Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes

Yuuya Sugita and Jun Tani

Adaptive Behavior 2005; 13; 33
DOI: 10.1177/105971230501300102

Adaptive **Behavior**

Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes

Yuuya Sugita, Jun Tani
BSI RIKEN, Japan

We present a novel connectionist model for acquiring the semantics of a simple language through the behavioral experiences of a real robot. We focus on the "compositionality" of semantics and examine how it can be generated through experiments. Our experimental results showed that the essential structures for situated semantics can self-organize themselves through dense interactions between linguistic and behavioral processes whereby a certain generalization in learning is achieved. Our analysis of the acquired dynamical structures indicates that an equivalence of compositionality appears in the combinatorial mechanics self-organized in the neuronal nonlinear dynamics. The manner in which this mechanism of compositionality, based on dynamical systems, differs from that considered in conventional linguistics and other synthetic computational models, is discussed in this paper.

Keywords embodied language · compositionality · recurrent neural network · self-organization · dynamical systems · robot

1 Introduction

Implementing language acquisition systems is an extremely difficult problem. Not only the complexity of the syntactic structure but also the diversity in the domain of meaning render this problem complicated and intractable. In particular, the manner in which linguistic meaning is represented in the system is crucial. This problem has been investigated in recent decades.

In this paper, we introduce a connectionist model that acquires the semantics of a simple finite language with respect to correspondences between sentences and the behavioral patterns of a real robot. An essential question to be answered is how compositional semantics can be acquired in the proposed connectionist model without providing any explicit representations of the meaning of a word or behavior routines a priori. By "compositionality," we refer to the fundamental human ability to understand a sentence from (1) the meanings of its constituents, and (2) the way in which these constituents are put together. It is possible for a language acquisition system that acquires compositional semantics to derive the meaning of an unknown sentence from that of known sentences. Consider the unknown sentence: "John likes birds." This could be understood by learning the following three sentences: "John likes cats," "Mary likes birds," and "Mary likes cats." That implies that the generalization of meaning can be achieved through compositional semantics.

Correspondence to: Y. Sugita, BSI RIKEN, Hirosawa 2-1, Wako-shi, Saitama 3510198, Japan.
E-mail: sugita@bdc.brain.riken.go.jp
Tel.: +81-48-462-1111 (ext.7416) , *Fax*: +81-48-467-7248.

(2005), Vol 13(1): 33–52.
[1059–7123(2005013) 13:1; 33–52; 050133]

From the point of view of compositionality, the symbolic representation of the meaning of a word is advantageous for processing the linguistic meaning of sentences. In general, AI-based models employ symbolic representation for the meaning of a word, which has a good affinity with compositionality in terms of the meanings of sentences (Thompson & Mooney, 1998). However, the symbolic approach has critical difficulties in acquiring language semantics that are based on the behavioral experiences of an agent. Associating semantic representations by symbol systems to corresponding sensory–motor profiles is not a trivial problem, as seen in the symbol grounding problem (Harnad, 1990). Even in a simulated block world, designing a symbolic semantic representation that unifies language and behavior is very complicated (Winograd, 1972). Although engineering approaches, employing handcrafted and task-specific representations, could be successful for certain application tasks, these would not explain the general cognitive mechanisms of embodied or situated language.

To solve these problems, various learning models have been proposed for acquiring the embodied semantics of language via a statistical method or a connectionist model. For example, some models learn semantics in the form of correspondences between sentences and nonlinguistic objects: i.e., visual images (Roy, 2002) or the sensory–motor patterns of a robot (Billard, 2000; Iwahashi, 2003; Steels, 2000; Sugita, 2002). However, these methods require relatively heavy manual preprogramming to realize compositional semantic representations. In the studies by Iwahashi (2003), Roy (2002) and Steels (2000), the syntactic aspects of a language are acquired through a preacquired lexicon. This implies that the acquired meanings of words (i.e., lexicon) are independent of the usages of those words in sentences (i.e., syntax). Although this separated learning approach seems to be plausible from the point of view of the requirements of compositionality, it causes inevitable difficulties in representing the grounded meaning of sentences (Winograd, 1972). A priori separation of words and syntax requires a predefined manner of combining the meanings of words to compose the meanings of sentences. In Iwahashi's model, the classifications of words are assigned prior to learning their meanings because different acquisition algorithms are required for nouns and verbs, see Siskind (2001). Roy's (2002) model does not require a priori knowledge of word classes, but requires the strong assumption that the meaning of a word can be assigned to some predefined attributes of nonlinguistic objects. This assumption is not realistic in more complex situations, wherein the meanings of words need to be extracted from nonlinguistic spatiotemporal patterns, such as learning verbs.

In contrast to using symbolic representations, some connectionist schemes have investigated the acquisition of both syntax and semantics in a co-dependent manner. It has been shown that a recurrent neural network (RNN) can acquire grammatical structures (Elman 1990; Pollack, 1991) and semantics (Miikkulainen, 1993) from examples of sentences or string sequences. A RNN also demonstrated the ability to emulate the symbolic structures of finite state machines, hidden in the interaction between a real robot and its environment, in a robot navigation task (Tani, 1996). These results suggest the possibility of self-organizing certain classes of combinatorial structures that are required in compositional semantics without using symbolic representation. In particular, Billard's (2002) model, as well as our previous model (Sugita & Tani, 2002) using RNNs, dealt with this issue. These models learn semantics as the association between sentences and action sequences, with motivations similar to those in the currently proposed model. However, these models can show only limited compositionality and they cannot deal with complex sensory–motor profiles. Instead, they only deal with macro-action sequences by employing predefined action primitives. Recently, Cangelosi (2004) employed a feedforward neural network for associating verb and noun pairs with sensory–motor sequences, using a simulated arm robot in object manipulation tasks. Although their analysis of the internal representation using a categorical perception scheme (Cangelosi, 2004) had certain implications for achieving possible situated semantics, the model is limited in its scheme for encoding word sequences into spatial input patterns rather than temporal patterns.

In this paper, a novel scheme for embodied language using RNNs is introduced, where no symbolic or structural representations are provided a priori, and the individual treatment of lexicon, syntax, and semantics are avoided. In this scheme, learning is achieved by means of mutual interactions between the linguistic process, dealing with given word sequences, and the behavioral process, dealing with the experienced sensory–motor flow. The hallmark of this approach is the self-organization of the necessary structures for embodied language as the result of such interactions. The pro-

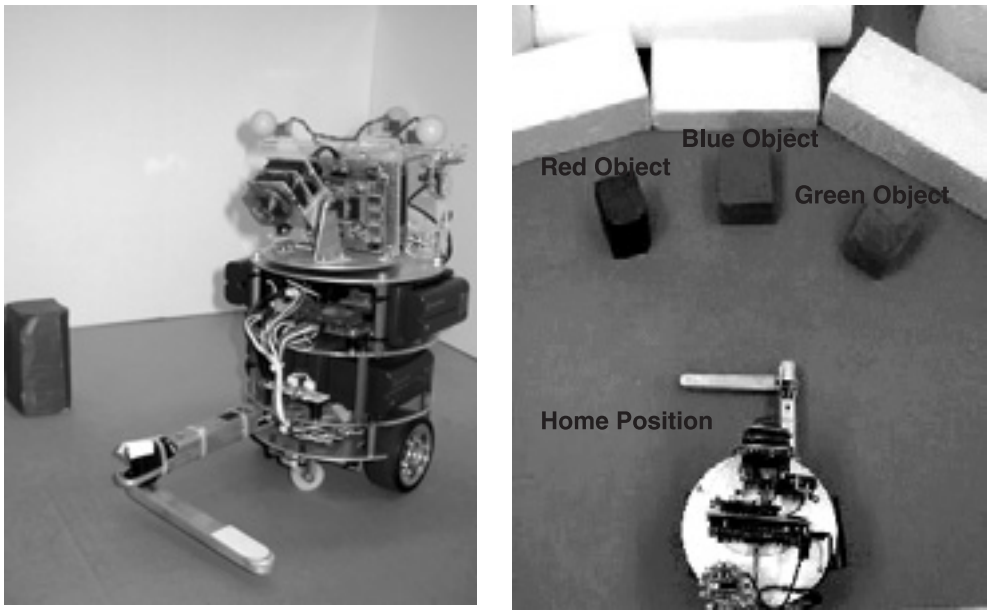

Figure 1 A mobile robot (a) starts from a fixed position in an environment and performs each behavior of pointing at, pushing, or hitting either a red, blue, or green object (b).

posed scheme is examined by conducting behavior–language acquisition experiments using a real mobile robot. We analyze the types of structures that should be self-organized in order to acquire situated semantics that can exhibit generalization in learning. Our discussion of the results leads to alternative interpretations of the symbol grounding problem and compositionality based on the dynamical systems perspective (Schoner & Kelso, 1988; Beer, 1995; Gelder, 1998).

2 Task Design

The experiments are conducted using a real mobile robot with an arm and various sensors, including a vision system. The robot learns a set of behaviors by acting on some objects associated with two-word sentences consisting of a verb followed by a noun. Although our experimental design is limited, it suggests an essential mechanism for acquiring situated compositional semantics through the minimal combinatorial structure of this finite language (Evans, 1981). Acting on objects is essential to our task, as inspired by Arbib's (2002) view of the evolutionary origin of language based on his mirror neuron hypothesis. The hypothesis is that the mirror neurons' ability to "conceptualize" object manipulation behaviors might lead to the origin of language, that initially consists of corresponding verbs and object nouns. In particular, our connectionist model is regarded as a mirror system (Rizzolatti, Fadiga, Galles, & Fogassi, 1996) as will be described later.

The robot experiments consist of a training phase and a testing phase. In the training phase, our neural network model learns possible associations between sentences and corresponding behavioral sensory–motor sequences of a robot in a supervised manner. In the testing phase, the network's ability to generate the corresponding correct behavior by recognizing the given sentences is examined. We also evaluate the system's generalization ability by examining whether appropriate behaviors can be generated from unlearned sentences based on learned sentences.

