Neural Networks, Vol.19, pp.323-337, 2006

Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model

Masato Ito Kuniaki Noda
Yukiko Hoshino
Sony Intelligence Dynamics Laboratories, Inc.
Takanawa Muse Building 4F, 3-14-13 Higashigotanda, Shinagawa-ku, Tokyo, 141-0022, Japan
Tel +81-3-5448-4777, Fax +81-3-5448-7378
{masato, noda, yukiko}@idl.sony.co.jp

Jun Tani
Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan
Tel +81-48-467-6467, FAX +81-48-467-7248
E-mail tani@brain.riken.go.jp

Key Words
learning of object handling behavior, dynamical systems approach, recurrent neural network

Abstract

This study presents experiments on the learning of object handling behaviors by a small humanoid robot using a dynamic neural network model, the recurrent neural network with parametric bias (RNNPB). The first experiment showed that after the robot learned different types of ball handling behaviors using human direct teaching, the robot was able to generate adequate ball handling motor sequences situated to the relative position between the robot's hands and the ball. The same scheme was applied to a block handling learning task where it was shown that the robot can switch among learned different block handling sequences, situated to the ways of interaction by human supporters. Our analysis showed that entrainment of the internal memory structures of the RNNPB through the interactions of the objects and the human supporters are the essential mechanisms for those observed situated behaviors of the robot.

1 Introduction

Learning object handling behavior by robots is a difficult problem since motor trajectories to achieve adequate handling behaviors could be diverse regarding various situations. Even when manipulating the same object, the motor time-development would be quite different depending on how the robot and the object are situated in the workspace. The current paper shows that a dynamic neural network model is effective in learning and generating such diverse and situational behaviors for object handling.

There are a substantial number of prior studies concerning the learning of object handling by robots. Recently, Bianco and Nolfi (Bianco & Nolfi, 2004) showed that a simulated robot arm can acquire object grasping behavior by evolving neural controllers. By evolving simple sensory-motor maps in layered networks, quite complex grasping behavior is generated dynamically even with a significant range of perturbations in position and direction of the object. However, it might be difficult to apply their evolutionary approach to a real robot task because it requires a substantial number of trials, which real robot situations cannot easily accommodate.

In some studies of reinforcement learning, behavior schemes are learned by combining predefined behavior primitives. For instance, for an object handling task, a robot learns to select among the predefined behavior primitives such as approaching, grabbing, carrying and releasing an object for each step appropriately. However, this approach can hardly be applied to a dynamic object handling behavior such as object grasping (Bianco & Nolfi, 2004) and juggling (Schaal, Sternad, & Atkeson, 1996) because it is difficult to divide the dynamic behavior scheme into a set of discrete behavior primitives manually. On the other hand, some researchers (Wolpert & Kawato, 1998; Tani & Nolfi, 1999) proposed models that can learn various behavioral skills from continuous sensory-motor flow without possessing any predefined behavior primitives. Recently, some of the authors proposed a neural network scheme, termed RNN with Parametric Bias (RNNPB) (Tani, 2003; Ito & Tani, 2004a), and applied it to the task of object manipulation by an arm-type robot(Tani, 2003). However, the task was quite simple since the object was manipulated only in a two-dimensional workspace and the interaction dynamics between the arm and the object were quite limited.

In the current study, complex tasks of a ball and blocks manipulations utilizing a humanoid robot are considered. In order to let the robot acquire these task skills, an imitation learning framework is introduced to avoid an unrealistic number of trial and error instances, which are often observed when applying reinforcement learning and genetic algorithms to complex behavior tasks. In our imitation learning method, manipulation of objects is directly taught by human supporters who guide the movements of the robot by grasping its arms. After repeated guidance and corresponding neuronal learning, the robot becomes able to generate the taught behavioral patterns with generalization. Although it is true that the introduction of direct teaching makes the task of imitation learning much easier(Billard, 2002), it has been reported that even chimpanzees cannot learn to imitate manipulatory actions by watching but can do so by direct teaching by human supporters (Myowa-Yamakoshi & Matsuzawa, 1999, 2000).

Imitation learning by watching may require human specific cognitive functions to solve the corresponding problems (Dautenhahn & Nehaniv, 2002; Nehaniv & Dautenhahn, 2001) with joint attention mechanisms(Baron-Cohen, 1996; Moore & Corkum, 1994) which our current robots as well as chimpanzees do not have.

The current study also investigates the issues surrounding interactive and cooperative behavior generation involving robots and human supporters. Interactive generation has been addressed in the research of human-robot cooperation. In the field of conventional engineering robotics, many have studied cooperative tasking such as carrying an object (K.Yokoyama et al., 2003) or dancing with a human (Kosuge, Hayashi, Hirata, & Tobiyama, 2003). In those studies, robots are controlled to keep desired states within the global task models, where the human assistance is incorporated. When a human supporter pushes or pulls an object, the robot can interactively behave by keeping its state trajectories within the pre-designed ones. However, in this approach, the controller of the robot has to be designed strictly as incorporated with the global task model. On the other hand, Ogata (Ogata, Masago, Sugano, & Tani, 2003) studied the cooperative robot-human navigation learning task without having such explicit task models. In their task, both of the human subjects and the robot learns to move to goal locations through repeated trials where the task skills of the robot are implicitly represented in the learned neural network. One of the crucial problems in interactive generation is how to coordinate the interactions between a robot's movements and the supporter's intentions of guidance. In order to accept guidance by human users, the robot's behavior generation has to be flexible enough to adapt to such external changes. On the other hand, the behavior generation has to be sufficiently robust in order to perform object handling behaviors stablely against various perturbations. Therefore, interactive generation involving human supporters requires a good balance between robustness and flexibility for adaptive behavior of the robot.

One specific goal of the current study is to show possible neuronal mechanisms that enable the robot to generate behavior adaptively corresponding to various situational changes of the robot, the object, and the human supporter. For this purpose, it is considered that reflex-type behavior generations for acquiring a simple sensory-motor mapping may not be sufficient since the recognition of situational changes in our task may require contextual information processing. In order to recognize current situations in a contextual manner, certain internal models might be required. The internal model, here, does not mean the global model of the task, but it refers to the capability to anticipate encountering sensory flow in the future by regressing sensory-motor flow of current and past time in a contextual manner. Much neuroscience research has identified that certain parts of prefrontal regions play an essential role in recognizing context switching. The Wisconsin card sorting task (WCST) (Milner, 1963) is one of the most popular schemes to investigate such mechanisms for the switching of cognitive sets. The subject is presented with cards of specific shapes, colors, and numbers. Then the subject has to sort the cards into different piles without having been explicitly given the current criteria for correct sorting. The subjects are then given feedback regarding the correctness of their current sorting results, which leads them to the correct sorting. Various neuro-imaging studies have indicated that the switching takes places with error monitoring in the anterior cingulate cortex (ACC) (Ito, Stuphorn, Brown, & Schall, 2003)and the resultant executive controls in the posterior parts of the bilateral inferior frontal sulcus (Nakahara, Hayashi, Konishi, & Miyashita, 2002). Although context switching in object manipulatory behavior and in the WCST dealing with cognition of abstract rules might be qualitatively different, they might share the same basic information flow of error-monitoring with anticipation and resultant executive control for switching.

In the current paper, our previously described scheme of the RNNPB (Tani, 2003; Ito & Tani, 2004b) is utilized as one possible neuronal network model to implement context switching. The ultimate challenge of the study is to clarify the essential mechanism of context switching for the task of object handling from the dynamical systems perspectives (Beer, 1995; Gelder, 1998). The dynamical structures that appear in the tight coupling among the body, the object and the internal neuronal processes will be explained by means of attractor dynamics and their parameter bifurcation characteristics.

2 Mechanism, model and algorithm

In order to achieve learning and the resultant interactive generation of learned behavior, a dynamic neural network model of RNNPB (Tani, 2003; Ito & Tani, 2004b) is utilized. In the following section, the basic cognitive modes of the RNNPB are introduced.

2.1 The basic mechanism

The following explains the basic idea for three different cognitive operational modes for a robot, which include learning, object handling, and object handling with human supporters. First, in the learning phase, sensory-motor patterns of guided behaviors are embedded in the RNNPB in the form of attractor dynamics. The attractor represents the essential spatio-temporal structure of the target behavior. Moreover, learning multiple behavior patterns is realized by switching among different attractor dynamics.

Second, the learned object handling behaviors are regenerated by means of interactions between the robot and the object. Here, the memory retrieval processes and the physical movements of the robot's arms and the objects are co-dependent. Mutual entrainment between the two of them enables the robot to generate the situated object manipulatory behaviors.

Third, the intentional acts of the human supporter are included in the interaction loop. The supporter can guide the robot to generate one of the learned patterns by physically forcing it to do so. This can be explained by the entrainment of the coupled dynamics of the RNNPB, the arms and the object movements by the supporter's intentional acts.

2.2 Model and algorithm

The three cognitive operational modes described in the previous subsection are implemented by using the RNNPB.

