerges from the interactions with the robot. Such emergent features are indispensable for producing future humanoid robots capable of achieving close interactions with humans.

Nevertheless, it is also true that local representation schemes have their advantages. They have fewer memory interference problems (McCloskey & Cohen, 1989). Such a characteristic is advantageous when the system is required to learn in a dynamic environment (Wang & Yuwano, 1996). One important future research direction is to explore an intermediate representation scheme between the two extremes of distributed and local representations. The degree of distribution in the representation might be controlled by modulating the sparseness of activated neurons in the network. If the activations become more sparse, the overlap of activated neurons among learned patterns becomes smaller, possibly reducing interference between them. The degree of distribution should be determined in the trade off between generalization and fast learning capabilities. Such learning schemes should be investigated in future studies.

An important issue that has not been addressed in the current paper is how hierarchical organization with level structures can be organized in the behavior learning context. When the agent attempts to generate complex motor behaviors, it is reasonable to assume that an abstract event sequence is generated in a higher level and

[1] It was observed that the diversity of behavioral responses becomes larger as the number of trained movement patterns is increased.

that its detailed motor program is generated in a lower level. The question is how this sort of abstraction of information in the processes of behavior generation and recognition can be achieved based solely on the sensory-motor experiences of the agent. The authors showed that continuous sensory-motor flow can be articulated hierarchically by self-organizing repeatedly utilized behavior primitives in the lower level. Combinations of primitives are learned in the higher level in both the localist scheme, using gated modules (Tani & Nolfi, 1998), and the distributed representation scheme, using the RNNPB (Tani, 2002, 2003). Doya, Samejima, Katagiri, and Kawato (2002) proposed multiple model-based reinforcement learning, which adaptively decomposes a task based on the predictability of the environmental dynamics using the localist modular representation. Ziemke and Thieme (2002) showed that their proposed extended sequential cascaded network can be evolved to solve delayed response tasks in robot navigation. However, all of these studies assume predetermination of explicit level structures in the network architecture in which essential parameters, such as time constant for each level, have to be carefully tuned depending on the task environment. The crucial question is whether such explicit level structures are necessary or not. If necessary, how could such hierarchical structures be self-organized through adaptation in the task environment without having much pre-programming. These questions are also left for future studies.

Another important issue which is missing in the current studies is the "goal-directedness" in generating or recognising behaviors. Although the current implementation has achieved only trajectory level repetitions of given movement patterns, its extensions to imitation through understanding others' goals as well as one's own (Tomasello, 1999) are important future research topics. It is also true that many mirror neurons are found in rather goal-directed task settings, where they seem to encode not exact movement patterns, but their abstraction or goals (Rizzolatti et al., 1996). In order to achieve goal-directed behavior generation or recognition, certain hierarchical architectures which are capable of abstracting and conceptualizing behavior patterns in multiple levels, as discussed previousely, might be required. This problem should be also open for future studies.

This article has discussed the advantages of the distributed representation scheme in building learning robots. It may be true that the authors are merely rephrasing what the PDP research group (Rumelhart, Mclelland, & PDP Research Group, 1986) discussed in the mid 1980s:

> "Distributed representations are efficient whenever there are underlying regularities which can be captured by interactions among microfeatures. By

encoding each piece of knowledge as a large set of interactions, it is possible to achieve useful properties like content-addressable memory and automatic generalization, and new concepts can be created without having to create new connections at the hardware level.(Hinton, Mclelland, & Rumelhart, 1986)".

This idea still seems worth considering today in order to understand and reconstruct behavioral cognitive systems.

References

Amit, R., & Mataric, M. (2002). Learning Movement Sequences from Demonstration. In *Proceedings of the IEEE International Conference on Development and Learning (ICDL-2002)* (pp. 203–208).

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.

Arbib, M. (2002). The mirror system, imitation, and the evolution of language. In *Imitation in animals and artefacts* (pp. 229–280). Cambridge: 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.

Bizzi, E., Acornero, N., Chapple, W., & Hogan, N. (1984). Posture control and trajectory formation during arm movements. *J. Neurosci.*, *4*, 2738–2744.

Demiris, Y., & Hayes, G. (2002). Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model. In K. Dautenhahn & C. L. Nehaniv (Eds.), *Imitation in animals and artefacts* (pp. 1448–1480). Cambridge: MIT Press.

Doya, K., Samejima, K., Katagiri, K., & Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural Computation*, *14*, 1347–1369.

