Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Experience Based Imitation Using RNNPB

Ryunosuke Yokoya†, Tetsuya Ogata†, Jun Tani‡, Kazunori Komatani†, and Hiroshi G. Okuno†
†*Graduate School of Informatics, Kyoto University, Kyoto, Japan*
{*yokoya, ogata, komatani, okuno*}*@kuis.kyoto-u.ac.jp*
‡*Brain Science Institute, RIKEN, Saitama, Japan*
tani@brain.riken.jp

***Abstract*— Robot imitation is a useful and promising alternative to robot programming. Robot imitation involves two crucial issues. The first is how a robot can imitate a human whose physical structure and properties differ greatly from its own. The second is how the robot can generate various motions from finite programmable patterns (generalization). This paper describes a novel approach to robot imitation based on its own physical experiences. Let us consider a target task of moving an object on a table. For imitation, we focused on an active sensing process in which the robot acquires the relation between the object's motion and its own arm motion. For generalization, we applied a recurrent neural network with parametric bias (RNNPB) model to enable recognition/generation of imitation motions. The robot associates the arm motion which reproduces the observed object's motion presented by a human operator. Experimental results demonstrated that our method enabled the robot to imitate not only motion it has experienced but also unknown motion, which proved its capability for generalization.**

I. Introduction

The final goal of this work was to develop a method that enabled robots to imitate human motion. Human adults can easily learn by watching the behavior of others, and imitate them. Even infants can learn through imitating facial and hand gestures. With this significant ability, human beings can acquire new behaviors from others within an incredibly short time. From the standpoint of robot learning, any method that enables robots to imitate humans can significantly speed up the learning process [1]. The learning load is crucial to real robots because of problems with durability. It is also almost impossible to program robots manually to make every conceivable motion.

With advances in hardware technologies, humanoid robots can now realize several kinds of motion: two-legged locomotion, running, and rising. Some of them have tried to imitate human motion. Nakazawa et al. developed a humanoid robot that imitates dancing using a motion capture system [2]. In their study, the robot imitated the trajectories for every part of the human body. The robot's joint angles are the almost same as humans and under control all the time. Therefore, the motors sometimes have to output extremely large torque because of the differences of the body dynamics. Thus, most conventional studies usually designed recognition process as pattern clustering, and the motion generating process was isolated from the recognition process.

For robot imitation, in this work, we focus on two factors. One is "mirror neurons" in the brain and the other is infant's "body babbling."

The mirror neurons were originally discovered in area F5 of the monkey premotor cortex, which discharge both when the monkey makes a particular action and when it observes another making a similar action [3]. The neurons suggest that both recognition and generation processes are conducted in the same structure in brain. This work uses the neural net model called the Recurrent Neural Network with Parametric Bias (RNNPB) [4] that can work as both recognition and generation functions. The detail of the model is described in section II.

The body babbling is experiential process where infants learn what muscle movements achieve a particular goal state [5]. This process enables infants to acquire a mapping between dynamic patterns of movement and a resulting body part configuration. Based on the fact, this work introduces the active sensing process as robot's experiential process where the robot acquires a mapping between their own motions and target motions based on real experiences.

Our target task is moving an object on a table. In our imitation architecture, recognition process is implemented not as the clustering of generated patterns but as the prediction of pattern generation (forward model). Based on this sense, Ogata et al. proposed the active recognition model using humanoid robot and RNNPB model [6]. The prediction of the object motion while manipulating enables the robot to generate the motion at the next moment (inverse model).

Section II describes our imitation architecture which is based on active sensing and a RNNPB. Section III describes detail implementation of the robot hardware and the neural net model. Section IV describes the imitation experiments and the obtained results. Section V discusses the prediction and generalization capabilities of our architecture as an imitation model. Section VI concludes this paper.

II. Imitation Method Based on Active Sensing

A. Overview of Our Imitation Process

Here, we present an overview of our method, which enables a robot to imitate human behaviors by using the experience of active sensing. For simplifying the verification of the effectiveness of the method, in this work, the imitation part, i.e. the trajectory of the object, is given in advance. Our imitation process consists of three phases: the learning, observation and motion-generating phases (See Fig. 1). We can overview it as follows.