(A) The architecture

The RNNPB model has the same architecture as the conventional Jordan-type RNN model (Jordan & Rumelhart, 1992) except for the PB nodes in the input layer. Figure 1 shows the configuration of the RNNPB model in the learning phase (a) where the RNNPB is trained with sensory-motor flow sequences, and in the interaction phase (b) where the trained RNNPB generates situated motor outputs according to incoming sensory inputs. For the normal input and output nodes, two types of operations are performed at the same time: open-loop and closed-loop. In the open-loop operation, outputs of the network $(\hat{s}_{t+1}, \hat{m}_{t+1})$ are calculated as a result of a prediction from the current inputs (s_t, m_t). In the closed-loop operation, copies of the previous prediction outputs are copied to the current inputs, and outputs are calculated according to the feedback. This feedback enables look-ahead prediction (rehearsal process) for an arbitrary number of future steps without perceiving the actual inputs.

There are context nodes c_t in both the input and output layers. The output of the context nodes is copied to the context nodes in the input layer. The internal state is recursively computed for future steps utilizing the recurrent feedback loop for the context nodes. There are PB nodes p_t in the input layer. These PB nodes are the additional network variables that can be manipulated to learn and generate diverse behavioral patterns.

The common structural properties of the training data sequences are acquired as connection weights by using the backpropagation through time (BPTT) algorithm (Rumelhart, Hinton, & Williams, 1986). On the other hand, the specific properties of each individual time sequence are simultaneously encoded as PB values. Therefore, the modulation of the PB values shifts the modes of the behavior pattern. In the processes of learning and recognition, the PB values are iteratively computed utilizing the error between the target sensory-motor sequence and the predicted sequence.

(B) Learning process

The learning algorithm for the parametric bias vectors 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 backpropagated 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 PB at the t in the sequence are as follows,

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

$$\Delta\rho_t = \epsilon\delta\rho_t \tag{2}$$

$$p_t = sigmoid(\rho_t/\zeta) \tag{3}$$

In Eq. (1), the $\delta\rho_t$ force for the update of the interval values of the PB ρ_t is obtained from the summation of two terms. The first term represents the delta error, δ_t^{bp}, backpropagated 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 ($0 \leq t - l/2$ and $t - l/2 < l$). Integrating the delta error prevents the local fluctuations in the temporal PB values. The second term is a low-pass filter that inhibits frequent rapid changes of the PB values. The k_{bp} and k_{nb} are the coefficient of the above two terms, respectively. The internal value ρ_t is updated using the delta force $\delta\rho_t$ by the learning rate ϵ, as shown in Eq. (2). Then, the current PB values p_t are obtained from the sigmoidal outputs of the internal values ρ_t.

(c) Generation of behaviors

This mechanism for the inverse computation of the PB values is performed online during the interactive behavior generations of the robot with/without the human supporter even after the learning process is terminated. When the dynamic characteristics of on-going sensory flow is alternated from one mode to another, the prediction error with the current PB values increases. The PB value is then updated in the direction of minimizing the error, which results in an update of motor patterns situated to the current sensory inputs. This parallel mechanism of inverse computation of the PB by means of the regression of the past sensory flow and the forward computation of motor flow in the future based on the current PB is essential for generating situated behaviors (Tani, 2003; Ito & Tani, 2004b).

3 The robot system configuration

The robot task of object handling involves two-arm ball handling and one-arm block building. Both the ball and the block are visually identified by using color segmentation. The robot acquires an adequate behavior scheme by learning a set of sensory-motor sequences obtained through direct human teaching. During behavior learning, the target sequence to be learned is the paired trajectory of sensory and motor values. The sensory information is obtained for arm movement and object movement. The arm movement is the trajectory of the joint angles measured by encoders in both arms. Object movement is obtained from the center 3-D positions of color-segmented regions of the objects. On the other hand, the motor values are the trajectories of the reference of joint angles in the robot's arms. In the current experiments, the target reference trajectory is simply obtained as a copy of the measured arm movement during the direct teaching by human supporters. A set of these paired sequences is learned by the RNNPB offline. In the interactive generation mode, the robot attempts to generate suitable behaviors depending upon situations by predicting incoming sensory sequences. In this phase, the robot perceives the sensory input the and generates its corresponding motor output while adapting the PB values using regression.

The experimental robot system using a remote computing environment has been built. This system consists of the sensory processing module, the RNNPB module and the motor control module. The sensor processing module integrates and sends data to the RNNPB module. The RNNPB module calculates motor commands according to received sensor information and then sends it to the motor control module. The sensor processing module and the motor control module are executed on the computer embedded in the robot. Since the RNNPB module requires relatively heavy computation, it is executed on a remote PC cluster connected by a wireless communication system.

4 Robot experiments

In this study, we conducted two different robot experiments. In the first experiment, we examined how the robot can learn to generate situated behaviors involving dynamic interactions between its body and objects for a ball handling task. In the second experiment, we focused on how human interactions can afford the robot to generate situated learned behaviors in a block building task.

4.1 Dynamic generation of ball handling behaviors

4.1.1 Learning of behaviors from human direct teaching

In the first experiment, the robot learns two different types of ball handling behaviors, as shown in Figure 2. One is "rolling a ball" in which the robot swings both arms alternately to roll a ball on a table from left to right and vice versa (a). The other is "lift up a ball", which is to put the robot's hands together to lift up a ball on a table vertically and then release its hands to drop the ball (b).

This task is performed by the small humanoid robot QRIO which is seated on a chair and handles the ball on a table. The ball is 6 cm in diameter. The table is 45 cm square which is equipped with 1 cm high, 3 cm wide guides to prevent the ball from falling off the table. The table is inclined about 4 degrees to make the near side low so that the ball returns to the reachable area even if the ball is pushed away. QRIO perceives the ball with a camera equipped on its head and handles the ball with both of its arms. The RNNPB in QRIO receives two types of information. One is the current ball position (BALLX,Y,Z) that is obtained by the robot vision system in terms of (x,y,z) positions in the task workspace, where the size information of the segmented region of the ball color represents z information. The other is the encoder value for each joint angle of both arms at the current time step (L,RSHP: left,right shoulder pitch joint, L,RSHR: left,right shoulder roll joint, L,RSHY: left,right shoulder yow joint and L,RELP: left,right elbow pitch joint) The RNNPB outputs two types of information. One is the motor commands in terms of the reference values of all joint angles of the next time step, and the other is the prediction of the ball position in the next time step. (Note that the reference value and encoder value for each joint can be different because of position error by the PID control in QRIO.)

In the learning phase, the robot learns two different ball handling behaviors from human direct teaching. In the teaching process, a human user grabs the robot's arms and guides them to perform the target ball handling behaviors using an actual ball and while the servo gain of the robot arms is almost set to zero. In this study, the reference trajectory is simply obtained as a copy of the measured arm movement in the direct teaching by human users. The training data for the RNNPB were recorded with the time interval of 50 msec which is the same frequency as the RNNPB's calculation interval. For the ball rolling behavior, two cycles of the behavior, which start from the right and left sides (3 samples each), were recorded. For the ball lifting task, one cycle of the behavior (6 samples) was recorded. The sequence length of the trajectories is about 120 steps (6 sec) for the ball rolling task, and about 90 steps (4.5 sec) for the ball lifting up task. It is important to note that during the teaching process these two behaviors are given as separate sequences. Thus the robot never learns the transition between them.

A set of these paired sequences is learned by the RNNPB off-line. For learning the forward model of the behavior sequences, we employed an RNNPB that has 13 inputs node and 13 prediction output nodes. It also has 2 parametric nodes, 50 hidden nodes, and 70 context nodes. This configuration was obtained by the parametric study where we examined various RNNPB with different number of hidden nodes and context nodes by using the PC cluster system. For the learning sample set, the learning is iterated for 50000 steps, starting from an initial random set of synaptic weights. In order to avoid the overfitting to noisy data, we introduced the artificial small random noise into the output of RNNPB in the learning process. The final root-mean-square error of the output nodes was less than 0.0003.

Subsequently, in the interaction phase, the RNNPB in the robot receives the current ball position and the current encoder values for all joint angles as inputs and generates its corresponding motor commands and prediction of the ball position in the next time step as outputs in an online manner. For the online recognition process (PB regression by utilizing the prediction error), 50 instances of forward and back-propagation iteration were conducted using a 30-step length window of the immediate past in order to determine the PB at each next time step. Along with this update of the PB, the motor references for the next step are also computed by means of forward computation using the window.

4.1.2 Dynamic generation and switching of learned behaviors

After the learning, we examined how the robot with the trained RNNPB could generate two different learned ball handling behaviors. We also observed how the ongoing behavior could alternate between them depending on the situational differences between the robot and the ball.