A mobile robot was built for this experiment in our laboratory. The mobile robot is equipped with three actuators for two wheels and a rotational joint on the arm, a colored vision sensor, and three torque sensors on both the wheels and the arm (Figure 1a). The robot operates in an environment where three colored objects (red, blue, and green) are placed on the floor (Figure 1b). The positions of these objects can be varied as long as the robot sees the red object (R) in the left-hand side of its field of view, the blue object in the middle (B), and

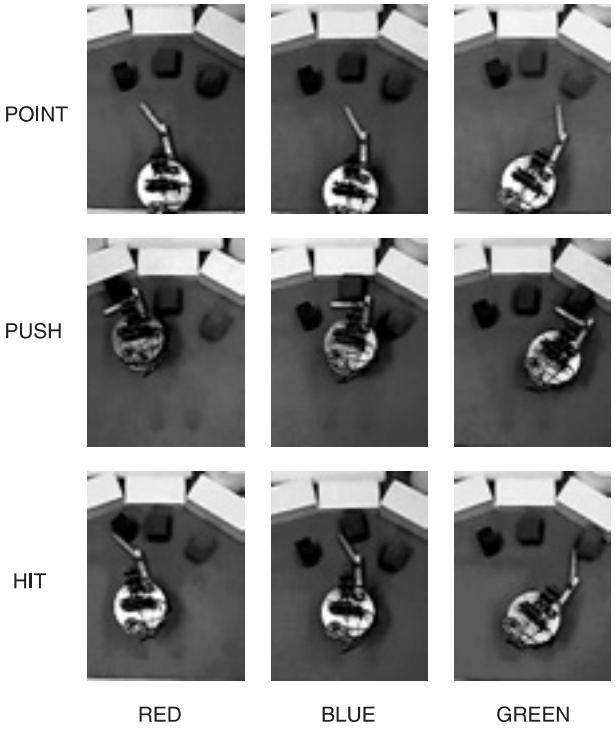

Figure 2 The robot learns nine categories of behavioral patterns, which are denoted as POINT-R, POINT-B, POINT-G, PUSH-R, PUSH-B, PUSH-G, HIT-R, HIT-B, and HIT-G from the top-left to the bottom-right.

the green object (G) on the right-hand side at the start of every trial of behavioral sequences. We adopt a fixed arrangement of the objects for simplifying behavioral learning, particularly to reduce the required training and computation time for learning. Despite this limitation, our experimental setting still preserves enough complexity to observe the minimal combinatorial properties in associations between sentences and behavioral patterns. The color information is still important for robust behavior generation because the narrow sight of the robot (about 60 degrees) ensures that at least one of the objects is out of sight, except near the starting position.

There are nine behavioral categories that the robot is expected to learn: pointing at, pushing, and hitting each of three objects located on the floor. These categories are denoted as POINT-R, POINT-B, POINT-G, PUSH-R, PUSH-B, PUSH-G, HIT-R, HIT-B, and HIT-G (Figure 2). The robot learns these behavioral categories through supervised learning. In order to gather data for supervised training, the sensory–motor sequences corresponding to each of these behavioral categories are generated through the manual-steering of the robot using a remote controller. It should be noted that no categorical cues are provided to the robot in learning; instead the categorical structures should be self-organized only through experiencing various sensory–motor sequences and associated sentences provided during training.

The robot learns sentences that consist of one of the three verbs: point, push, and hit followed by one of the six nouns: red, left, blue, center, green, and right. We note that the labels are introduced for the ease of our understanding. From the robot's point of view, they are merely nine unknown lexical symbols and they should be labeled as w_1, w_2, …, w_9. Therefore, the robot cannot get any information regarding the meaning of the word from the word itself. The meanings of these 18 possible sentences are given in terms of fixed correspondences with the nine behavioral categories (Figure 3). For example, "point red" and "point left" correspond to POINT-R, "point blue" and "point center" to POINT-B, and so on.

In these correspondences, because of the fixed arrangement of the objects in the environment, "left", "center", and "right" have exactly the same meaning as "red", "blue", and "green", respectively. These synonyms are introduced to observe how the behavioral similarity affects the acquired linguistic semantic struc-

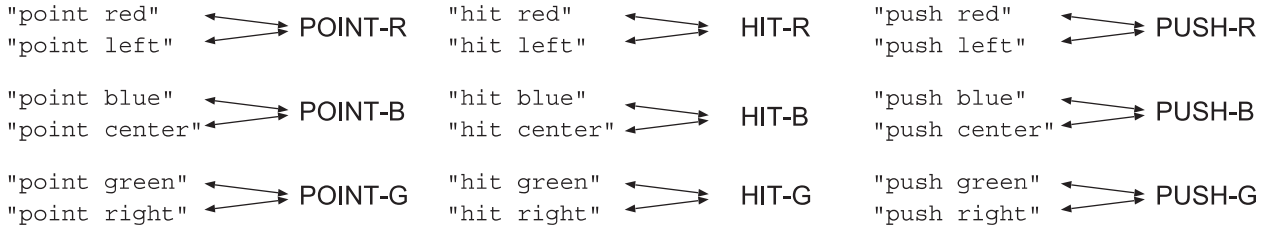

Figure 3 The correspondences between sentences and behavioral categories. For each behavioral category, there are two corresponding sentences.

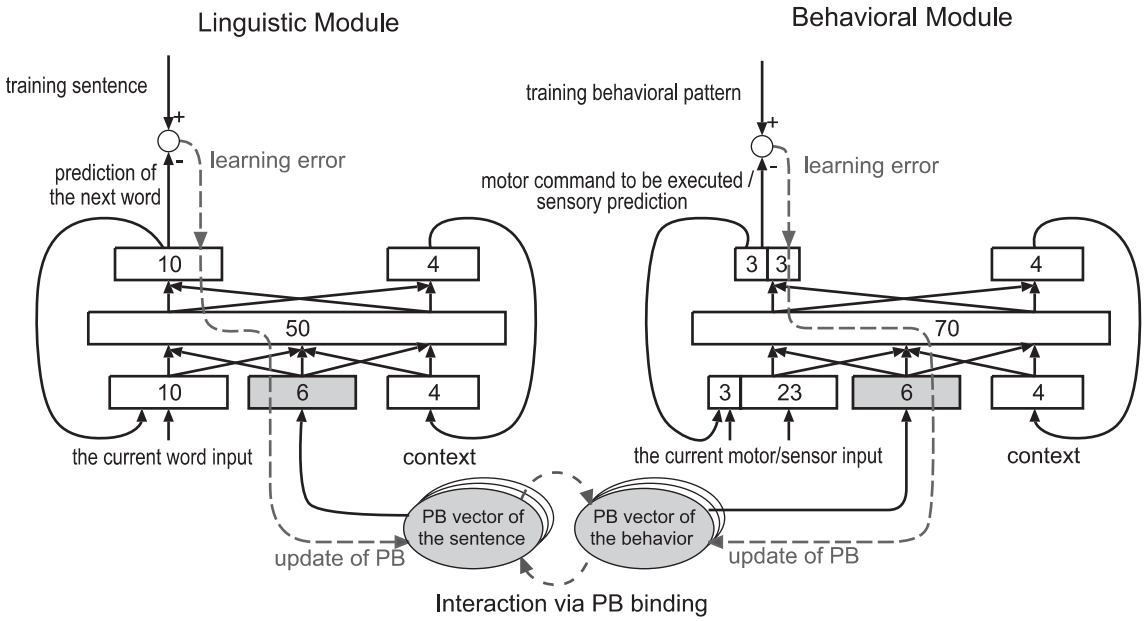

Figure 4 Our model is composed of two RNNPBs, one for a linguistic module and the other for a behavioral module. Each square represents a set of nodes, and the associated digits denote the number of nodes used in the task. The solid lines denote the information flow in the forward computation and dotted lines denote the flow of the learning error back-propagated to the PB nodes. In the learning process, the PB nodes are iteratively computed through interactions between both the modules.

ture. Moreover, it should be noted that any isolated concepts concerning the objects are not presented to the robot. The objects are taught as targets of actions in which their information, such as color, shape, and weight, is inseparably embedded in the sensory–motor flow associated with behaviors. The robot should understand the meanings of the objects in terms of the possible actions carried out upon them, such as pointing at, approaching, pushing, and hitting.

3 Proposed Model

3.1 General Scheme

We propose a connectionist model which acquires the embodied semantics of a simple language on the task design outlined in the previous section. First, this section describes the basic ideas of our proposed connectionist model. The details of each computational algorithm, as well as the module architectures employed in the proposed scheme, will be described in the subsequent sections.

