Full Paper

Inter-modality Mapping in Robot with Recurrent Neural Network

Tetsuya OGATA[1)] Shun NISHIDE[1)] Hideki KOZIMA[2)]
Kazunori KOMATANI[1)] Hiroshi G. OKUNO[1)]

1) Graduate School of Informatics, Kyoto University
Yoshida-honmachi Sakyo-ku, 606-8501 Kyoto, Japan
{ogata, nishide, komatani, okuno}@i.kyoto-u.ac.jp

2) School of Project Design, Miyagi University
1 Gakuen, Taiwa-cho, Kurokawa-gun, 981-3298, Miyagi, Japan
xkozima@myu.ac.jp

Keywords:

Dynamical Systems, Inter-modal Mapping, Recurrent Neural Network with Parametric Bias, Generalization

1st Revised manuscript submitted on July 8, 2009

2nd Revised manuscript submitted on February 12, 2010

Figures

Abstract

A system for mapping between different sensory modalities was developed for a robot system to enable it to generate motions expressing auditory signals and sounds generated by object movement. A recurrent neural network model with parametric bias, which has good generalization ability, is used as a learning model. Since the correspondences between auditory signals and visual signals are too numerous to memorize, the ability to generalize is indispensable. This system was implemented in the “Keepon” robot, and the robot was shown horizontal reciprocating or rotating motions with the sound of friction and falling or overturning motion with the sound of collision by manipulating a box object. Keepon behaved appropriately not only from learned events but also from unknown events and generated various sounds in accordance with observed motions.

1. Introduction

Various kinds of robot systems that interact with humans have recently received a great deal of attention, represented by increased interest in humanoid robots [1, 2], particularly human assistance robots. These robots have to react to multi-modal sensory inputs in order to execute tasks and communicate with human operators. Most humanoid robots developed so far handle the sensory data from different modes independently. After information processing for each modality, the results are synchronized and integrated, a process that is quite difficult to design. An alternative approach is for the robot to handle all the data simultaneously, which is the approach we have taken.

People deal with “cross-modal information” by, for example, expressing auditory information (e.g. sounds of collision) by using visual expressions like gestures (e.g., moving the hand quickly and stopping it sharply). These gestures are apparently related to the development of onomatopoeia [3]. We call this process “inter-modality mapping.”

Arsenio and Fitzpatrick proposed an interesting method for object recognition using “periodic dynamics” in multi-modal information [4]. Using this method, a humanoid robot called Cog recognizes objects by coupling data from different modes. For example, a hammer is recognized

from its visual image, ringing sound, and hitting motion. The crucial concept of this method regards recognition as the extraction of common dynamics from multi-modal sensory information. However, the targets are restricted to rhythmic patterns.

Our ultimate goal is to design and implement a method for inter-modal mapping. The method should enable a robot to generate motion from various types of sound signals and to generate sound appropriate to various types of images. Such a method should lead to various interesting findings in the field of cognitive sciences.

Section 2 presents our model of inter-modality mapping—a robot acquires the relationships between different items of modal information by observing various events. Section 3 introduces the neural network model used for association/translation between inter-modalities and for the generalization of multi-modal sensory dynamics obtained from observation experience. Section 4 describes the implementation of our system in a small robot called Keepon. Section 5 presents the experimental results for inter-modality mapping, and Section 6 discusses the generalization ability of our method on the basis of the results of experiments using environmental sound. Section 7 summarizes the key points and mentions future work.

2. Model of Inter-modality Mapping

As mentioned above, conventional robot systems typically process sensor modalities separately. However, various modes of sensory information are usually received simultaneously. We have developed a procedure for interpretation of inter-modality mapping that is divided into two main phases, a “learning phase” and a subsequent “interaction phase.”

a) Learning Phase (“Looking at sound source”)

In the learning phase, the robot observes an event that can have various kinds of sound, such as a bouncing sound, a friction-induced sound, a continuous sound, or a rhythmic sound (See Fig. 1). The robot memorizes these sounds along with the motions of the sound source. We call this the “robot looking at sound source” phase.

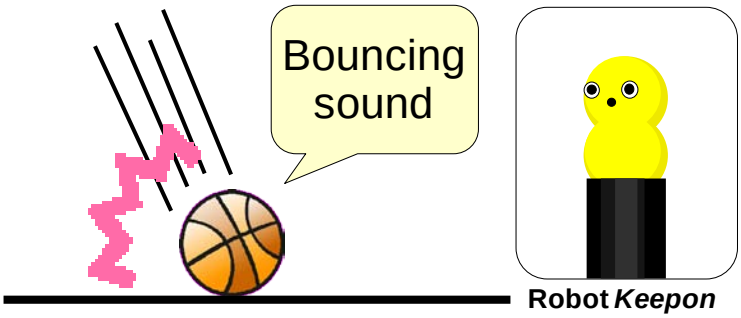


Figure 1 Learning phase: robot looking at sound source

b) Interactive Phase (“Mapping from sound to motion”)

In the interactive phase, the sensory information from a single modality (image or sound) is input into the robot’s system. The robot associates this information with the information from the other modalities and expresses it by, for example, moving its body to create the same motion as the sound source (Figure 2). Conversely, the robot observes a motion and outputs the sound memorized for that motion (Figure 3).