Figure 3 (a) shows the snapshots of the ball rolling behavior generated by the robot. When the ball was rolling from the front of the robot to the left side, the robot hit it by the righthand. Then the ball rolled to the opposite side and the robot hit it with its left hand. This rolling a ball behavior was stably repeated for several times. Figure 3 (b) shows the snapshots of the ball lifting up behavior generated by the robot after the ball rolling behavior. When the human user stopped the ball in front of the robot, after a short while the robot started to hold it with both arms without any irregular movements and then lifted it up to a specified height. After this, the robot released the ball and then the ball was dropped in front of the robot. The robot started to hold it again. This ball lifting up behavior was also autonomously repeated for several times.

Figure 4 shows the time course of the whole interaction and the parametric bias values of the RNNPB. In Figure 4, the plot at the top and the second row show the actual ball positions and the ones predicted by the RNNPB. The third row plot shows the robot joint angles generated by the RNNPB (Only 2 DOF are plotted among a total of 8 DOF). The plot at the bottom shows the parametric bias of the RNNPB.

We observed a transient status where two different behaviors were switched between (from rolling to lifting up) as a result of PB online adaptation according to modulations in the sensory sequence pattern. In this case, the ball position was changed by human intentionally. At around 200 steps, the ball's motion was stopped in front of the robot body. This resulted in one of the PB values being decreased. From about 340 steps, ball lifting up behavior was generated according to the PB values.

In this experiment, we observed that the learned ball handling behaviors were well generated through interaction between the RNNPB dynamics and the ball movement dynamics. Remember that the actual ball movement does not necesserally repeat exactly the same as the learned one. Even under such noisy conditions, the learned behaviors were stably generated. The system seems to maintain a certain robustness against unknown irregularities. We speculate that such robustness originates from the characteristics of attractor dynamics that emerges in the coupling between the RNNPB dynamics and the ball movement dynamics. It was also observed that the behavior switching could be performed smoothly which the robot had never learned to do. (Remember that in the learning process, these two behaviors were just trained as separate patterns.) It can be said that novel behavior in terms of behavior transitions were generated spontaneously utilizing emergent dynamical structures self-organized in the system.

4.1.3 Analysis of the memory structure of RNNPB

To clarify the relationship between the memory structure organized in the RNNPB and the behavior appeared in the previous robot experiment, we analyzed the structure of the PB space and then examined the behavior dynamics embedded in it.

First, to visualize the memory structure of learned behaviors in the RNNPB, we calculated the distribution of prediction error for the ball movements of two learned ball handling behaviors over all the PB space. Each point in the PB space was set in the RNNPB as a constant value and then the RNNPB predicted ball movements without PB adaptation in the simulation. Ball movements from both rolling a ball and lifting up a ball were shown independently and the prediction error was calculated for each sequence.

Figure 5 shows the contour map of the prediction error obtained by the above calculation over the two dimensional PB space. Two of the error distributions were overlaid to show the smaller prediction error value distribution. In Figure 5, it is observed that there are two continuous bowl shaped error distribution and two minimum values exist for each structure. It is also observed that there is a boundary of two structures in the region where PB1 around 0.70 and PB2 around 0.95. By relating the PB trajectory shown in the figure 4 to this contour map, we speculate that continuous variation of the PB values between the two different regions in the PB space contributed in the smooth behavior switching.

Next, to examine the behavior dynamics embedded in the PB space where two different types of ball handling behaviors have been learned, we chose the three typical PB vectors from the distribution map and examined how the RNNPB could perceive the ball movements of each behavior for them and generate the prediction and the corresponding arm movements in the offline simulation. The PB vectors chosen to correspond with each behavior are as follows: (PB1: 0.99, PB2: 0.99) for the ball rolling task, (PB1: 0.59, PB2: 0.97) for the ball lifting up task and (PB1: 0.73, PB2: 0.96) for the intermediate between the two behaviors. The ball movements of the two learned behaviors were shown to the RNNPB for each PB vector and then the RNNPB predicted the ball movement and generated its corresponding arm movement without PB adaptation.

Figure 6 shows the ball movement predicted and the arm movement generated by the RNNPB with three typical PB vectors (a), (b), (c) for ball movements of rolling a ball. Figure 7 shows the same ones for the ball movement of lifting a ball.

In these two figures, we found that the RNNPB with the intermediate PB value performed as well as the RNNPB with the proper PB as compared to the RNNPB with the opposite PB for both rolling a ball and lifting a ball. This means that two different behavior dynamics were incorporated into the single dynamical structure which was represented by a constant PB values at the boundary region.

4.2 Interactive generation of block handling behaviors

4.2.1 Learning of behaviors from human direct teaching

In the second experiment, the robot learns to put one block on top of another one from among three color blocks, using the right hand arm using human direct teaching. The teaching process is the same as that of the first experiment except the hand action for grabbing and releasing a block. The action of grabbing a block is activated with the tactile sensor on the palm of the robot's hand. Then, when the palm contacts with a block, the grabbing action is reflexively generated. On the other hand, the action of releasing a block is activated with a touch sensor on the robot's shoulder. When human users want the robot to release a block, they touch the robot's shoulder and then the robot releases it.

In this task, the robot perceives the arm movement and the movement of each of the three color block and generates the arm movement including the hand action. In the current experiments, the tactile sensor was only used in the teaching phase and not used in the interaction phase. The sensory input vector consists of the current step encoder values for joint angles of the robot arms (RSHP: right shoulder pitch joint, RSHR: right shoulder roll joint, RSHY: right shoulder yaw joint and RELP: right elbow pitch joint) and the center 3D positions of the three color blocks (REDX,Y,Z, BLUEX,Y,Z and YELLOWX,Y,Z). The motor output is the next step reference values of the joint angles of the robot arms and the binary hand action such as grabbing or releasing. For learning the forward model of the behavior sequences, we employed the RNNPB which has 20 input nodes and 20 prediction output nodes. It also has 2 parametric nodes, 40 hidden nodes, and 80 context nodes.

4.2.2 Interactive generation of learned behaviors

In the experiment with the building block task, it is examined how the robot can adaptively situate its behavior relative to guidance from human users.

In this experiment the robot learns two alternative behaviors to be selected in the same block layout situation where the yellow block, the blue one and the red one are located on the lefthand side, in the center and on the righthand side respectively, in front of the robot. One behavior is to put the yellow block onto the blue one (B1). The other behavior is to put the red block on top of the blue one (B2).

This teaching process was repeated twice for each behavior to prepare a set of samples for robust learning. Each sample data was obtained by sampling the behavior sequence each 0.5 seconds for 50 steps. For the learning sample set, the learning is iterated for 100000 steps, starting from randomly set initial synaptic weights. The final root-mean-square error of the output nodes was less than 0.0003.

After learning, we examined if the robot's behavior could be switched alternatively by means of partial human guidance. In this experiment, the control gains of the robot's arm were set to the relatively low values where the robot can be moved by itself and by a human. Figure 8 shows the snapshots of behavior switching guided by the human user. In this case, at first the robot autonomously tried to grab the yellow block for B1 due to the initial PB values and succeeded it. After the human user restored the yellow block to the initial position, when the robot tried to generate B1 again, the human user guided the robot by grabbing its arm to switch to B2. The human user continues the guidance to switch it while she or he senses the robot's "against-force" (resistance) back to her or his hand. After awhile the robot starts to follow the user's guidance while the user feels joint movement force instead of the against-force at their hand. (This part of the force feedback from robots to human resembles the experimental studies of the RNN-implemented force feedback joystick by Tani (Tani & Ito, 2005).)

Although the human user stopped providing guidance when the robot's hand was approaching the red block for B2, the robot's arm kept on moving to it instead of moving back to the yellow block for B1. After this, the robot tried to grab the red block but failed once. However, after the robot's arm moved back to the initial position which is far from the blocks, the robot retried to generate B2 and finally the robot successfully achieved B2.

Figure 9 shows the time course of the entire behaviors and the parametric bias values of the RNNPB. In Figure 9, it is observed that the generation of a complex motor command sequence by the RNNPB is adaptively switched by human guidance. After the human user guided the robot arm by direct teaching from 80 to 95 steps, the robot behavior was switched from B1 to B2. During this, one of the PB values was gradually increased according to the prediction error of the sensory inputs and then the behavior switching was done. Since the

PB was still changing toward the value corresponding to B2, the robot failed to grab the block once. After this, the PB achieved it and the robot successfully achieved behavior B2.

This robot experiment showed that human users can allow the robot to generate their intended actions through the force-based bodily interactions while the internal neuronal dynamics and the body dynamics of the robot continues without stopping.

4.2.3 Analysis of the interaction structure and the memory structure of RNNPB

In order to clarify the characteristics of the interactive generation based on the learned behaviors, we conducted further experiments for the purpose of analysis. In this experiment, we examined the timing of successful interactions which led the robot to achieve a user's intended actions. More specifically, we examined how the human user could alternate the robot's behavior by direct guidance using three different timings during the ongoing behavior: T1 (far from the block), T3 (near the block) and T2 (between T1 and T3). This experiment was conducted in the situation as in the previous experimental settings, when the robot performs B1, the user intends to switch to B2. We conducted this trial three times for each timing. Table 1 shows whether the robot's behavior was switched or not and the prediction error for the three trials in the three timings. This error indicates the accumulated prediction error during human direct guidance. In the direct guidance process, when the human user grabs the robot's arm and leads it in a different direction from the ongoing behavior, prediction error is generated in the RNNPB which predicts the encoder values of joint angles for the next time step. This could indicate the against-force of human direct guidance. We could thus say there is no significant difference in the force level.