Our model is composed of two loosely coupled connectionist networks referred to as the recurrent neural network with parametric bias nodes (RNNPB) (Tani, 2003; Tani & Ito, 2003; Ito & Tani, 2004a), one for the linguistic module and the other for the behavioral module as shown in Figure 4. The RNNPB is based on the Jordan-type recurrent neural network (RNN) (Jordan & Rumelhart, 1992) but is specialized with a mechanism for modulating its own dynamic function using the so-called parametric bias (PB) nodes allocated in the input layer (Figure 5). The RNNPB can both generate and recognize sequences, and therefore these functions of RNNPB can be interpreted as an abstract modeling of mirror systems (Rizzolatti et al., 1996). In the current setting, the RNNPB generates word sequences or sensory–motor sequences in terms of forward models (Kawato, Furukawa, & Suzuki, 1987) predicting the next state from the current state. When the PB vectors are set to different values, the RNNPB exhibits different forward dynamics, i.e., generating different output sequences, whose mechanism is equivalent to the parametric bifurcation, which is well known in nonlinear dynamical systems theory (Wiggins, 1990). A set of different target output sequences are embedded in an RNNPB by self-organizing an adequate mapping between the PB vector and the output sequences in the learning process. All the training sequences are learned

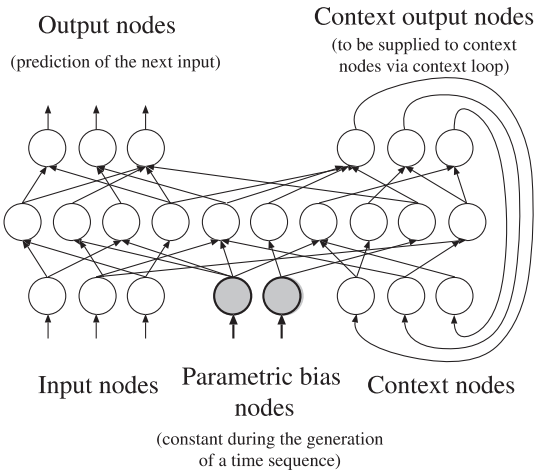

Figure 5 The RNNPB is a variant of the conventional RNN that has PB nodes in its input layer. The values of the PB nodes are set constant throughout each generation of a time sequence, and function as the modulator of the dynamics of the network. The other nodes function in the same manner as the conventional RNN.

simultaneously through batch training. It should be noted that each PB vector value of a target sequence is self-determined rather than assigned by an experimenter. Once such a mapping is generated, the corresponding PB vector values for a given output sequence can be computed inversely by minimizing the prediction error in the output sequence. This can be regarded as the recognition of given sequences in terms of the associated PB vector. This PB computation scheme will later be described in greater detail.

The linguistic module learns to recognize a set of sentences, which is represented as sequences of words, while the behavioral module learns a set of sensory–motor sequences, as shown in Figure 4. The association between a sentence and its corresponding behavior is achieved by means of self-organization of both modules through their mutual interactions. The learning process utilizes the error signal back-propagated (Rumelhart, Hinton, & Williams, 1986) to the PB nodes, as will be described in detail later. We implement the constraint that the PB vector for corresponding behavioral patterns and sentences should converge to approximately the same value in both modules. In the testing phase, a sentence is first passed to the linguistic module and then the corresponding PB vector value is inversely computed. The obtained PB value is then set in the behavioral module and the corresponding behavioral pattern is generated by the robot. Although the opposite process, i.e., recognizing behavioral patterns and then generating the corresponding sentences, is actually possible, it is beyond the scope of the current paper.

3.2 Algorithmic Description of RNNPB

We review the algorithmic description of the RNNPB (Tani, 2003; Tani & Ito, 2003) before describing the details of each module. The description focuses on how the PB nodes function in learning and recognizing a sequence. In this experiment, every node of the RNNPB yields a real-numbered value between 0.0 to 1.0.

The RNNPB learns multiple sequences in a supervised manner through two different mechanisms: connection weight modification and PB vector modification. The connection weights are common to all sequences in the training set, while the PB vector is different for each sequence. Both are simultaneously computed using the conventional back-propagation through time (BPTT) algorithm (Jordan & Rumelhart, 1992; Rumelhart et al., 1986) to minimize the value of the total learning error function E over all the training sequences $\mathbf{q}_0,\ldots,\mathbf{q}_{s-1}$ defined as follows:

$$E(W, p_0, \ldots, p_{s-1}) = \sum_{k=0}^{s-1} E_k(W, p_k) \tag{1}$$

and

$$E_k(W, p_k) = \sum_{t=0}^{l_k-1} \left\| r_k(t) - o_k(W, p_k, t) \right\|^2 \tag{2}$$

where E_k is the learning error function of the training sequence $\mathbf{q}_k$, W is a set of all the connection weight values of the network, p_k is a PB vector corresponding to a specific training sequence $\mathbf{q}_k$, s is the number of training sequences, l_k is the length of the training sequence $\mathbf{q}_k$, and $r_{kn}(t)$ and $o_k(W, p_k, t)$ are the target and output vectors in the training sequence $\mathbf{q}_k$ at a time step t, respectively. It should be noted that the output values of the network depend on both the connection weight values and the PB vectors.

The connection weight values are iteratively computed to minimize the total learning error E as in the conventional RNN. A connection weight value $w_{nm} \in W$ from node m to node n is initialized randomly and then iteratively updated at every training iteration T as follows:

$$\delta^2 w_{nm}^{(T)} = -\frac{\partial E(W^{(T)}, p_0^{(T)}, \ldots, p_{s-1}^{(T)})}{\partial w_{nm}} \quad (3)$$

$$\delta w_{nm}^{(T)} = \eta_w \cdot \delta w_{nm}^{(T-1)} + \epsilon_w(1-\eta_w) \cdot \delta^2 w_{nm}^{(T)} \quad (4)$$

$$w_{nm}^{(T)} = w_{nm}^{(T-1)} + \delta w_{nm}^{(T)}. \quad (5)$$

The delta error (Rumelhart et al., 1986) $\delta^2 w_{nm}^{(T)}$, back-propagated to the connection weight from node n to node m at a training iteration T, is computed from the current connection weight values $W^{(T)}$ and the current PB vectors $p_0^{(T)}, \ldots, p_{s-1}^{(T)}$ using the BPTT algorithm. The current update $\delta w_{nm}^{(T)}$ of the current connection weight value $w_{nm}^{(T)} \in W^{(T)}$ is computed from the previous update $\delta w_{nm}^{(T-1)}$ and the delta error as described in Equation 4 where ϵ_w and η_w are positive coefficients that determine the learning rate and the time constant of the update modification, respectively.

In contrast, the PB vector p_k is computed to minimize the learning error E_k of each training sequence $\mathbf{q}_k$. Each jth element p_{kj} of a PB vector p_k is initially set to 0.5, and then it is iteratively updated at every training iteration T as follows:

$$\delta^2 p_{kj}^{(T)} = -\frac{\partial E(W^{(T)}, p_0^{(T)}, \ldots, p_{s-1}^{(T)})}{\partial p_{kj}} = -\frac{\partial E_k(W^{(T)}, p_k^{(T)})}{\partial p_{kj}} \quad \left(\because \frac{\partial E_k}{\partial p_{k'j}} = 0, \forall k' \neq k\right) \quad (6)$$

$$\delta p_{kj}^{(T)} = \eta_p \cdot \delta p_{kj}^{(T-1)} + \varepsilon_p(1-\eta_p) \cdot \delta^2 p_{kj}^{(T)} \quad (7)$$

$$p_{kj}^{(T)} = p_{kj}^{(T-1)} + \delta p_{kj}^{(T)} \quad (8)$$

where $\delta^2 p_{kj}^{(T)}$ is the delta error back-propagated to the jth PB node at a training iteration T, which is computed by using the BPTT algorithm. ϵ_p and η_p are positive coefficients that determine the learning rate and the time constant of the modification of the current update, $\delta p_{kj}^{(T)}$, of $p_{kj}^{(T)}$, the jth element of the PB vector p_k.

The recognition algorithm basically follows the same update rules for the PB vectors, shown in Equations 6–8, where it only updates the PB vector for a given sequence while the connection weight is constant. For the purpose of avoiding local minima, it is effective to introduce a Gaussian noise term that is proportional to the prediction error in Equation (8).

3.3 Linguistic Module

The linguistic RNNPB learns and recognizes the sentences. Similar to Elman's previous work employing the conventional RNN (Elman, 1990), our linguistic module is trained to predict the next words in the output nodes from the current word received in the input nodes. A set of sentences can be learned by differentiating the PB vector for each different sentence. This module has 10 input nodes, 6 PB nodes, 4 context nodes, 50 hidden nodes, and 10 prediction output nodes (see Figure 4).

The sentences are represented as sequences of words, which always start with a fixed starting symbol. The module has 10 input nodes allocated for nine words (point, push, hit, red, left, blue, cen-ter, green, and right) and one starting symbol. Each word is locally represented, such that each input node corresponds to a specific word exclusively activated with 1.0 as shown in Figure 6. Although this input representation scheme is almost similar to that of Elman's model (Elman, 1990), the internal representations of the word sequences are very different. The Elman model learns the probabilistic distribution of the next possible words while our model learns each sentence as a deterministic sequence encoded in a distinct PB vector.

3.4 Behavioral Module

The behavioral module learns the behavioral patterns in order to regenerate them. The module is trained to produce as an output a prediction of the next motor

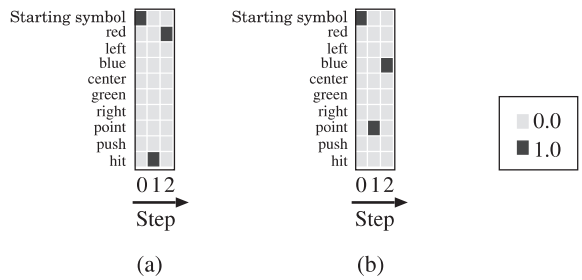

Figure 6 A sentence is represented as a sequence of words, which corresponds to each dimension of the input vector. The representation of the sentences "hit red" (a) and "point blue" (b) is presented.

