Incremental Online Learning of Robot Behaviors From Selected Multiple Kinesthetic Teaching Trials

Sumin Cho and Sungho Jo, *Member, IEEE*

***Abstract*—This paper presents a new approach to the incremental online learning of behaviors by a robot from multiple kinesthetic teaching trials. The approach enables a robot to refine and reproduce a specific behavior every time a new teaching trial is provided and to decide autonomously whether to accept or reject each trial. The robot neglects bad teaching trials and learns a behavior based on adequate teaching trials. The framework of this approach consists of the projection of motion data to a latent space and the description of motion data in a Gaussian mixture model (GMM). To realize the incremental online learning, the latent space and the GMM are refined incrementally after each proper teaching trial. The trial data are discarded after being used. The number of Gaussian components in the GMM is not initially fixed but is autonomously selected by the robot over the trials. The proposed method is more suitable for practical human–robot interaction. The experiments with a humanoid robot show the feasibility of the approach. We demonstrate that the robot can incrementally refine and reproduce learned behaviors that accurately represent the essential characteristics of the teaching trials through our learning algorithm and that it can reject erroneous teaching trials to improve learning performance.**

***Index Terms*—Gaussian mixture model, incremental learning, learning by imitation, learning from demonstrations, robots.**

I. Introduction

Teaching a robot using multiple demonstrations, also referred to as learning by imitation, is a useful and convenient approach to teaching a robot new behavioral skills [1]–[4]. Several advantages to the approach are well summarized in the literature [5]. In learning by imitation, a robot observes multiple demonstrations of a repeated behavior or is taught kinesthetically over several trials. The robot responds by reproducing the generalized motions.

Most of the previous approaches require full teaching information at once [6]–[9]. However, to be more adequate and applicable in real human–robot interactions, teaching methods for skillful behaviors should be dynamic and in real time. As if a human is taught incrementally, teaching trials would be performed sequentially, and the robot should start learning a behavior online from the first trial. The robot should be able to further refine its behavior with repeated trials. In addition, during the learning process, the robot should be able to evaluate the data actively rather than accepting all of the data passively. The robot can thus focus exclusively on good teaching information to improve its performance. A teacher is also informed of his or her teaching performance.

The EPFL LASA group [8]–[12] has presented an approach to behavioral learning from full demonstrations. Their approach is based on principal component analysis (PCA) and a Gaussian mixture model (GMM) to build a probabilistic representation of behaviors. The approach can classify motion segments and reproduce smooth generalized movements. They proposed an incremental version of the batch-learning algorithm [10]. However, their version requires a fixed number of components in the GMM and the latent space prior to the learning procedure, and it does not take into account data alignment.

This paper describes an approach to the incremental online learning of behaviors by a robot from properly selected kinesthetic teaching trials. A robot encodes an expected behavior and projects it onto a latent space of motion and a GMM. The robot modifies the latent space evenly and progressively refines the number of Gaussian mixture components in the GMM whenever a new teaching trial is provided. Each trial data is not stored once used, and no preliminary assumption is required for the GMM and the latent space. Furthermore, for teaching sessions of the same motion, the teaching data are aligned incrementally. The temporal alignment clarifies the coherence of the data trajectories of the repeated behavior, and it normalizes the raw data temporally. Our approach enables a robot to recognize and reject bad teaching trials autonomously. The autonomous recognition is made possible by comparing with a generalized behavior from teaching trials in terms of spatial coherence and pattern similarity. By rejecting bad teaching trials, the robot can learn a motion more skillfully and proficiently. The robot will also be able to provide feedback to a teacher on his/her teaching performance, thus adding another dimension to the interactive learning scheme.

II. Algorithm

This paper proposes an incremental online learning of behaviors from proper kinesthetic teaching trials, as described in Fig. 1. To select proper teaching trials for learning, the data evaluation step is employed. This work uses the GMM in the latent space as a motion descriptor and aims to select model parameters in the GMM and to construct the latent space properly soon after each proper teaching trial is provided.

GMM has widely been used to modeling a probabilistic density function and has been applied to various applications such as motion pattern classification [8]–[12] and speech recognition [13]. GMM is known to be good at describing the essential variations of complex and nonlinear patterns even with noisy data, while its analysis can be done easily [14]. The probabilistic density function can be represented as

$$p(x_i) = \sum_{k=1}^{K} \pi_k p(x_i|k) \tag{1}$$

where

$$p(x_i|k) = \mathcal{N}(x_i;\mu_k,\Sigma_k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_k|}} \exp\left(-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right)$$

π_k is the prior probability, and μ_k and Σ_k are the mean vector and the covariance matrix of the kth Gaussian component, respectively.

Manuscript received August 7, 2011; revised January 2, 2012 and May 14, 2012; accepted May 24, 2012. Date of publication September 12, 2012; date of current version April 12, 2013. This work was supported by National Research Foundation of Korea under Grant 2011-0003572, and by the Korea Government (MKE) under the Human Resources Development Program for Convergence Robot Specialists. This paper was recommended by Associate Editor G. C. Calafiore.

The authors are with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: jerachiel7@kaist.ac.kr; shjo@kaist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2012.2207108

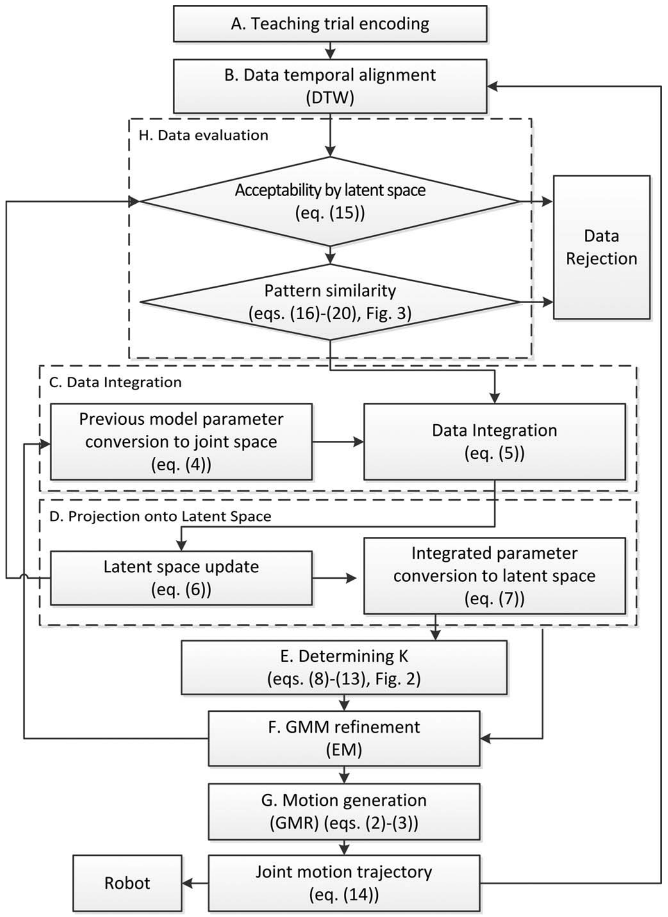

A. Teaching trial encoding
B. Data temporal alignment (DTW)
H. Data evaluation
Acceptability by latent space (eq. (15))
Pattern similarity (eqs. (16)-(20), Fig. 3)
Data Rejection
C. Data Integration
Previous model parameter conversion to joint space (eq. (4))
Data Integration (eq. (5))
D. Projection onto Latent Space
Latent space update (eq. (6))
Integrated parameter conversion to latent space (eq. (7))
E. Determining K (eqs. (8)-(13), Fig. 2)
F. GMM refinement (EM)
G. Motion generation (GMR) (eqs. (2)-(3))
Joint motion trajectory (eq. (14))
Robot

Fig. 1. Overall learning algorithm procedure. "H. Data evaluation" step is skipped when a robot does not evaluate the teaching trial.

