2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Human-Robot Cooperation using Quasi-symbols Generated by RNNPB Model

Tetsuya Ogata, Shohei Matsumoto, Jun Tani, Kazunori Komatani, and Hiroshi G. Okuno

Abstract - **We describe a means of human robot interaction based not on natural language but on "quasi symbols," which represent sensory-motor dynamics in the task and/or environment. It thus overcomes a key problem of using natural language for human-robot interaction – the need to understand the dynamic context. The quasi-symbols used are motion primitives corresponding to the attractor dynamics of the sensory-motor flow. These primitives are extracted from the observed data using the recurrent neural network with parametric bias (RNNPB) model. Binary representations based on the model parameters were implemented as quasi symbols in a humanoid robot, Robovie. The experiment task was robot-arm operation on a table. The quasi-symbols acquired by learning enabled the robot to perform novel motions. A person was able to control the arm through speech interaction using these quasi-symbols. These quasi symbols formed a hierarchical structure corresponding to the number of nodes in the model. The meaning of some of the quasi-symbols depended on the context, indicating that they are useful for human-robot interaction.**

I. INTRODUCTION

The meaning of the words and sentences used in daily conversations depend on the situation and/or context. This property of natural language enables flexible and effective communication. However it leads to the *symbol-grounding* problem [1]. Because most computational systems including intelligent robots can process only predefined symbolic representations, they do not have the human ability to handle natural languages.

We have been working on this problem and have investigated the use of *quasi-symbols* as a prelude to human robot communication using natural language. As quasi-symbols we use "motion primitives," which we define as components of a complex motion. They represent the boundaries that form the attractor dynamics (not shape) of the sensory-motor flow. We identified two conditions that quasi-symbols should satisfy for them to be used effectively in human-robot communication.

- They should be plausible enough for users to be able to interpret them easily.
- They should be sufficiently rich in information for the users to be able to use them for interactions.

Some studies generating symbols for use in robot systems use techniques like the self-organizing map [2] that can handle only static vectors. The environment, objects, and task are thus encoded as static images, so the generated representations usually satisfy only the first conditions above. They cannot satisfy the second condition because the static image encoding cannot handle dynamic features like motions and events, which are essential in communication.

Studies on symbol emergence have dealt with sequential processes by using stochastic models (hidden Markov, Bayesian, and so on) [3]. While the symbols generated are generally sufficient for analyzing human behavior and controlling humanoid robots, they are usually restricted to handling sensory-motor sequences consisting of nodes (i.e., states) in a defined HMM and/or Bayesian model. It is thus difficult for these models to generate novel motions that reflect non-linear dynamic coupling. These models also require huge amounts of data for learning, and it is problematic for actual robot systems to collect lots of data due to the durability problem.

Other studies used designed dynamic attractors [4]. First, "goal" trajectories are designed in the sensory-motor space and many uniform vectors are added around the trajectories to create convergent dynamics. However, from the dynamical systems perspective, the key property is the form of the basin around the attractors rather than the converged trajectories. In other words, the transient dynamics are more crucial than the shape of the attractors. It is difficult to design these basins appropriately by adding only uniform vectors. The final convergent trajectories are emerged through interaction with the other dynamical systems like the body and environments. We think motion primitive is not a fixed goal but flexible function according to the coupled dynamical systems.

A previous study of human-robot interaction based on *quasi-symbols* generated by the recurrent neural network with parametric bias (RNNBP) model [5] showed that the quasi-symbols were plausible enough for people to use them to operate and cooperate with robots. However, since the RNNPB in the robot acquired only four kinds of symbols representing forwarding and rotating motions, the results were easy to predict and it was difficult to use them for complex cooperation.

We have conducted trial experiments using a humanoid robot with an artificial neural network that can acquire

T. Ogata, Y. Hattori, K. Komatani, and H. G. Okuno is with the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, Japan {ogata, komatani, okuno}@kuis.kyoto-u.ac.jp

J. Tani is with the Brain Science Institute, RIKEN, Saitama, Japan tani@brain.riken.jp

quasi-symbols automatically from the environment. It can also generalize observed experiences.

Section II summarizes the concept of motion primitives, and Section III describes the RNNPB model we used as the learning algorithm. Section IV describes the design of quasi-symbols using the parameters output by the neural network model and their implementation. Section V describes our experiments and presents some of the results. Section VI concludes the paper with a summary of the key points and a look at future work.

