The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Associating and Reshaping of Whole Body Motions for Object Manipulation

Hirotoshi Kunori, Dongheui Lee* and Yoshihiko Nakamura

***Abstract*— Since humanoid robots have similar body structures to humans, a humanoid robot is expected to perform various dynamic tasks including object manipulation. This research focuses on issues related to learning and performing object manipulation. Basic motion primitives for tasks are learned from observation of human's behaviors. An object manipulation task is divided into two types of motion primitives, which are represented as hidden Markov models (HMMs): one for a body motion primitive and the other for the relation between the object and body parts, which manipulate the object. When performing a task, a natural whole body motion is associated from an object motion by using learned motion primitives. Furthermore, the associated body motion is reshaped in both spatial and temporal space, in a more precise way. The reshaping in spatial space is realized in two stages by a feedback control policy learned with reinforcement learning and by constrained inverse kinematics. Key features like end-effectors for manipulation and timing for a task are extracted and used for the feedback control policy learning. The reshaping in temporal space is realized by comparing a predicted and observed object motion speed.**

I. INTRODUCTION

In order to coexist with humans in human society, robots need to understand human motions. In particular, for humanoid robots which have similar body structures to humans, the ability to learn motion patterns by imitating humans and to perform tasks instead of humans is highly desirable. Because tasks in daily life often involve tool-usage or interaction with objects, this paper studies on humanoid robot's intelligence for manipulation of objects including tools.

As robots move from industrial to human environments, robot learning will become an increasingly important skill for robots to master. Several researchers provide reviews in this area [1] [2] [3]. The imitation learning mechanism provides a means of automatic programming of complex systems without extensive trials or complex programming. Since the neuroscience evidence of motor primitives and mirror neurons in humans and other primates have been discovered [4], a number of researchers have developed models for robot imitation learning, inspired by the neuroscience evidences. Inamura et al. [5] proposed the mimesis model which is inspired by the bidirectional structure of mirror neurons for motion recognition and generation.

This research is supported by Japan Society for the Promotion of Science, "Category S of Grant-in-Aid for Scientist Research" and by Special Coordination Funds for Promoting Science and Technology, "IRT Foundation to Support Man and Aging Society".

The authors are with Department of Mechano-Informatics, University of Tokyo. 7-3-1 Hongo Bunkyo-ku, 113-8656 Tokyo, Japan.

∗ Corresponding author. Email: dhlee@ynl.t.u-tokyo.ac.jp

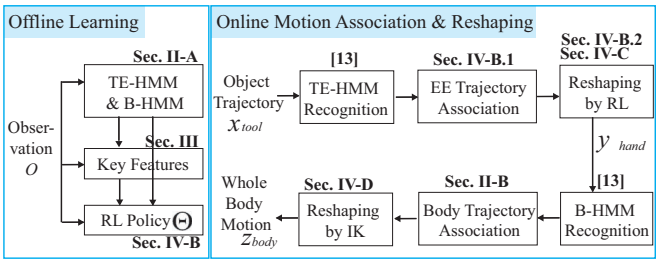

Fig. 1. Overall architecture for object manipulation tasks, consisting learning and online motion recognition-association-reshaping.

Most previous works on humanoid's imitation learning focus on robot's free body motion only [5] [6]. Relatively small amount of research has been carried out for humanoid tool-usages [7][8] or object manipulation [9]. Ogura et al. [7] presented geometric models of tools for humanoid motion generation. The information for grasping and for attention are embedded in the tool models. Nabeshima and Kuniyoshi [8] constructed a tool-use model for a blind retrieving manipulation task.

Recent neuroscience research [10] [11] revealed that, when primates use a tool to reach for a distant object, the extended motor capability is followed by changes in specific neural networks that hold an updated map of body shape and posture. This changes are compatible with the notion of the inclusion of tools in the 'body schema', as if our own effector (e.g. the hand) were elongated to the tip of the tool. Based on the above evidence that the tool is considered as a part of the body, the authors assume that during tool-use motion a tool motion triggers the appropriate motion for the rest of body.

The authors proposed a mechanism to associate a natural-looking whole body motion from a given tool motion [12]. When the tool trajectory is known, the tool trajectory becomes a trigger to generate an appropriate hand motion. The hand motion becomes the other trigger to associate a whole-body motion. By adopting the mimesis method from partial observations [13], an appropriate whole body motion (high dimensional movement) is generated from partial information, such as a tool trajectory (small dimensional movement).

Tasks which require interaction with objects including tools can be divided into two categories. One is the reaction to an object self motion and the other is object manipulation. In the former case, the object already has its own motion and a human reacts to the object motion. In the latter, the human manipulates the object in order to achieve a desired

object motion. For instance, the case when a human catches a flying paper-plane is an example of the former. The case that the human places the caught paper-plane on a table is an example of the latter. Our previous work which associates a whole body motion from an object motion corresponds to the latter. This paper extends the association mechanism [12] in order to solve the both cases.