1) Learning (Object Recognition)
The robot connects its arm motions and the object

1-4244-0259-X/06/$20.00 ©2006 IEEE

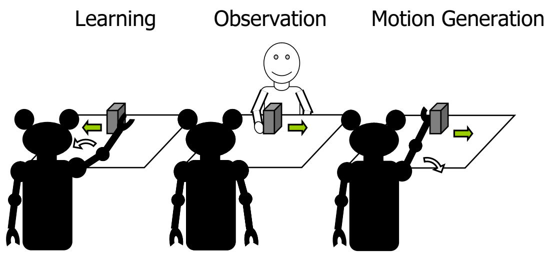


Fig. 1. Imitation process.

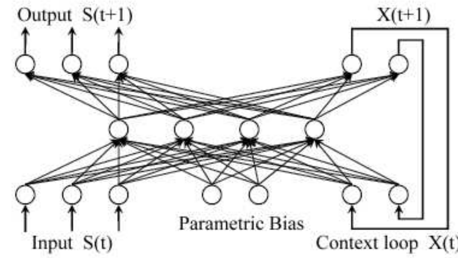


Fig. 2. RNNPB.

motions while it manipulates the object (active sensing). The experience of active sensing enables the robot to predict the object motion.

2) Observation (Motion Planning)
The robot observes a target object manipulating generated by a human teacher focusing not on the teacher's motion but on the object motion. The robot planned its arm motion which can generate the similar object motions.

3) Motion Generation (Imitation)
The robot actually generates the arm motion planned in the previous phase.

In this process, one of the problems is how appropriate motion is fixed from the object motion presented. There needs to be some kind of method to connect robot motion with object motion. The robot motion has to be generated using only limited patterns of learnable object manipulations which are limited due to real robots having problems with durability.

The RNNPB model has advantages in that it can acquire self-organized behavioral primitives as the "PB values". The most significant feature of the model is its generalization capabilities. By taking advantage of the RNNPB model, in this work, the robot motion was associated with the object motion with PB values.

B. Learning Model

This section describes the learning model used in our method, the RNNPB model, and its learning algorithm.

1) RNNPB: The RNNPB model is the FF-model (forwarding forward model) proposed by Tani and Ito. The RNNPB model works as a prediction system: its input data is current sensory state S(t) and its output data is predicted sensory state S(t+1) in the next step. The network configuration for the RNNPB model is outlined in Fig. 2. This model has the same architecture as the conventional hierarchical neural network model except for the context layer and the PB nodes in the input layer. Unlike the other input nodes, these PB nodes take a constant value throughout each time sequence. The context layer has a loop that inputs current output as input data in the next step. An advantage of this layer is that the RNNPB model can learn the time sequences taking advantage of past contexts. After learning time sequences, the RNNPB model self-organizes the PB values at which the specific properties of each individual time sequence are encoded.

The RNNPB model learns with a particular learning algorithm. Although the learning algorithm for the conventional hierarchical neural network is back propagation, the RNNPB model cannot learn with this algorithm because it does not have a teacher signal to the context layer. Consequently, a novel learning algorithm called the BPTT (back propagation through time) [7] is employed.

2) Learning PB Value: The PB values are generally calculated during the learning process as follows.

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

$$p_t = sigmoid(\rho_t), \quad (2)$$

where k_{bp} and k_{nb} are constants; ρ_t is the internal value of the PB node at t; p_t is the PB value of the PB node at t; and l is the step length of a sequence. In (1), the first term represents the delta error, δ_t^{bp}, back-propagated from the output nodes to the PB nodes; it is integrated over a period from t-l/2 to t+l/2 steps. Integrating delta error in multiple steps prevents local fluctuations in 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. In (2), the current PB values are obtained from the sigmoidal outputs of the internal values. The PB values after threshold processing can also be utilized as *quasi-symbols* for human-robot interaction [8].

In this work, the delta force of the PB values have been calculated as follows because our goal was to acquire specific PB values corresponding to each object manipulation.

$$\delta\rho_t = k_{bp} \cdot \sum_{0}^{T} \delta_t^{bp}, \quad (3)$$

where T is the sensory sequence length. In (3), the delta errors are not integrated errors in the constant steps but in all the steps.

C. Calculation in Observation and Motion Generating Phases

After the RNNPB model is organized in the BPTT and the PB values are calculated in the learning phase, the RNNPB model is used in the observation and motion generating phases. This section describes how the RNNPB model is used in the observation and motion generating phases.