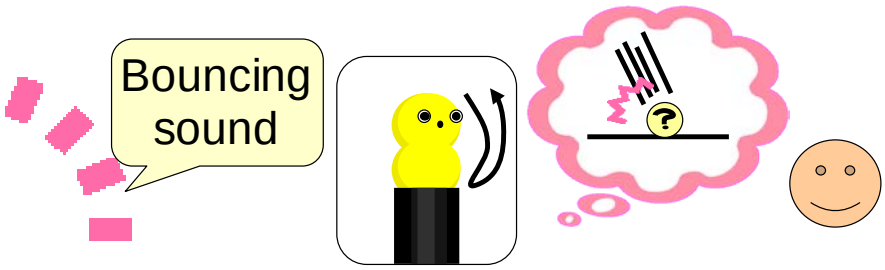


Figure 2 Interactive phase: mapping from sound to motion

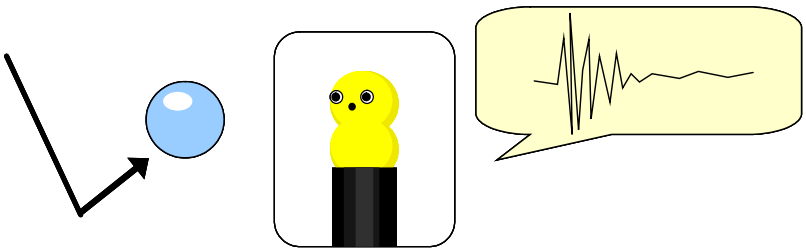


Figure 3 Interactive phase: mapping from motion to sound

3. Recurrent Neural Network Model

3.1 Introduction

There are numerous sounds in the environment around us. It is impossible to construct a database that can systematically store all environmental sounds. Therefore, to achieve inter-modal mapping, the robot must be able to generalize various sounds from a limited collection of memorized sounds. That is, the robot should be able to adapt to unknown stimuli.

To meet this requirement, we use the artificial neural network model proposed by Tani and Ito [5]. The main characteristic of this recurrent neural network model with parametric bias (RNNPB model) is that chunks of sequence patterns of the sensory-motor flow can be represented by a vector of small dimension. This vector plays the same role as bifurcation parameters in nonlinear dynamic systems. That is, different vector values result in different dynamic patterns being generated by the system. The main advantage of using parameter bifurcation is that ideally the RNNPB model can encode an infinite number of dynamic patterns with modulated analog values of the vector.

An RNNPB model is usually designed as a predictor ("forwarding forward model") for which input is current condition $S(t)$ and output is next condition $S(t+1)$. Its network has the same structure as the Jordan-type RNN [6] except that it has parametric bias (PB) nodes in the input layer (See Fig. 4). Unlike other input nodes, these PB nodes have a constant value throughout each time sequence. The context layer has a loop that inputs the current output as input data into the next step. This enables the RNNPB model to learn the time sequences on the basis of past contexts.

The RNNPB model has three activation modes: learning, recognition, and generation based on prediction. Learning is a process that modulates the network weights and PB values by using output error. Recognition is a process that inputs the whole sequence to output the PB representing the sequence. Prediction is a process that inputs a certain state of a given sequence to output the next state.

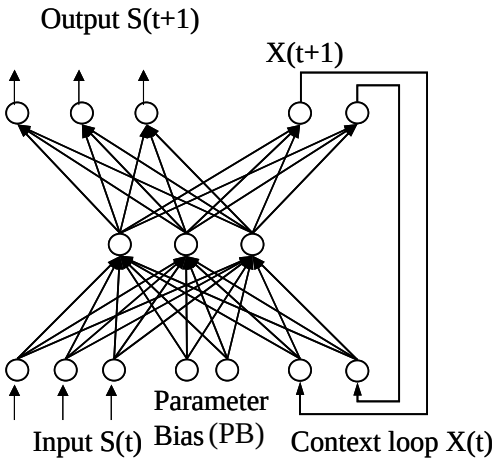


Figure 4 Network configuration of RNNPB model

3.2 Learning mode

In the learning mode, it updates its weights and the value for the "parametric bias' simultaneously using the back-propagation through time (BPTT) method [7] with prediction error. Each update is carried out using the equations below. The step length of a sequence is denoted by *l*. For each sensory-motor output, back-propagated errors with respect to PB nodes are accumulated and used to update the PB values. The update equations for the *i*th unit of the parametric bias at the *t* in sequence are

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

$$\rho_{t+1} = \rho_t + \varepsilon \cdot \delta\rho_t, \quad (2)$$

$$p_t = sigmoid(\rho_t / \zeta). \quad (3)$$

In Eq. (1), the $\delta\rho_t$ for updating the internal values ρ_t of the PB (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 in

the PB values. Internal value ρ_t is updated using the $\delta\rho_t$, as shown in Eq. (2). The k_{bp} (> 0), k_{nb} (< 0), and ε (> 0) are coefficients. The current PB values are obtained from the sigmoidal outputs of the internal values. After learning the time sequences, the RNNPB model self-organizes the PB values at which the specific properties of each individual time sequence are encoded and can generate a sequence from the corresponding PB values. Our goal is to identify the specific parameter values corresponding to each event. Therefore, to fix the parameter values during the motion recognition, parameter k_{nb} in Eq. (1) was set to 0 in our RNNPB model training:

$$\delta\rho_i = \varepsilon \sum_{t=0}^{T} \delta_t^{bp^i} \quad (4)$$

.

RNNPB model usually can acquire the generalized structure of the learning data with less than 20 training sequences. RNNPB is a predictor of which input/output are current/next states. Therefore, the RNNPB has to learn *n* input-output mappings for one training sequence of *n* steps length. That is, the RNNPB model is trained with over a hundred input-output patterns with under a hundred training sequences.

3.3 Recognition/Generation mode

In the recognition mode, the PB value corresponding to a given sequence can be obtained by using the update rules for the PB values (Eqs. (1) to (3)) without updating the connection weight values. In the generation mode, the PB value for a desired sequence is set to the PB node. The desired sequence is obtained by carrying out forwarding-forward calculation of the RNNPB.

An important characteristic of the RNNPB model is that the relational structure among training sequences can be acquired in the PB space through the learning process. This enables the RNNPB model to generate and recognize previously unseen sequences without the need for additional learning.

3.4 Modality mapping using RNNPB model

In the learning phase, discussed in Section 2-(a), a robot in which the RNNPB model has been implemented learns various events (sensor sequences) by using equations (1), (2), and (3). Here we describe how the model is used in the interaction phase discussed in Section 2-(b) by presenting an example of modality translation from sound to motion.