- In [12], we assumed a fixed relation between a tool and a tool-driving end-effector: the right hand manipulates the tool and holds the tool firmly. However, end-effectors which manipulate objects are changing for different tool-tasks. A method to find important body parts and timing for a manipulation task is proposed.
- In [12], an associated whole body motion is synchronized to the tool motion. In this paper, the chosen motion primitive can be adapted in the spatial domain.
- In [12], a whole body motion is computed for a desired tool motion sequence. When considering "humanoid motion reaction to the object self motion", real-time motion adaptation should be realized by prediction based on current sensory data.

Therefore, this paper aims at solving two main issues:

- learning key features (e.g. important body parts and timing) of the tasks (Sec. III), and
- adaptation of motion trajectory and speed (Sec. IV).

The overview of this paper is shown in Fig. 1, consisting of offline learning and online motion recognition-association-reshaping. By observing human's demonstrations, an object manipulation task is learned (Sec. II-A). Key features for the manipulation task are extracted (Sec. III). Then, a control policy, how to achieve the task successfully, is learned by reinforcement learning (Sec. IV-B). Once object manipulation tasks are learned, the humanoid robot can perform the object manipulation tasks in a new situation. From a given object motion, the tool-manipulation motion is recognized and the appropriate end-effector motion is associated (Sec. IV-B.1). Then, the learned control policy is used to adapt the associated end-effector trajectory (Sec. IV-B.2). Also, motion speed is adapted (Sec. IV-C). From the reshaped end-effector motion, an appropriate whole-body motion is associated (Sec. II-B). In order to satisfy the reshaped end-effector motion, the associated body motion is reshaped by the constrained inverse kinematics method (Sec. IV-D).

II. BODY MOTION ASSOCIATION FROM TOOL MOTION [12]

A. Knowledge of Tool-Use Motion

Basic tool-use motion primitives are learned from real human data of tool-use motions. The time sequence of observed human motion patterns are abstracted through hidden Markov models (HMMs) by the EM algorithm [14].

Tool-use motion knowledge consists of two models; *tool manipulation knowledge* and *body motion knowledge*. Tool manipulation knowledge contains information of a tool and a corresponding end-effector (e.g. a grasping hand). The tool manipulation knowledge is embodied into "Tool-Effector HMMs" (TE-HMMs). Body motion knowledge contains motion information of full body including the grasping hand without tool information. The body motion knowledge is embodied into "Body HMMs" (B-HMMs). One can interpret that TE-HMMs and B-HMMs consider task space and joint space in imitation respectively, similar to [15].

By separating tool-use motion into two models[1], changes of the body schema with and without a tool can be implemented easily by including and excluding the tool manipulation model. Another advantage to having two models is richness of motion representability by combination. Since body motion knowledge, in which tool specific information is excluded, is general, same body motion knowledge can be used for different types of tools. Therefore, with the limited number of demonstrations, possible motion representation can be increased.

B. Whole Body Motion Estimation From Tool Motion

The authors developed an inference mechanism to associate a whole body motion from tool knowledge during the tool-usage task [12]. This mechanism can generate a natural high dimensional movement from the low dimensional information like a tool trajectory. The main concept of the method is illustrated in Fig. 2. To achieve a successful task during a tool-use motion, the trajectory of the tool is estimated. Then, the tool trajectory becomes a trigger in associating a whole-body motion. The best matching TE-HMM for the desired tool trajectory is found among a dataset of TE-HMMs by the algorithm of *motion recognition from partial observation*, which is proposed in [13]. An appropriate end-effector (a grasping hand) motion for the desired tool trajectory is estimated by the algorithm of *proto-symbol based duplication of an observed motion* [13].

The same inference method is applied for association of a whole-body motion from the end-effector motion. The best matching B-HMM for the estimated hand motion is found among a dataset of B-HMMs by the algorithm of *motion recognition from partial observation*. An appropriate full body motion is generated from the best B-HMM by the algorithm of *proto-symbol based duplication of an observed motion*.

If computing a whole body motion which satisfies the constrained trajectory only by inverse kinematics, there might be many possible solutions because human/humanoid model is highly redundant. Some solutions might not be natural-looking. Yamane et al. [16] addressed a similar problem in animation of manipulation tasks of human figures. In order to satisfy geometric constraints during manipulation tasks and to generate natural-looking motions, they proposed to combine an iterative inverse kinematics method and a data-driven method. In a similar sense, our proposed algorithm provides a natural whole body motion, because TE-HMMs and B-HMMs are trained via both a model-driven (based on kinematics) and a data-driven way (based on captured human

[1]Representing a tool-use motion with one HMM by combining a TE-HMM and a B-HMM is also technically possible. In such a case, the proposed method is still usable.

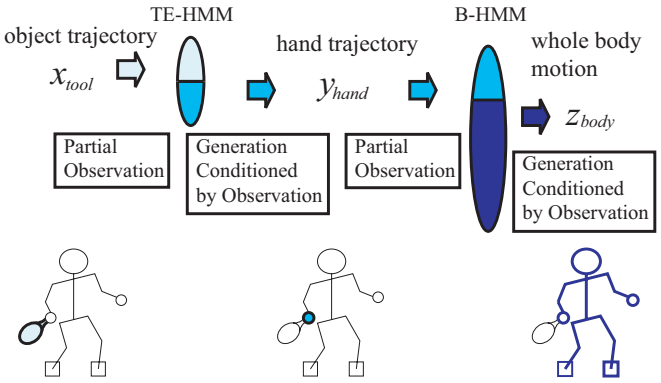

Fig. 2. Overview of the association model [12]. This model contains two kinds of HMMs. One is TE-HMM that represents tool-handling knowledge. The other is B-HMM that represents whole body motion. By the association model, a natural-looking whole body motion is associated from a tool motion.

motion patterns). Moreover, this is an efficient method for learning and reproducing tool-use tasks.