Table 1: Performance of behavior switch and the prediction error during it at each timing.

Timing	Behavior Switch	Error
T1 (far from the block)	SUCCESS	0.002548
	SUCCESS	0.002061
	SUCCESS	0.002044
T2 (between T1 and T3)	FAIL	0.003419
	FAIL	0.001766
	FAIL	0.002838
T3 (near the block)	FAIL	0.004631
	FAIL	0.004251
	FAIL	0.002761

From this result, we found that behavior switching by human direct guidance could not be done at any time but at particular timing in the ongoing behavior. It is speculated that this interaction structure is related to the self-organized dynamic memory structure of the RNNPB. To clarify this relationship, we examined the robustness of the behavior by adding perturbations to the sensory input of the RNNPB at the three timings. The robustness of the behavior must be related to its persistence. The persistence of an ongoing behavior and the ease of human intervention are opposites. If the persistence of a behavior is high, it is difficult to switch the behavior by an external force such as human direct guidance. On the other hand, if the persistence of a behavior is low, it is easy to switch by external guidance.

In this experiment, we measured the robustness of the behavior at the three timings by adding several levels of perturbation: 0.005, 0.05 and 0.15 of the value normalized for the RNNPB. The perturbation was to offset the measurement of joint angles by such levels for 15 steps. Table 2 shows whether the ongoing block building behavior succeeds or not after the several level perturbation has been added at three timings: T1, T2 and T3.

Table 2: Performance of behavior switch and the prediction error during it at each timing.

Timing	Rate of perturbation		
	0.5	5.0	15.0
T1 (far from the block)	SUCCESS	FAIL	-
T2 (between T1 and T3)	-	SUCCESS	FAIL
T3 (near the block)	-	-	SUCCESS

In the case of T1, if the ongoing behavior succeeded when 0.5 % perturbation was added, the ongoing behavior failed when the 5 % perturbation was added. In the case of T2, although the ongoing behavior succeeded when 5 % perturbation was added, the ongoing behavior failed when 15 % perturbation was added. In the case of T3, even if 15 % perturbation was added, the ongoing behavior succeeded.

From this result, it is found that the behavior's robustness against the perturbation increased as the robot's arm approached the block. This suggests that the memory dynamics of the RNNPB which would determine the persistence of the ongoing behavior was dynamically changing as it proceeded. Thus we conclude that the robot could accept guidance by human users was dynamically changed depending on the dynamical change of the memory dynamics of the RNNPB.

5 Discussion

5.1 Dynamical systems explanation

Our experiments showed that the robot can learn to generate multiple patterns of object handling both robustly and flexibly by utilizing the RNNPB. The dynamical relationships between the objects and the body were learned as the memory dynamic structure self-organized in the RNNPB. The behaviors learned were autonomously generated through mutual entrainment between the internal neuronal dynamics and the external environmental dynamics, which enabled the robot to maintain stable behavioral performance. Moreover, in the case of performing different types of object handling behaviors, behavior patterns were flexibly alternated by means of utilizing the dynamic characteristics of the parameter bifurcation of the RNNPB. Such behavior switching could be done not only by environmental changes but also by human direct guidance. The structure of interactive generation was related to the behaviors' persistence, which was characterized by the memory structure of the RNNPB.

In the dynamic and interactive generation of object handling behaviors, the characteristics of robustness and flexibility are crucial, which seems to be contradictory. In order to examine how such two conflicting natures can be incorporated into the presented scheme, we review the dynamical mechanism of the RNNPB and the resultant information flow that appeared in the coupling between the robot, the object, and the human user.

Figure 10 shows the sketch of the information flows in the ball handling task (a) and in the block handling task (b). In this figure (a), the RNNPB can be divided into two subsystems, the standard RNN itself and the PB units associated with it. The former handles fast dynamics while the latter handles slow dynamics, remembering that the neural activations in the RNN modulate step by step while the PB values modulate with much slower time constant on the order of every 10 steps.

The robot's arm is controlled by the motor command vector $\hat{m}_{t+1}$ generated by the RNN and affects the object in terms of the force F_t. The RNN receives current sensor vectors regarding the relative positions of the arm s_t^a, the object s_t^o, and the encoder reading of joint vectors m_t as its inputs. The output of the RNN is the prediction of the next sensor vector $\hat{s}_{t+1}^a$ and $\hat{s}_{t+1}^o$ and the next motor command vector $\hat{m}_{t+1}$. The PB is modulated depending upon the prediction error regarding the sensor inputs. In addition, as shown in Figure in 10 (b), for interactive generation with human users in the block handling task, direct guidance by human users affects the robot's arm movements in terms of the force G_t and resultantly the RNN receives the affected encoder values for all the joints. The PB is slowly modulated depending on the error between the predicted values (target motor outputs) and their outcomes (encoder readings). In summary, the ongoing robot's behavior is determined by the fast dynamics in the RNN while behavior can be alternated in the coupling with the objects and the human user by means of the slow dynamics in the PB. It is important to note that the "relatively" slow dynamics of the PB as compared to the RNN dynamics is useful to balance between the robustness of behavior and the flexibility of behavioral switching depending on situation changes. In the case of faster motor behavior control such locomotory gaits, as compared to the dynamics of gait pattern, the switching of it could be the "relatively" slow dynamics. However, in the current study the PB dynamics is "absolutely" slow due to the heavy computation time for calculating the PB value in on-line. In order to apply our model to the "absolutely" faster task such as locomotion, it requires to improve the implementation or the algorithms of the RNNPB.

Next, we consider what sorts of dynamic mechanism are responsible for the alternations of the learned behaviors. Figure 11 (a) shows a schematic of transition from the learned behaviors B1 to B2 regarding changes in environmental situations. The figure at the top shows an illustration of the pseudo-potential field in which the learned behavior B1 is generated as stabilized in the global minimum. Here, the potential is defined in terms of the prediction error associated with the current ongoing behavior. In this figure, the prediction error is minimized when the learned behavior of B1 is stablely generated as situated to the current environment i.e., the current ball movement pattern. Since the combination between the behavior B1 and the current ball movement pattern makes the potential as the global minimum, B1 can be robustly generated regardless of certain contingencies of irregular ball movements. When the environmental situations change in terms of ball movement patterns, the landscape of the potential field gradually modulates as the PB slowly changes in the direction of minimizing the error. During this change, another local minimum appears which represents the alternative behavior pattern B2. As the PB continues to change, this new local minimum grows in depth and finally becomes the global minimum in the potential field, as shown from (2) to (4) in figure 11. Then, the behavioral pattern of B2 is generated robustly.

On the other hand, Figure 11 (b) shows how the pseudo-potential field modulates in the interactive generation of the two learned building block behaviors B1 and B2 when responding to external support by the human user. When the robot's arm is located far from both blocks, there are two local minima that correspond to each block building behavior in P1. However, from P2 to P3 in the generation of one of the behavior patterns in the course of approaching to the block, the local minimum for B1 becomes steeper while the other local minimum for the B2 becomes shallow. Th is occurs because prediction becomes more accurate as the arm approaches the block in the on-going behavior. During this course if the human supporter attempts to switch the behavior pattern from B1 to B2, it requires more force for the shift as the behavior proceeds. It actually becomes impossible for the human supporter to switch the ongoing behavior in P4 and P5.

These two examples illustrate how both robustness and flexibility in behavior generation can be achieved in the presented scheme. Each learned behavior pattern can be generated in terms of attractor dynamics defined in the pseudo-potential field that appeared in the coupling among the robot, the ball, and the human supporter. The behavior can be generated the most stablely when the attractor becomes the global attractor with the global minimum in the potential field. However, changes in the relation between these three initiate gradual modulations in the landscape of the potential field which finally results in the catastrophe of switching the global minimum, which corresponds to the switching from one behavioral pattern to another.