Then, a sequence of states $\{x_i\}_{i=1}^N$ describes a motion, which consists of N data points. Previous works [7], [8] have proposed that a generalized motion which maximizes $\prod_{i=1}^N p(x_i)$ when the number of Gaussian components is fixed can be computed by the Gaussian mixture regression (GMR). The expected position $\hat{\mu}$ and the spatial covariance matrix $\hat{\Sigma}$ for representing the generalized motion in a latent space can be expressed as functions of time t using the model parameters, μ_k and Σ_k, of the GMM. $\mu_k = [\mu_{t,k}, \mu_{s,k}]^T$ consists of a temporal component $\mu_{t,k}$ and a spatial vector $\mu_{s,k}$. In this paper, we let the subscripts t and s indicate the time component and the posture vector, respectively. Σ_k can be partitioned according to the two components as follows: $\Sigma_k = \begin{bmatrix} \Sigma_{tt,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{ss,k} \end{bmatrix}$.

Then, the following functions are given in [8], [10]:

$$\hat{\mu}_s(t) = \sum_{k=1}^{K} \frac{\pi_k \mathcal{N}(t; \mu_{t,k}, \Sigma_{tt,k})}{\sum_{i=1}^{K} \pi_i \mathcal{N}(t; \mu_{t,i}, \Sigma_{tt,i})} \times \left(\mu_{s,k} + \Sigma_{st,k}\Sigma_{tt,k}^{-1}(t - \mu_{t,k})\right)$$

$$\hat{\Sigma}_{ss}(t) = \sum_{k=1}^{K} \left(\frac{\pi_k \mathcal{N}(t; \mu_{t,k}, \Sigma_{tt,k})}{\sum_{i=1}^{K} \pi_i \mathcal{N}(t; \mu_{t,i}, \Sigma_{tt,i})}\right)^2 \times \left(\Sigma_{ss,k} - \Sigma_{st,k}\Sigma_{tt,k}^{-1}\Sigma_{ts,k}\right). \quad (2)$$

Including the time value, the parameters are respectively set to

$$\hat{\mu}(t) = [\,t \quad \hat{\mu}_s(t)\,]^T$$
$$\hat{\Sigma}(t) = \begin{bmatrix} 0 & 0 \\ 0 & \hat{\Sigma}_{ss}(t) \end{bmatrix}. \quad (3)$$

The regressed spatial trajectories in the latent space $\hat{x}_s(t)$ are retrieved from the probability distribution of $\mathcal{N}(x_s(t); \hat{\mu}_s(t), \hat{\Sigma}_{ss}(t))$, and then, the generalized motion in a latent space is $\hat{x}(t) = [t\ \hat{x}_s(t)]^T$. In fact, $\hat{x}(t) = \hat{\mu}(t)$.

The accumulated information after the jth teaching trial is compactly expressed by $M^j = \{G^j, L^j\}$, where $G^j = \{\pi_k^j, \mu_k^j, \Sigma_k^j\}_{k=1}^K$ is a collection of the GMM parameters in the latent space and $L^j = \{m^j, P^j\}$ represents parameters that describe the latent space according to the procedure in Section II-D.

The procedural steps of the proposed framework are explained in the following sections.

A. Teaching Trial Data Encoding

During each teaching trial, the motion is encoded in terms of joint angle trajectories. The jth collected joint angle trajectory information is represented by $\{\zeta_i^j\}_{i=1}^{N^j} = \{[\zeta_{t,i}^j\ \zeta_{s,i}^j]^T\}_{i=1}^{N^j}$, where N^j represents the total number of data points from the jth trial. Each ζ_i^j consists of a time value $\zeta_{t,i}^j \in R$ and a posture vector $\zeta_{s,i}^j \in R^d$, where d is the dimensionality of the joint space.

B. Incremental Temporal Alignment

Each motion of the same task is similar, but not exactly identical. The duration and speed of one trial may be different from previous ones. Therefore, motions are sampled at different lengths by the sensors. To make the motion data comparable, the robot must align the motion trajectories temporally. We apply dynamic time warping (DTW) which has been designed for aligning one curve with respect to another [6], [8], [15], [16]. A slope constraint, as suggested in [16], is selected to alleviate the problem of singularities. We use DTW to align each newly encoded trial trajectory to a learned motion trajectory from previous teaching session. In addition, time values are newly estimated to be accordant with the alignment. To produce an aligned trajectory with a constant time interval and a fixed number of data points N, linear interpolations are applied. As a result, aligned teaching data $\{\xi_i^j\}_{i=1}^N$ are obtained. Furthermore, the regression parameters $\{\widehat{\mu}_i^{j-1}, \widehat{\Sigma}_i^{j-1}\}_{i=1}^N$ in the latent space from (3) are adjusted by taking into account the new time alignment.

C. Data Integration

The temporal alignment step generates the aligned data trajectory $\{\xi_i^j\}_{i=1}^N$ and a parameter set $\{\hat{\mu}_i^{j-1}, \hat{\Sigma}_i^{j-1}\}_{i=1}^N$ for GMR, which represents the previous data inclusively. The parameter set is updated to include the most recent data $\{\xi_i^j\}_{i=1}^N$. The two are described in different spaces. Therefore, the parameter set is converted to $\{\widehat{m}_i^{j-1}, \widehat{C}_i^{j-1}\}_{i=1}^N$ in the joint space as follows:

$$\widehat{m}_i^{j-1} = P^{j-1}\hat{\mu}_i^{j-1} + m^{j-1}$$
$$\widehat{C}_i^{j-1} = P^{j-1}\hat{\Sigma}_i^{j-1}P^{j-1^T} \quad (4)$$

where $L^{j-1} = \{m^{j-1}, P^{j-1}\}$, the latent space parameters, is provided from the previous teaching trial by the procedure described in Section II-D.

Then, (5) shown at the bottom of the page yields the update of the set in the joint space which results in $\{\widetilde{m}_i^j, \tilde{C}_i^j\}_{i=1}^N$.

D. Projection Onto Latent Space

It is inefficient to manage complicated whole-body motions in high-dimensional joint angle space. The description of motions in a latent space of reduced dimensionality with uncorrelated components has been employed in many works due to its usefulness and simplicity. PCA is a popular method for projecting the original joint space onto a latent space of reduced dimensionality [17]–[19]. In this paper, the latent space changes every time new teaching data are acquired. After the jth kinesthetic teaching trial, $\{\widetilde{m}_i^j, \tilde{C}_i^j\}_{i=1}^N$ is computed from (5), where the temporal and spatial components are separated such as $\widetilde{m}_i^j = [\widetilde{m}_{t,i}^j, \widetilde{m}_{s,i}^j]^T$ and $\tilde{C}_i^j = \begin{bmatrix} \tilde{C}_{tt,i}^j & \tilde{C}_{ts,i}^j \\ \tilde{C}_{st,i}^j & \tilde{C}_{ss,i}^j \end{bmatrix}$. Then, the mean m_s^j and the covariance matrix C_{ss}^j in the joint space are computed as follows:

$$m_s^j = \frac{\sum_{i=1}^N \widetilde{m}_{s,i}^j}{N}$$
$$C_{ss}^j = \frac{\sum_{i=1}^N \tilde{C}_{ss,i}^j + \left(\widetilde{m}_{s,i}^j - m_s^j\right)\left(\widetilde{m}_{s,i}^j - m_s^j\right)^T}{N}. \qquad (6)$$

The eigenvectors and eigenvalues λ_i of the real symmetric covariance matrix C_{ss}^j are computed: $C_{ss}^j W^j = W^j \Lambda^j$, where Λ^j is a diagonal matrix whose elements are eigenvalues and W^j is a matrix whose columns are the corresponding eigenvectors. To construct the reduced dimensional latent space, we selected q number of eigenvectors as $W_{d\times q}^j$ which satisfy the condition $(\sum_{i=1}^q \lambda_i / \sum_{i=1}^d \lambda_i) > \gamma$ which indicates that the projection to the latent space covers $100 \times \gamma\%$ of the data's spread. The new latent space L^j is defined as $L^j = \{m^j, P^j\}$, where $m^j = [0\ m_s^j]^T$ and $P^j = \begin{bmatrix} 1 & 0 \\ 0 & W_{d\times q}^j \end{bmatrix}$.