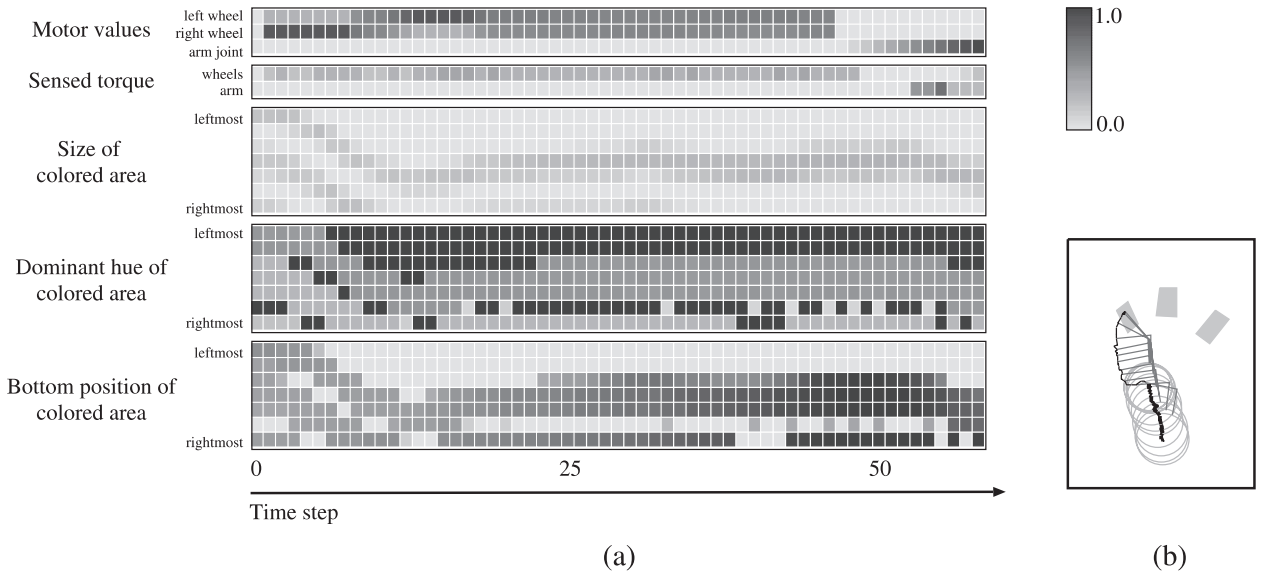

Figure 7 The sensory–motor sequence representation (a) and the corresponding robot trajectory (b) for HIT-R is shown as an example. The robot starts from the home position (step 0). As the robot turns to the red object, the green object soon disappears (step 4), and the red object is at the center of the view (step 10–25). The blue object is still to the right of the view. Subsequently, as the robot moves directly towards the red object (step 25–47), the distance between the red object and robot decreases (the size and the bottom position increases). After this, the robot stops (step 48) and HITs the red object with its arm (step 49–58).

values as well as part of the sensory inputs when it receives the current sensory and motor values as input. All the training sequences are manually prepared by hand-steering the robot in the workspace. They are then used in the off-line learning phase. A training behavioral sequence is sampled with three sensory–motor steps per second during the manual-steering of the robot. The duration of typical behavioral sequences are of approximately 5–25 s, and therefore consist of approximately 15–75 sensory–motor steps as shown in Figure 7.

A sensory–motor vector is a real-numbered 26-dimensional vector consisting of 3 current motor values denoting the angular velocities of the 2 wheels and an angle of the arm joint, 2 measured torque values (an average torque value of both wheels, and a torque value of the arm), and 21 values encoding the visual image. The visual field is divided vertically into seven regions, and each region is represented by (1) the fraction of the region covered by the object, (2) the dominant hue of the object in the region, and (3) the bottom border of the object in the region, which is proportional to the distance of the object from the camera. For the region in which there is no colored area, the hue takes a predefined constant value 1.0, and the bottom border position takes 0.0, which is designated as distant. In particular, we note that the visual information is not a priority for the acquisition of semantics. It occupies 21 of the 26 dimensions in the sensory–motor vector only due to the characteristic nature of visual information.

The module has 26 input nodes for the sensory–motor vector, 6 PB nodes, 4 context nodes, 70 hidden nodes, 6 output nodes for the 3 motor commands, 2 predicted torque values which will be sensed, and a predicted hue value for the center region of the visual field (see Figure 4). The rest of the values of the sensory vector are not predicted in order to reduce the learning time. The module can enable the robot to generate behavior appropriately without predicting the entire sensory–motor vector.

In order to robustly generate each behavioral category, each category has to be trained with multiple samplings of manually guided robot trajectories in which each trajectory is slightly different from the others. This training variability is needed because the robust generation of behavior requires generalization in learning sensory–motor sequences. In order to generate different sensory–motor sequences within the same behavioral categories, the positions of the objects in the workspace are slightly varied (within 20 percent of the distance traveled by the robot) to generate each training sensory–motor sequence.

After successful learning, the robot can generate a learned behavioral pattern from an obtained PB vector. In the actual behavior generation process, the module takes the actual sensory–motor vector as input three times per second and generates the motor commands on the fly. The predicted motor values are used as the actual motor commands for the robot in the next time step.

3.5 PB Binding Method

We have already discussed how the linguistic and behavioral modules learn sentences and behavioral patterns, respectively. In this section, the novel associative learning mechanism referred to as PB binding is explained in detail. Both modules are trained at the same time and interact with each other during the learning process. As noted above, the PB binding method imposes the constraint that the PB vectors, for a sentence in the linguistic module and for the corresponding behavioral sequence in the behavioral module, should converge as close as possible to the same value. This constraint is implemented by introducing an interaction term into part of the update rule for the PB vectors in Equation 8. During the learning process, the PB vector $p_{s_k}^{(T)}$ of the sentence $\mathbf{s}_k$ and the PB vector $p_{b_k}^{(T)}$ of the corresponding behavioral sequence $\mathbf{b}_k$ are updated at every training iteration T by means of both the back-propagated error and the mutual interaction as follows:

$$p_{s_k}^{(T)} = p_{s_k}^{(T-1)} + \delta p_{s_k}^{(T)} + \gamma_L \cdot (p_{b_k}^{(T-1)} - p_{s_k}^{(T-1)}) \qquad (9)$$

$$p_{b_k}^{(T)} = p_{b_k}^{(T-1)} + \delta p_{b_k}^{(T)} + \gamma_B \cdot (p_{s_k}^{(T-1)} - p_{b_k}^{(T-1)}) \qquad (10)$$

where γ_L and γ_B are positive coefficients that determine the strength of the binding. Equations 9 and 10 are the constrained update rules for the linguistic module and the behavioral module, respectively. Under these rules, the PB vectors of sentence $\mathbf{s}_k$ and behavioral sequence $\mathbf{b}_k$ attract each other. In particular, the corresponding PB vectors need not be completely equalized to acquire a correspondence at the end of the learning process. The epsilon errors of the PB vectors can be neglected because of the continuity of the PB spaces.

This binding learning is performed off-line, where the training of both modules is conducted by using all the presented pairs of sentences and the corresponding behavioral sensory–motor time sequences in a single batch. At each iteration in the training, the forward computation and the subsequent backward computation for the BPTT are conducted for all linguistic and behavioral sequences, one at a time. Subsequently, the PB vector for each linguistic and behavioral sequence is updated by Equations 9 and 10 and the connection weights of both module networks are updated. This forward and the backward computation for each linguistic and behavioral sequence, computed one at a time, does not necessarily require the time step of the sensory–motor sequence and of the word sequence to be synchronized.

In the testing phase, the linguistic and the behavioral module do not work simultaneously. First, the recognition of a given sentence is performed in the linguistic module by iteratively computing the PB vector. Subsequently, the obtained PB vector is set in the behavioral module in order to generate the corresponding behavior.

4 Experiments

In order to qualitatively examine the self-organization of the internal structures in the proposed modular neural network scheme, we designed three experiments with different learning conditions for comparison. In this section, we briefly explain each experiment.

In experiment I, only the linguistic module was used to investigate the acquisition of the pure syntactic structure of language. The linguistic module is trained with 14 out of 18 possible sentences. In experiment II, only the behavioral module was used to investigate the acquisition of the pure embodied structure of the behaviors. The behavioral module was trained with the sensory–motor sequences of all nine behavioral categories in a supervised manner. In this training of the behavioral module, 10 different sensory–motor sequences were allotted to each behavioral category, and 90 training sequences in total were manually prepared using a remote controller. This approach was used to enhance generalization in the learning of the sensory–motor sequences, as was described previously.

Finally, in experiment III, the associations between the behaviors and the sentences were learned by utilizing both modules. As in the previous experiment, the linguistic module learned with 14 out of 18 possible sentences. The behavioral module learned with 90 sen-

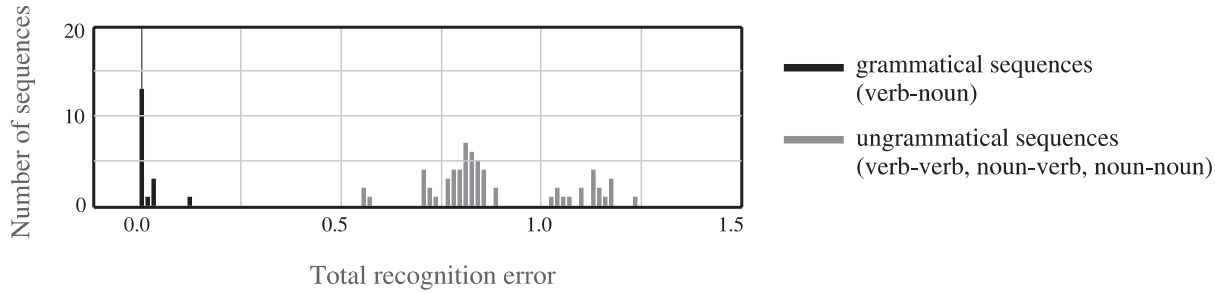

Figure 8 Distributions of the total recognition error of regenerated word sequences.

sory–motor sequences, covering 9 behavioral categories. These two modules were trained simultaneously using the binding scheme described previously. During this training, each different sentence was bound five times with five slightly different sensory–motor sequences within its corresponding behavioral category. As a result, both modules were trained with 70 pairs of bound word and sensory–motor sequences and 20 sensory–motor sequences that were not bound to sentences. In addition, the behavioral module learned the same 90 behavioral sequences without binding. Without this additional unbound training, the acquired structure in the behavioral module tends to be fragile. This approach was necessary because the linguistic regularity is much stronger than the behavioral regularity. In these three experiments, the generation capabilities of sentences as well as behavioral patterns are investigated by analyzing the internal structures self-organized under different learning conditions.

5 Results and Analysis

The following sections describe the results of each experiment.