II. Motion Primitive

As mentioned above, we used motion primitives as the quasi-symbols. They should encode not only static states but also dynamic patterns in the environment. They are usually defined as components of complex motions.

One way to represent these primitive is based on motion trajectories [6]. Such methods usually cut off the parts of trajectories where motion velocity is close to zero. Each part is approximated as a straight line, circle, and/or spline function. These "primitive" parts are practical, especially for motion imitation by a humanoid robot, which has morphology quite similar to that of human. However, the problem created by the difference in body dynamics between human and robot becomes more serious as the motion speed increases.

Another ways is based on the dynamics that generate the trajectories [7][8]. Systems using this approach usually consist of several dynamic recognizers that predict target sequences individually (local expression). Each primitive is determined as a part of a trajectory that one dynamic recognizer can follow reasonably well. If the recognizer loses its ability to follow the target trajectory, it is replaced by the other dynamic recognizer that is best able to do so. In this approach, the characteristics of each recognizer are fundamental rather than the velocity and trajectory shape.

III. Learning Algorithm

The RNNPB model we use to enable robots to deal with the dynamic features of sensory-motor information is the one proposed by Tani and Ito [9]. It articulates complex motion sequences into motion units, which are encoded as the limit cycling dynamics and/or the fixed-point dynamics of the RNN.

A. RNNPB Model

The RNNPB model we used has the same architecture as the conventional Jordan-type RNN model [10] except that it has the PB nodes in the input layer. Unlike the other input nodes, the PB nodes have a constant value throughout each time sequence. They are used to implement a mapping between fixed length values and time sequences. The network configuration of the RNNPB model is shown in Figure 1.

Like the Jordan-type RNN model, the RNNPB model learns data sequences in a supervised manner. The difference is that in the RNNPB model, the values that encode the sequences are self-organized in the PB nodes during the learning process. The common structural properties of the training data sequences are acquired as connection weights by using the back propagation through time (BPTT) algorithm [11], and the specific properties of each individual time sequence are simultaneously encoded as PB values. As a result, the RNNPB model self-organizes the mapping between the PB values and the time sequences.

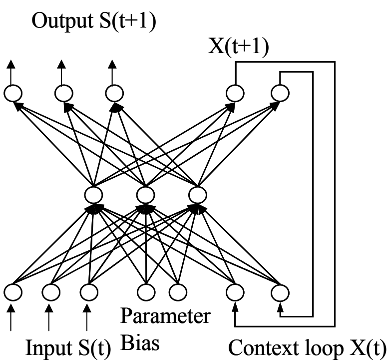

Figure 1 Network configuration of RNNPB model

B. Learning of PB Vectors

The learning algorithm for the PB values is a variant of the BPTT algorithm. The step length of a sequence is denoted by l. For each of the sensory-motor outputs, the back-propagated errors with respect to the PB nodes are accumulated and used to update the PB values. The update equations for the ith unit of the parametric bias at the step t in the sequence are as follows.

$$\delta\rho_t = k_{bp} \cdot \sum_{t-l/2}^{t+l/2} \delta_t^{bp} + k_{nb}(\rho_{t+1} - 2\rho_t + \rho_{t-1}) \qquad (1)$$

$$\Delta\rho_t = \varepsilon \cdot \delta\rho_t \qquad (2)$$

$$p_t = sigmoid(\rho_t / \zeta) \qquad (3)$$

In Eq. (1), the δ force for updating the internal values of the PB nodes p_t is obtained from the summation of two terms. The first term represents the delta error, δ_t^{bp} back propagated from the output nodes to the PB nodes: it is integrated over the period from the $t–l/2$ to the $t+l/2$ steps. Integrating the delta error prevents local fluctuations in the output errors from significantly affecting the temporal PB values. The second term is a low-pass filter that inhibits frequent rapid changes of the PB values. Internal value ρ_t is updated using the delta force, as shown in Eq. (2). The k_{bp}, k_{nb}, and ε are coefficients. The current PB values are obtained from the sigmoidal outputs of the internal values. After learning the sequences, the RNNPB model generates a sequence from the corresponding input PB values.