Figure 5 outlines the concept of the BPTT algorithm used in the RNNPB model for recognition. When the robot detects only an auditory signal ("Auditory data–Input" in Figure 5), the PB values are calculated using only the prediction error in the auditory signal. The input/output layers for the visual signal are handled the same as the context layer.

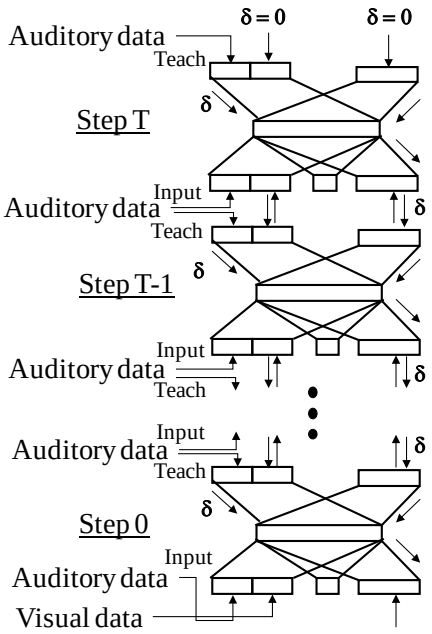


Figure 5 BPTT algorithm for RNNPB model

After the PB values are obtained, the sequence of visual signals is obtained using the following process. The PB values and the auditory signal are input into the network in each step and forwarding-forward calculation is carried out. The input/output layer for the visual signal is regarded as the context layer; so the visual sequence is obtained.

4. Implementation into Robot System

4.1 Interaction robot, Keepon

We used a robot called “Keepon” for our experiments. It was developed at the National Institute of Information and Communication Technology (NICT) mainly for communicative experiments with infants [8]. Its body is approximately 12 cm high and has four degrees of freedom, as shown in Fig. 6. It has two CCD cameras and one microphone in its head.

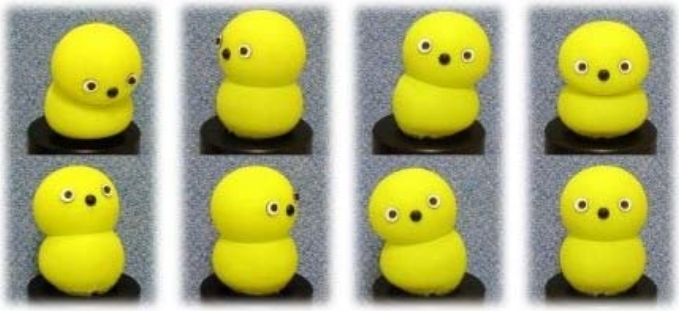


Figure 6 Robot Keepon and its motions

4.2 Audio-visual Processing

In an experiment, the robot observed events represented by the manipulation of a box by a person. The box was 165 mm long, 110 mm wide, and 33 mm high (Fig. 7) and was made of plastic. The corner positions of the box were detected during box manipulation by visual processing. We applied Kawato and Ohya’s method to our color detection [9]. The method is designed for real-time detection of the human face.

For audio processing, we used the values for the Mel filter bank multiplied by the four triangular windows shown in Fig. 8. The Mel filter bank has been shown to be effective for expressing environmental sounds [10]. The values were normalized and synchronized within 50 ms for input to the RNNPB model.

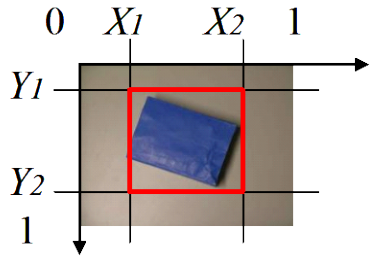


Figure 7 Box used for event observation

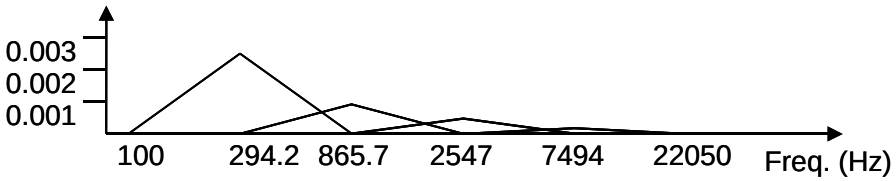


Figure 8 Four triangular windows used for audio processing

4.3 Generation Processing

An inverse translation process from the audio-visual signal obtained by the RNNPB model to actual robot motion and sound is required for modality mapping.

In the motion generation phase, Keepon uses its pitch and yaw axis to reproduce the trajectory of the blue box obtained from the model output. In the sound generation phase, Keepon outputs colored noise by multiplying white noise with the Mel filter bank value obtained from the model output. The actual frequency and time data were obtained using a linear approximation due to poor resolution in the model output.