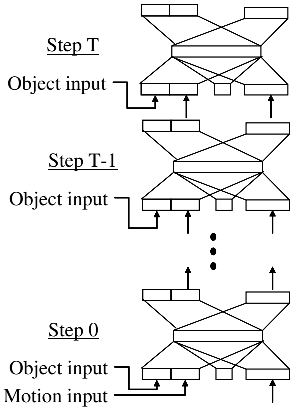


Fig. 3. Forward calculation of PB values.

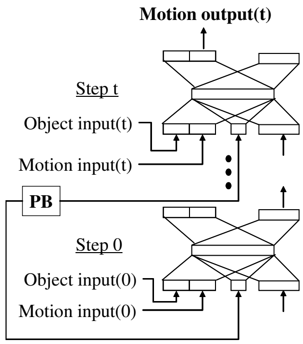


Fig. 4. Motion generation.

1) Method for Recognizing Manipulation: This section describes how the manipulation presented by the teacher in the observation phase is recognized, i.e., how the PB values in the observation phase are obtained. The PB values are calculated based on (2) and (3) by the organized RNNPB model without updating the connection weights. However, there are no arm motor data because the robot is just looking at the target, and does not move unlike in the learning phase. The initial arm motor values are then input to the motion input layer in step 0, and the outputs are calculated forward in the close looping mode from step 1; the outputs in the motion output layer in step t-1 are the input data in the motion input layer in step t (See Fig. 3). To put this simply, the motion input layer plays the same role as the context layer does.

2) Method for Generating Motion: This section describes how directive motor values transferred to the robot to move its motors in the motion generating phase are calculated (See Fig. 4). The motion output of the RNNPB model is obtained in a forward calculation. The PB values obtained in the observation phase and each item of real input data are input in real time to the RNNPB in each step. The motion output signal, the predicted directive motor value, of the RNNPB model in step t-1 is transferred to the robot as the directive motor value in the next step, step t.

III. Model and System

A. Humanoid Robot Robovie-IIs

Our experimental platform was a humanoid robot, Robovie-IIs, a refined model of Robovie-II developed at ATR [9]. Robovie has three DOFs (degree of freedom) in its neck and four DOFs in each arm. Each motor angle value is measured with potentiometers. It also has stereoscopic CCD cameras on its head. The potentiometers and the camera collected the data required for the experiment.

B. Target Object

The manipulation targets are a cylinder-shaped object and a box-shaped object. The cylinder-shaped object moves in parallel when the robot lays its hand on the low position, and it tumbles when the robot lays the hand on the high position. The box-shaped object was moved by the robot hand. The top of the object is separated into two colors, red and blue, which enable the robot to easily detect the rotation of the object.

C. Experiment System

Fig. 5 is the system diagram. The camera tracks the target object by controlling the neck motor keeping the centroid of the object centered on the camera. Since the robot is required to move in real time, the module for the moving motors has been constituted on a PC embedded in the robot, and the processes for translating data and calculating the directive motor values run on an external PC. The size of the RNNPB model differed according to experiments.

The following sensory data were collected in the experiment for use in the RNNPB model.

a) Visual Information:
Only the left eye camera was used. The trajectory of an object was selected from the image information by a CCD camera with a resolution of 500 $\times$ 400 pixels. The center position of each colored top face, the X-Y coordinates in the camera ([0-1]), was estimated by extracting the object from the color information.

b) Motor Information:
The neck (2 DOFs: pitch and yaw axis) and the left arm were used. Note that DOFs of used arm motors differed according to experiments, and unused motors were fixed.

Those values were synchronized between different modalities, and were normalized in [0.1-0.9]. The sensory data were stored every 400 msec for each manipulation, and their lengths were all twenty steps.

In the learning phase, the robot first collected the camera data and the motor data from its own neck and arm during active sensing. The connection weights for the RNNPB model were updated off-line using collected data simultaneously. In the observation phase, the robot then collected the neck motor data and the camera data. The corresponding PB values were calculated for the given sequence by the RNNPB model without updating the connection weight values. Finally, in the motion generating phase, the robot generated its motion by inputting the PB values obtained in the observation phase into the organized RNNPB model.