5.1 Linguistic Module: Syntactic Structure

In this section, we analyze the results of experiment I, in which only the word sequences were learned in the linguistic module. The final averaged output error of each node was 0.0060 after 50,000 steps of learning. An analysis of the linguistic module reveals that the syntax is successfully learned by extracting combinatorial properties hidden in the partial set of sentences given as training data. This generalization characteristic in learning is investigated by examining the obtained PB mapping structures.

In this experiment, 14 of the 18 two-word sentences have been used as training sentences. The remaining four sentences, “point green”, “point right”, “push red”, and “push left” are used to evaluate the generalization capability in the proposed learning scheme. It is found that the linguistic module acquires the underlying syntax correctly from the given sentences. The module generates grammatically correct sentences. The following analysis of the recognition errors confirms this finding. In this analysis, the recognition errors are computed for all the possible two-word combinations, including those that are illegal. The recognition error for a sentence is obtained by attempting to recognize the word sequence. As described in the earlier section, a given word sequence is recognized by means of searching the optimal PB vector for minimizing the error between the target sequence and the output sequence. If the recognition error for a given word sequence is large, the sequence is considered to be nongenerable. Figure 8 shows the distribution of the recognition error of two-word sequences for each sequence type, including the learned type, the unlearned but legal type, and the illegal type. The linguistic module can recognize unlearned sentences as well as learned sentences by minimizing the recognition error. However, it tends to generate a larger error for recognizing illegal sentences. This analysis concludes that only legal word sequences, including unlearned ones, can be recognized/generated in the linguistic module.

Subsequently, we show the analysis of the PB mapping that is generated during learning. For this purpose, the PB vectors corresponding to all the legal sentences are obtained through their recognition processes. Figure 9a shows a plot of the obtained PB vectors using two representative PB nodes. (Unlike the analysis shown later in Figures 11 and 12, we did not employ the principal component analysis (PCA) method to determine the axes at this stage. The plot projected on the surface

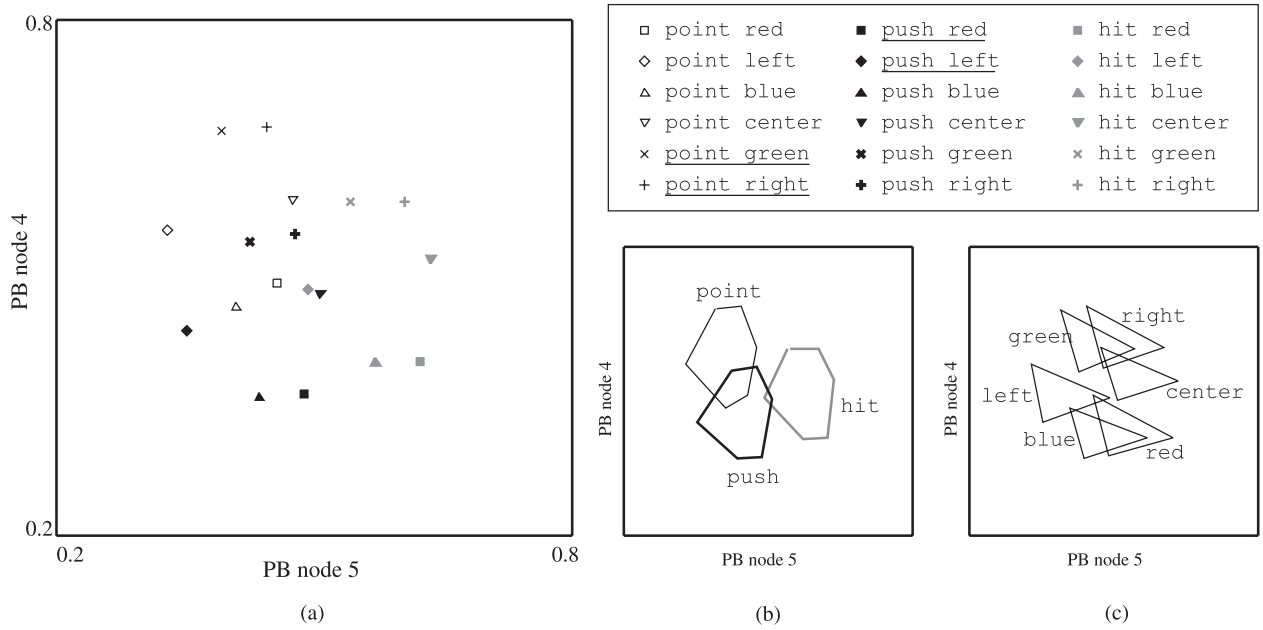

Figure 9 A plot of the linguistic PB mapping acquired without binding (a). The unlearned sentences are underlined. The plot shows three congruent sub-structures for each verb (b), and six congruent sub-structures for each noun (c).

obtained by the PCA is not suitable for observing congruent sub-structures because the PCA tends to enlarge the variances within the whole structure rather than in the local substructures.) In this plot, three congruent sub-structures are observed for each verb (Figure 9b), and six congruent sub-structures for each noun (Figure 9c). It is noted that congruence can be observed by using any of the dimensions of the PB space. From a mathematical point of view, the relationship between n words that have an identical syntactic role can be represented in an $n - 1$ dimensional regular hyper-polyhedron. This implies that the relationship among three verbs and that among six nouns can be represented in two- and five-dimensional spaces, respectively. Therefore, the relationship among the sentences in the experiment should be represented in a $2 + 5 = 7$ dimensional space in an ideal case. The acquired PB structure might be defective because the linguistic module has only six PB nodes. However, the defect seems to be negligible in the observed results.

From the apparent congruence in the sub-structures for verbs and nouns, it is inferred that the combinatorial characteristics of verbs and nouns are extracted in the PB mapping using only a partial set of possible legal sentences. In particular, it is interesting to observe that the PB vectors for unlearned sentences are observed to be in the correct positions in the configuration of the congruence. For example, the PB vector of "push green" comes at the cross-over point between the sub-structures for push and green. These results confirm that the sentences are learned and generalized by extracting the possible combinatorial characteristics from the example sentences.

Further, an interesting phenomenon is revealed by observing the connection weights from the input nodes to the hidden nodes, which could be related to discussions regarding the method in which syntax structures undergo self-organization. The pattern of the connection weight values from each input node is clearly divided into two groups, depending on the type of the corresponding word: Namely, for all the hidden nodes, the connection from each verb node to a specific hidden node has an identical weight value. Similarly, the connection from each noun node to a specific hidden node has an identical weight value. In other words, the values propagated from the word input nodes to the hidden nodes do not carry information about the word itself, but about the class of the word, i.e., a verb or a noun. This would explain the reason for the appearance of the congruence in the PB mapping.

It should be noted that in the first part of the experiment, the linguistic module can learn only the syntactic aspects of the sentences without their meanings because it is trained without any behavioral con-

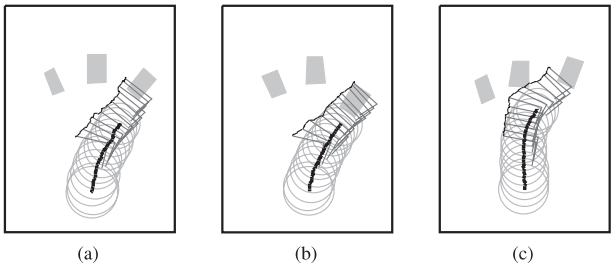

Figure 10 The three trajectories (a, b, and c) show that the robot can achieve PUSH-G behavior robustly with the green target object placed in positions that differ slightly.

text. In the linguistic module, it is seen that each of the three verbs and each of the six nouns is syntactically equivalent within each group. Each noun appears immediately after the same set of verbs, and each verb appears immediately before the same set of nouns. This sort of syntactic equivalence between words is acquired as a result of learning generalization, even with the asymmetry in the training set consisting of partial word sequences.

5.2 Behavioral Module: Embodied Structure

In this section, we analyze the results obtained from experiment II, in which only the behavioral module is trained. The final averaged output error of each output node is 0.012 after 50,000 steps of learning. All nine behavior categories are successfully generated using the PB vector values obtained in the learning process. (In this case, an averaged PB vector obtained for multiple training sequences within each behavioral category is used for each behavior generation test.) An additional experiment showed that the robot can successfully act on objects that are located in arbitrarily different positions, within 20% of the travel distance, with the PB vector obtained in learning (see Figure 10). Furthermore, as long as the object is within its sight, the robot can continue the target behavior even if the object is slightly shifted from its position during the behavior.

The sensory–motor sequences have different step lengths even within the same behavioral categories. This characteristic of the sensory–motor sequences is