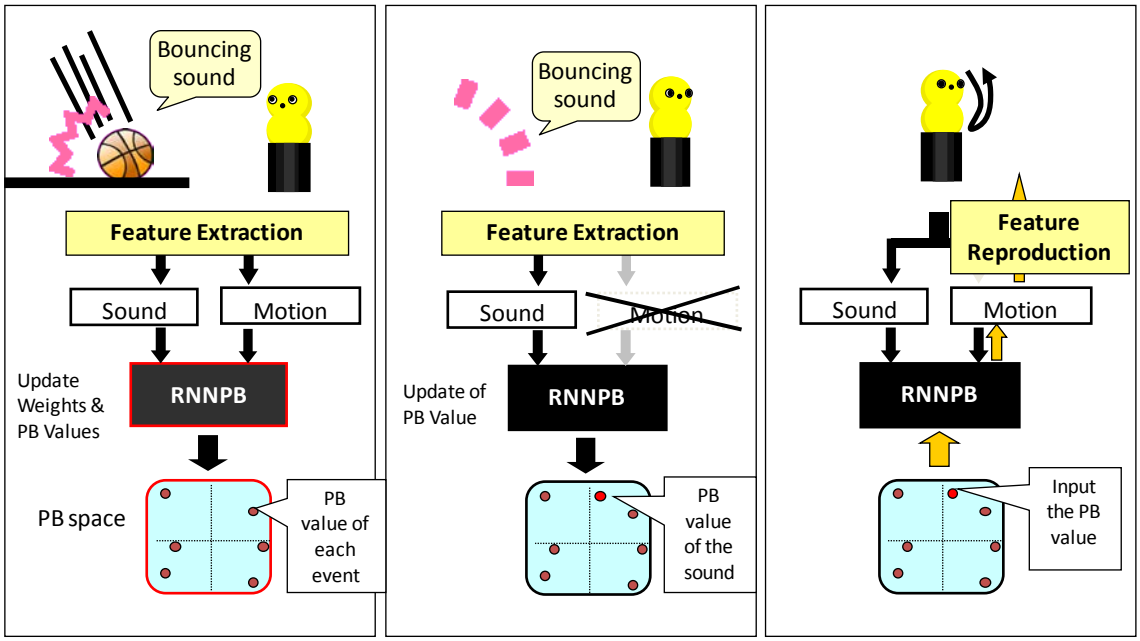


Figure 9 Illustration of multi-modal information translation by RNNPB model

Figure 9 illustrates the translation of the multi-modal sensory information. The left panel illustrates the learning phase in which the robot observes various events with various motions and sounds. The middle one illustrates the observation phase in which the robot estimates the parameters using the prediction errors of the RNNPB model for a modal sensory flow. And the right one illustrates the generation phase in which the robot reproduces the multi-modal sensory flow using the estimated RNNPB parameters.

5. Experiments

5.1 Preliminary Experiment using Environmental Sounds

To investigate the learning ability and characteristics of RNNPB, we conducted an experiment using only environmental “sounds” (Bell, CycleBell, Glass, Gun, and Spray). The acoustic signal for each sound was transformed into 12 Mel-frequency cepstrum coefficient (MFCC) features. For model training, we selected two patterns consisting of 50–90 steps for each acoustic signal.

We conducted a cross-validation experiment in a specific manner. In five-fold cross-validation

experiment, the model is usually trained on 80% of all recognition classes, and tested on other 20%. On the other hand, we use the data of four types/classes of auditory data for training neglecting one type. The performance of RNNPB is tested not only with unknown data in known class but also with "unknown data in unknown class".

The configuration of the RNNPB model was 12 input/output nodes for the MFCC features, 50 middle nodes, 40 context nodes, and 2 PB nodes. After training, the performance was evaluated using ten environmental sounds for which the model was untrained.

RNNPB model is not a recognition classifier that maps input patterns to defined classes, but is a generator that predicts patterns. The generation ability is one of most notable features of RNNPB. Therefore, we evaluated its performance from two aspects: pattern clustering and pattern generation.

5.1.1 Results for Pattern Clustering

The PB space obtained for each training result is shown in Fig. 10. Some clusters were self-organized corresponding to the environmental sounds. This clustering ability can be regarded as recognition ability if recognition classes are assigned to these clusters by existing classifiers like Gaussian Mixture Model (GMM).

(a), (b), (c), (d), and (e) each show the result when training without Bell, CycleBell, Glass, Gun, and Spray, respectively. The results shown in (a), (d), and (e) clearly show the cluster of the distribution even for the sounds of unknown type. The result in (b) also shows well formed clusters except one of the ten data for CycleBell. Although RNNPB has a limitation in scalability, it can generate novel clusters representing novel types/classes of sounds. This is a notable feature of our model. The result in (c) shows that the cluster of the distribution for the unknown Glass data could not be created well. The Glass data, glass breaking sound, differs greatly even among the same Glass category.

These results indicate that the RNNPB model can self-organize not only training types of sound but also unknown type of sound thanks to its generalization capability.

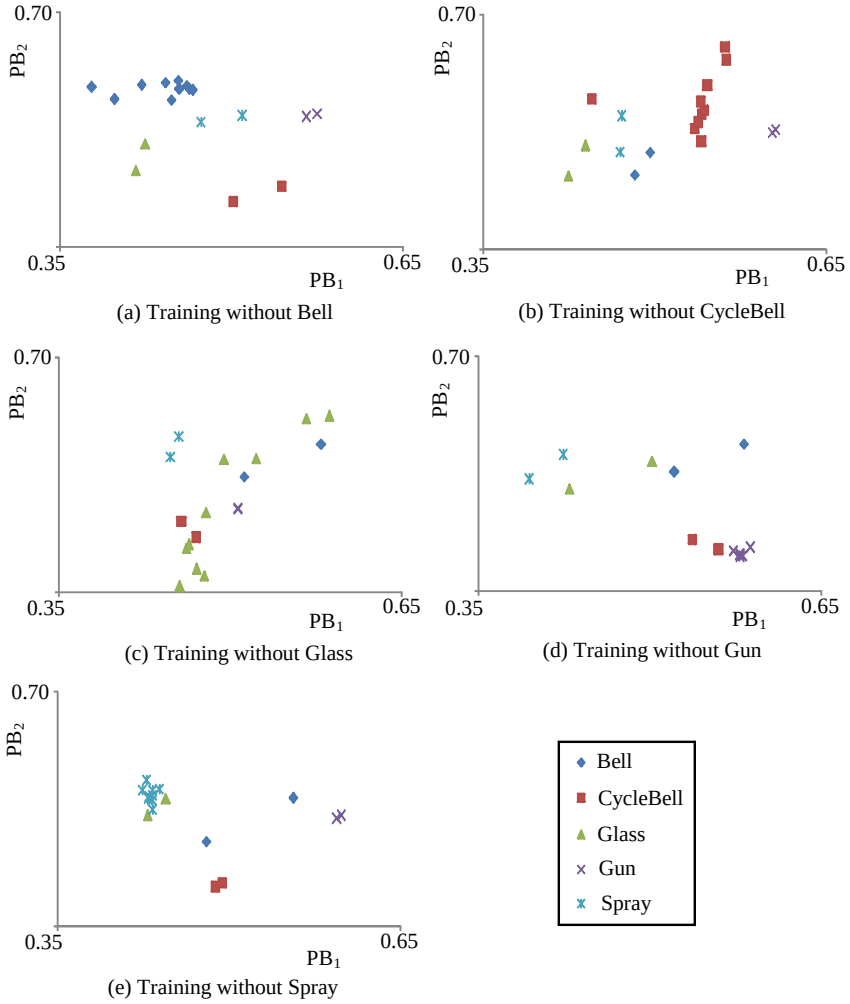
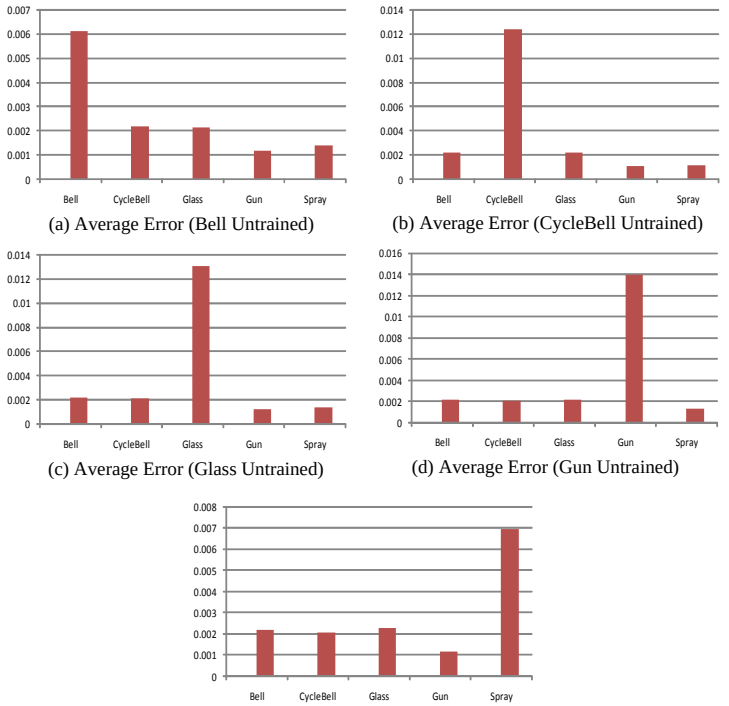