III. EXTRACTION OF KEY FEATURES

Depending on the task, the end-effector with which the robot manipulates objects varies. For instance, both hands are important for a golf full swing motion, and feet are important for kicking a soccer ball. This section proposes a method to find important *body parts* and *timing* for a certain manipulation task. The key insight is that important features are invariant for the multiple patterns of the same task. In other words, the important features have small variance at a certain moment. The detailed procedure is as follows.

Step1 Let M_i be the human motion sequence and O_i be the object motion sequence in the i-th observation of a certain manipulation task. From M_i and O_i, a time-sequence of the relative object position $X_{i,m}$ with respect to each body part is calculated. Here, m is the index of a body part.

Step2 For each body part, the parameters of an HMM λ_m are optimized by the EM algorithm from multiple observations $X_{i,m}$ of the same manipulation task.

Step3 Check the variance of each state of each HMM λ_m. If the variance is smaller than the predefined threshold, the corresponding body part to the HMM is an important body part and the state is an important timing for the task.

By the above procedure, the key features, in particular body parts and timing, for the given task can be extracted.

The proposed method was implemented to extract the important manipulating end-effector and timing for a ball catching task. Ball catching motions of a right-handed person are captured with a sampling time of 5 [ms]. Pre-segmented eleven motion patterns, whose average duration is 341 [ms], are used. Six body parts like head, hip, right hand, left hand, right foot, and left foot are considered as candidates of manipulating end-effectors. The relative ball position and orientation with respect to each body part is used for training

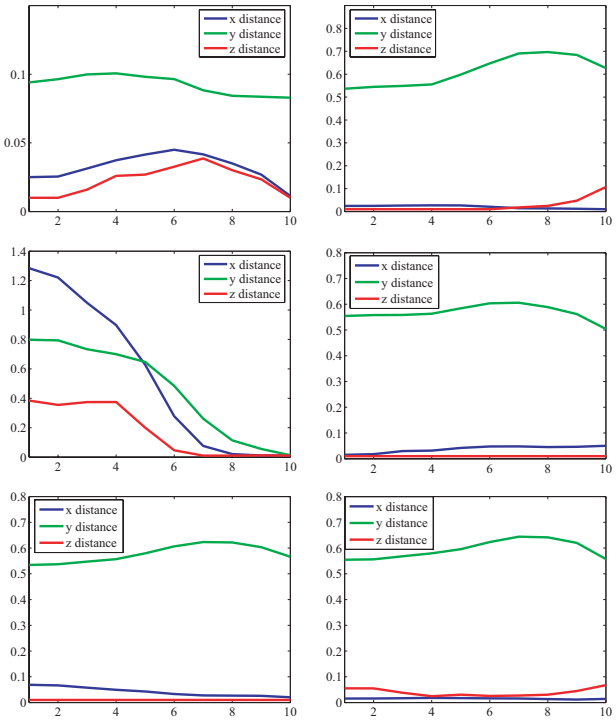

Fig. 3. Variance in [m] at each state of each HMM, which is learned from relative object positions to each body part. The body parts are head (top left), hip (top right), right hand (middle left), left hand (middle right), the right foot (bottom left), and the left foot (bottom right).

each HMM. A left-to-right type HMM with 10 states is used. The threshold is set to 0.02 [m].

Figure 3 illustrates the variance of output probability at each state of each HMM. The horizontal axis is the state index and the vertical axis is the variance value for the relative ball position in x, y, and z axis. This shows that variances for the relative ball position (x,y,z) with respect to the right hand are almost zero at the 10th state. This implies that the right hand is the important body part, and the 10th state, which corresponds to the ball catching moment, is the important timing for the ball catching task. If applying this strategy for a right-foot-kicking task, the right foot will be detected as the important body part.

IV. REACTIVE ASSOCIATION MODEL FOR DYNAMIC TASKS

A. Reactive Association Model

For general manipulation tasks, a humanoid robot requires to interact with objects in an environment. For instance, in order to catch a flying paper-plane, the trajectory and speed of the whole body motion should be adapted to the paper-plane in real-time.