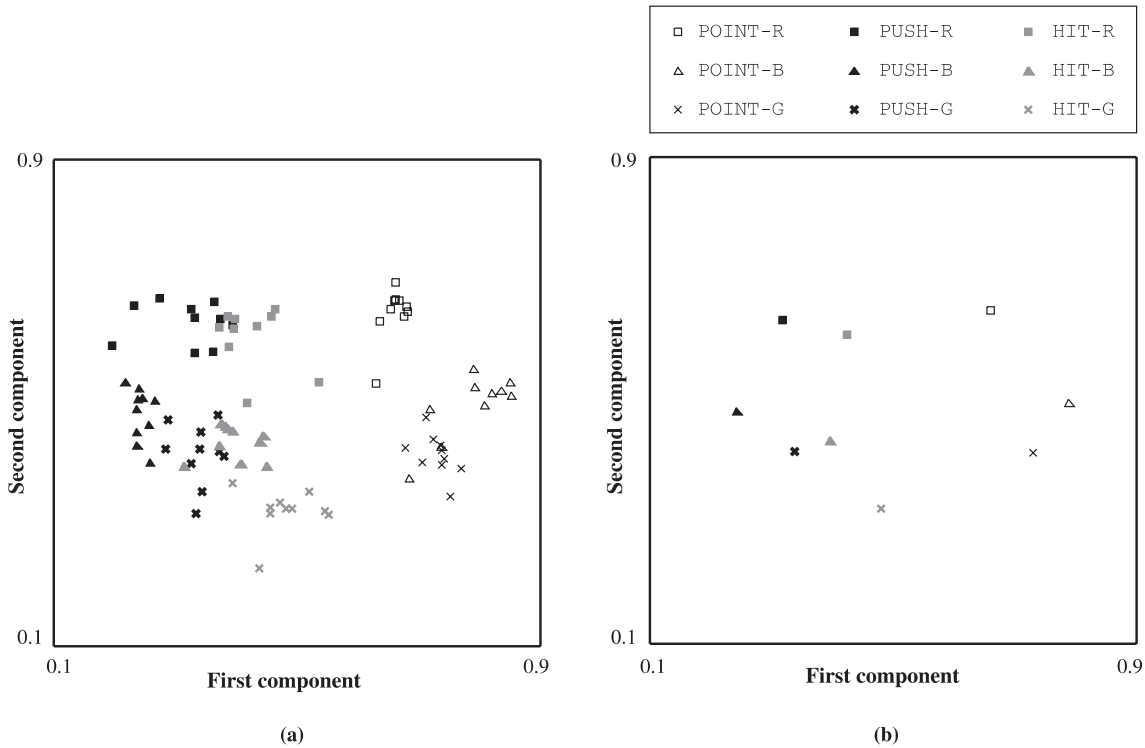

Figure 11 A plot of the PB vectors acquired through the learning of 90 behavioral training sequences without binding (a) and the averaged PB vectors for each behavioral category (b). The six-dimensional PB space was projected onto a surface that maximizes the deviation of the plots by using the PCA method. The accumulated contribution ratio of the two axes is 70%.

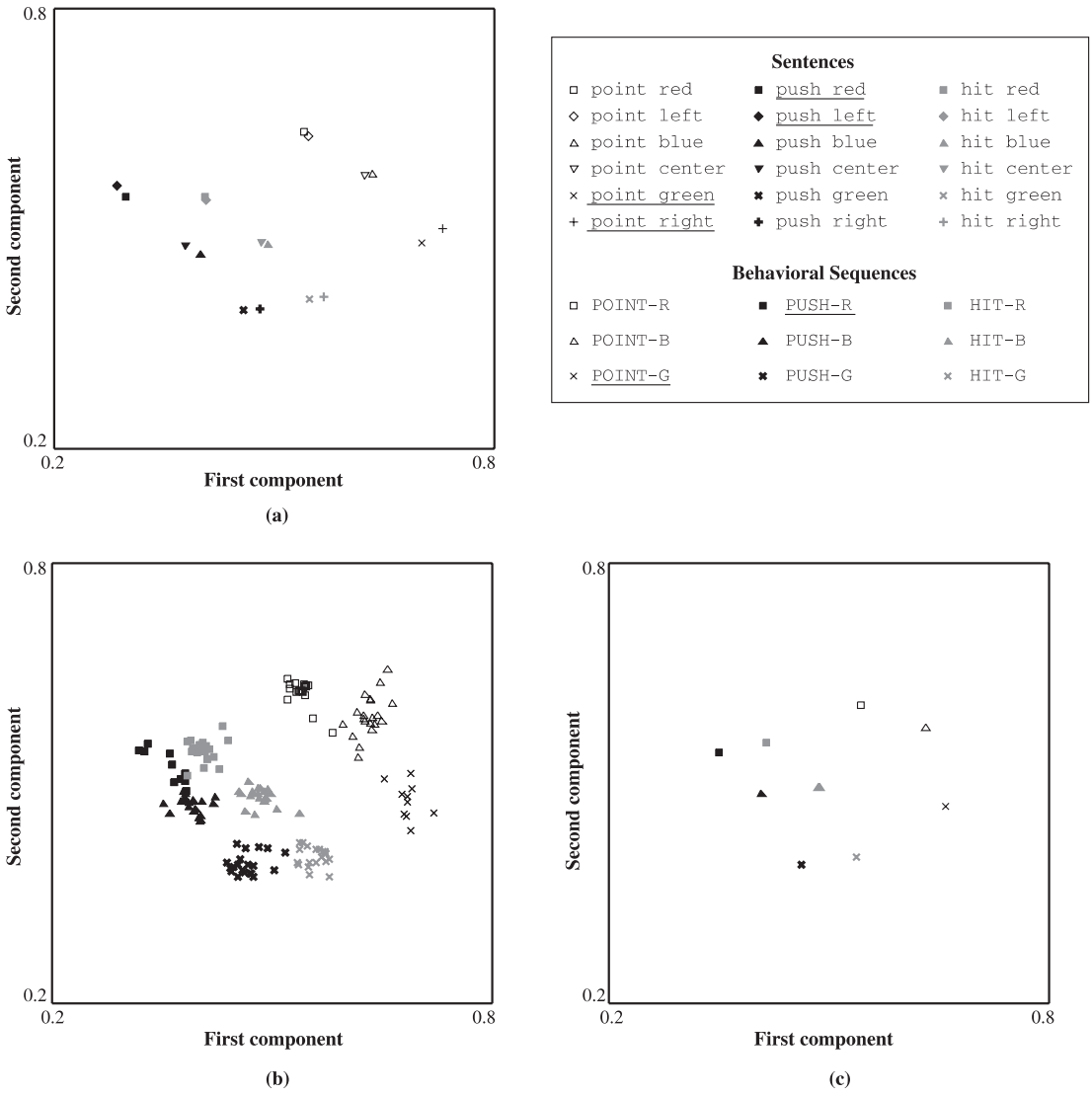

Figure 12 The plots show (a) the PB vectors for recognized sentences in the linguistic module, (b) the PB vectors for training behavioral sequences in the behavioral module, and (c) the averaged PB vector for each behavioral category.

different from that of the word sequences, which are restricted to having two words for each sentence. The behavioral module learns these various behavioral sequences by embedding them into dynamic functions parameterized with the PB vector rather than learning each by rote. This is why the robot can robustly generate the appropriate behaviors under certain perturbations of the environment.

Figure 11 shows the PB space obtained in this experiment. In Figure 11a, the acquired PB vectors for all 90 behavioral training sequences are presented (10 vectors for each behavioral category). The 6-dimensional PB space was projected onto a surface that maximizes the deviation of the plots using the conventional principle component analysis (PCA) method. Figure 11b shows the PB vectors averaged over 10 vectors within the same behavioral categories. As nine clusters are observed in the PB space, which correspond to the nine behavioral categories, it can be concluded that the PB mapping is self-organized with categorical structures that preserve the similarity among the sensory–motor sequences.

This part of the result is analogous to what we observed in the self-organization of the PB mapping in behavioral pattern learning using an arm robot (Tani & Ito, 2003) as well as a Sony humanoid robot (Ito &

Tani, 2004b), where it was shown that each behavioral category was embedded in each corresponding attractor dynamics parameterized by the PB vector. The self-organization of the categories for the relevant dynamic structures accounts for the robustness observed in behavior generation. In this experiment, however, no clear combinatorial structures among the behavioral categories can be observed in the PB space. This result is different from that of experiment I, wherein only linguistic sequences are learned.

5.3 Unified Structure

In this section, we analyze the results of experiment III, in which the sensory–motor sequences and the word sequences were learned by the binding of the two modules. The final averaged output error of each node is 0.0091 for the linguistic module and 0.025 for the behavioral module after 50,000 steps of learning. It is found that the corresponding behaviors are successfully generated after recognizing any learned sentences.

As mentioned in Section 4, the training data for this experiment do not include all the correspondences. Four sentences ("point green", "point right", "push red", and "push left") are not included in the training data for the linguistic module. Therefore, despite the behavioral module being trained with the behavioral sequences of all behavioral categories, two behavioral categories (POINT-G and PUSH-R) are trained without binding the corresponding sentences. The rest are trained with binding.

The most important result of this experiment is that the robot can generate the appropriate behaviors robustly for all the sentences including unlearned ones. In other words, the meanings of all four unlearned sentences can be recognized correctly to generate the corresponding behaviors. This implies that a certain generalized structure appeared in the results of the interactive learning between the two modules.

In order to investigate the self-organized internal structure, the PB mappings for both modules are examined, as shown in Figure 12.

All the plots are projections of the PB spaces onto the same surface determined by the PCA method. In this case, the accumulated contribution rate is approximately 73%. Figure 12a shows the PB vectors obtained as the result of recognizing all 18 legal sentences. Figure 12b shows the PB vectors obtained as the result of training the 90 behavioral sequences. Figure 12c shows the averaged PB vectors for all behavioral categories. The comparison of the PB mappings between both modules, as shown in Figures 12a and 12c, shows that they indeed share a common structure as a result of the binding. Figure 12a shows the congruent substructures similar to those observed in experiment I in the linguistic PB mapping. The PB vectors of four unlearned sentences are also observed in the correct positions in the linguistic PB space, and their positions coincide with the positions of the corresponding PB vectors in the behavioral module. Another interesting observation is that the PB vectors for pairs of sentences with identical meanings but with different nouns, such as "point red" and "point left", appear to be close to each other in the linguistic PB mapping, as shown in Figure 12a.

This analysis concludes that the structure in the PB mapping with binding is acquired by means of combining (1) the combinatorial structure from the linguistic module with (2) the metric space, based on the similarities among the sensory–motor sequences, that originated from the behavioral module. The interactions between the two modules enable linguistic PB mapping (Figure 12a) to be self-organized as an embodied combinatorial structure. The embodiment has affected self-organization in linguistic PB mapping. Similar linguistic PB vectors share the same behavioral concept, as described previously. The modules also affect each other in opposite directions, i.e., the linguistic structure affects the self-organization of the behavioral structure. The combinatorial structure organized in the linguistic module is introduced into the behavioral module (Figure 12b). In contrast to the PB mapping acquired from the behavioral module in experiment II (Figure 11a), we can find geometric regularity in the relationships among behavioral categories in the plot of Figure 12c.