Figure 10 Self-organized PB space for five-fold cross-validation training

5.1.2 Results for Pattern Generation

The average errors for generating each training data point are shown in Fig. 11. The errors were calculated by accumulating the error for the RNNPB model for each node, step, and pattern and then dividing the total by the number of nodes, steps, and patterns. Therefore, the values shown represent the average error for one node at one step for one pattern. The average error for the untrained data was substantially larger than that for the trained data. However, the maximum error was smaller than 0.015 for one node at one step.

These results indicate that the RNNPB model can generate unknown patterns well thanks to its generalization capability.



(a) Average Error (Bell Untrained)

(b) Average Error (CycleBell Untrained)

(c) Average Error (Glass Untrained)

(d) Average Error (Gun Untrained)

(e) Average Error (Spray Untrained)

Figure 11 Open loop error for each data type for five-fold cross-validation training

5.2 Learning of Motion and Sound and Acquired PB Space

In the learning phase, we had Keepon observe four kinds of manipulations of the blue box with different types of sound. They were 1) rotating on a wall with the sound of continuous friction, 2) reciprocating on a table with sound of periodic friction, 3) overturning on a table with the sound of collision, and 4) falling to a table with the sound of collision. Keepon observed each event three times (4 x 3 = 12 sequences in total), and the RNNPB model was trained using the data collected. The model network consisted of 8 neurons in the input layer, 35 neurons in the middle layer, 25 neurons in the context layer, and 2 neurons as the parametric bias. The training sequence for the model was segmented when the variation of all sensory inputs were less than a threshold. The event lengths were 10 to 40 steps (0.5–2 s).

Figure 12 plots the acquired PB space. The two parametric values correspond to the X-Y axes. The reciprocating motions were mapped in the upper area, the rotating motions were mapped in the

left area, the overturning motions were mapped in the left-bottom area, and the falling motions were mapped in the right area. The areas for the rotating and overturning motions were close together because the overturning motion can be regarded as a type of rotating motion.

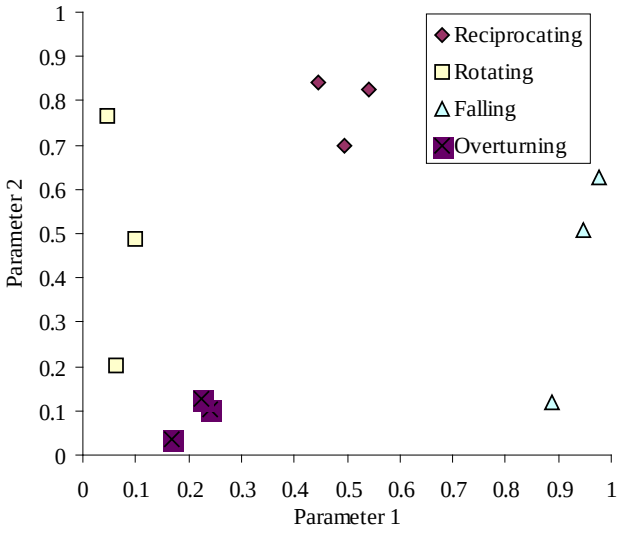


Figure 12 PB space acquired in learning phase

5.2.1 Mapping from Sounds to Motions

Figure 13 plots the sound recognition results for the RNNPB model. Though the circled plot points denote same four kinds of events described in previous section, these were not used to train the model. That is, these plots show the recognition results of untrained sounds in known categories. We also investigated the PB values corresponding to completely novel sounds randomly generated a few times by 1) spraying, 2) clapping, and 3) plastic bag shaking. The PB values corresponding to these events are also plotted in Fig. 13.