This paper proposes a new method how to associate and modify a whole body motion for an object manipulation task (Fig. 4). In order to apply the proposed association and reshaping method, the robot should learn the object manipulation task a prior. By learning from observations,

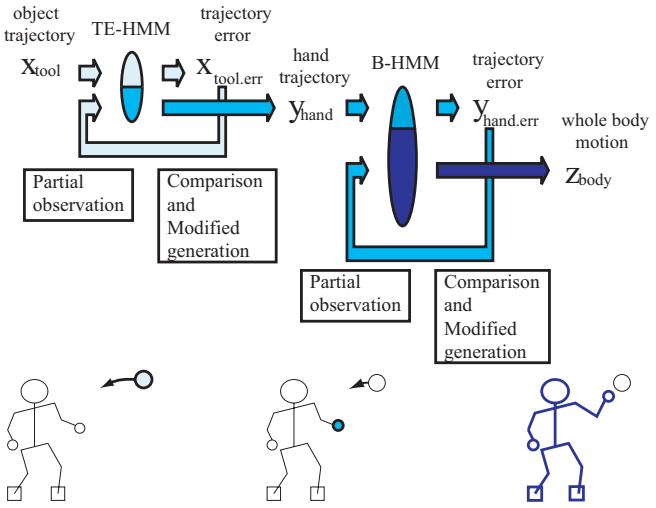

Fig. 4. Overview of the proposed model of association and reshaping. This contains two inference steps based on TE-HMMs and B-HMMs. Both HMMs calculate the error between predictive and actual trajectories and generate time series data based on the error. TE-HMM's feedback is realized by the natural actor-critic algorithm and B-HMM's is realized by the constrained inverse kinematics.

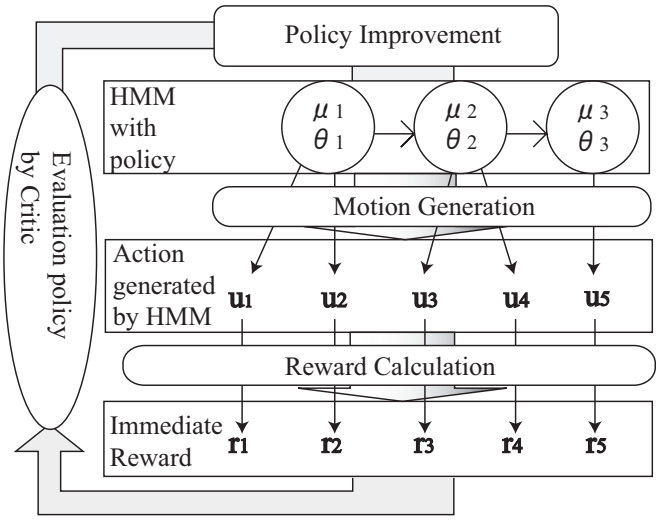

Fig. 5. Flowchart of policy improvement by a natural actor-critic algorithm, which is a fast reinforcement learning method. First, the actor generates a whole body motion pattern according to the policy. Second, the actor gets immediate rewards. Third, the critic evaluates actor's behavior and improves actor's policy.

the robot learns basic motion primitives (TE-HMM and B-HMM) (Sec. II-A) and key features (Sec. III) of the task. Also, a control policy, how to adapt an end-effector motion primitive to achieve the task successfully, is learned by reinforcement learning (Sec. IV-B). This control policy learning is implemented, aiming at embodying "transferred skill improvement by practice" and dealing with object motions which are hard to predict (e.g. a paper-plane motion and a swimming fish motion).

After learning the motion primitives, the key features, and the control policy, the humanoid robot can perform the object manipulation task in a new situation. From a given object motion, the correct TE-HMM is found by the algorithm of *motion recognition from partial observation* [13] and the appropriate end-effector motion is associated (Sec. IV-B.1). Then, the learned control policy is used to adapt the associated end-effector motion in the current dynamic situation (Sec. IV-B.2). Also, motion speed is adapted by modifying the state transition probabilities of the HMM based on difference of the predicted and observed speed (Sec. IV-C).

From the reshaped end-effector motion, an appropriate whole-body motion is associated by the inference mechanism to associate whole body motion (Sec. II-B). In order to satisfy the reshaped end-effector motion, the associated body motion is reshaped by the constrained inverse kinematics method [17] (Sec. IV-D).

B. End-Effector Trajectory Adjustment by Reinforcement Learning

This section introduces how to learn a policy to modify an end-effector trajectory and how to use the learned policy to adjust an end-effector trajectory in action. The control policy is improved by the natural actor-critic (NAC) algorithm [18], which is a fast reinforcement learning method. As shown in Sec. II-A, an object manipulation motion is represented as a TE-HMM, which is corresponding to the key end-effector (Sec. III), and a B-HMM. Based on the TE-HMM, the key features, and observations, the policy is learned by the NAC. In this paper, the TE-HMM is the actor to generate an end-effector motion trajectory based on the control policy. Simply saying, from a current object trajectory (sensory input), the parameters of the TE-HMM (output) are modified. From the modified HMM parameters, the end-effector motion is reshaped.

1) Motion Generation from an HMM: A motion pattern is decoded from a selected TE-HMM either in a deterministic [6] or stochastic way [13]. The motion generation has two processes: generation of the state sequence and generation of the output motion. The state sequence, which is temporal information of the motion primitive, is decoded from the state transition probability matrix and the initial state probability vector. Once the state sequence has been generated, the spatial information at each state is decoded from the output probability distribution, which is represented as a Gaussian distribution.