It has been observed that the PB vectors of the unlearned sentences and those of the corresponding behavioral sequences successfully coincide without binding during learning. This has been achieved because the shared structure between the PB mappings in the two modules has self-organized with the underlying relationship within sentences and behavioral categories as well as the relationship across these two modalities. This is the mechanism for generating the situated compositional semantics through the generalization process in learning.

It is also noted that the same experiment was repeated three times with different sets of four unbounded sentences, where a similar topological organization

was observed in the PB mapping of both the modules. However, it was found that the performance declines gradually when more than four sentences are removed from the training data.

6 Discussion

Our simple experiments showed that the structures for the situated semantics of primitive language can self-organize in an attempt to acquire a generalized correspondence between the word sequences and the sensory–motor flow of a robot. Although the present study has not yet shown the strong systematicity of Hadley (1994), it has addressed at least the minimal combinatorial characteristics of language. This implies that the robot could understand relatively simple sentences in a systematic way and could understand novel sentences by generating the appropriate corresponding behaviors. Our two-word language is very similar to the language $\mathcal{L}_0$ in Evans (1981). The argument regarding the compositional semantics of $\mathcal{L}_0$ holds true for our language as well. However, the compositionality discussed in conventional linguistics and that assumed to be organized in our dynamical systems approach seem to have crucial differences. Similar differences can also be seen in the treatments of the symbol grounding problem between the conventional computational approach and the dynamical systems approach. The remaining sections will first proceed by comparing our approach to others, and then the issues of symbol grounding and compositionality will be addressed.

6.1 Comparison to Related Works

It has been shown that some other models (Roy, 2002; Iwahashi, 2003) can acquire situated compositional semantics provided that certain computational data structures, such as trees or graphs, are predefined for representing and manipulating phonological expressions and conceptual knowledge. In their schemes, the learning is divided into several computation modules: (1) articulating a sentence into a series of words; (2) learning the underlying syntactic rules; (3) transforming the information about a referent of a sentence into a combination of predefined conceptual elements; and (4) acquiring a meaning of a word from the co-occurrence relation between words and conceptual elements. By means of these divisions, compositional semantics is acquired by associating the syntactic rules and the situated meanings of a word, each of which is learned rather independently. Although this modular approach is advantageous for scaling of the training data size (Roy, 2002) and in the introduction of additional complexity, for example, contextual dialog processing across sentences (Iwahashi, 2003), the scheme requires substantial task-specific designs and programming. In contrast, our model required less task-specific preprogramming because the computation involving (2)–(4) is seamlessly implemented by employing the capability of the RNNPB, although the articulation of sentences into words is presumed.

Vogt (2003) and Steels (2002) examine the emergence of compositional phonological representation of an embodied language by using evolutionary models, referred to as the iterated learning model (ILM) (Kirby & Hurford, 2002) and the language game approach (LGA) (Steels & Vogt, 1997), respectively. The concept of a sentence is represented in the form of a fixed-dimensional feature vector, where each feature dimension is assigned predefined properties of a colored geometric figure such as color and shape. A meaning of a word is represented as a vector specifying a certain subset of the features. According to the definition of the conceptual representation, both sentences and words are guaranteed to have grounded meanings. Through an evolutionary process, syntactic rules and words are acquired in an interdependent manner. A new (syntactic) composition rule is invented according to the heuristics, while a given concept is decomposed into a set of concepts that undergo rule-based assembling into a concept. If the new concept fragment has no corresponding word, a new word is introduced in the model. It should be noted that a concept can always be decomposed in a straightforward manner because of its predefined combinatorial representation.

Vogt's and Steels' studies seem to approach the issues of situated compositional semantics in a manner opposite to that of ours. They start from the predefined structure of the concept space, in which every possible concept can be represented as a combination of predefined features of the perceived visual image, and they gradually promote the combinatorial phonological representation by inventing new words and their corresponding concepts such that the concepts of the given objects can be fully expressed. On the other hand, our model employs a predefined set of words and avoids the problems of word segmentation in sentences. We

examine how dynamic combinatorial structures that embed concepts can self-organize in a manner that is situated to behavior, which is represented as a spatio-temporal pattern of sensory–motor images. Both models can work in a complementary manner to investigate the origin of compositionality.

The study by Kirby and Hurford (2002) on the role of the communicative bottleneck in acquiring compositional semantics also seems to complement the present study. This study shows that communicative constraints, which cannot be reduced to the capabilities of each individual agent, can work as an important driving force of the evolutionary self-organization of the linguistic combinatoriality over generations. In their model, the survivability of a syntactic rule depends on the diversity of its generable sentences. In other words, the more utterances a rule generates, the more the probability for the rule to be reproduced in the next generation. Therefore a language that can be covered with a smaller number of rules is preferred.

This argument concerning the communicative bottleneck is related to the issues of learning generalization in our model. Our experiment showed that novel combinations of words can be recognized rationally by analogy with other learned combinations. It was also shown that the PB mapping is self-organized such that the combinatorial relations between verbs and nouns are well represented in the low-dimensional PB space. The generalization in learning using our neural network model of the present study and the evolution of a smaller number of rules with a rich combinatoriality in Kirby's model have similar ideas.

The ideas for organizing situated semantics presented in the studies by Cangelosi (2004), using a three-layered feed-forward network, seem to be very similar to ours. However, the problem is whether a simple feed-forward network can deal with the combinatorial and context-dependent characteristics of both the linguistic and behavioral processes. In order to achieve such complex linguistic and behavioral processes, an architecture capable of acquiring internal models rather than merely adapting to sensory–motor maps might be required. Our proposed architecture is characterized by its acquisition of internal models in terms of forward models (Kawato et al., 1987), both in the word and the sensory–motor sequences, utilizing the contextual loop in the RNNPBs. It can be said that the observed complexity as well as the contextual nature of the system originate from the iterative forward and inverse computation in the generation and the recognition processes through the nonlinear dynamics of the RNNPB.

6.2 Rethinking the Compositionality and Symbol Grounding Problems

Finally, we examine the issues of the symbol grounding problem and compositionality. We attempt to provide alternative interpretations of the issues based on the dynamical systems approach employed in our scheme. For this purpose, prior studies on the symbol grounding problem in conventional behavior-based robotics and our alternative of using the dynamical systems approach are first reviewed. Compositionality is then discussed as a natural extension of our approach.

The original discussions of the symbol grounding problem by Harnad (1990) focused on the method to achieve seamless connections between various concepts, represented by symbol systems that consist of arbitrarily shaped tokens, and the physical world, defined in a metric space such as size, weight, speed, etc. Harnad's idea for solving the problem was to introduce categorizers for the symbols. The categorizers categorize real-world patterns and attach them to corresponding symbols.

This approach has been widely employed by the behavior-based robot community (Arkin, 1998). Landmark-based navigation using a topological trajectory (Kuipers & Byun, 1987; Mataric, 1992) might be a good explanatory example. The topological trajectory is represented by a finite-state machine (FSM) where the nodes represent encountered landmarks and the arcs represent possible paths between two landmarks. Pattern categorizers are employed in order to recognize the encountered landmarks. The result of the recognition is matched with that expected from the FSM representation during navigation.

Problems can occur when the recognition of a landmark is perturbed by noise. The FSM simply halts when it receives illegal input symbols, and the robot is lost in the workspace. Although various engineering exception handling techniques could partially remedy the situation, the potential problem would persist. The crux of the problem resides in the fact that the symbol systems cannot interact closely with physical systems since they are not defined in the metric space shared with the physical world. When certain conflicts or disparities occur between the expectations in symbol systems

and the physical reality, the conflicts cannot be well absorbed since the symbol systems do not maintain the required elasticity.

In this situation, an alternative proposed by Tani (1996) is to use adaptive dynamical systems, such as RNNs, which can mimic certain symbolic computations as shown by Elman (1990) and Pollack (1991). In particular, Tani (1996) showed that an equivalent to the FSM topological trajectory can be learned as they are embedded in the RNN attractor, in terms of a Cantor set encoding through the exploratory navigation of the robot. It was also shown that the robot, lost and wandering aimlessly after being perturbed by noise, can determine its position in the environment by the entrainment of its internal neuronal dynamics through familiar sensory input sequences. This type of auto-recovery mechanism becomes possible because the internal neuronal dynamics and external physical dynamics can share the same metric space in their structural coupling.

By referring to the studies of embodied language or situated semantics conducted by other groups, such as Roy (2002), Iwahashi (2003), and Steels (2002), it may be observed that these are similar to the conventional behavior-based robotics approach in terms of their attempts to ground their semantic representations. They employ tree or graph computational structures for representing semantics and attempt to ground them by using feature vectors for their interfaces to the physical world.

In our approach, the necessary mechanism of combinatoriality can be self-organized in the RNNPB by utilizing the nonlinear dynamical characteristics of the parametric bifurcation. In particular, Tani and Ito (2003) as well as the current study have shown that diverse sequential patterns can be generated in a combinatorial manner by manipulating the PB vector. Our objective is not to achieve seamless connections from the representational entities of concepts to the physical world, but rather to introduce active interactions among different modalities: Namely, the linguistic processes and behavioral processes. Consequently, the necessary internal dynamical structures, that account for the situated combinatorial semantics, self-organize through such dense interactions between these two processes.

Our essential argument is that semantics be regarded as inseparable processes in the whole dynamics of embodied cognition. In contrast, others (Roy, 2002; Iwahashi, 2003; Steels, 2002) treat semantics as separable modules: Semantic symbols to be grounded and concepts to be associated with the semantic symbols. It should be noted that this separation results in the symbol grounding problem. We could argue that there exist no representational entities to be grounded in our scheme. There exist only dynamical structures that emerge as a consequence of the interactions between the behavioral and linguistic processes. Since there are no symbols to be grounded, the symbol grounding problem might be regarded as a pseudo-problem in our dynamical system views.