$\{\widetilde{m}_i^j, \tilde{C}_i^j\}_{i=1}^N$ can be transformed onto the newly updated latent space L^j and expressed by $\{\widetilde{\mu}_i^j, \tilde{\Sigma}_i^j\}_{i=1}^N$, where

$$\widetilde{\mu}_i^j = P^{j^T}\left(\widetilde{m}_i^j - m^j\right)$$
$$\tilde{\Sigma}_i^j = P^{j^T} \tilde{C}_i^j P^j. \qquad (7)$$

E. Model Parameter Initialization for GMM Refinement

The number of Gaussian components K in the GMM is computed during learning. Previous investigations have proposed the selection of an optimal number of components K by minimizing the Bayesian information criterion (BIC) score S_{BIC} which evaluates both model performance and complexity [8]–[12]

$$S_{\mathrm{BIC}} = (\text{model fitting term}) + (\text{model complexity term})$$
$$= -\mathcal{L} + \frac{n_p}{2}\log(N_{tot})$$

where n_p denotes the total number of free parameters required for a mixture of Gaussian components and N_{tot} is the total number of data points. $\mathcal{L} = \sum_{i=1}^{N_{tot}} \log(p(x_i)) = \sum_{i=1}^{N_{tot}} \log(\sum_{k=1}^K p(k)p(x_i^j|k))$ is the log-likelihood of the model described in (1).

The number of the free parameters is equal to

$$n_p(K,q) = (K-1) + K\left(q + \frac{1}{2}q(q+1)\right). \qquad (8)$$

The first term indicates the total number of prior probabilities, and the second term indicates the total number of means and elements in the symmetric covariance matrices.

Suppose that the jth kinesthetic teaching is currently encoded and a robot stores all of the data from the first to the jth teachings in its memory. Let $\{x_i^j\}_{i=1}^N$ represent temporally aligned data points in the latent space from the jth teaching trial. Then, the total encoded data points in the latent space are expressed by $\{\{x_i^l\}_{l=1}^j\}_{i=1}^N$. A GMM is set as $G = \{\pi_k^j, \mu_k^j, \Sigma_k^j\}_{k=1}^K$ from (1). Let $\{x_{k,i}^j\}_{i=1}^{L_k^j}$ denote the data points governed dominantly by the kth Gaussian component, where L_k^j indicates the number of the governed data points. By assuming that $p(x_{k,i}^j|k) \gg p(x_{k,i}^j|/k)$, where $/k \in \{1,2,\ldots,K\} - \{k\}$ indicates any number other than k, an approximated estimation of $\mathcal{L}$ is computed by considering the expected value of $\mathcal{L}$

$$E[\mathcal{L}] = E\left[\sum_{l=1}^{j}\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k^j p\left(x_i^l|k\right)\right)\right]$$
$$\approx E\left[\sum_{k=1}^{K}\sum_{i=1}^{L_k^j}\log\left(\pi_k^j p\left(x_{k,i}^j|k\right)\right)\right]. \qquad (9)$$

The above approximation is taken based on the propensity of each Gaussian density component to be dominant over partial data points where the data points tend to be linearly spread in a local space. Each $x_{k,i}^j$ is regarded as a random vector distributed according to $\mathcal{N}(\mu_k^j, \Sigma_k^j)$

$$E\left[\sum_{k=1}^{K}\sum_{i=1}^{L_k^j}\log\left(\pi_k^j p\left(x_{k,i}^j|k\right)\right)\right]$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{L_k^j} E\left[\log\left(\pi_k^j p\left(x_{k,i}^j|k\right)\right)\right]$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{L_k^j} E\left[\left(-\frac{1}{2}\left(x_{k,i}^j - \mu_k^j\right)^T \Sigma_k^{j^{-1}}\left(x_{k,i}^j - \mu_k^j\right)\right)\right]$$
$$+ \sum_{k=1}^{K} L_k^j\left(\log\left(\pi_k^j\right) - \frac{1}{2}\log\left(\left|\Sigma_k^j\right|\right) - \frac{1}{2}q\log(2\pi)\right).$$

By assigning $z_i = \Sigma_k^{j^{-(1/2)}}(x_{k,i}^j - \mu_k^j)$

$$E\left[\left(-\frac{1}{2}\left(x_{k,i}^j - \mu_k^j\right)^T \Sigma_k^{j^{-1}}\left(x_{k,i}^j - \mu_k^j\right)\right)\right] = -\frac{1}{2}E[z_i{}^T z_i].$$

$$\widetilde{m}_i^j = \frac{(j-1)\widehat{m}_i^{j-1} + \xi_i^j}{j}$$
$$\tilde{C}_i^j = \frac{(j-1)\left(\hat{C}_i^{j-1} + \left(\widehat{m}_i^{j-1} - \widetilde{m}_i^j\right)\left(\widehat{m}_i^{j-1} - \widetilde{m}_i^T\right)^T\right) + \left(\xi_i^j - \widetilde{m}_i^j\right)\left(\xi_i^j - \widetilde{m}_i^j\right)^T}{j} \qquad (5)$$

Due to its definition, the random vector z_i is distributed according to $\mathcal{N}(0, I_{q \times q})$. Then

$$E[z_i{}^T z_i] = E\left[\sum_{l=1}^{q} z_{i,l}^2\right] = \sum_{l=1}^{q} E\left[z_{i,l}^2\right] = \sum_{l=1}^{q} 1 = q$$

where $z_{i,l}$ represents each element in the vector, $l = 1, \ldots, q$. Therefore

$$\sum_{k=1}^{K}\sum_{i=1}^{L_k^j} E\left[\left(-\frac{1}{2}\left(x_{k,i}^j - \mu_k^j\right)^T \Sigma_k^{j^{-1}}\left(x_{k,i}^j - \mu_k^j\right)\right)\right]$$

$$= -\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{L_k^j} q = -\frac{1}{2}\sum_{k=1}^{K} qL_k^j = -\frac{1}{2}qjN$$

since it is true that $\sum_{k=1}^{K} L_k^j = jN$ by definition. As a result, we have the following assumption:

$$E[\mathcal{L}] = \sum_{k=1}^{K} L_k^j\left(\log\left(\pi_k^j\right) - \frac{1}{2}\log\left(\left|\Sigma_k^j\right|\right) - \frac{1}{2}q\left(1 + \log(2\pi)\right)\right). \quad (10)$$

Equation (10) indicates that no explicit data point information is required to compute the log-likelihood, which makes an incremental approach more attractive. Through the procedure from Section II-B–D, the integrated trajectory parameters $\{\tilde{\mu}_i^j, \tilde{\Sigma}_i^j\}_{i=1}^N$ are generated [see (7)]. Using $\{\tilde{\mu}_i^j, \tilde{\Sigma}_i^j\}_{i=1}^N$, for $k = 1, \ldots, N/2$, a set of parameters in GMM, defined in (11) shown at the bottom of the page, is initially declared by grouping two successive time steps to avoid zero variance along the time axis.

Equation (11) does not require explicit data points. Therefore, the robot does not need to save past data in its memory.