The RNNPB model can be used for recognition processes as well as for sequence generation processes. For a given sequence, the corresponding PB value can be obtained by using the update rules for the PB values without updating the connection weight values. This inverse operation for generation is regarded as recognition.

IV. Implementation of Quasi-Symbols into Human-Robot Interactions

A. Design of Quasi-symbols and Interaction

We designed the quasi-symbols using the PB values of the RNNPB model used in previous research [8], since original PB nodes output analog values. However the characteristic of each value is similar to the symbol and/or word in the sense that it represents the event (attractor dynamics) in the actual environment. Therefore, we generate the quasi-symbols by binarizing the PB values with a threshold. Each PB value is translated into 0 or 1 using a threshold of 0.5. For example, if the RNNPB model has two parametric nodes, it has 4-bit representation. This means that the more PB nodes the RNNPB model has, the more precise the dynamics of the environment the quasi-symbols encode. A detailed analysis of this interesting relationship between the number of the PB nodes and the number of variations in the meaning of the quasi-symbols will be presented in subsection V-D.

The user guesses the relationship between these symbols and the actual robot motions. He or she then adjusts the PB values by using speech commands corresponding to the quasi-symbols so as to move the robot as desired. The speech commands were the numerical numbers corresponding to quasi-symbols in our experiments. The PB value is then switched to the value corresponding to the command input. This process is implemented by modifying Eq. (1) to

$$\delta\rho_t^i = k_{bp}\cdot\sum_{t-l/2}^{t+l/2}\delta_t^{bp^i} + k_{nb}(\rho_{t+1}^i - 2\rho_t^i + \rho_{t-1}^i) + k_{input}\cdot\eta^i \quad (4)$$

where, η^i is either +1 or –1 depending on the input, and k_{input} is the influence degree of the effect.

C. Robovie-IIs and Experimental Environment

We used a modified version of the Robovie-IIs humanoid robot [12] as the platform for our experiments. Robovie-IIs itself is a refined version of Robovie-II, which was developed at ATR [13]. The original Robovie-II has three degrees of freedom (DOF) to control its neck and four to control each arm. It also has two CCD cameras on its head. Robovie-IIs has tactile sensors in soft silicon covering its entire body. Furthermore, we added functions: two external ears on its head and two 1-DOF hands.

During the experiment, the robot moved its head so as to capture its right-hand by forward control with the joint angles of the arm. We used two computers: one in the robot and used mainly for motion control, and one outside the robot and used for voice recognition, image processing, and neural network training. The sensory data collected were the area ratio of each color (red, blue, green, and white) captured by a CCD camera with a resolution of 320 x 240 pixels (four dimensions) and the joint angles of the right arm and head (four dimensions). They were normalized ([0-1]) and synchronized (10 frame/s) for use by the RNNPB model.

A system diagram is shown in Figure 2. The outside computer continuously recognized the situation, and calculated the PB values in real time during the interaction. When the change values of all PB nodes fell below the threshold, the robot judged that the command motion had ended. It then stopped the arm, turned its face to the user, and waited for the next command.

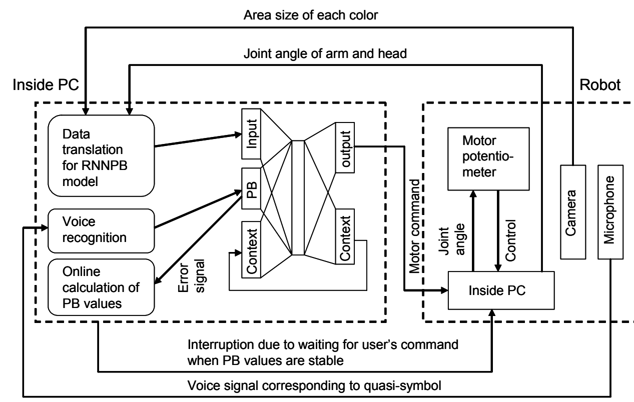

Fig. 2 System Diagram

V. Experiments and Results

A. Task Design and RNNPB Model Configuration

We carried out an experiment in which the robot moved its right hand to one of four areas in turn on the table; the areas were marked red (R), blue (B), yellow (Y), and white (W). Figure 3 shows the actual image of the experiment in progress. Four motion sequences were used as training data.