2) Reshaping the output probability function of a TE-HMM: An end-effector motion primitive, which is generated from a TE-HMM, should be reshaped in accordance to an object motion in a current situation. The expected motion pattern from the TE-HMM is compared to the observed sensory data. From the difference, the motion primitive is reshaped. The concept is described in Fig. 6 and its detailed procedure is described as follows.

Step1 Calculate the expected mean vectors for the j-th state in the TE-HMM, by simply averaging the

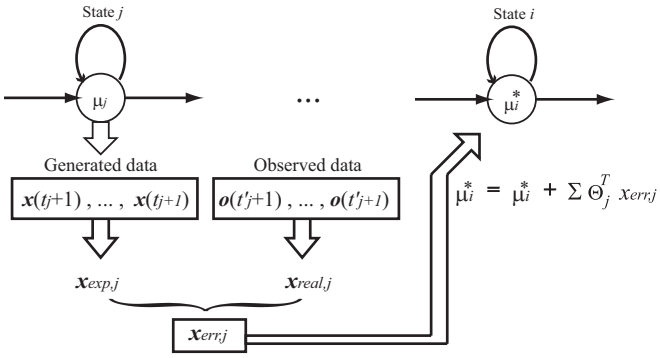

Fig. 6. The mean vector of the *i* th state (extracted key timing) is modulated by the difference between the observed and expected data of the *j* th state (states for prediction) of the HMM.

generated motion sequence from the state.

$$\boldsymbol{x_{exp,j}} = \frac{1}{D_j^*} \sum_{\forall q_t = j} \boldsymbol{x_t} \quad ,$$

where D_j^* is the duration that the generated motion pattern corresponds to the j-th state. Term q_t denotes the state at time t for the data sequence. Here, the j-th state implies the state for prediction.

Step2 Estimate the state-sequence for the observed motion pattern. Then, calculate the observed mean vectors for the j-th state, by simply averaging the observed motion sequence corresponding to the j-th state.

$$\boldsymbol{x_{real,j}} = \frac{1}{D_j} \sum_{\forall q_t = j} \boldsymbol{o_t} \quad ,$$

where D_j is the duration that the observed motion pattern corresponds to the j-th state.

Step3 Calculate the error between the expected and observed motion pattern for the j-th state by

$$\boldsymbol{x_{err,j}} = \boldsymbol{x_{real,j}} - \boldsymbol{x_{exp,j}} \quad .$$

Step4 Let i be the state corresponding to extracted key timing, which is explained in Sec. III. Based on the error $\boldsymbol{x_{err,j}}$ of the j-th state, the mean vector of the output probability at the i-th state is modified by

$$\mu_i^* = \mu_i + \sum \boldsymbol{\Theta_j^T x_{err,j}} \quad , \tag{1}$$

where $\boldsymbol{\Theta_j}$ is a feedback gain matrix. The parameter $\boldsymbol{\Theta_j}$ is the trained control policy by the NAC.

3) Reward Calculation: The reward is an essential part of reinforcement learning. From the modified TE-HMM by (1), a motion pattern $X_{mod} = \{x_{mod,t}\}$ is generated. Then, the actor TE-HMM gets immediate rewards r_t at time t. The reward is calculated based on the extracted key features (body parts and timing) of the task, which is explained in Sec. III. If the key timing is the i-th state of the TE-HMM, the generated motion sequence $x_{mod,t,i}$ from the i-th state of the modified TE-HMM and the mean vector μ_i of the originally trained TE-HMM are compared. The reward is designed as a penalty function like

$$r_t = \sum_{\forall q_t = i} |x_{mod,t,i} - \mu_i| \quad . \tag{2}$$

Since the key motion elements have small variance at the key moment, simply the mean vector μ_i can be compared instead of considering the exact output probability function of the i-th state. By this reward function, the NAC reshapes the motion primitive to follow the key features of the task. From the immediate reward r_t at time t, the average reward r_e for each episode is calculated by

$$r_e = \frac{1}{D_i^*} \sum_{\forall q_t = i} r_t \quad , \tag{3}$$

where D_i^* is the time duration corresponding to the i-th state.

One can see that the reward is not provided manually by a supervisor, but automatically based on the extracted key features. Thus, the same mechanism can be applied for complicated tasks without the need of re-designing a different optimizing function for a different task.

4) Updating parameters: The control policy is updated by the NAC algorithm [18]. The NAC is chosen because it is expected to handle high dimensional movements and to provide faster convergence to the nearest local minima. The outline of policy improvement mechanism is illustrated in Fig. 5. The natural gradient w and the fisher information matrix F are designed as

$$w = F^{-1} J \tag{4}$$

$$F = \sum_e \hat{\phi}_e \hat{\phi}_e^T \tag{5}$$

$$J = \sum_e \hat{\phi}_e r_e \tag{6}$$

$$\hat{\phi}_e = \sum_{t=T_i}^{T_{i+1}} \gamma^t [\nabla_{\Theta_j} \log \pi(x_{mod,t,i}|q_t)^T]^T \tag{7}$$

where γ is a forgetting factor.

$$\nabla_{\Theta_j} \log \pi(x_{mod,t,i}|q_t) = \frac{(x_{mod,t,i} - \mu_i^*)\boldsymbol{x_{err,j}}}{\sigma_i}$$