By considering the possible repeated merging of sequential groups, a correct number of K is sought. Suppose that the two kth and $(k+1)$th components are merged into a single component (i.e., a new kth component). The change of the fitting term value is then computed as follows:

$$\begin{aligned} E[\Delta\mathcal{L}] =& L_k^{j'}\left(\log\left(\pi_k^{j'}\right) - \frac{1}{2}\log\left(\left|\Sigma_k^{j'}\right|\right) - \frac{1}{2}q\left(1 + \log(2\pi)\right)\right) \\ &- L_k^j\left(\log\left(\pi_k^j\right) - \frac{1}{2}\log\left(\left|\Sigma_k^j\right|\right) - \frac{1}{2}q\left(1 + \log(2\pi)\right)\right) \\ &- L_{k+1}^j\left(\log\left(\pi_{k+1}^j\right) - \frac{1}{2}\log\left(\left|\Sigma_{k+1}^j\right|\right)\right. \\ &\left. - \frac{1}{2}q\left(1 + \log(2\pi)\right)\right) \\ =& \log\frac{\pi_k^{j'}{}^{L_k^{j'}}}{\pi_k^j{}^{L_k^j}\pi_{k+1}^j{}^{L_{k+1}^j}} - \frac{1}{2}\log\frac{\left|\Sigma_k'\right|^{L_k^{j'}}}{\left|\Sigma_k^j\right|^{L_k^j}\left|\Sigma_{k+1}^j\right|^{L_{k+1}^j}} \end{aligned} \quad (12)$$

where $G = \{\pi_k^{j\prime}, \mu_k^{j\prime}, \Sigma_k^{j\prime}\}_{k=1}^{K-1}$ after merging and $L_k^{j\prime} = L_k^j + L_{k+1}^j$.

```
Initial inputs: ${\{\tilde{\mu}_i^j, \tilde{\Sigma}_i^j\}}_{i=1}^N$
initialize $G = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ from ${\{\tilde{\mu}_i^j, \tilde{\Sigma}_i^j\}}_{i=1}^N$
while true do
    compute $\Delta S_{BIC}$'s from any two sequential groups
    $\Delta S_{BIC}^* :=$ min $\Delta S_{BIC}$
    $k_m :=$ arg $\Delta S_{BIC}^*$
    if $\Delta S_{BIC}^* < 0$ then
        $K := K - 1$ // merge is confirmed
        Update $G :=$ merge($G, k_m, k_m + 1$)

    // $G = {\{\pi_k^{j\prime}, \mu_k^{j\prime}, \Sigma_k^{j\prime}\}}_{k=1}^{K-1}$
    else
        Break
    end if
end while
return $K$, $G$ // initial value and data for EM
```

Fig. 2. Algorithm of the model initialization for EM.

The change of the complexity term value is computed from (8)

$$\Delta n_p = -1 - q - \frac{1}{2}q(q+1). \quad (13)$$

All possible merges of two sequential groups are considered to compute the largest reduction of $\Delta S_{\mathrm{BIC}} = -E[\Delta\mathcal{L}] + (\Delta n_p/2)\log jN$. For the case of a merge resulting in the largest reduction, the merge is confirmed if the sign of ΔS_{BIC} is negative. This process is repeated until the sign is no longer negative. After the termination, the number of remaining components becomes K. We note that using (12) and (13) to compute ΔS_{BIC} does not include explicit data point values, but only model parameters. Therefore, no memorization of data points is required. The procedure is summarized in Fig. 2.

F. GMM Refinement

Once K is selected, the expectation–maximization (EM) algorithm can be applied to refine the model parameters [20]. From $\{\hat{\mu}_i^j, \hat{\Sigma}_i^j\}_{i=1}^N$ which generalizes the data information so far, data samples used to run EM can be easily generated. Suppose that α random samples are stochastically generated from each distribution of $\mathcal{N}(z; \hat{\mu}_i^j, \hat{\Sigma}_i^j)$ and are represented by $z^j = \{z_i^j\}_{i=1}^{\alpha N}$. Using the samples, EM can be implemented:

0. Initialize the model parameters (see Section II-E)

1. E-Step:

$$p_{k,i}^{(t+1)} = \frac{\pi_k^{(t)}\mathcal{N}\left(z_i^j; \mu_k^{(t)}, \Sigma_k^{(t)}\right)}{\sum_{l=1}^{K}\pi_l^{(t)}\mathcal{N}\left(z_i^j; \mu_l^{(t)}, \Sigma_l^{(t)}\right)}$$

$$\begin{aligned} \pi_k^j &= \frac{2}{N} \\ \mu_k^j &= \frac{\widetilde{\mu}_{2k-1}^j + \widetilde{\mu}_{2k}^j}{2} \\ \Sigma_k^j &= \frac{\left(\widetilde{\Sigma}_{2k-1}^j + \left(\widetilde{\mu}_{2k-1}^j - \mu_k^j\right)\left(\widetilde{\mu}_{2k-1}^j - \mu_k^j\right)^T\right) + \left(\widetilde{\Sigma}_{2k}^j + \left(\widetilde{\mu}_{2k}^j - \mu_k^j\right)\left(\widetilde{\mu}_{2k}^j - \mu_k^j\right)^T\right)}{2} \end{aligned} \quad (11)$$

2. M-Step:

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^{\alpha N} p_{k,i}^{(t+1)}}{\alpha N}$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{\alpha N} p_{k,i}^{(t+1)} z_i^j}{\alpha N \pi_k^{(t+1)}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{i=1}^{\alpha N} p_{k,i}^{(t+1)} \left(z_i^j - \mu_k^{(t+1)}\right)\left(z_i^j - \mu_k^{(t+1)}\right)^T}{\alpha N \pi_k^{(t+1)}}$$

where t is the iteration number.

Repeat 1 and 2 until convergence $|(\mathcal{L}^{(t+1)}/\mathcal{L}^{(t)}) - 1| < \varepsilon \ll 1$, where $\mathcal{L} = \sum_{i=1}^{N} \log(p(z_i^j))$.

G. Motion Generation

The GMR method mentioned earlier can reproduce a generalized motion through (2) and (3) using the converged parameter values from EM. The generalized motion in the latent space $\{\hat{x}_i^j\}_{i=1}^N$ after the jth teaching trial is projected onto the joint space to produce the generalized (learned) joint motion $\{\hat{\xi}_i^j\}_{i=1}^N$ by

$$\hat{\xi}_i^j = P^j \hat{x}_i^j + m^j. \tag{14}$$

The motion $\{\hat{\xi}_i^j\}_{i=1}^N$ represents a learned motion soon after the jth kinesthetic teaching trial.

H. Teaching Trial Data Evaluation

So far, it is assumed in the algorithm that every teaching trial is accepted. In this section, we explore the option of allowing the robot to decide autonomously whether the most recent teaching trial is suitable for learning or if it should be rejected. If the teaching trial is inappropriate, it is neglected, and the learning procedure does not proceed; otherwise, the remaining steps are taken for learning.

Suppose that the aligned jth teaching trial $\{\xi_i^j\}_{i=1}^N$ is evaluated. The evaluation proceeds in two steps: acceptability by latent space and similarity with GMM, under the assumption that the teaching trials should be similar to include the essential characteristics of a specific behavior.