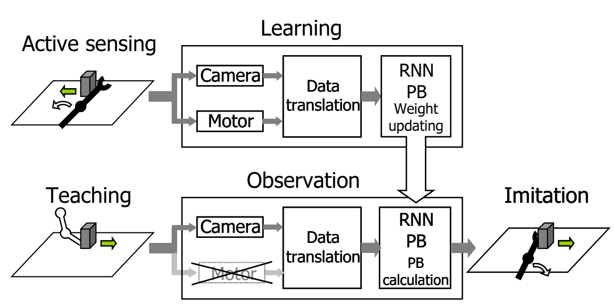


Fig. 5. System diagram of imitation.

IV. Experiment

A. Imitation of Known Manipulation

We carried out the experiment to confirm that whether the robot can associate its motions only with object motions.

1) Task: A target object is a cylinder-shaped object. In the experiment, there were two kinds of manipulation, parallel translation and tumbling (See Table I). Each manipulation has moving directions, "Left to Right (L → R)" and "Right to Left (R → L)". Learning 1 in Table I is, for example, parallel translation from the left to the right. Learning 1-4 in Table I represent manipulation that the robot learned in the learning phase. Observations 1-4 in the table represent the manipulation that the robot observed in the observation phase.

2) Procedure: In the learning phase, the robot first conducted motions programmed to manipulate the object as listed in Table I, Learning 1-4, and collected sensory data. It manipulated the object three times for each learning, and collected twelve patterns of data. The RNNPB model was then trained with the collected data 300,000 times, which spent approximately forty minutes. The RNNPB model, consisted of fifty-eight neurons: in the input layer, fifteen in the middle layer, ten in the context layer, and two as parametric bias.

In the observation phase, the robot then observed four types of manipulation, Observation 1-4 in Table I, presented by a human teacher. The robot collected data twice for Observation 1-4; there was a total of eight patterns. With the collected sensory data, the PB values were renewed 5,000 times, which spent approximately sixty seconds.

Finally, in the motion generating phase, the robot generated its motion.

3) Results: Fig. 6 has examples of sequences of input and output data after the RNNPB model learned in the learning phase. The solid lines describe RNNPB output (prediction) and the broken lines describe input (real data). We confirmed that RNNPB could predict the sequences accurately.

Fig. 7 shows two-dimensional PB space acquired in the learning and observation phases, which consisted of pairs of PB values. The PB values obtained in the learning phase were self-organized corresponding to the categories of object manipulations. The PB values resulting from observations of known manipulations are plotted close to the PB values resulting from the same manipulations being learned.

Fig. 8 shows motor values generated in the observation and motion generating phases. Learned manipulations were reproduced almost accurately. Fig. 9 has examples of sequential photographs that capture the robot generating motion.

TABLE I

Object manipulation (cylinder).

	Moving direction	Contact position
Learning 1	L → R	Low
2	L → R	High
3	R → L	Low
4	R → L	High
Observation 1	L → R	Low
2	L → R	High
3	R → L	Low
4	R → L	High

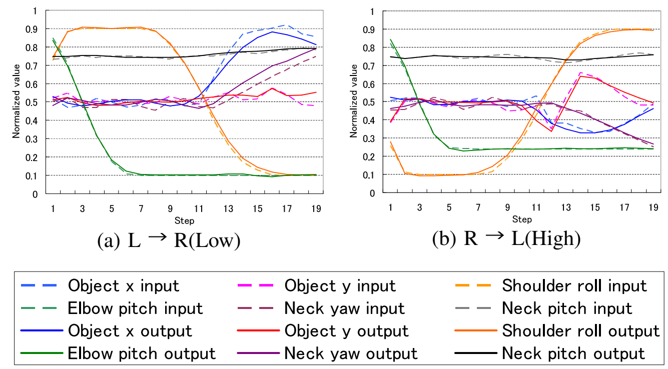


Fig. 6. Prediction of time sequences.