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).

Elman, J. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Feldman, A. (1980). Superposition of motor programs, I. Rhythmic forearm movements in man. *Neuroscience*, *5*, 81–90.

Fujita, M., Kuroki, Y., Ishida, T., & Doi, T. (2003). A Small Humonoid Robot SDR-4X for Entertainment Applications. In *Proc. of international conference on advanced intelligent mechatronics* (pp. 435–442).

Haruno, M., Wolpert, D., & Kawato, M. (2001). MOSAIC model for sensorimotor learning and control. *Neural Computation*, *13*, 2201–2220.

Hinton, G., Mclelland, J., & Rumelhart, D. (1986). Distributed representation. In D. Rumelhart & J. Mclelland (Eds.), *Parallel distributed processing* (pp. 77–109). Cambridge, MA: MIT Press.

Ijspeert, A. J., Nakanishi, J., & Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems 17* (pp. ??–??). Cambridge, MA: MIT Press.

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).

Ito, M., & Tani, J. (2003a). *Analysis of generalization in dynamic pattern learning.* (submitted)

Ito, M., & Tani, J. (2003b). *On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system.* (Tech. Rep. No. RIKEN-BSI-BDC-TR2003-005). RIKEN, BSI.

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.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist network. *Psych. Learning Motivat.*, *24*, 109–165.

Miikkulainen, R. (1999). Text and discourse understanding: The DISCERN system. In R. Dale, H. Moisl, & H. Somers (Eds.), *A handbook of natural language processing: Techniques and applications for the processing of language as text.* New York: Marcel Dekker.

Miyake, Y. (2002). Co-creation systems. *Cognitive Processing*, *3*, 131–136.

Oztop, E., & Arbib, M. A. (2002). Schema Design and Implementation of the Grasp-Related Mirror Neuron System. *Biological Cybernetics*, *87*, 116–140.

Pellegrino, G. di, Fadiga, L., Fogassi, L., Galless, V., & Rizzolatti, G. (1992). Understanding motor events: a neurophysiological study. *Experimental Brain Research*, *91*, 176–180.

Pollack, J. (1991). The induction of dynamical recognizers. *Machine Learning*, *7*, 227–252.

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. Mclelland (Eds.), *Parallel distributed processing* (pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, D., Mclelland, J., & PDP Research Group the. (1986). Cambridge, MA: MIT Press.

Squire, L., Cohen, N., & Nadel, L. (1984). The medial temporal region and memory consolidation: A new hypothesis. In H. Weingartner & E. Parker (Eds.), *Memory consolidation* (pp. 185–210). Hillsdale, N.J.: Erlbaum.

Sugita, Y., & Tani, J. (2003). Holistic approach to compositional semantics: a connectionist model and robot experiments. In *Advances in Neural Information Processing Systems 17.* Cambridge, MA: MIT Press.

Taga, G. (1996). A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics*, *12*, 1131–1141.

Tani, J. (2002). Articulation of sensory-motor experiences by " forwarding forward model ": from robot experiments to phenomenology. In *From animals to animats 7* (pp. 171–180). Cambridge, MA: MIT Press.

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., & 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., & 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)

Tomasello, M. (1999). *The caltural origins of human cognition.* Harvard University Press.

Wang, D., & Yuwano, B. (1996). Incremental learning of complex temporal patterns. *IEEE Trans. on Neural Networks*, *7*(6), 1465–1480.

Werbos, P. (1990). A menu of designs for reinforcement learning over time. In W. Miller, R. Sutton, & P. Werbos (Eds.), *Neural networks for control* (pp. 67–95). Boston, MA: MIT Press.

Wolpert, D., & Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, *11*, 1317–1329.

Yamauchi, K., Ohta, M., & Ishii, N. (1999). A Self-supervised Learning System for Pattern Recognition by Sensory Integration. *Neural Networks*, *12*, 1347–1358.

Ziemke, T. (2000). On 'Parts' and 'Wholes' of Adaptive Behavior: Functional Modularity and Diachronic Structure in Recurrent Neural Robot Controllers. In *From animals to animats 6* (pp. 171–180). Cambridge, MA: MIT Press.

Ziemke, T., & Thieme, M. (2002). Neuromodulation of Reactive Sensorimotor Mappings as Short-Term Memory Mechanism in Delayed Response Tasks. *Adaptive Behavior*, *10*(3/4), 185-199.