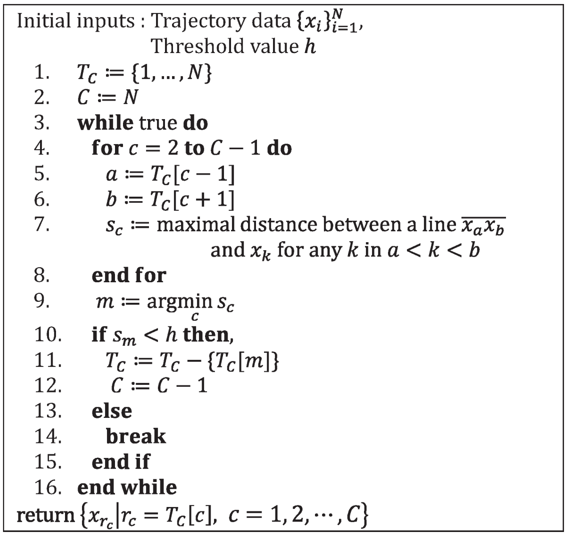
Suppose that the parameters of the latent space $L^{j-1} = \{m^{j-1}, P^{j-1}\}$ and GMM with K Gaussian components $G^{j-1} = \{\pi_k^{j-1}, \mu_k^{j-1}, \Sigma_k^{j-1}\}_{k=1}^K$ are currently calculated from the previous trial.

1) Acceptability by Latent Space: The projection matrix onto the latent space $W_{d\times q}^{j-1}$, as explained in Section II-D, has been computed after the $(j-1)$th teaching trial. We check whether the jth teaching trial can be represented sufficiently according to the most recently updated latent space. Using the spatial posture vectors in ξ_i^j, the information loss through the projection to the latent space is defined as

$$S_e = \frac{\sum_{i=1}^{N} e_i^{j^T} e_i^j}{\sum_{i=1}^{N} \left(\xi_{s,i}^j - m_s^{j-1}\right)^T \left(\xi_{s,i}^j - m_s^{j-1}\right)} \tag{15}$$

where $e_i^j = (\xi_{s,i}^j - m_s^{j-1}) - W_{d\times q}^{j-1} W_{d\times q}^{j-1^T} (\xi_{s,i}^j - m_s^{j-1})$.

If $S_e > \delta$, then $\{\xi_i^j\}_{i=1}^N$ is rejected, and this learning procedure is terminated; otherwise, we proceed with the next step. Here, δ is a constant that represents a permissible deviation. Since the latent space is constructed to cover $100 \times \gamma\%$ of data spread, $S_e \le 1 - \gamma$ is expected if $\xi_{s,i}^j$ lies strictly on the current latent space. If $\delta = 1 - \gamma$, no effective learning is experienced for future teaching trials. An adequate δ in the range of $1 - \gamma < \delta \ll 1$ is selected to balance data acceptability and learning opportunity because the model is not a final version.



```
Initial inputs : Trajectory data $\{x_i\}_{i=1}^N$,
                 Threshold value $h$
$T_C := \{1, \dots, N\}$
$C := N$
while true do
  for $c = 2$ to $C - 1$ do
    $a := T_C[c-1]$
    $b := T_C[c+1]$
    $s_c :=$ maximal distance between a line $\overline{x_a x_b}$
             and $x_k$ for any $k$ in $a < k < b$
  end for
  $m := \operatorname{argmin}_c s_c$
  if $s_m < h$ then,
    $T_C := T_C - \{T_C[m]\}$
    $C := C - 1$
  else
    break
  end if
end while
return $\{x_{r_c} | r_c = T_C[c],\ c = 1, 2, \cdots, C\}$
```

Fig. 3. Algorithm of the characteristic point assignment.

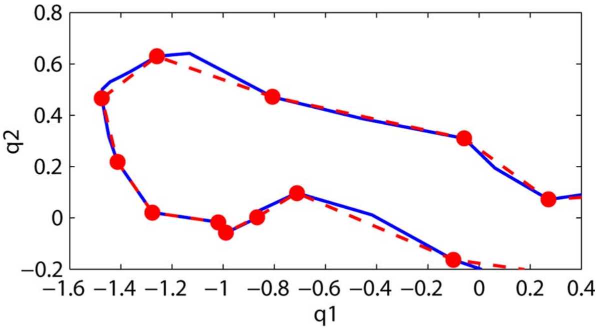


Fig. 4. Example of characteristic point assignment and linear segments. Red filled circles: Assigned characteristic points. Dashed lines: Linear segments. Represented only in the first and the second principal (q1 and q2) in the latent space for simplicity.

2) Pattern Similarity Measure: This step examines the similarity of the jth teaching trial to the reproduced trajectory obtained based on the teaching trial information up to the $(j-1)$th trial. First, characteristic points of the teaching trial trajectory are selected through the procedure shown in Fig. 3. The algorithm finds a list of characteristic points (the red filled circles in Fig. 4 for example) from the teaching trial trajectory (the blue line in Fig. 4). The characteristic points divide the trajectory into a set of segments. Every data point between two sequential characteristic points is placed in the range defined by a threshold h, and it is regarded approximately as a linear segment. The characteristic points of the reproduced trajectory are found using the same procedure. For the two trajectories to be comparable, both trajectories are linearly segmented via all of the selected characteristic points (the red dotted line segments in Fig. 4). The corresponding linear segments from both trajectories are compared to evaluate pattern similarity. To enable the pattern similarity comparison, the velocity expression is designated.

Suppose that $\{r_c\}_{c=1}^C$ represents the sample indexes of the total C characteristic points. The $(j-1)$th aligned teaching trajectory in the latent space $\{x_i^j\}_{i=1}^N$ is partitioned into a time value $x_{t,i}^j$ and a posture vector $x^j_{s,i}$. Then, the velocity of the cth segment from the teaching trajectory is approximated as

$$v^j_c = \frac{x^j_{s,r_{c+1}} - x^j_{s,r_c}}{x^j_{t,r_{c+1}} - x^j_{t,r_c}}, \quad c = 1, 2, \ldots, C-1. \tag{16}$$

The cth segment from the reproduced trajectory in the latent space can be represented by $\{\hat{\mu}^{j-1}(x^j_{t,i}), \hat{\Sigma}^{j-1}(x^j_{t,i})\}^{r_{c+1}}_{i=r_c}$ using GMR [see (2) and (3)].

Let $\bar{\mu}^{j-1}_c = \sum^{r_{c+1}}_{i=r_c} \widehat{\mu}^{j-1}(x^j_{t,i})/(r_{c+1} - r_c + 1)$ and $\bar{\Sigma}^{j-1}_c$ be defined as shown at the bottom of the page.

By partitioning $\bar{\Sigma}^{j-1}_c = \begin{bmatrix} \bar{\Sigma}^{j-1}_{tt,c} & \bar{\Sigma}^{j-1}_{ts,c} \\ \bar{\Sigma}^{j-1}_{st,c} & \bar{\Sigma}^{j-1}_{ss,c} \end{bmatrix}$, the velocity of the cth segment $\mu^{j-1}_{v,c}$ and its corresponding covariance $\Sigma^{j-1}_{v,c}$ are stated as follows:

$$\begin{aligned} \mu^{j-1}_{v,c} &= \bar{\Sigma}^{j-1}_{st,c}\left(\bar{\Sigma}^{j-1}_{tt,c}\right)^{-1} \\ \Sigma^{j-1}_{v,c} &= \left(\bar{\Sigma}^{j-1}_{ss,c} - \bar{\Sigma}^{j-1}_{st,c}\left(\bar{\Sigma}^{j-1}_{tt,c}\right)^{-1}\bar{\Sigma}^{j-1}_{ts,c}\right)\left(\bar{\Sigma}^{j-1}_{tt,c}\right)^{-1}. \end{aligned} \tag{17}$$

A score of the pattern similarity between the cth segments is defined as

$$S_c = \frac{\sum^{C-1}_{c=1}(r_{c+1} - r_c + 1)\left(v^j_c - \mu^{j-1}_{v,c}\right)^T\left(\Sigma^{j-1}_{v,c}\right)^{-1}\left(v^j_c - \mu^{j-1}_{v,c}\right)}{N-1}. \tag{18}$$

The aforementioned score measures how closely the two trajectories travel. When each pair of segments points in the same direction, the value of S_c is small.