5.2 Comparison with related works

The learning of object handling behaviors from human direct teaching in the current study is largely related to the recent studies of robot learning such as "programming by demonstration" (Calinon, Guenter, & Billard, 2005) and "computational approach to imitation learning" (Schaal, 1999; Schaal, Ijspeert, & Billard, 2003). We will examine the essential characteristics of our proposed scheme through comparisons with those schemes. Recently, Calinon, Guenter and Billard (Calinon et al., 2005) studied the imitation of goal-directed behavior in humanoid robot. In their study, a robot can imitate human demonstrations in terms of not only the reproduction of observed arm movements but also the reproduction of observed goal state in the task of reaching behavior by using the combination of the multiple cost functions regarding to them. Also, in our study, a robot learn perceived arm movements as not just imitations of trajectories but a goal-directed behavior since the learned ball handling behaviors are stably generated against irregular ball movements. In our task, the cyclic movements of rolling ball or holding and dropping it themselves can be regard as the goal state which are embedded in stable attractor structures. However, our model does not incorporate with explicit evaluation criteria for the goals although the models by (Calinon et al., 2005) does this. In our case, the invariant sets in the generated attractors represent the goals and all the vector flows converging to the invariant sets may represent the ways to achieve the goals where their distinctions are implicit from the outside. We will clarify the mechanisms of our learning model by referring to the classification about imitation learning from the computational point of view (Schaal, 1999; Schaal et al., 2003). Our behavior learning from human direct guidance corresponds to "imitation by direct policy learning" scheme in their classification. In this scheme, without task specific evaluation criteria, the policy of behavior control can be learned from observed trajectories, though achievement of task is not guaranteed. In our study, such direct policy learning is the learning of the forward dynamics of observed sensory-motor trajectories. And the forward dynamics learned by the RNNPB could incorporate the task goal state as dynamical attractor. Thus, our model can learn the goal-directed behavior without task specific evaluation function regarding to task goal.

From the other point of view, our learning of multiple behaviors is the learning of switching dynamics for motor schemes. For this problem, there are several models besides the RNNPB. Here, we make the comparisons between the RNNPB and the other models with focusing on the following three points.

First, Ijspeert, Nakanishi and Schaal (Ijspeert, Nakanishi, & Schaal, 2003) proposed a dynamical systems model which composes of learnable movement primitive based on two types of canonical dynamical systems, namely fixed point dynamics and limit cycle dynamics. The computational cost of their model is quite low and the stability in its learning is high. However, RNN can learn both of fixed point dynamics and limit cycle dynamics stably. And Tani and Ito (Tani & Ito, 2003) showed that the RNNPB can learn both of them simultaneously in its distributed representation. Moreover, it has been studied that RNN can learn more complex structures such as chaos to learn complex non-periodic patterns (Tani & Fukumura, 1995) and fractal structure to learn grammatical recursive structure (Pollack, 1991; Elman, 1990) without having prior structural assumptions of the networks.

Secondly, in order to learn multiple behaviors as switching dynamics, two distinct types of learning scheme have been proposed. One is the local representation scheme such as MOSAIC (Wolpert & Kawato, 1998) and Mixture of RNN experts (Tani & Nolfi, 1999). On the other scheme is the distributed representation scheme such as RNNPB. RNNPB has remarkable generalization capability on learning multiple dynamic patterns since the shared structure are extracted through the interference among memorized patterns (Tani & Ito, 2003; Ito & Tani, 2004a; Sugita & Tani, 2005). In the RNNPB, as the number of training patterns increases, the PB space is organized as more complex structures where not only training patterns but also various novel patterns can be generated. When there are certain common characteristics among training patterns, the PB space becomes smooth where patterns interpolated among the training patterns can be generated. On the other hand, when the RNNPB can not find such characteristics, the PB space can be distorted in nonlinear way where the quite different novel patterns from trained patterns can be generated. However, RNNPB has difficulty in increasing in the number of memorized patterns due to such memory interference. On the other hand, in the local representation scheme, it is easy to increase the number of memorized patterns by adding local modules at the cost of loosing generalization among the memorized patterns. Future researches might consider how to combine both of the distribute representation and the local representation scheme in order to exploit both of their advantages. For example, the current model could be extended to have multiple gated modules each of which affords distributed representation by having associated PB units.

Thirdly, there have been three different schemes which enable switching among learned dynamics while all of them use maximum likelihood estimation such as EM algorithm and back-propagation algorithm in whether the local representation scheme or the distribute representation scheme. The first scheme is to introduce the gating module which selects local module according to external inputs (Weigend, Mangeas, & Srivastava, 1995). In this scheme, although dynamics switching can be done before generating outputs unlike the other two schemes, the whole performance can not be maximized for unknown inputs. The second scheme is to select the most likelihood module among the local modules depending on their performance such as the prediction error of each modules (Wolpert & Kawato, 1998). The third scheme is to have the internal state in a switching mechanism and update it so that the whole performance can be maximized by EM algorithm or back-propagation algorithm in online manner, which RNNPB is classified into. The switching of the third scheme could be more contextual than that of the second scheme since the internal state preserves the past history of switching. Such contextual switching is more important for switching dynamics for motor schemes than that for classification problems to balance both of robustness and flexibility.

5.3 Imitation learning with more natural setting

There are many open problems to be addressed in the current imitation learning scheme. One interesting question might be how the presented approach to imitation by direct guidance can be improved to a more natural one. Although the movements are taught explicitly in the current method, real imitation of infants can be done in quite an ambiguous setting. A hard problem associated with this is the correspondence problem (Dautenhahn & Nehaniv, 2002; Nehaniv & Dautenhahn, 2001) that asks how parts of bodies of others attended by the self can be matched with those of the self. It is not just the problem of coordinate system transfers but it should include the problem of joint attention (Dautenhahn & Nehaniv, 2002; Nehaniv & Dautenhahn, 2001). Joint attention is also crucial to determine when and what to imitate without explicit segmentation of interaction flow between imitators and tutors(Andry, Gaussier, Moga, Banquet, & Nadel, 2001). (Andry et al., 2001; Ito & Tani, 2004b) argued that joint attention in the case of imitating rhythmic movement patterns can be achieved from a continuous interaction flow by means of synchronization. Also the current experiment with the human supporter shows a similar line as the intention of the supporter and the robot's movement is dynamically matched by means of entrainment. This can be interpreted as joint attention to one specific learned behavior patterns by both the robot and the supporter. However, such joint attention can appear only after explicit training of the robots, both in the current study as well as in our previous work (Ito & Tani, 2004b). One specific future research goal is to consider how to introduce this sort of joint attention mechanism during the learning period. In order to proceed in this direction, learning should be conducted in a developmental way instead of using the current offline batch training method. Our future research will focus on these problems.

6 Summary

Our robot experiments have demonstrated the dynamic and interactive generation of learned multiple object handling behaviors by a small humanoid robot using the RNNPB. We have shown and discussed that our approach can provide both robustness and flexibility in such behavior generation by utilizing the fast and the slow dynamics in the RNNPB.

Acknowledgments

The authors are grateful to Dr. Toshitada Doi, Masahiro Fujita and Hideki Shimomura for giving us this research opportunity and useful comments. The authors are also grateful to Dr. Ronald C. Arkin for correcting the final draft. The authors thank the members of Entertainment Robot Company, Information Technology Laboratories, Sony Corporation and Sony Intelligence Dynamics Laboratories.

References

Andry, P., Gaussier, P., Moga, S., Banquet, J., & Nadel, J. (2001). Learning and communication in imitation: An autonomous robot perspective. *IEEE Transaction on Systems, Man and Cybernetics. Part A : Systems and Humans*, *31*(5), 431–444.

Baron-Cohen, S. (1996). *Mindblindness: An essay on autism and theory of mind.* MIT Press.

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

Bianco, R., & Nolfi, S. (2004). Evolving the neural controller for a robotic arm able to grasp objects on the basis of tactile sensors. *Adaptive Behavior*, *12*(1), 37–45.

Billard, A. (2002). Imitation: a means to enhance learning of a synthetic proto-language in an autonomous robot. In *Imitation in animals and artifacts* (pp. 281–311).

Calinon, S., Guenter, F., & Billard, A. (2005). Goal-directed imitation in a humanoid robot. In *Proceedings of the ieee international conference on robotics and automation (icra).*

Dautenhahn, K., & Nehaniv, C. L. (Eds.). (2002). *Imitation in animals and artifacts.* Cambridge, MA: MIT Press.

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

Gelder, T. van. (1998). The dynamical hypothesis in cognitive science. *Behavior and Brain Sciences.*

Ijspeert, A., Nakanishi, J., & Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *In advances in neural information processing systems 15* (pp. 1547–1554).

Ito, M., & Tani, J. (2004a). Generalization in learning multiple temporal patterns using rnnpb. In *Proc. of the 11th international conference on neural information processing (iconip2004)* (pp. 592–598).

Ito, M., & Tani, J. (2004b). On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, *12*(2), 93–115.

Ito, S., Stuphorn, V., Brown, J. W., & Schall, J. D. (2003). Performance monitoring by the anterior cingulate cortex during saccade countermanding. *Science*, *302*(5642), 120–122.

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

Kosuge, K., Hayashi, T., Hirata, Y., & Tobiyama, R. (2003). Dance partner robot -ms dancerr-. In *Proceedings of the 2003 ieee/rsj international conference on intelligent robots and systems* (pp. 3459–3464).

K.Yokoyama, H.Handa, T.Isozumi, Y.Fukase, K.Kaneko, F.Kanehiro, Y.Kawai, F.Tomita, & H.Hirukawa. (2003). Cooperative works by a human and a humanoid robot. In *Proceedings of the 2003 ieee international conference on robotics and automation* (pp. 2985–2991). Taipei, Taiwan.