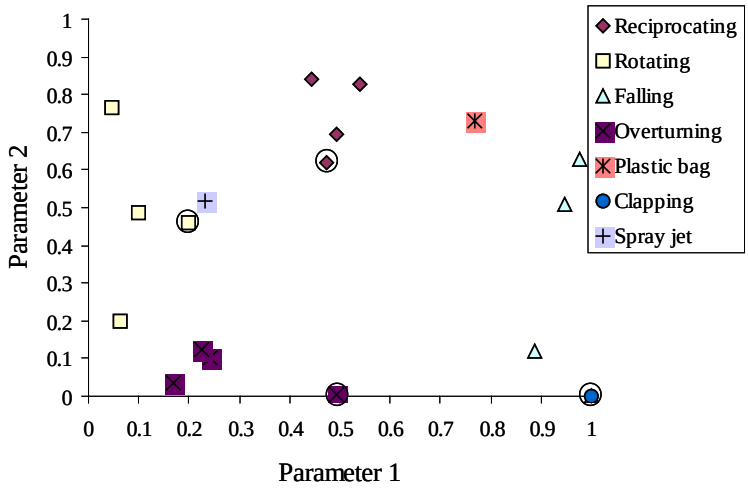
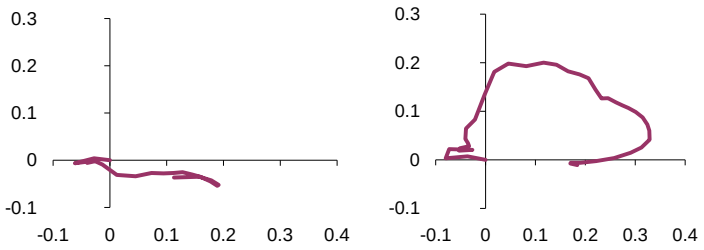
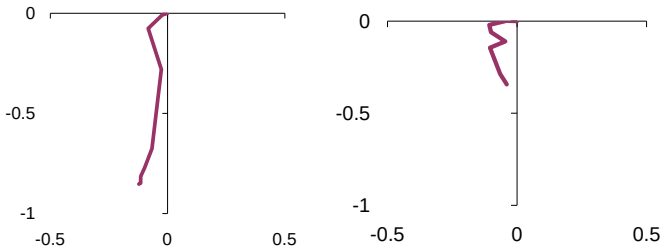


Figure 13 Sound recognition results

Figure 14 plots the motion trajectories corresponding to 'known' events. Keepon generated motions (trajectories and velocities) that were similar to the manipulations of the blue box in the learning phase.



1) Reciprocating (friction) 2) Rotating (friction)



3) Falling (collision) 4) Overturning (collision)

Figure 14 Trajectories generated by sounds corresponding to four events

(horizontal axis shows normalized degree of pan axis; vertical axis shows tilt axis)

Figures 15, 16, and 17 show the signals of the novel sounds and plots of the generated motions. The motion generated for the clapping sound was similar to that for the collision sound possibly because they were common in the sense they were “collision sounds.” Similarly, the motion generated for the spraying sound was similar to that for the rotating sound possibly because they were common in the sense they were “continuous friction sounds.”

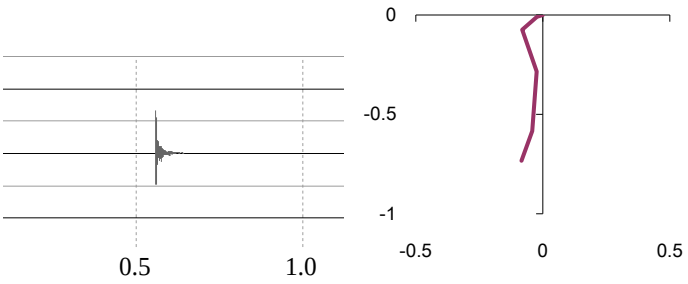


Figure 15 Sound signal of clapping and plot of generated motion

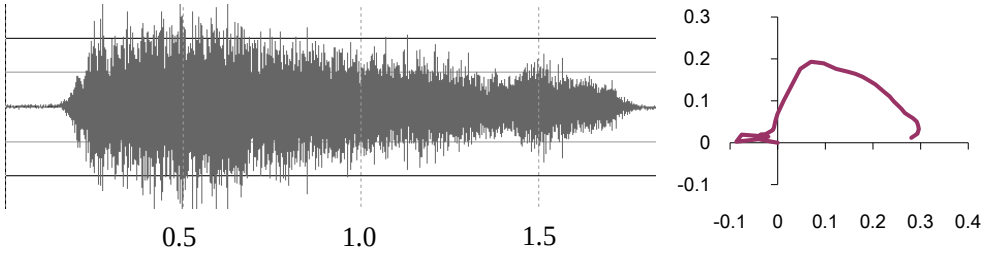


Figure 16 Sound signal of spraying and plot of generated motion

Note that the motion generated from the sound of the plastic bag shaking was not as simple as the previous two examples, as shown in Fig. 17. This sound contained not only friction and collision sounds but had various other features. The motion reflected the complex characteristics of shaking up and down and small rotations. The ability of the RNNPB model to generalize enables such novel motion patterns to be generated from novel sounds.

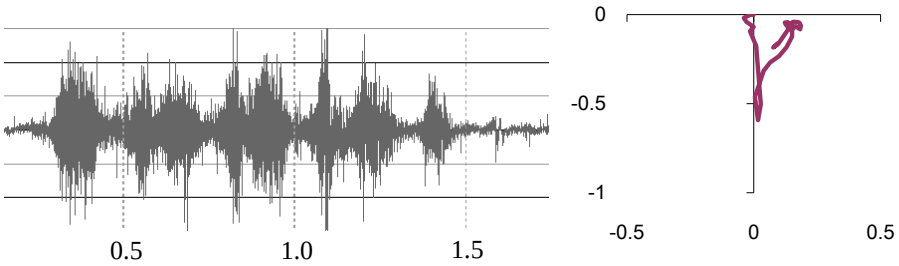


Figure 17 Sound signal of plastic bag shaking and plot of generated motion

5.2.2 Mapping from Motions to Sounds

Figure 18 plots the motion recognition results for the RNNPB model. Although the circled plot points in the PB space denote the same four kinds of events described in Section 5.2, these were not used to train the model. That is, these plots show the recognition results of untrained motions in known categories. We also investigated the PB values corresponding to novel motions without sound: 1) quickly sliding the box horizontally, 2) slowly sliding the box horizontally, and 3) shaking the box up and down a few times. The PB values corresponding to these events are also plotted in Fig. 18.