Although the two motion trajectories are similar, it is possible that the two motions are not coincident spatially. Therefore, a regressive estimation score, defined in (19) shown at the bottom of the page, is incorporated to check the coincidence using GMR, where ϵ is a nonnegative number.

Covariance matrices are adjusted by adding a term $\epsilon I_{q\times q}$ so as to not strictly rely on the currently learned model in terms of spatial uncertainty because the model is not a final version.

The good teaching trial trajectory tends to have both small S_c and S_ϵ values. Because both S_c and S_ϵ are nonnegative, a teaching trial evaluation rule is designed to be

$$S_c \cdot S_\epsilon < \rho \tag{20}$$

where ρ is a threshold value.

If the above condition is satisfied, the teaching trial is accepted for learning; otherwise, it is rejected, and this learning session is terminated. It is worth noting that the two evaluation steps aim to determine whether a teaching trial includes the essential characteristics of a specific behavior with respect to the latent space and the model. The evaluations do not imply that the teaching trial should be rigorously close to the most likely generalized behavior.

TABLE I
PARAMETER SETTINGS

Parameter	Value	Description
γ	$1 - \frac{0.02}{j+1}$	latent space coverage ratio
ε	0.01	EM Convergence threshold
δ	0.05	"acceptability by latent space" threshold
h	0.1	threshold for characteristic point assignment
ϵ	$0.01\sum^d_{i=1}\lambda_i$	covariance matrix adjustment
ρ	1.5	pattern similarity acceptance threshold

III. Experiments

A. Experimental Setup

We used an Aldebaran Nao humanoid robot with 25 degrees of freedom (DOFs) [21] to test our approach. To learn specific behaviors, the robot motors were set to passive mode. In this mode, during kinesthetic teaching trials, the trajectories of the angle values of each joint were recorded at a rate of 20 Hz through motor encoders. The 14 DOFs of the upper torso were used in the experiment, and the DOFs of the lower body were set to a constant position, maintaining a stable posture. Before the first teaching trial, no assumptions about any parameters were prepared in the GMM. The learning procedure was run in a computer equipped with an Intel Core2 Quad CPU Q6600 at 2.40 GHz with the Windows XP operating system.

Table I summarizes the parameter values used in the experiments. Due to the incremental learning property, the coverage ratio of effective principal components over the whole data with a fixed γ tends to decrease as further teaching trials are provided. Therefore, γ was chosen as a function of the teaching session, which helps retain the coverage ratio. For the first teaching session, γ was 0.99. ε is selected to guarantee the EM convergence. As mentioned in Section II-H2, δ is determined not to reject new data too tightly for learning while checking their acceptability. $\delta = 0.05$ implies that data trajectories should be accepted at least 95% by the current latent space. ϵ relies on the summation of total variances. It makes it possible to score the pattern similarity independently of motion scales. For more widely spread motions, the larger covariance matrices in (19) were considered. We decided h and ρ empirically to be generally applicable regardless of motion scales and patterns from the humanoid robot. The selection of parameter values also affects how strictly a robot is to evaluate a teaching trial. Every experiment in this work used the same parameter setting in Table I.

$$\bar{\Sigma}^{j-1}_c = \frac{\sum^{r_{c+1}}_{i=r_c}\left(\widehat{\Sigma}^{j-1}\left(x^j_{t,i}\right) + \left(\widehat{\mu}^{j-1}\left(x^j_{t,i}\right) - \bar{\mu}^{j-1}_c\right)\left(\widehat{\mu}^{j-1}\left(x^j_{t,i}\right) - \bar{\mu}^{j-1}_c\right)^T\right)}{r_{c+1} - r_c + 1}$$

$$S_\epsilon = \frac{\sum^N_{i=1}\left(x^j_{s,i} - \widehat{\mu}^{j-1}_s\left(x^j_{t,i}\right)\right)^T\left(\widehat{\Sigma}^{j-1}_{ss}\left(x^j_{t,i}\right) + \epsilon I_{q\times q}\right)^{-1}\left(x^j_{s,i} - \widehat{\mu}^{j-1}_s\left(x^j_{t,i}\right)\right)}{N} \tag{19}$$

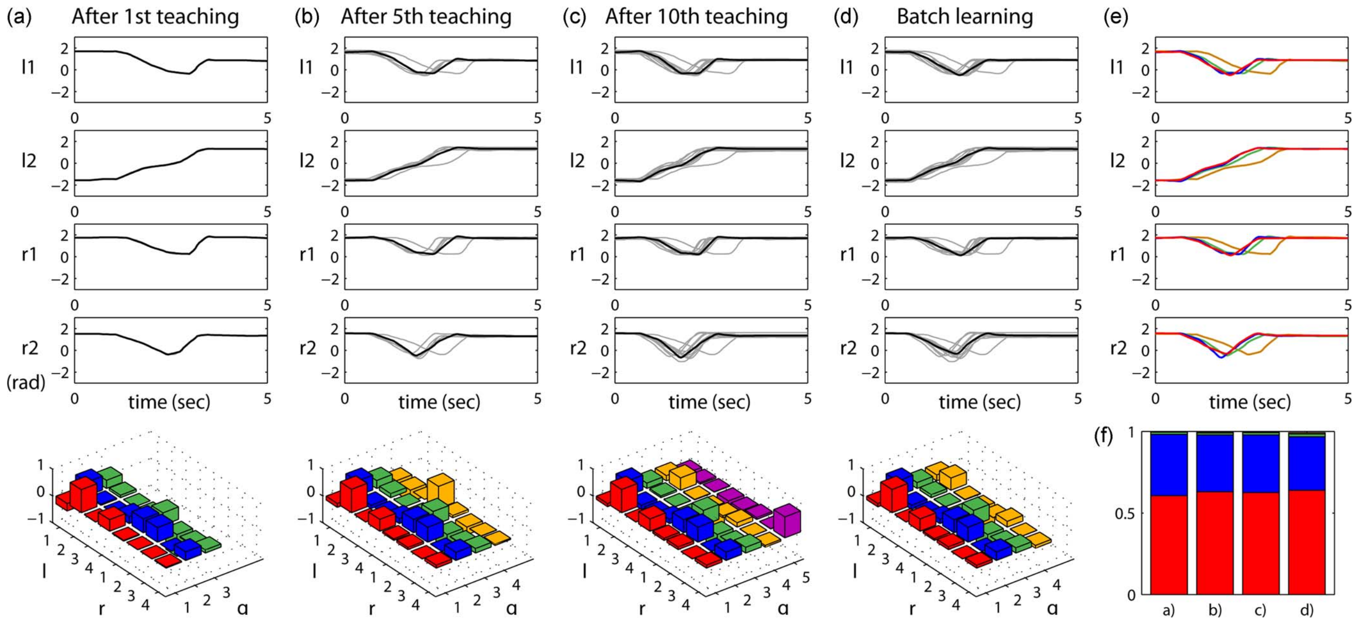


Fig. 5. Incremental learning of "underneath blocking" behavior over the ten teaching trials. Only selected joint trajectories are illustrated. l1 and l2 indicate two joints, shoulder roll and elbow roll, in the left arm, and r1 and r2 indicate those in the right arm. (a)–(c) represent the learned joint trajectories after the first, fifth, and tenth teaching trials, respectively. (d) represents the learned joint trajectories through the batch-learning algorithm from [8], and (e) shows them together. In the bottom of (a)–(d), major principal directional components (eigenvectors) are plotted correspondingly. For clarity, only eight element values are shown from each principal direction. (f) illustrates the covering portions of principal components in (a)–(d).