Milner, B. (1963). Effects of brain lessons on card sorting. *Archives of Neurology*, *9*, 90–100.

Moore, C., & Corkum, V. (1994). Social understanding at the end of the first year of life. *Developmental Review*, *14*(4), 349–450.

Myowa-Yamakoshi, M., & Matsuzawa, T. (1999). Factors influencing imitation of manipulatory actions in chimpanzees (pan troglodytes). *Journal of Comparative Psychology*, *113*(2), 128–136.

Myowa-Yamakoshi, M., & Matsuzawa, T. (2000). Imitation of intentional manipulatory actions in chimpanzees (pan troglodytes). *Journal of Comparative Psychology*, *114*(4), 381–391.

Nakahara, K., Hayashi, T., Konishi, S., & Miyashita, Y. (2002). Functional mri of macaque monkeys performing a cognitive set-shifting task. *Science*, *295*, 1532–1536.

Nehaniv, C., & Dautenhahn, K. (2001). Like me? - measures of correspondence and imitation. *Cybernetics and Systems*, *32*(1-2), 11–51.

Ogata, T., Masago, N., Sugano, S., & Tani, J. (2003). Interactive learning in human-robot collaboration. In *Proc. 2003 ieee/rsj int. conf. on intelligent robots and systems (iros2003), las vegas, usa* (pp. 162–167).

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

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning internal representations by error propagation. In *Parallel distributed processing.* MIT Press.

Schaal, S. (1999). Is imitation learning the route to humanoid robots ? *Trends in Cognitive Sciences*, *3*, 233–242.

Schaal, S., Ijspeert, A., & Billard, A. (2003). Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, *358*, 537–547.

Schaal, S., Sternad, D., & Atkeson, C. G. (1996). One-handed juggling: A dynamical approach to a rhythmic movement task. *Journal of Motor Behavior*, *28*(2), 165–183.

Sugita, Y., & Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior*, *13*(1), 33–52.

Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction process. *Neural Networks*, *16*, 11–23.

Tani, J., & Fukumura, N. (1995). Embedding a grammatical description in deterministic chaos: an experiment in recurrent neural learning. *72*(Biological Cybernetics), 365–370.

Tani, J., & Ito, M. (2003). Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. *IEEE Transactions on System, Man and Cybernetics Part B*, *33*(4), 481–488.

Tani, J., & Ito, M. (2005). Interacting with neurocognitive robots: A dynamical system view. In *Proc. 2nd int. workshop on man-machine symbiotic systems, kyoto, japan* (pp. 123–134).

Tani, J., & Nolfi, S. (1999). Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. *Neural Networks*, *12*, 1131–1141.

Weigend, A. S., Mangeas, M., & Srivastava, A. N. (1995). Nonlinear gated experts for time series: discovering regimes and avoiding overfitting. *International Journal of Neural Systems*, *6*, 373–399.

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

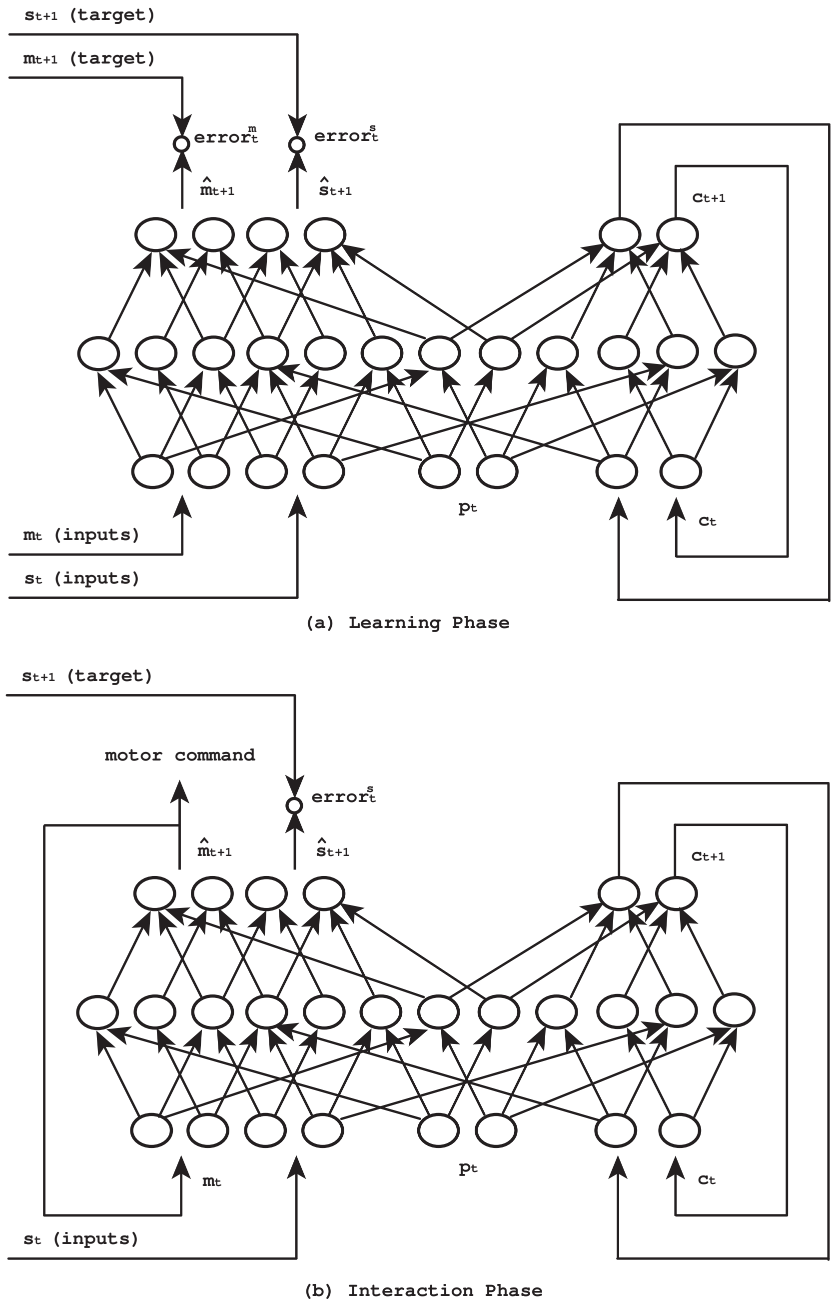
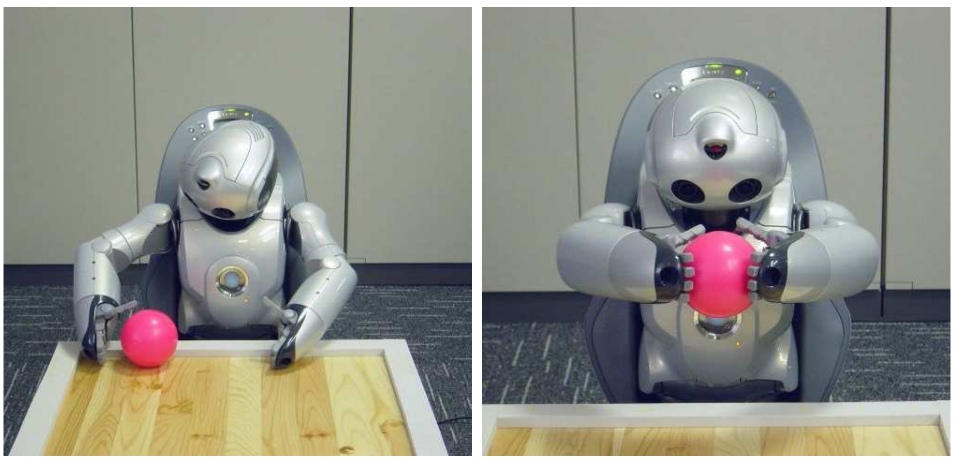


Figure 1: The configuration of RNNPB in learning phase (a) and interaction phase (b).



(a) (b)

Figure 2: The rolling a ball behavior (a) and the lifting up a ball behavior by QRIO.

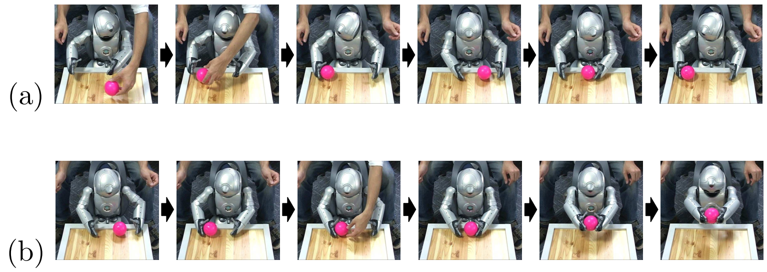


Figure 3: Snapshots of the rolling a ball behavior (a) and the lifting up a ball behavior (b).