Figure 19 shows the sounds generated corresponding to the four ‘known’ events. Keepon generated sounds that were almost the same as those in the learning phase. The reason for the two power peaks in 2) and 4) is that the RNNPB model also learned the rebound sound from the falling event.

Figures 20, 21, and 22 show the sound signals generated for the novel motions. The sound generated for quickly sliding was similar to that for falling although no sound had been given for quickly sliding. This may be because they are common in the sense that they are both “collisions”. Similarly, the sound signal generated for slowly sliding was similar to that for rotating possibly because they are common in the sense that they are “continuous motions.” The sound generated for shaking differed from that for the reciprocating used for training (Fig. 19). It was synchronized with the observed motions.

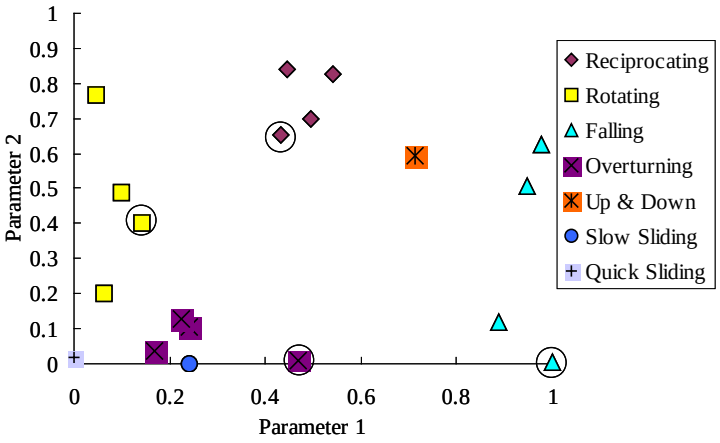
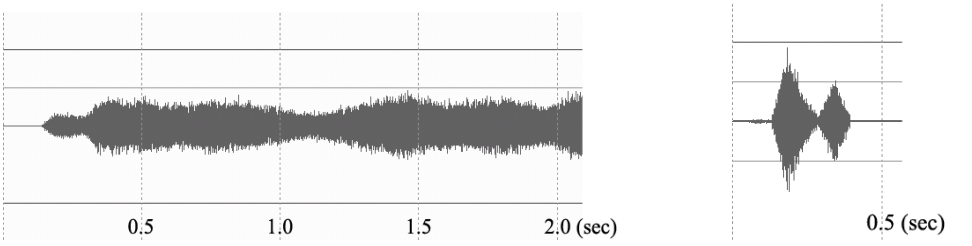
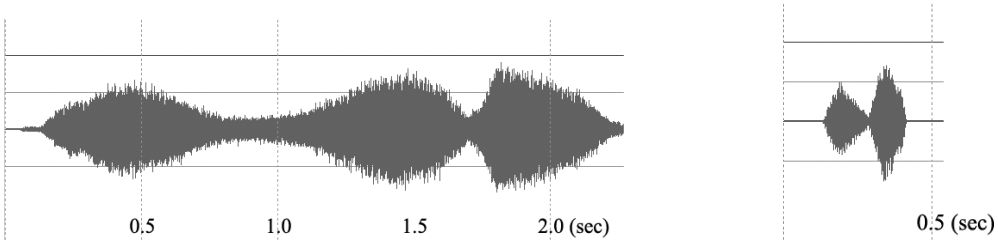


Figure 18 Motion recognition results



1) Rotating

2) Falling



3) Reciprocating

4) Overturning

Figure 19 Sound signals generated by four known motions

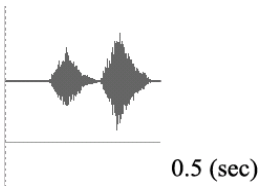


Figure 20 Sound signal generated by quickly sliding motion

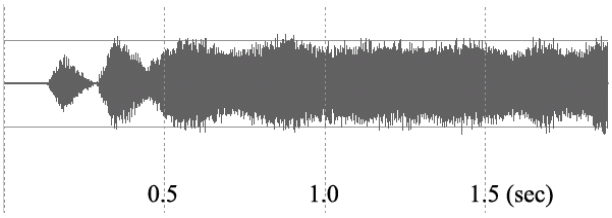


Figure 21 Sound signal generated by slowly sliding motion

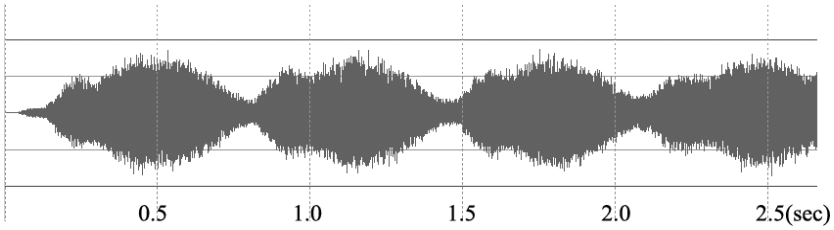


Figure 22 Sound signal generated by shaking motion

6. Discussion

6.1 Technique for Learning Temporal Sequence Patterns

Various techniques are available for dealing with time sequence patterns. One representative method is the temporal difference (TD) scheme of Barto et al. [11]. This scheme represents temporal sequences by using automatons in a reinforcement learning manner. Most TD frameworks deal with discrete actions. When the action space is discrete, the implementation of reinforcement learning works well. However, this approach is not well suited for smooth motion, such as object dynamics, because dividing such a motion into a set of states is difficult. Somewhat modified methods without quantization [12] are needed for dealing with continuous motions. In fact, the RNNPB model has been applied to a TD scheme dealing with continuous motor sequences [13].