B. Incremental Learning of Taekwondo Behaviors

This experiment aimed to verify the incremental online learning capability (without rejection). Six behaviors of the Korean martial art, Taekwondo, were selected as test examples of a rich behavior vocabulary: underneath blocking, trunk blocking, face blocking, trunk punching, fist back hitting, and hand blade neck hitting. A human taught each martial art behavior to the Nao robot kinesthetically. Ten teaching trials were provided for each behavior. As an example, Fig. 5 shows the evolution of the learned motions of underneath blocking over the teaching sessions. To evaluate the performance of our algorithm, it was compared with the performance of the batch algorithm as proposed in [8]. In the batch case, the ten teaching trial data are used simultaneously. It has been asserted that batch learning [8] can reproduce the trajectories that preserve the essential characteristics of the demo data set. To compare the batch algorithm performance with ours, the DTW proposed in [8] was applied to the raw teaching data trajectories to align them temporally.

In the batch case, the template for each behavior is obtained by an iterative method that computes a structural average from the data as the template and updates the average iteratively [16]. An optimal number of K is selected for the batch algorithm, as explained in [8] and [10]. The batch-learning process employs a fixed γ, whose value is 0.99. Fig. 5(a)–(c) shows the reproduced trajectories of the selected joints and the major principal directional components of the latent space after the first, fifth, and tenth teaching sessions, respectively; (d) shows them obtained through batch learning. The grayed trajectories represent raw teaching data in the joint space.

The orange, green, and blue lines indicate the reproduced trajectories after the first, fifth, and tenth teaching sessions through the proposed learning algorithm, respectively. Fig. 6(a) shows the robot performances of the behaviors after the tenth teaching, respectively. The red line shows the reproduced trajectories through the batch-learning algorithm. The reproduced joint trajectories are nearly identical to and temporally synchronous with those from the batch learning in (e). The result including unshown rest cases implies that our algorithm can perform the learning comparably to the batch-learning algorithm even though it is incremental online.

Our incremental time alignment can achieve temporal coherence with DTW for batch learning even without a full data set. Our learning algorithm results in nearly the same dimensionality q of the latent space over the teaching sessions as the batch algorithm, as shown in Table II. The major principal directions are fairly similar between the two cases, as shown at the bottom of Fig. 5(a)–(d), where the major principal directional components of each latent space are plotted. For clarity, only eight element values are shown. The influence of the other principal directions is negligible, as shown in (f), where the ratios of principal directional motion spreads $(\lambda_i/\sum_{i=1}^{d}\lambda_i)$ from (a)–(d) are illustrated. The motion spreads over the major principal directions in (c)–(d) are fairly comparable. For the learning of all six behaviors, the incrementally updated latent spaces cover about 98% of the full original teaching data set after three teaching trials. After the final teaching, the latent spaces represent 99% of the full original teaching data set.

Table II shows the changing number of Gaussian components K in GMM during incremental learning. K tends to converge almost immediately after the first teaching trial. The final K obtained through our algorithm is generally less than that obtained through the batch algorithm, which implies that our algorithm prefers a simpler model. Table II also indicates that, on the whole, the joint space of 14 DOFs is reduced to the latent space of four to six dimensions. Occasionally, q increases over the teaching trials. The increasing q is possible because of varying γ. A principal direction that was less significant initially could become more influential in later teaching trials. The experimental results demonstrate that behaviors can be incrementally learned after each teaching trial without significant loss of reproducibility and that the motion model can successfully represent the teaching data set.

Fig. 6 shows sequential snapshots of the robotic behavior reproductions after the final teaching trial using our algorithm.

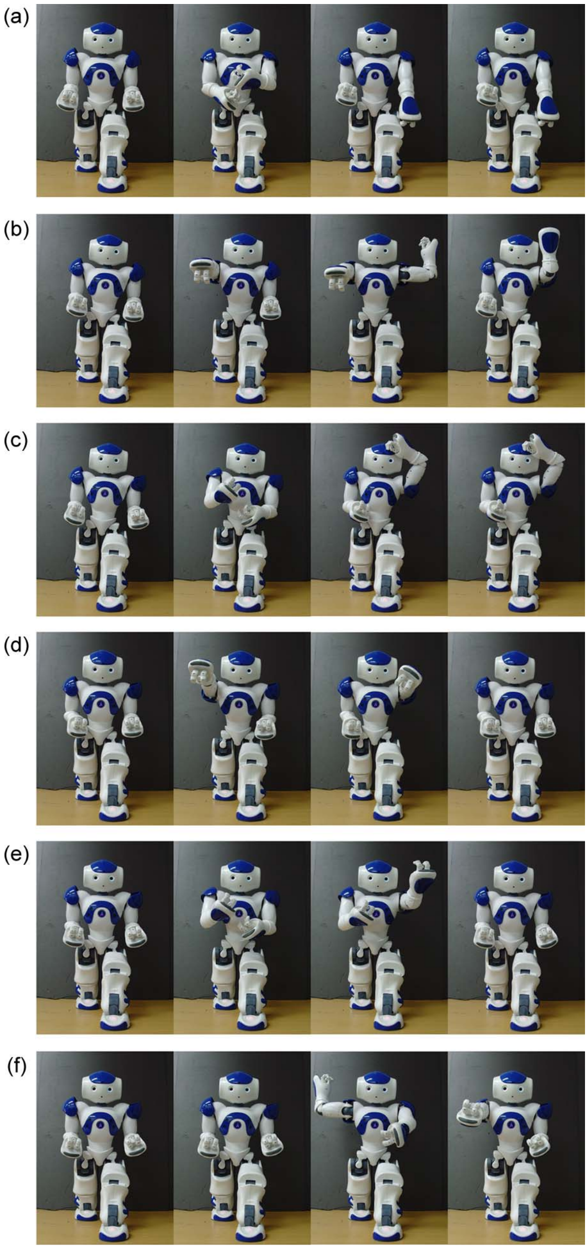


Fig. 6. Sequential snapshots of the six Taekwondo behaviors reproduced by the robot after final teaching: (a) Underneath blocking, (b) trunk blocking, (c) face blocking, (d) trunk punching, (e) fist back hitting, and (f) hand blade neck hitting.

TABLE II

CHANGE OF THE NUMBER OF GAUSSIAN COMPONENTS (K) AND THE DIMENSIONALITY OF THE LATENT SPACE (q) OVER TRIALS DURING LEARNING OF THE SIX TAEKWONDO BEHAVIORS (K/q), WITH "BT" INDICATING DATA OBTAINED THROUGH BATCH LEARNING: (a) UNDERNEATH BLOCKING, (b) TRUNK BLOCKING, (c) FACE BLOCKING, (d) TRUNK PUNCHING, (e) FIST BACK HITTING, AND (f) HAND BLADE NECK HITTING

Trial No.	1	2	3	4	5	6	7	8	9	10	BT
(a)	13/3	7/4	7/4	7/4	7/4	7/4	7/4	7/5	7/5	7/5	7/4
(b)	10/4	8/4	8/4	8/4	8/4	8/4	8/4	8/4	8/4	8/4	10/4
(c)	11/2	7/3	8/3	7/3	7/3	6/3	6/3	7/3	7/4	7/4	7/4
(d)	11/4	8/5	8/6	7/6	7/6	7/6	7/6	7/6	7/6	7/6	10/7
(e)	12/4	8/5	8/5	8/5	8/5	8/5	8/5	8/5	8/5	8/5	9/5
(f)	12/4	8/5	7/5	7/5	7/5	7/5	7/5	7/6	7/6	7/6	13/6