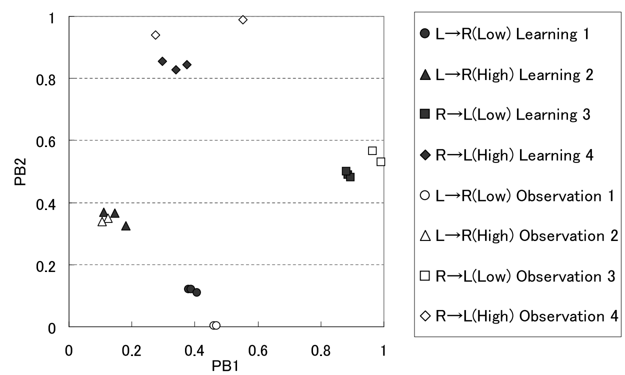


Fig. 7. PB Space (cylinder).

B. Imitation of Unknown Manipulation

We carried out another experiment testing some imitation motions involving not only the trained motions in active sensing process but also unknown motions.

1) Task: A target object is a box-shaped object. In the experiment, there were two kinds of manipulation, parallel translation from the left to the right ("L → R"), and rotation to the right ("Rrot"). Each manipulation was divided into three levels of moving distance: short "S", medium "M", and long "L" (See Table II). Due to several levels being set for each manipulation, we expected that the robot could learn about the gradual shift in its motor value. Learning 1-6 in Table II represent manipulation that the robot learned in the learning phase. Observations 1-3 in the table represent the manipulation that the robot observed in the observation phase. Observation 3, which is "moving from the left to the right while rotating to the right", is manipulation unknown to the robot.

2) Procedure: In the learning phase, the robot first conducted motions programmed to manipulate the object as listed in Table II, Learning 1-6, and collected sensory data. It manipulated the object once for each learning, and collected six patterns of data. The RNNPB model was then trained with the collected data 200,000 times, which spent approximately ten minutes. The RNNPB model, consisted of thirty-six neurons:

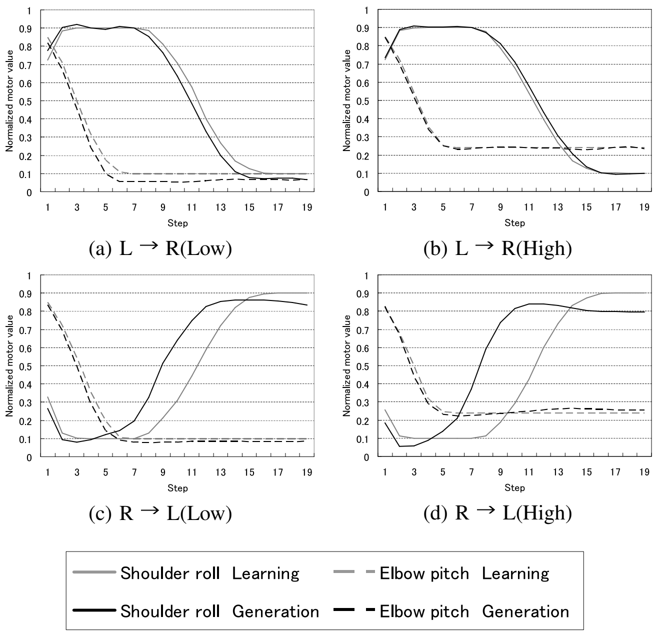


Fig. 8. Motor values generated in motion generating phase.

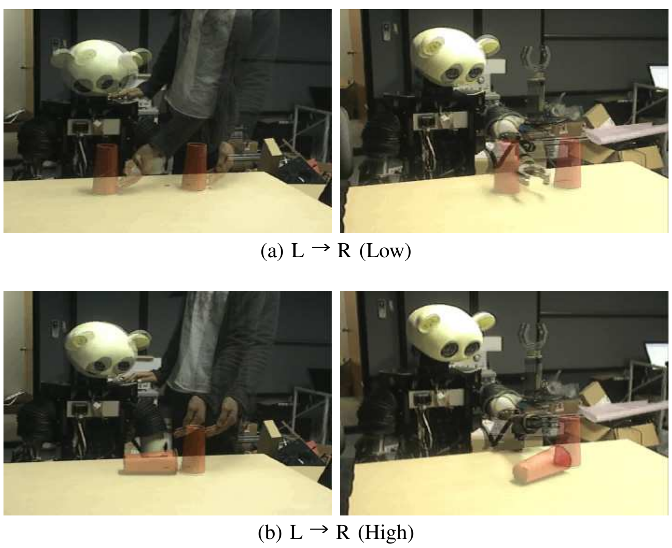


Fig. 9. Observation and motion generation (cylinder).