Sequence 1: Red > White > Blue > White (50 step length)
Sequence 2: Red > White > Blue > Yellow (68 step length)
Sequence 3: Red > White > Yellow > Red (67 step length)
Sequence 4: Red > White > Yellow > Blue (74 step length)

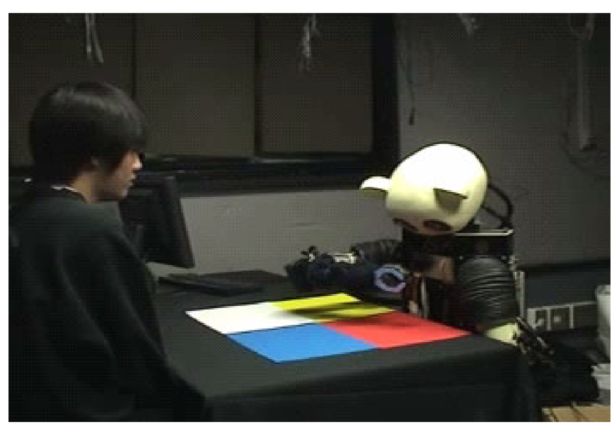

Figure 3 Image of experiment in progress

The RNNPB model used had 8 neurons in the input/output layer, 40 in the middle layer, and 10 in the context layer. The number of PB nodes was 1, 2, 3, 4, or 5.

B. Result of the Articulation

Figure 4 shows examples of the articulation by the RNNPB model. The top graph shows the sensory flow of the robot; it includes the size of each color area and the joint angles.

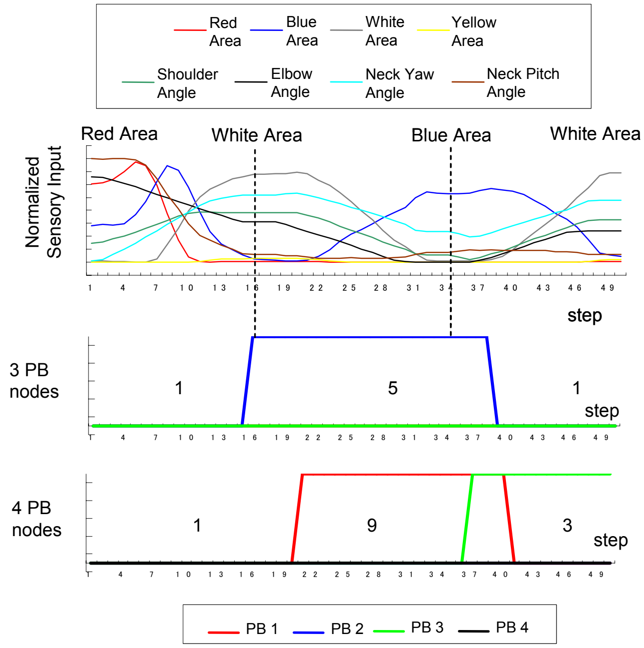

Figure 4 Example articulation of sensor flow by RNNPB models

Two vertical dotted lines show the steps at which the robot hand reached the white and blue areas respectively. Since each sensor input changed continuously and irregularly, it is difficult to estimate the exact position/condition of the robot's hand from this graph.

The two bottom graphs show the output of the PB value *after binarization* for the RNNPB model with three and four PB nodes. The symbol number output is also indicated on each graph; for example, four PB node model changed 1 ("0001"), 9 ("0101"), and 3 ("0011") in the bottom graph corresponding to the dotted lines in the top graph. That is, the RNNPB model segmented and labeled complex sensory data to fit our intuition.

C. Human-Robot Interaction

We carried out several experiments on human-robot interaction using our quasi-symbols. Here we describe an example. The operator gave the robot three speech commands for controlling its right arm. Figure 5 shows the joint angles of the robot arm and the PB outputs *before binarization* during the interaction.

The initial position of the robot hand was on the red area. The operator spoke 'one' corresponding to a quasi-symbol '1', which encodes the movement from the red to the white area at step 1 in the graphs. This command input changed the PB values as specified in Eq. (4). However, the shift in the values was not smooth - they showed complex fluctuations typical of dynamical systems – as they converged to the next condition. The vertical lines at steps 31 and 61 indicate the time at which the operator produced commands '9' and '3' respectively. Again, the PB values showed complex fluctuations.