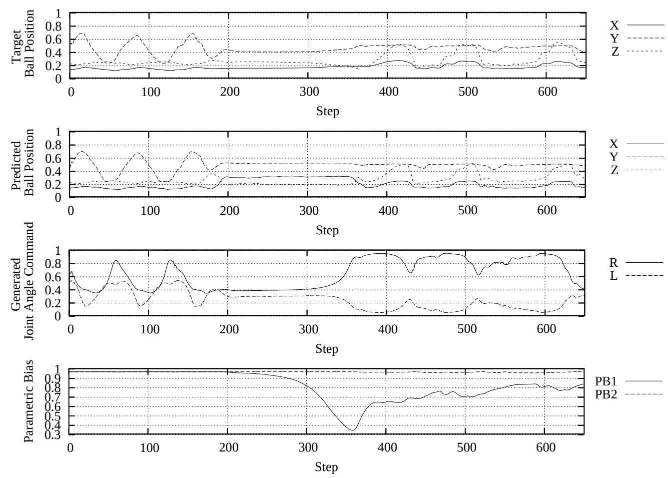


Figure 4: Dynamic generation and switching of two learned behaviors in the ball handling task. Plot in the top row represents the measured position of the ball. Plot in the second and the third row represent the ball position predicted and the robot joint angles generated by the RNNPB. Plot in the bottom row represents the parametric bias of the RNNPB.

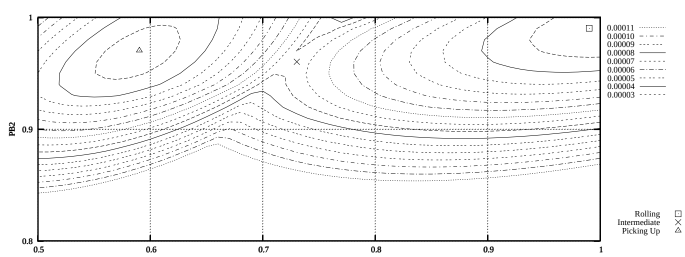


Figure 5: Prediction error distribution in the PB space. Only the area where the prediction error is less than 0.00011 is shown. Th right half contour indicates the area to realize ball rolling behavior and the left half contour indicates the area to realize the ball lifting up behavior.

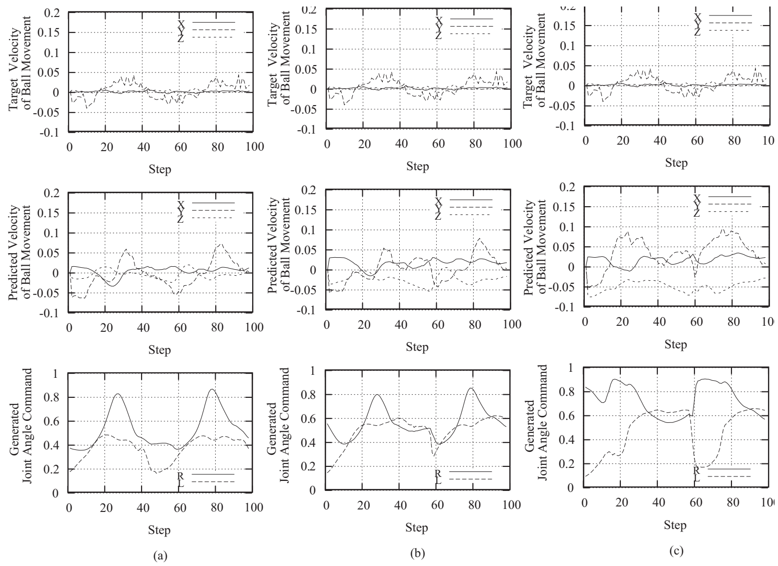


Figure 6: Ball rolling behavior according to constant PBs by the RNNPB which has learned two different types of ball handling behaviors. Each graph labeled with (a), (b) and (c) indicates the results that were acquired from the constant PB selected from the PB space where the prediction error corresponds with rolling, intermediate and lifting up behavior consequently. Plots in the top and second row represent the calculated target velocity of ball movement and the calculated velocity based on prediction by the RNNPB for comparing the performance of one-step prediction by the RNNPB. Plots in the bottom row represents the joint angles generated by the RNNPB.

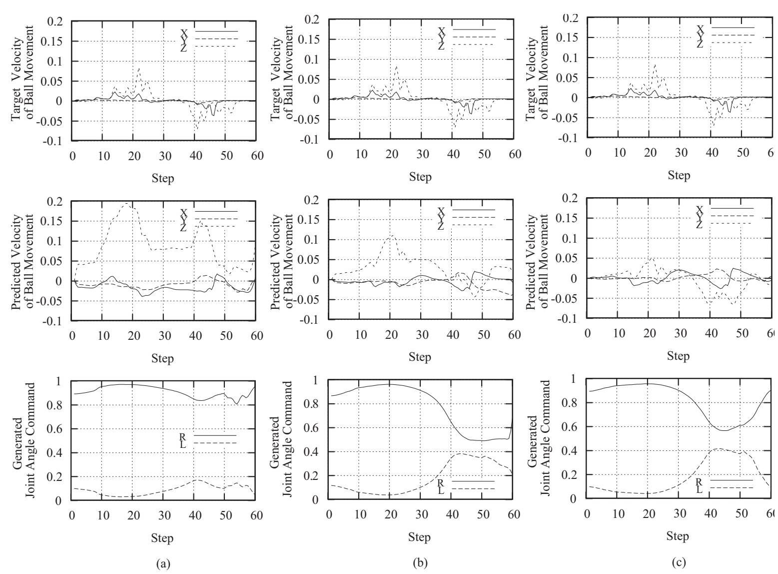


Figure 7: Ball lifting up behavior according to constant PBs. The labels below the graphs have the same meaning as figure 6.

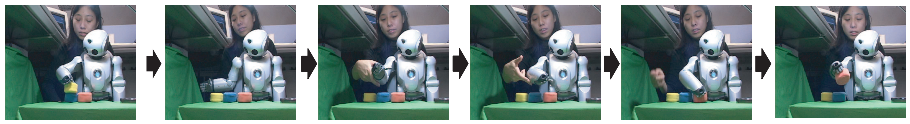


Figure 8: Snapshots of the behavior switching guided by the human user.