TABLE II

OBJECT MANIPULATION (BOX).

		Moving direction	Moving level
Learning	1	L → R	S
	2	L → R	M
	3	L → R	L
	4	Rrot	S
	5	Rrot	M
	6	Rrot	L
Observation	1	L → R	L
	2	Rrot	L
	3*	Rrot + L → R	L + L

*Unknown

in the input layer, fifteen in the middle layer, ten in the context layer, and two as parametric bias.

In the observation phase, the robot then observed three manipulations, Observation 1-3 in Table II, presented by a human teacher. The robot collected data once for each manipulation; there was a total of three patterns. With the collected sensory data, the PB values were renewed 5,000 times, which spent approximately fifteen seconds.

Finally, in the motion generating phase, the robot generated its motion.

3) Results: Fig. 10 shows PB space acquired in the learning and observation phases. The PB values obtained in the learning phase were self-organized corresponding to the categories of object manipulations and moving levels. The PB values resulting from observations of known manipulations are plotted close to the PB values resulting from the same manipulations being learned. However, PB values corresponding to the unknown manipulation labeled as "$*$" are plotted to the center position between "L → R" and "Rrot".

Fig. 11 plots the trajectories for the robot's hand seen from above the table in the learning and motion generating phases. Fig. 12 has sequential photographs that capture the robot observing and generating motion. The unknown manipulation was imitated as a combination of known manipulations.

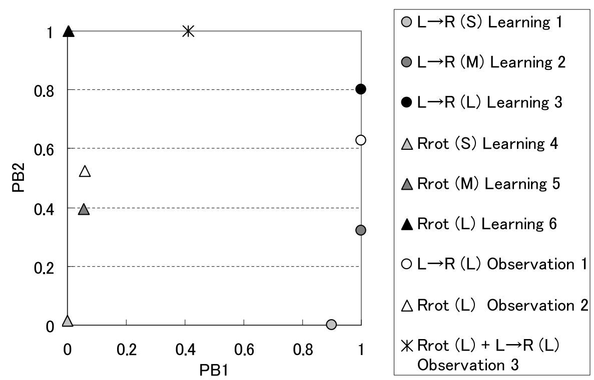


Fig. 10. PB Space (box).

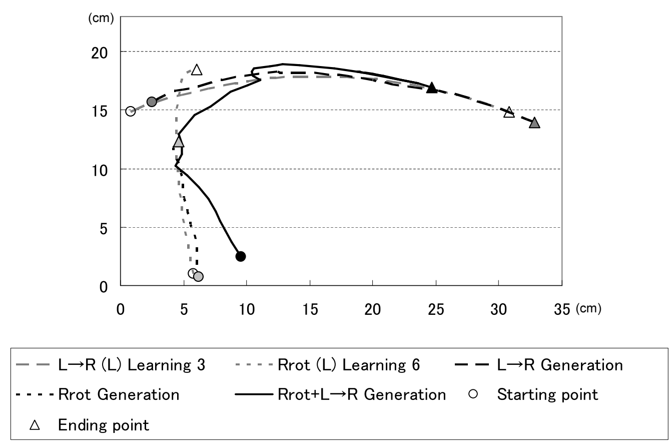
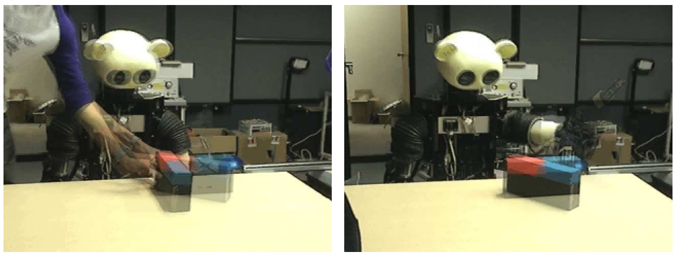
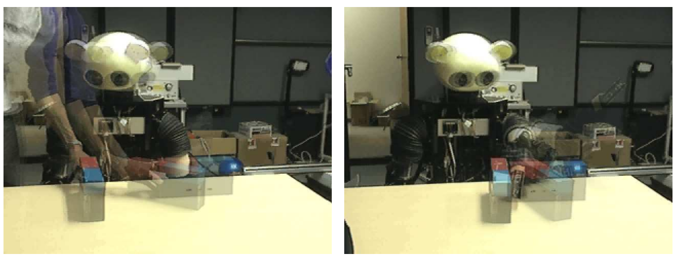


Fig. 11. Robot hand trajectory.