Although the interactions in our experiments seemed to be symbolic processes from the viewpoint of the operator, there are no explicit symbolic processes in our system, as shown by the bottom graph in Figure 5. The PB values encoding the motion dynamics enabled the operator and robot to interact using symbols without the use of explicitly designed dialog patterns.

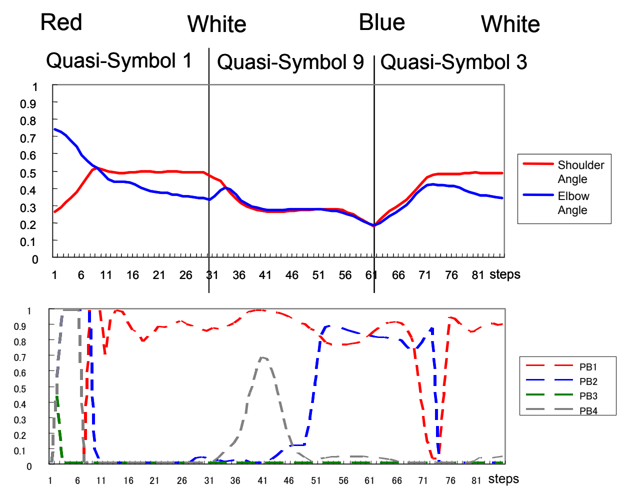

Figure 5 Fluctuation PB values during the interaction

D. Hierarchical Structure of Quasi-Symbols

As mentioned in section II, there is an interesting relationship between the number of PB nodes and the number of variations of the quasi-symbols. For example, Figure 4 shows the articulation results when there were three and four PB nodes.

The RNNPB model with four nodes distinguished the "red to white" part (and labeled '1') from the "blue to white" part (and labeled '3'). The one with only three nodes could not discriminate them and labeled them both '1'. We investigated the relationship between the number of PB nodes and the number of variations of the quasi-symbols. Our findings are represented by the hierarchical tree structure shown in Figure 6. In this structure, the movement from the white area to the blue area, for example, is indicated "W – B."

As shown in Figure 6, the variation in the number of meanings of the quasi-symbols increased with the number of nodes. For example, the model with only one node divides the condition into two states: 'right to left' motion and 'left to right' motion. However, the number of the quasi-symbols did not always increase according to the increase of nodes. For example, although the "W - B" movement appeared in the model with two nodes, it remained even when the number of nodes was increased. The division property of each quasi-symbol was determined by the context. If it appeared in the training tasks in various contexts, it was stable and remained even when the number of nodes was increased. On the other hand, if it appeared in only a few contexts, its meaning easily changed with changes in the order of hand positions, motions, and postures.

The "subdivision" in Figure 6 indicates the motion in too short time for the operator to catch. Since the time length of subdivision quasi-symbols is about 0.5 sec on average, ignoring them was not a problem for robot operation.

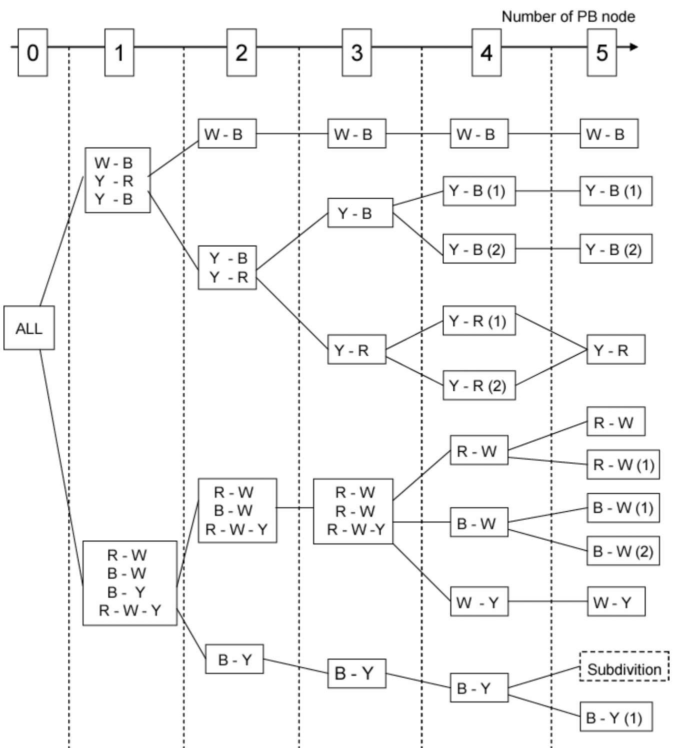

Figure 6 Hierarchical tree showing relationship between number of PB nodes and meaning of quasi-symbols