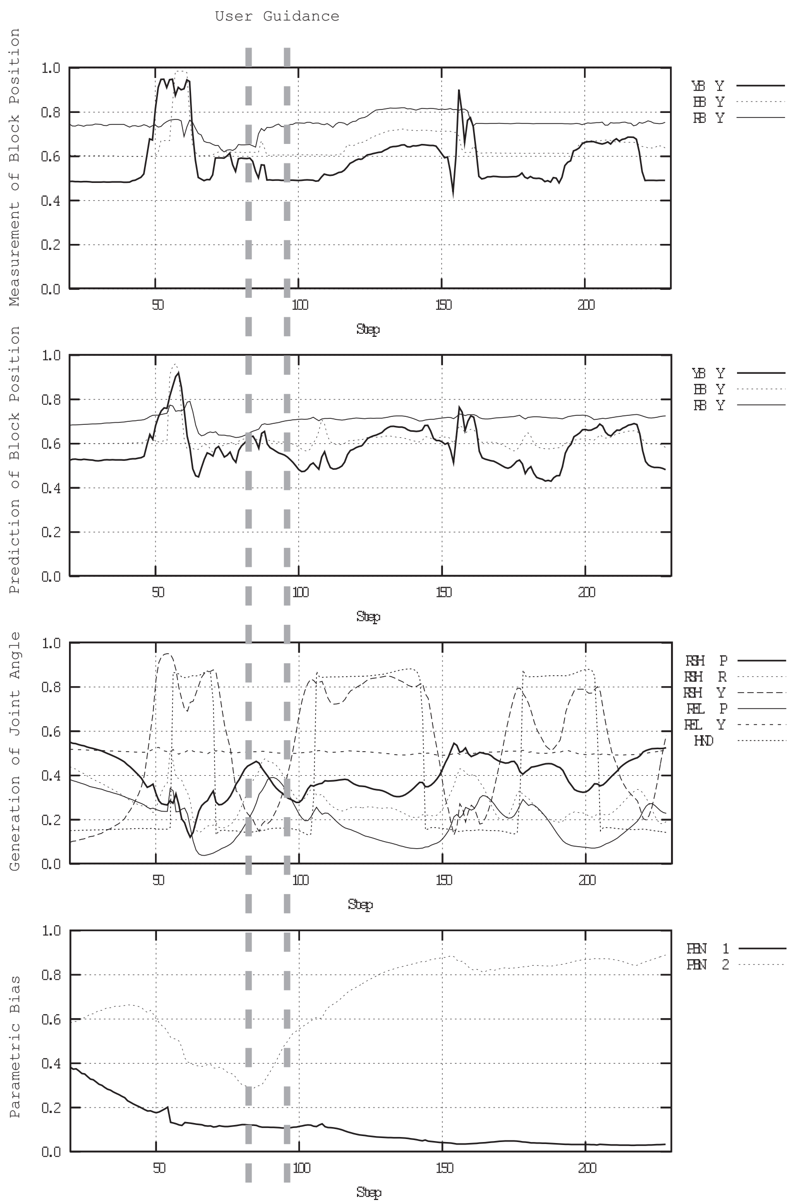
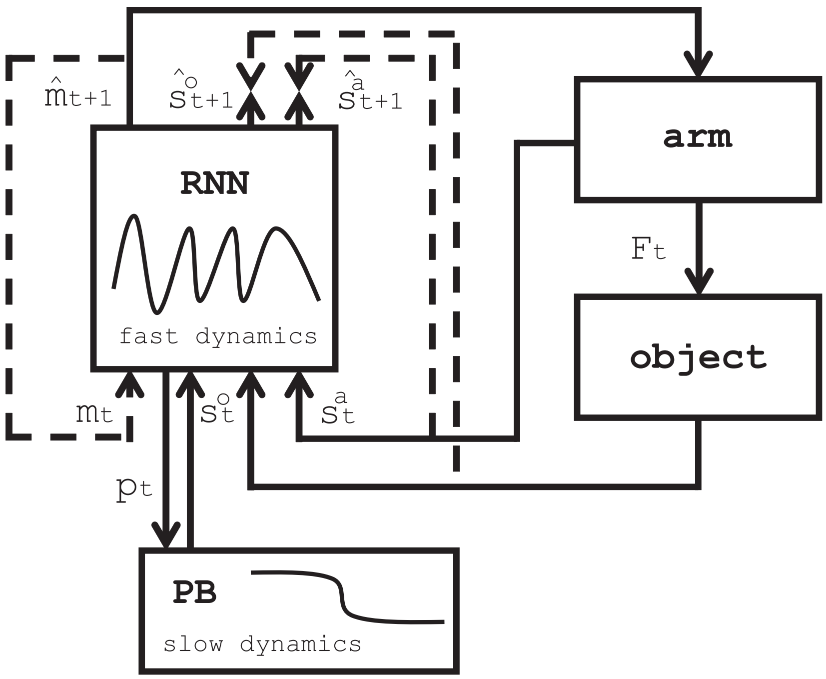
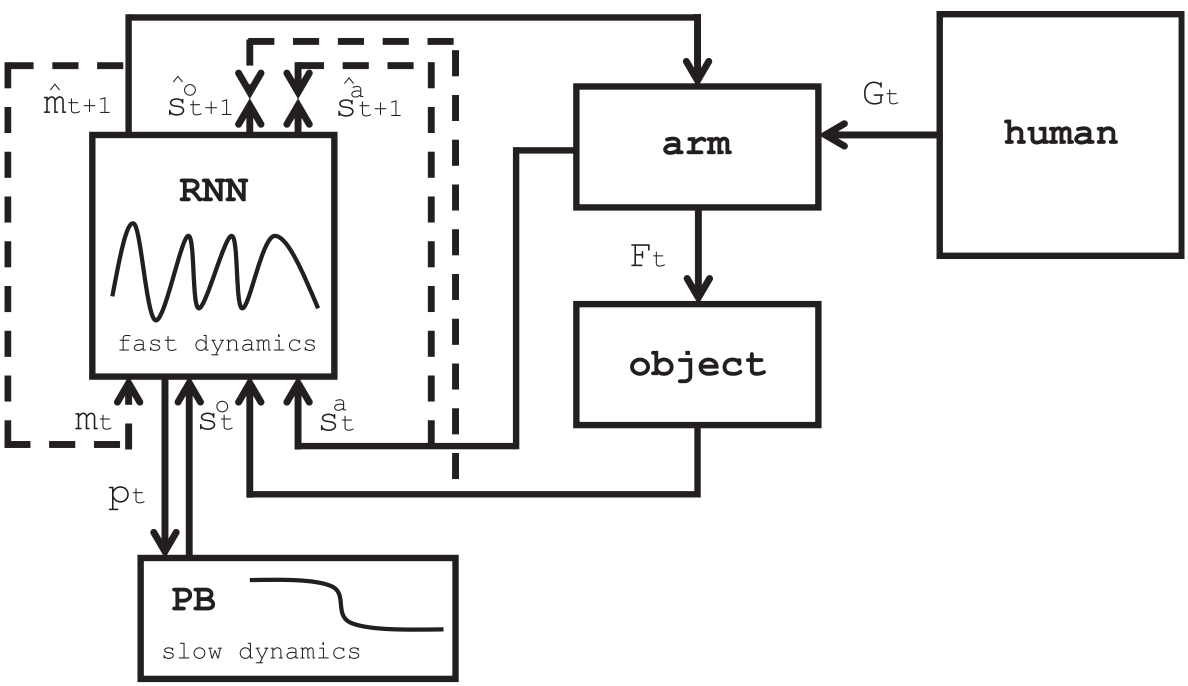


Figure 9: Guided behavior switching in the block building task. Plot in the top row represents the measured position of 3 blocks. Plot in the second and the third row represent the block position predicted and the robot joint angles generated by the RNNPB. Plot in the bottom row represents the parametric bias of the RNNPB.



(a)



(b)

Figure 10: Information flows in the ball handling task (a) and in the block handling task.

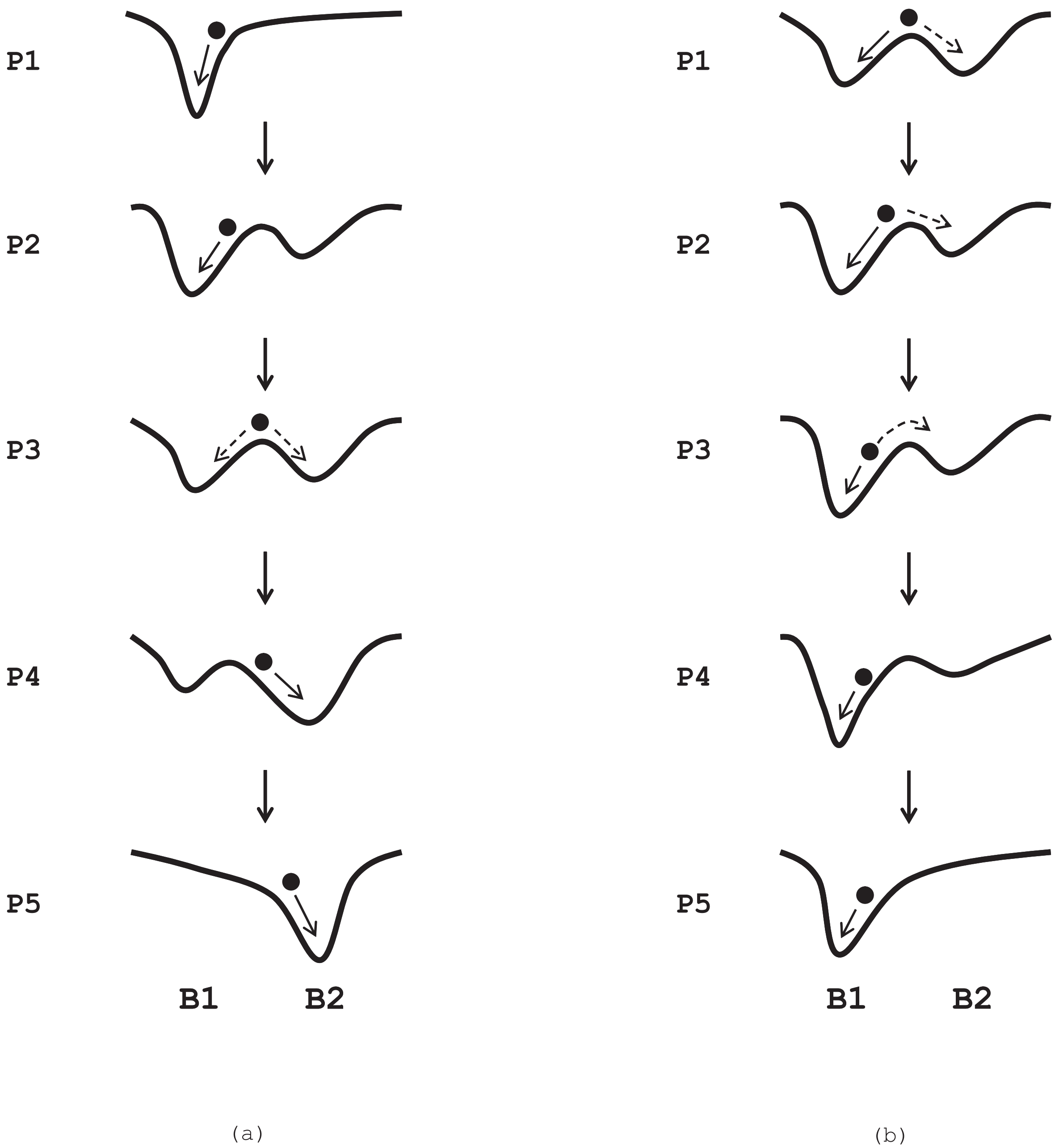


Figure 11: The transition of landscape of dynamical structure in the ball handling task (a) and in the block handling task (b).