Similar discussions could be constructed for issues related to compositionality. According to the conventional definition of compositionality, the meaning of a sentence is a function of the meanings of its parts: namely, words. This definition is in keeping with the model by Roy (2002) and Iwahashi (2003) since in their model each word has a certain representation by a feature vector and sentences actually have representations composed of those of the words. However, in our approach, it is difficult to determine the location where the meaning of each word is stored. Since the robot can recognize the possible combinations of verbs followed by nouns and can generate the corresponding behaviors, it appears as if the meanings of verbs and nouns existed as independent entities, and as if the robot recognized sentences by combining those meanings.

However, this interpretation simply arises from computational views based on reductionism, where the whole is regarded as divisible into parts, as has been shown by the models of Roy (2002) and Iwahashi (2003). On the other hand, the dynamical systems perspective provides a holistic view, where neither the whole can be divided into its parts, nor can the meaning of a sentence be divided into those of its individual words. We could argue that the meaning of "red" can be understood only as being associated with the relevant actions of "point", "push", and "hit", but not by "red" itself. Our examination of the PB mapping has shown that the meaning of each sentence is structured relative to other sentences. The observation of the congruence in the PB mapping indicates that the combinatorial characteristics between verbs and nouns are well extracted and embedded in the neuronal dynamics by means of self-organizing combinatorial mechanics.

This observation does not support the view that the meaning of a sentence is composed of the meanings of its words. When the meaning appears only in

the relative structures among learned sentences and their corresponding sensory–motor sequences, as has been shown in our proposed dynamical systems formulation, the conventional linguistics' idea of compositionality might deserve reconsideration.

7 Conclusions and Future Studies

The present studies have shown how the situated/embodied concept of linguistic competence can be achieved by using the synthetic robotic approach, where the results were articulated in terms of dynamical systems. In our proposed model, coupled RNNs were trained on a structured mapping between a simple linguistic representation and a sensory–motor system. This training resulted in generalization in learning of the linguistic representation. Thus, the robot was able to access unlearned sentences and generate the correct corresponding behaviors. At the same time, the robot performed goal-directed behaviors robustly against various perturbations. The analysis showed that unified structures are self-organized by means of iterative interactions between linguistic and behavioral structures, in which the combinatorial characteristics of language as well as situatedness in sensory–motor contexts are preserved. We conclude that the situated compositional semantics can be achieved solely through the self-organizing processes of dynamical structures rather than relying on the conventional ideas of symbols to be grounded in many of the behavior-based robotics approaches or in the compositionality of linguistic theory.

Future research will include theoretical studies of the proposed scheme, scaling of the system, and extensions of the scheme to the problem of the origin of language (Kirby & Hurford, 2002; Steels, 2002). Theoretical studies should be conducted to investigate the more basic properties of the RNNPB. In particular, the issues of generalization in learning should be examined by introducing carefully controlled studies that vary the number of training sets, the amount of noise in each training set, the number of PB nodes, and the complexity of the self-organized structures. It is also important to investigate the possible limitations in organizing the structural mapping in the proposed scheme. There could be classes of structures which cannot be bound in different modalities.

The scaling of systems is also an important issue that needs to be addressed. The linguistic part should cover larger language sets, including the recursive syntax, which is another essential feature of human language. Although it has been proven that a simple RNN has the capability of learning nested sentences (Wiles, Blair, & Boden, 2001), it is not yet known whether such highly structured sentences can be successfully mapped to behavioral patterns. Further, it would be natural to expect that additional linguistic complexity should be accompanied by a corresponding increase in behavioral complexity. However, the present simple architecture would not allow the behavioral module to adapt to more complex behavioral situations. The behavioral module might require certain hierarchical structures, as has been shown in Tani and Nolfi (1999) and Tani (2003), in order to gain the necessary behavioral complexity.

The scheme could be extended to conduct simulation studies of the origin of language(Kirby & Hurford, 2002; Steels, 2002; Vogt, 2003). Although the language set is given in the present study, it could be evolved by introducing communicative interactions among a set of agents. For example, certain collaborative tasks that require multiple agents to communicate with each other could be considered. In the course of evolving the RNNPB in a linguistic module of each agent, an appropriate class of linguistic competency might emerge which would allow achievement of the collaborative tasks given to the system.

References

Arbib, M. (2002). The mirror system, imitation, and the evolution of language. In K. Dautenhahn & C. L. Nehaniv (Eds.), *Imitation in animals and artifacts* (pp. 229–280). Cambridge, MA: MIT Press.

Arkin, R. C. (1998). *Behavior-based robotics.* Cambridge, MA: MIT Press.

Beer, R. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence, 72*(1), 173–215.

Billard, A. (2002). Imitation: A means to enhance learning of a synthetic proto-language in an autonomous robot. In K. Dautenhahn & C. L. Nehaniv (Eds.), *Imitation in animals and artifacts* (pp. 281–311). Cambridge, MA: MIT Press.

Cangelosi, A. (2004). The sensorimotor bases of linguistic structure: experiments with grounded adaptive agents. In S. Schaal et al. (Ed.), From animals to animats 8: Proceedings of the Eighth International Conference on Simulation of Adaptive Behavior (pp. 487–496). Cambridge, MA: MIT Press.

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

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.

Gelder, T. van. (1998). The dynamical hypothesis in cognitive science. *Behavior and Brain Sciences, 27*(5), 615–628.

Hadley, R. (1994). Systematicity revisited: Reply to Christiansen and Chater and Niklasson and van Gelder. *Mind and Language, 9,* 431–444.

Harnad, S. (1990). The symbol grounding problem. *Physica D, 42,* 335–346. Ito, M., & Tani, J. (2004a). Generalization in learning multiple temporal patterns using RNNPB. In *Proceedings of the 11th International Conference on Neural Information Processing (ICONIP' 04)* (in press).

Ito, M., & Tani, J. (2004b). On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior* (in press).

Iwahashi, N. (2003). Language acquisition by robots—towards a new paradigm of language processing. *Journal of Japanese Society for Artificial Intelligence, 48*(1), 49–58.

Jordan, M., & Rumelhart, D. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science, 16,* 307–354.

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.

Kirby, S., & Hurford, J. (2002). The emergence of linguistic structure: An overview of the iterated learning model. In D. Parisi & A. Cangelosi (Eds.), *Simulating The Evolution of Language* (pp. 121–148). London: Springer.

Kuipers, B., & Byun, Y. T. (1987). A qualitative approach to robot exploration and map-learning. In *IEEE workshop on spatial reasoning and multi-sensor fusion* (pp. 390–404). Los Altos, CA: IEEE.

Mataric, M. (1992). Integration of representation into goal-driven behavior-based robot. *IEEE Transactions on Robotics and Automation, 8*(3), 304–312.

Miikkulainen, R. (1993). Cambridge, MA: MIT Press.

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

Rizzolatti, G., Fadiga, L., Galles, B., & Fogassi, L. (1996). Promotor cortex and the recognition of motor actions. *Cognitive Brain Research, 3,* 131–141.

Roy, D. (2002). Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language, 16,* 353–385.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In D. Rumelhart & J. Mclelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 318–362). Cambridge, MA: MIT Press.

Schoner, S., & Kelso, S. (1988). Dynamic pattern generation in behavioral and neural systems. *Science, 239,* 1513–1519.

Siskind, J. (2001). Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. *Artificial Intelligence Research, 15,* 31–90.

Steels, L. (2000). The emergence of grammar in communicating autonomous robotic agents. In W. Horn (Ed.), *Proceedings of ECAI 2000: 14th European Conference on Artificial Intelligence* (Vol. 54, pp. 764–769). Amsterdam: IOS Press.

Steels, L. (2002). Grounding symbols through evolutionary language games. In A. Cangelosi & D. Parisi (Eds.), *Simulating the evolution of language* (pp. 211–226). London: Springer.

Steels, L., & Vogt, P. (1997). Grounding adaptive language games in robotic agents. In P. Husbands & I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life, ECAL'97* (pp. 474–482). Cambridge, MA: MIT Press.

Sugita, Y., & Tani, J. (2002). A connectionist model which unifies the behavioral and the linguistic processes: Results from robot learning experiments. In M. Stamenov & V. Gallese (Eds.), *Mirror neurons and the evolution of brain and language* (pp. 363–376). Amsterdam/Philadelphia: John Benjamins.

Tani, J. (1996). Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems, Man and Cybernetics, Part 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., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. *IEEE Transactions on Systems, Man and Cybernetics, Pan A, 33*(4), 481–488.

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.

Thompson, C. A., & Mooney, R. J. (1998). *Semantic lexicon acquisition for learning natural language interfaces* (Tech. Rep. No. AI98–273). The University of Texas at Austin.

Vogt, P. (2003). Iterated learning and grounding from holistic to compositional languages. In S. Kirby (Ed.), *Language evolution and computation, Proceedings of the Workshop/ Course at ESSLLI* (pp. 76–86).

Wiggins, S. (1990). *Introduction to applied nonlinear dynamical systems and chaos.* New York: Springer.

Wiles, J., Blair, A., & Boden, M. (2001). Representation Beyond Finite States: Alternatives to Push-Down Automata. In J. F. Kolen & S. C. Kremer (Eds.), *A field guide to dynamical recurrent networks* (pp. 129–142). New York: IEEE Press.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology, 3*(1), 1–191.

About the Authors

Yuuya Sugita is a Ph.D. candidate at the Department of General Systems Studies of the University of Tokyo, Japan. Currently, he works as an Associate Researcher at Brain Science Institute of RIKEN, Japan. His research interests include embodied cognition, complex adaptive systems, and the origin of symbol manipulation ability.

Jun Tani was born in 1958 in Tokyo, Japan. He received the B.S. degree in mechanical engineering from Waseda University, M.S. degree in electrical engineering from Michigan University and Dr. Eng. from Sophia University. He is currently a Team Leader in Lab. For Behavior and Dynamic Cognition, Brain Science Institute, RIKEN in Tokyo, Japan. He is interested in embodied cognition, complex adaptive systems and brain science. **Address**: BSI RIKEN, Hirosawa 2–1, Wako-shi, Saitama 351019 Japan.
E-mail: tani@bdc.brain.riken.go.jp