E. *Generation of Novel Motion*

We investigated the generation of quasi-symbols in novel situations. Figure 7 shows, for example, the trajectories generated by the speech command of quasi-symbol '1,' which encoded movement from the red to the white area in the training phase for the model with four nodes.

Even when the initial position of the hand was changed, the hand still eventually reached the center of white area. The trajectory was not always smooth like the PB fluctuations shown in Figure 5.

The key point here is that the movement from the yellow to the white area was not included in any of the training patterns (described in subsection V-A). This means that the quasi-symbol did not encode just a trajectory from the red to the white area but instead encoded convergence dynamics to the white area from all areas through the use of only four training patterns.

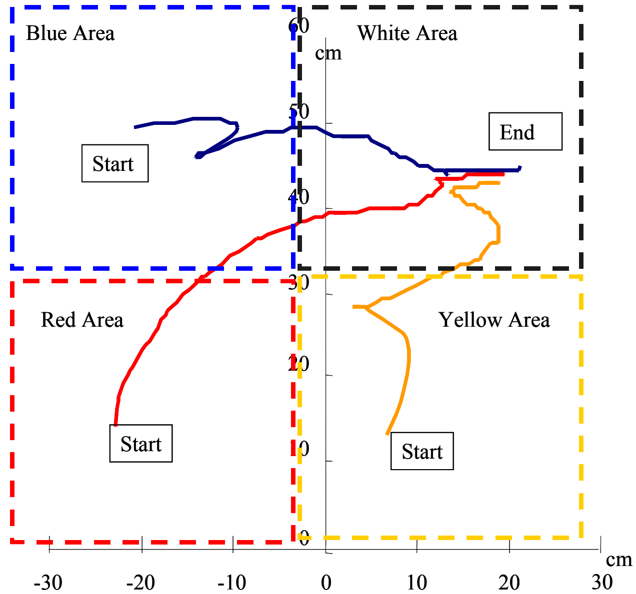

Figure 7 Three motion trajectories generated by quasi-symbol '1' (R - W) from different initial positions

F. *Self-Organizing of Meaning in Unused Quasi-Symbols*

Some quasi-symbols were not used for training the motion patterns. For example, although the RNNPB model with three PB nodes can have eight quasi-symbols, it used only five (see Figure 6). We investigated the motions when the operator gave speech commands using these unused quasi-symbols. Most of them did not encode significant dynamics with motions. However some of them self-organize interesting meanings like convergence movement to specific positions. Table 1 shows, for example, the motions generated by the RNNPB model with three nodes for used and unused quasi-symbols.

Table 1 Generated motions by used and unused quasi-symbols

	(PB1,PB2 ,PB3)	Motion in Training Phases	Initial Position: Red	Initial Position: White	Initial Position: Yellow
Quasi-Symbol 2	(0,0,1)	Yellow to Blue	Stay Red	Move to Blue	Move to Blue
Quasi-Symbol 3	(0,1,0)	White to Blue	Move to Blue	Move to Blue	Stay Yellow
Quasi-Symbol 4	(0,1,1)	<u>*None*</u>	Move to Blue	Move to Blue	Move to Blue

Quasi-symbol 2, expressed as '0, 0, 1,' encoded the movement from yellow to blue. Quasi-symbol 3, expressed as '0, 1, 0,' encoded the movement from white to blue. Both symbols represent convergence dynamics to the blue area. Although quasi-symbol 4, expressed as '0, 1, 1,' was not in the training data, it encoded the convergence dynamics to the blue area as shown by the bottom row of the table. This is because this vector is similar to the other two vectors. This is quite interesting because it shows that quasi-symbols can emerge novel significant meaning even if they are not used in the training.