When the natural gradient $\boldsymbol{w}$ is converged over a window, the policy parameter Θ_j is updated by

$$\Theta_j \leftarrow \Theta_j + \alpha \boldsymbol{w} \tag{8}$$

with a learning factor α. Some of sufficient statistics are forgotten with a forgetting factor β.

$$F \leftarrow \beta\ F \tag{9}$$

$$J \leftarrow \beta\ J \tag{10}$$

If simply using the natural gradient, sometimes the changes of the policy parameter are very big and the updated policy moves beyond the local minima. Therefore, we added the upper limit on the natural gradient w. Another issue for learning the control policy is selection of training data set. If the observations in the training set are very close to

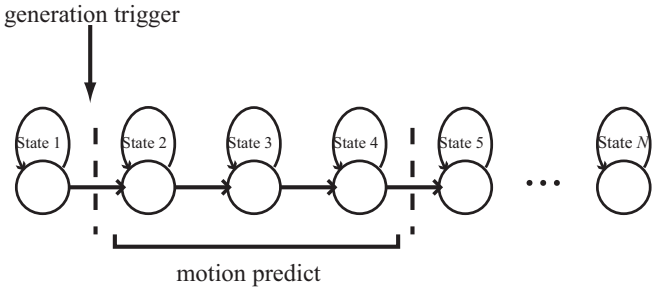

Fig. 7. An HMM starts to generate a motion pattern when recognizing a state transition from the 1st state to the 2nd state.

each other, the updated parameters via NAC are prone to overfitting. Thereafter, when a new object motion (sensory input) is very different from the training data, the motion primitive (output) might not be reshaped well.

C. End-Effect Motion Speed Adjustment

In order to achieve a successful task, the robot needs to adapt its motion with respect to temporal and spatial variability. This section explains how to adjust for temporal changes. In particular, two main issues should be considered: when to start the motion and how to change motion speed.

Since left-to-right type HMMs are used for representation of motion primitives, the motion patterns always start from the most left state and finish at the most right state. When recognizing the state transition from the jth state to the $j+1$th state, the moment is taken as a trigger for generating a motion, as shown in Fig. 7.

In the left-to-right type HMMs, the expected state duration at the ith state is $1/(1-a_{i,i})$, where $a_{i,i+1}$ is the probability to transit from the ith state to the $i+1$th state [6]. By estimating the state-sequence for the observed motion pattern, the real state duration D_i for the ith state is known. By comparing the expected and real state duration, the motion speed can be adjusted. The motion speed scale s^* is calculated by

$$s^* = (D_{i-1} + D_i)/(\frac{1}{1-a_{i-1,i-1}} + \frac{1}{1-a_{i,i}}) \quad . \qquad (11)$$

With the calculated scale factor s^*, the state transition matrix is modified to $a^*_{j,j}$. Since the modified state duration is required to be the same as the real state duration, which implies that

$$\frac{1}{1-a^*_{j,j}} = \frac{1}{1-a_{j,j}} s^* \quad ,$$

the state transition matrix is modified by

$$a^*_{j,j} = 1 - \frac{1-a_{j,j}}{s^*} \quad , \qquad (12)$$

where j is the state index. Equation (12) is calculated for all states and

$$a^*_{j,j+1} = 1 - a^*_{j,j}$$

is calculated for all states except the last state.

D. Whole Body Motion Adjustment by IK

So far, we explained how to adjust the trajectory and speed of the key body part (end-effector) motion to achieve the successful task in accordance to the dynamic object. This section describes how to generate a whole body motion which is consistent with the adjusted end-effector trajectory.

If using the conventional association model shown in Fig. 2, the best matching B-HMM for the adjusted end-effector trajectory is found and an appropriate full body motion is generated. However, the associated whole body motion cannot guarantee to follow exactly the same adjusted end-effector trajectory. Therefore, the associated whole body motion is modified with two constraints.

- Keep the adjusted trajectory of the end-effector exactly as long as robot kinematics allows.
- Two feet are contacted on the ground. If considering only the previous constraint, the generated motion might be a floating motion in the air.

The associated whole body motion is reshaped to satisfy the above constraints by applying the constrained inverse kinematics method [17].

V. EXPERIMENTAL RESULTS

The proposed concepts are tested on a motion data containing three different types of ball catching motions: under catch, over catch, and side catch. Eleven motion patterns of human's ball catching task are obtained through a motion capture system. The data set contains multiple observations of under catch (4 observations whose average length is 73 frames), over catch (5 observations whose average length is 61 frames), and side catch (2 observations whose average length is 76 frames). The sampling rate for each frame is 5 [ms].

From randomly selected observations among the data-set of each ball catching type, a TE-HMM and a B-HMM are learned via the EM algorithm, namely 3 TE-HMMs and 3 B-HMMs in total. Each HMM consists of 10 states and is a left-to-right type. Key features are extracted for the ball catching task. Section III shows that the right hand configuration at the 10-th state is the key feature for the ball catching task[2].