A TD scheme represents temporal sequences using discrete states while the RNNPB model represents them as dynamic systems. Because our objective, modality mapping, includes the actual physical dynamics of objects, we used the RNNPB model for the mapping. The generalization ability of the RNNPB model is a unique feature. It enables, for example, Keepon to associate quite complex motions to unknown sounds, like the sound of a plastic bag being shaken.

6.2 Design of PB Nodes and Scalability of Learning Data

This section discusses the design of the RNNPB network, such as the size of the PB nodes and the scalability of the learning data. The RNNPB model can handle multiple visual-auditory events in the PB space self-organized with a few examples.

There is another deterministic learning system called “mixture of experts” [14] that also handles multiple dynamic patterns (attractors) well. Deterministic learning systems usually consist of several dynamic predictors that learn target sequences in parallel. We call this architecture “local expression.” In contrast, the RNNPB model acquires multiple attractors in a single network by changing the parameters that represent the bias condition. We call this architecture “distributed expression.” In local expression, interference between patterns is not a serious problem because the network allocates a novel pattern to an additional predictor. In a distributed expression, memory interference occurs since the memories share the same network resources.

The RNNPB model has an advantage in terms of generalization of not only unknown data but also unknown classes, while scalability of learning data is an issue. As a result of multiple attractors being encoded in a distributed network, a global structure can be acquired using only a few learning patterns. For example, in the experiment described in section 5.1, the RNNPB model acquired a global space for five types of environmental sounds by using only eight sounds. On the other hand, the RNNPB model has a disadvantage in terms of learning data scalability. The learning capability could be improved by increasing the number of PB nodes. However, the effect is not significant compared to the addition of a predictor in local expression. Although there was a study investigating the relationship between the number of PB nodes and the learning capability [15], it is still an open problem. The number of PB nodes (two) was determined heuristically in our experiments.

7. Summary and Future Work

A method has been described for mapping between different sensory modalities that will enable a robot system to generate motions expressing auditory signals or sounds on the basis of the movements of objects. Since complete memorization of the correspondences between auditory

signals and visual signals is practically impossible, the ability to generalize is indispensable. We developed a neural circuit model called the “recurrent neural network model with parametric bias” (RNNPB model), which has good generalization ability, for use as the learning model.

We implemented this model in the “Keepon” robot. Keepon was then shown horizontal reciprocating and rotating motions with the sound of friction and falling or overturning motions with the sound of collision by manipulating a box object. Keepon behaved appropriately not only for learned events but also for unknown events. It also generated sounds appropriate for observed motions. We also conducted a cross-validation of RNNPB model with five types of environmental sounds. The results show that the obtained PB space can represent not only known types of sounds but also unknown types of sounds.

An interesting challenge for future work is to apply our method to a humanoid robot with many degrees of freedom. One crucial problem is how to select the joints for use in expressing the sounds; the joints used for Keepon were selected in advance. Human infants learn which muscles to move to achieve a particular goal by using a process called “body babbling.” This process enables infants to acquire the mapping between movement and a particular body-part configuration. We plan to introduce such a process into a future humanoid robot.

Acknowledgments

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

References

[1] http://www.honda.co.jp/ASIMO/

[2] Ishiguro, H., Ono, T., Imai, M., Maeda, T., Kanda, T., and Nakatsu, R. (2001), “Robovie: an

interactive humanoid robot," International Journal of Industrial Robotics, Vol. 28, No. 6, pp. 498–503.

[3] Werner, H. and Kaplan, B.: Symbol Formation: An Organismic Developmental Approach to the Psychology of Language, John Wiley and Sons, New York, 1963.

[4] Arsenio, A. and Fitzpatrick, P., "Exploiting Cross–Modal Rhythm for Robot Perception of Objects," Proceedings of the 2nd International Conference on Computational Intelligence, Robotics, and Autonomous Systems, 2003.

[5] Tani, J. and Ito, M. (2003), "Self–Organization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment," IEEE Transactions on Systems, Vol. 33, No. 4, pp. 481–488.

[6] Jordan, M. (1986), "Attractor dynamics and parallelism in a connectionist sequential machine," Proceedings of the 8th Annual Conference of the Cognitive Science Society (Erlbaum, Hillsdale, NJ), pp. 513–546.

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

[8] Kozima, H., Nakagawa, C., Yasuda, Y., and Kosugi, D. (2004): A toy-like robot in the playroom for children with developmental disorders, in Proceedings of the 8th International Conference on Development and Learning (ICDL-2004; San Diego, CA).

[9] Kawato, S. and Ohya, J. (2000): Automatic Skin–color Distribution Extraction for Face Detection and Tracking, Proceedings of the 5th International Conference on Signal Processing, Vol. II, pp. 1415–1418.

[10] Cowling, M. and Sitte, R. (2003): Comparison of techniques for environmental sound recognition, Pattern Recognition Letters, Vol. 24, No. 15, pp. 2895–2907.

[11] Barto, A. G., Sutton, R. S. and Anderson, C. W. (1983): "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Trans. Syst. Man Cybernet., Vol. 13, pp. 834–846.

[12] Doya, K. (2000): “Reinforcement learning in continuous time and space,” Neural Comput., Vol. 12, 219–245.

[13] Arie, H., Ogata, T., Tani, J., and Sugano, S. (2007): “Reinforcement learning of continuous motor sequence with hidden state,” Advanced Robotics, Special Issue on Robotic Platforms for Research in Neuroscience, VSP and Robotics Society of Japan, Vol. 21, No. 10, pp. 1215–1229.

[14] Pavlovic, V. Rehg, J. M., and MacCormick, J. (2001): “Learning Switching Linear Models of Human Motion,” Advances in Neural Information Processing Systems, MIT, pp. 981–987.

[15] Ogata, T., Matsumoto, S., Tani, J., Komatani, K., and Okuno, H. G. (2007): Human-Robot Cooperation using Quasi-symbols Generated by RNNPB Model,” Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2156–2161.