VI. Summary and Future work

We have described a means of human robot interaction based not on natural language but rather on "quasi symbols," which represent the sensory-motor dynamics in the task and/or environment. As the quasi symbols, we use motion primitives, which correspond to attractor dynamics of the sensory-motor flow. We use the recurrent neural network with parametric bias model to extract these primitives from observed data. Binary representations using the model parameters were implemented as quasi symbols in a humanoid robot, Robovie. The experimental task was robot-arm operation on a table. The quasi-symbols acquired by learning enabled the robot to produce novel motions. A person could control the arm through speech interaction using these quasi-symbols. These quasi-symbols formed a hierarchical structure corresponding to the number of nodes in the model.

These quasi-symbols are not regarded as pure logic-symbols of course. They simply encode boundary conditions for the motion dynamics. While we could handle some of them as if they were actual symbols, quite complex phenomena were actually produced in a dynamical system in which a neural network, robot hardware, and experimental environment were coupled. The meaning of some of the quasi-symbols depended on the context, such as the initial hand position. Even the meaning of natural language in daily conversation often depends on the communication context, so it can be said that these quasi-symbols are useful for human-robot interaction.

An interesting challenge for future work is to connect quasi-symbols to natural language. This is impossible if natural language is a pure stochastic system that can be described by only HMM or Bayesian methods. However, Sugita et al. proposed an attractive technique [14] binding two RNNPB models, which is trained with the sensory-motor data and the English sentences. Here, the language is also regarded as a dynamical system in the sense of Elman's studies [15]. We plan to implement the system to a real communication robot and investigate its characteristics.

Acknowledgment

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (A) (No. 17680017, 2005-2007), and Kayamori Foundation of Informational Science Advancement.

References

[1] S. Harnad, The symbol grounding problem. *Physica D*, *42*, 335–346, 1990.

[2] T. Kohonen, "Self-Organizing Maps," *Springer Series in Information Science, Vol. 30*, Springer, Berlin, Heidelberg, New York, 1995.

[3] T. Inamura, I. Toshima, H. Tanie and Y. Nakamura, "Embodied Symbol Emergence based on Mimesis Theory", *International Journal of Robotics Research*, Vol.23, No.4, 2004.

[4] M. Okada, Yoshihiko Nakamura, "Design of the Continuous Symbol Space for the Intelligent Robots using the Dynamics-based Information Processing", *IEEE ICRA2004*, pp.3201-3206, New Orieans, LA, April, 2004.

[5] T. Ogata, M. Matsunaga, S. Sugano, and J. Tani, "Human Robot Collaboration Using Behavioral Primitives," *IEEE/RSJ IROS 2004*, pp. 1592-1597, 2004.

[6] A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi, "Imitating Human Dance Motions through Motion Structure Analysis," Proc. IROS 2002, pp. 2539-2544, 2002.

[7] H. Kawashima and T. Matsuyama, "Multiphase Learning for an Interval-based Hybrid Dynamical System", *IEICE Trans. Fundamentals*, Vol.E88-A, No.11, pp.3022-3035, 2005.

[8] M. Haruno, D. Wolpert, and M. Kawato, "MOSAIC model for sensorimotor learning and control," *Neural Computation* 13, pp. 2201-2220, 2001.

[9] J. Tani and M. Ito, "Self-Organization of Behavioural Primitives as Multiple Attractor Dynamics: A Robot Experiment," *IEEE Transactions on SMC Part A*, Vol. 33, No. 4, pp. 481-488, 2003.

[10] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *Eighth Annual Conference of the Cognitive Science Society* (Erlbaum, Hillsdale, NJ), pp. 513-546, 1986.

[11] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by error propagation," *in D.E. Rumelhart and J.L. McLelland, editors, Parallel Distributed Processing (Cambridge, MA: MIT Press)*, 1986.

[12] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, "Robovie: an interactive humanoid robot." *Int. Journal of Industrial Robotics*, Vol. 28, No. 6, pp. 498-503, 2001.

[13] T. Miyashita, T. Tajika, K. Shinozawa, H. Ishiguro, K. Kogure and N. Hagita, "Human Position and Posture Detection based on Tactile Information of the Whole Body," *IEEE/RSJ IROS 2004 Work Shop*, 2004

[14] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes", *Adaptive Behavior*, Vol.13, No.1, pp.33-52, 2005.

[15] J. Elman, Finding structure in time. *Cognitive Science*, *14*, 179–211, 1990.