A. Policy Learning

A policy how to reshape a motion primitive for end-effector motion is learned by the NAC, which is explained in Sec. IV-B. We tested the policy learning on the "under catch" motion. In the experiments, the two states (the 2nd and 3rd state) of the HMM are used for prediction. From the difference between the expected and observed data for these two states, the mean vector of the output probability function of the 10th state (extracted key timing) is modified. The policy is learned from multiple observations: three observations of the under catch motion, which are used for TE-HMM and B-HMM. The penalty during the policy learning is illustrated in Fig. 8. The figure shows that penalty becomes smaller as learning proceeds.

[2]The demonstrator was a right-handed person.

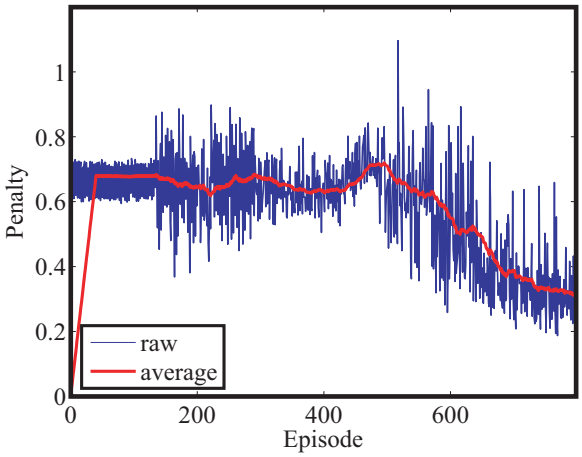

Fig. 8. Policy Learning result.

B. Adaptation of Motion Trajectory

The learned policy Θ_j is tested on a new "under catch" motion, which is used for training neither the "under catch" TE-HMM nor the policy Θ_j. Ten trials are carried out with and without the learned policy. For the expected motion pattern, ten different motion patterns are generated from the TE-HMM in a stochastic way. The penalty for each trial and the average penalty are shown in table I. When using the learned policy, the average penalty is about 0.40. When not using the policy, the average penalty is about 0.71. This results show the soundness of the policy learning. The trail 10 shows the relatively small improvement because in this trail the stochastically generated motion pattern was not a good expectation.

In the case of ball catching, the penalty function (3) implies the positional and rotational distance between the right hand and the ball. In other words, the penalty values in table I implies that the humanoid robot moves its right hand much closer to the ball by using the learned policy, compared to without the policy.

TABLE I
PENALTY OF EACH EPISODE

Trial	With learned policy	Without learned policy
1	0.397136	0.710987
2	0.247843	0.707972
3	0.392468	0.710516
4	0.279734	0.719023
5	0.466914	0.714592
6	0.452610	0.712177
7	0.503462	0.714383
8	0.379811	0.716159
9	0.193874	0.724841
10	0.710112	0.716237
average	0.4023964	0.7146887

From the reshaped right-hand motion according to the ball trajectory, whole body motion is also reshaped by the the inverse kinematics method with constraints, explained in Sec. IV-D. The snapshots of a generated motion with reshaping strategies are shown in Fig. 9 (top). The snapshots of a generated motion without reshaping strategies, only by the association model (Sec. II-B), are shown in Fig. 9 (middle). The reshaped motion according to the ball motion is commanded to a human-size humanoid robot. The snapshots are shown in Fig. 9 (bottom). In order to ensure the dynamic stability of the humanoid robot, a COG (center of gravity) based balance controller [19] is adopted for the control of lower body joints, which simultaneously realizes the hip motion.

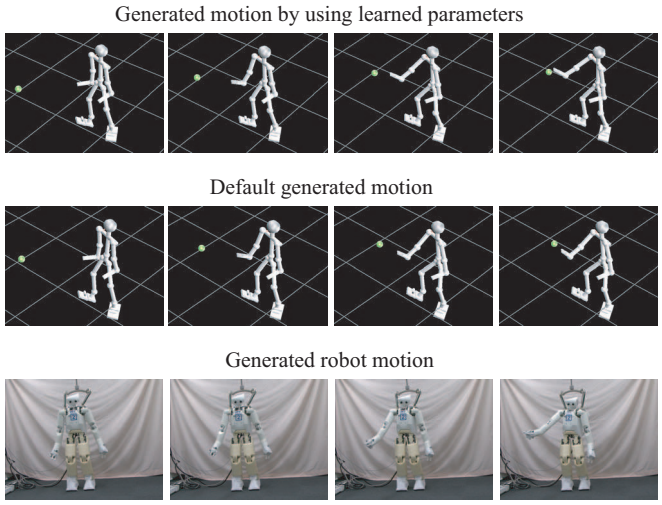

Fig. 9. Snapshots during catching a ball. (Top) A robot generated a modified motion reacting to a ball in simulation. (Middle) A robot generated a default motion from an HMM in simulation. (Bottom) The reshaped motion according to the ball motion is commanded to a humanoid robot.

C. Adaptation of Motion Speed

This section shows the evaluation of motion speed adaptation. The motion speed adaptation is tested on two over-catch movements with different speed. Both observations are new patterns, which are not used for training the corresponding HMMs and policy, to the robot.