(a) L → R (Left to Right)



(b) Rrot (Right rotation)



(c) Rrot+L → R (Right rotation + Left to Right)

Fig. 12. Observation and motion generation (box).

V. Discussion

A. Prediction Capability

As we can see from Fig. 6, the RNNPB model has prediction capabilities. The robot can predict what kind of object motion its own motion would generate. This enables the robot to associate motion with object motion in the observation phase. In recognizing observed manipulation, the robot predicts the motion and object sequence, and recognizes the PB values that generate appropriate motion. In the motion generating phase, the robot predicts the sequence in real time, and selects motion for the next step with the RNNPB model.

B. Generalization Capability

The robot acquired behavioral primitives implicitly through learning in the second experiment: moving the hand from the left to the right for manipulation "L → R", and extending its arm for manipulation "Rrot". The unknown manipulation was recognized as a combination of the primitives. This clearly proved the generalization capabilities the proposed method had.

VI. Conclusions

This paper proposed a method of imitation focusing on object motion generated while a humanoid robot was actively sensing objects. The task was moving objects on a table, the first step in object manipulation. The method consists of three phases i.e., the learning, observation, and motion generating phases. The RNNPB model, which has generalization capabilities, was used as the learning model to reduce the learning load. By specifically taking advantage of the RNNPB model, the robot self-organized connection between its own arm motions and the object motions, associated a motion with an observed object motions. A learning system that gathered visual data and motor data during manipulations was implemented on the humanoid robot Robovie-IIs. An experiment using a cylinder-shaped object and an experiment using a box-shaped object were conducted. The first experiment demonstrated that the robot could associate its motions only with the object motions. The second experiment demonstrated that this method enabled the robot to imitate the unknown manipulation of object as well as learned patterns.

Although the task set for the experiment was object manipulation, our method can be used for different tasks. Our method plays a role of connecting the actor's operation and the target response. If the target is a part of body, it is also possible for robots to imitate body motions.

In this work, experiments were conducted with limited and few learning patterns. Acquiring a greater variety of motions requires resolution of the trade-off between generalization and differentiation of motion dynamics. Our future work will confirm the general effectiveness of the method for a variety of motions, and resolve the issue stated above to develop a sophisticated method which enables robots to generate more motion patterns.

Acknowledgment

This research was supported by RIKEN Brain Science Institute and the Scientific Research on Priority Areas: "Informatics Studies for the Foundation of IT Evolution".

References

[1] S. Schaal, "Is Imitation Learning the Route to Humanoid Robots?," Trends in Cognitive Sciences, vol. 3, no. 6, pp. 233-242, 1999.

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

[3] G. Rizzolatti and L. Craighero, "The mirror-neuron system," Annu. Rev. Neurosci., 27:169-192, 2004.

[4] J. Tani and M. Ito, "Self-Organization of Behavioral Primitives as Multiple Attractor Motions: A Robot Experiment," IEEE Trans. on SMC Part A, vol. 33, no. 4, pp. 481-488, 2003.

[5] R. Rao, A. Shon, and A. Meltzoff, "A Bayesian Model of Imitation in Infants and Robots," Imitation and Social Learning in Robots, Humans, and Animals, Cambridge University Press, 2005.

[6] T. Ogata, H. Ohba, J. Tani, K. Komatani, and H. G. Okuno, "Extracting Multi-Modal Motions of Objects using RNNPB," Proc. IROS 2005, pp. 160-165, 2005.

[7] 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.

[8] T. Ogata, S. Sugano, and J. Tani, "Acquisition of Motion Primitives of Robot in Human-Navigation Task: Towards Human-Robot Interaction based on 'Quasi-Symbols'," Journal of Japanese Society for AI, vol. 20, no. 3, pp. 188-196, 2005.

[9] H. Ishiguro, et al, "Robovie: an Interactive humanoid robot," Int. Journal of Industrial Robotics, vol. 28, no. 6, pp. 498-503, 2001.