Figure 10 illustrates how body motions are modified by the ball speed. In particular, a trajectory of the right shoulder pitch joint angle and a ball position as it moves toward the human are shown. The solid lines correspond to the fast ball motion and the dashed lines correspond to the slow ball motion. Until 0.03 [sec], the shoulder movements are almost same in both slow and fast speed. It shows that this initial 0.03 [sec] is the period for prediction of the current situation. After the prediction period, the motion speed is adjusted for the ball speed. One can see that the shoulder joint moves faster for the fast ball motion compared to the slow ball motion.

VI. CONCLUSIONS AND FUTURE WORKS

This paper proposes a new method to associate and reshape a whole body motion according to an object movement in object manipulation tasks. Motion reshaping methods for

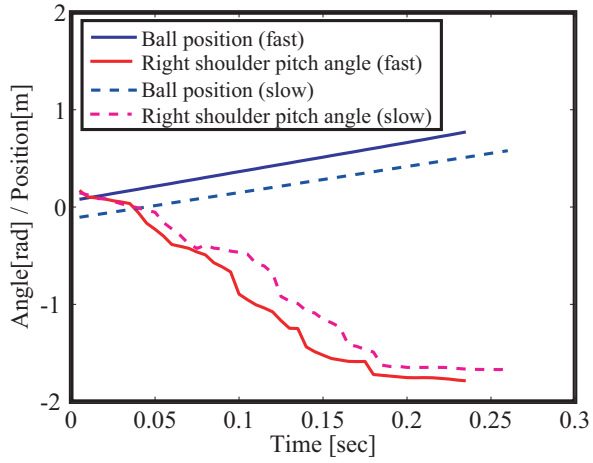

Fig. 10. Changes of joint angle affected by ball speed. Solid blue and red line is position of fast ball and the corresponding joint angle of right shoulder. Broken purple and pink line is position of slow ball and the corresponding joint angle of right shoulder.

dynamic changes are integrated with the model of body motion association from tool motion. The motion reshaping is carried out in both spatial and temporal variabilities. The reshaping is realized by integrating reinforcement learning for end-effector motions and by constrained inverse kinematics for whole-body motions. For temporal variabilities, issues like when to start the motion and how to change speed are solved. A method to extract key features for a task is proposed. The extracted key features become an important basis to design reward functions during the reinforcement learning and this leads to unsupervised learning.

In future work, the proposed method will be implemented on a complicated task (e.g. catching a flying paper-plane and placing it on a table) with a human-size humanoid robot in real-time. The task will be a combination of robot's reaction to a moving object, whose motion is hard to predict, and manipulation of an holding object. When generating a whole body motion pattern, the active use of the previous knowledge (e.g. the preference of motion types and the appropriation of motion types according to situations) will be considered. During a tool-use task, the constraints between an end-effector and a tool are changing time-to-time. Such time-variant constraints will be studied.

REFERENCES

[1] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, vol. 358, pp. 537–547, 2003.

[2] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Science*, vol. 6, no. 11, pp. 481–487, 2002.

[3] V. Krueger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: A review on action recognition and mapping," *Advanced Robotics*, vol. 21(13), pp. 1473–1501, 2007.

[4] G. Rizzolatti and L. Craighero, "The mirror-neuron system," *Annual Reviews of Neuroscience*, vol. 27, pp. 169–192, 2004.

[5] T. Inamura, Y. Nakamura, and I. Toshima, "Embodied symbol emergence based on mimesis theory," *Int. Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[6] D. Kulić, W. Takano, and Y. Nakamura, "Representability of human motions by factorial hidden markov models," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.

[7] T. Ogura, Y. Sagawa, K. Okada, and M. Inaba, "Humanoid tool operating motion generation by model with attention points," in *Annual Conf. of the Robotics Society of Japan*, 2005, p. 1F15, in Japanese.

[8] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Towards a model for tool-body assimilation and adaptive tool-use," in *The 6th IEEE Int. Conf. on Development and Learning*, 2007, pp. 288–293.

[9] K.Sugiura and N. Iwahashi, "Learning object-manipulation verbs for human-robot communication," 2007.

[10] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, vol. 8, no. 2, pp. 79–86, 2003.

[11] A. Iriki, M. Tanaka, and Y. Iwamura, "Coding of modified body schema during tool use by macaque postcentral neurons," *Neuroreport*, vol. 7, no. 14, pp. 2325–2330, 1996.

[12] D. Lee, H. Kunori, and Y. Nakamura, "Association of whole body motion from tool knowledge for humanoid robots," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 2867–2874.

[13] D. Lee and Y. Nakamura, "Mimesis from partial observations," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 1911–1916.

[14] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77(2), pp. 257–286, 1989.

[15] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 367–372.

[16] K. Yamane, J. Kuffner, and J. Hodgins, "Synthesizing animations of human manipulatioin tasks," vol. 23, no. 3, pp. 532–539, 2004.

[17] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," in *IEEE Transactions on visualization and computer graphics, Vol. 9, No. 3, pp.352-360*, 2003.

[18] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement Learning for Humanoid Robotics," 2003.

[19] C. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2008, pp. 399–405.