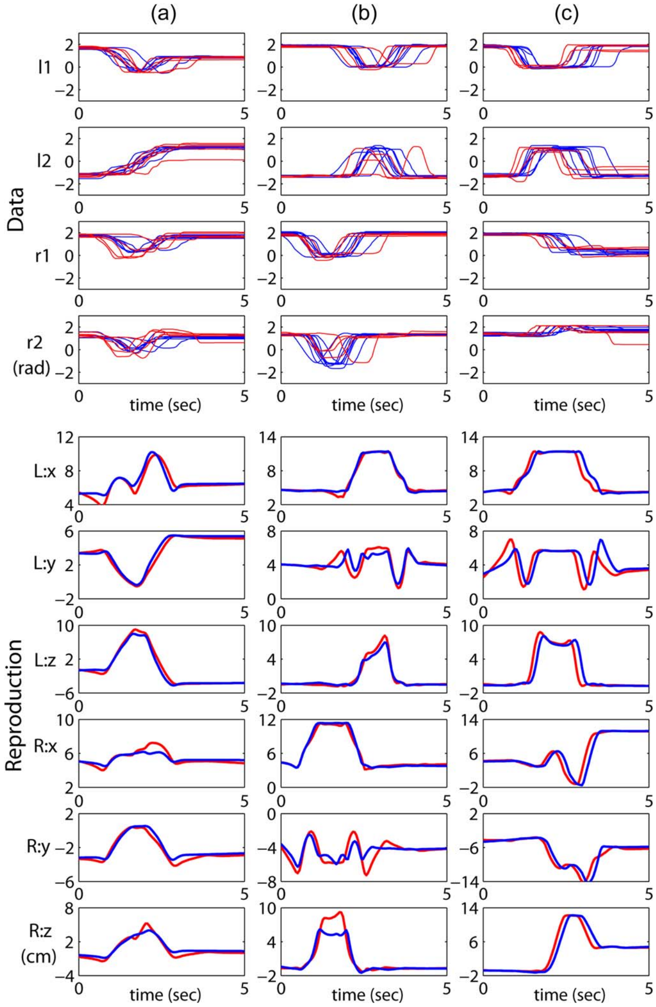


Fig. 7. Raw teaching data and reproduction of the selected Taekwondo behaviors using the learning with and without rejection: (a) Underneath blocking, (b) trunk punching, and (c) hand blade neck hitting. Raw teaching data in terms of some selected joints are drawn. Reproduced hand end point trajectories (R: right hand; L: left hand) are shown in the Cartesian coordinate (x, y, z).

C. *Comparison of Learning With/Without Rejection*

Another experiment was undertaken to evaluate the robot's ability to judge the quality of the teaching trials. If a teaching trial is judged to be inconsistent with others, neglecting the trial may help secure a stable learning process and enhance the generalized representation of a learned behavior over the accepted teaching trials. Again, a human teacher taught the six Taekwondo behaviors incrementally through 12 kinesthetic teaching trials. The learning algorithms with and without data evaluation (see Section II-H) were compared for the same sequence of teaching trials. For the learning with rejection, the robot accepts the first three teaching trials without judgment to establish its knowledge on a behavior. Then, it starts evaluating the next teaching trials.

Fig. 7 shows the unaligned teaching trials used for learning. The blue and red trajectories indicate the accepted and rejected teaching trials through the data evaluation algorithm, respectively. The learning without rejection is incrementally implemented using all of the teaching data, and the learning with rejection uses the blue-colored data alone. The reproduced joint trajectories are commanded to the robot. Fig. 7 also shows the robot's behaviors in terms of the left and right hand end position trajectories in the Cartesian coordinate to clarify different behaviors. The reproduced behaviors from the

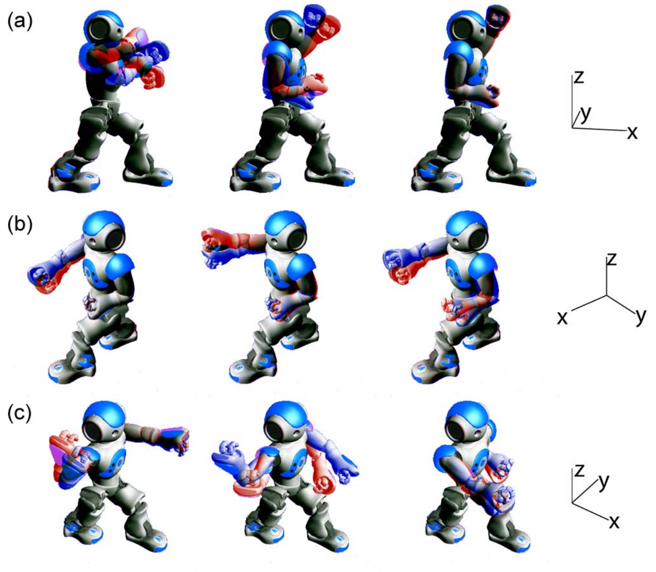


Fig. 8. Comparison between reproductions by the robot through the learning with and without rejection: (a) Face blocking, (b) trunk punching, and (c) hand blade neck hitting.

two cases are differently colored (blue: with rejection; red: without rejection). The trajectories from the two cases are distinct in terms of phase and amplitude. Even a little gap is of significant difference in behavior pattern. The essential differences between the two reproduced trajectories are more clearly recognized in Fig. 8 which shows the snapshots of reproduced behaviors by the robot. In Fig. 8, the selected reproduced behaviors with and without rejection are overlapped synchronously (blue: with rejection; red: without rejection). The improved performance resulting from learning with rejection can be demonstrated upon closer examination. For example, while performing face blocking [see Fig. 8(a)], the robot blocks the face more rapidly after learning with rejection. During trunk punching [see Fig. 8(b)], the straightforward punching is executed after learning with rejection, but the punching is upward after learning without rejection. During hand blade neck hitting [see Fig. 8(c)], the robot does not raise its right hand backward enough before hitting in the case of learning without rejection.

Table III shows the changes of the number of Gaussian components K and the dimensionality of the latent space q over the trials. For each behavior, the upper row indicates learning with rejection, and the bottom row indicates learning without rejection. In the case of learning with rejection, "r" indicates the rejected trial sessions. On the whole, the q of the learned model tends to be smaller when learning with rejection is implemented, which may be because the accepted trial trajectories are more spatially coherent. The results demonstrate that the inconsistent teaching trajectories that violated the general pattern characteristics were rejected.

Figs. 9 and 10 show examples of the rejected teaching trajectories. Fig. 9 shows an "acceptability by latent space" evaluation. A blue dotted trajectory is projected to the most recently updated latent space, and the projected trajectory is then reprojected onto the joint space. The reprojection produces the red trajectory. The red trajectory fails to exhibit the influential pattern characteristics (see l3 in Fig. 9) in comparison with the original blue dotted trajectory. This result indicates that the teaching trajectory requires another principle component, which the current latent space does not incorporate. In Fig. 10, the rejected trajectory is red, and the accepted trajectory is blue. The arrows illustrate the velocity directions. The ellipsoids represent the

TABLE III
CHANGE OF THE NUMBER OF GAUSSIAN COMPONENTS (K) AND THE DIMENSIONALITY OF THE LATENT SPACE (q) OVER TRIALS DURING LEARNING OF THE TAEKWONDO BEHAVIORS (K/q) (UPPER ROW) WITH AND (BOTTOM ROW) WITHOUT REJECTION: (a) UNDERNEATH BLOCKING, (b) TRUNK BLOCKING, (c) FACE BLOCKING, (d) TRUNK PUNCHING, (e) FIST BACK HITTING, AND (f) HAND BLADE NECK HITTING

Trial No.	1	2	3	4	5	6	7	8	9	10	11	12
(a)	11/3	7/4	7/4	7/4	r	r	7/5	r	r	5/5	5/5	r
	11/3	7/4	7/4	7/4	7/6	7/6	7/6	6/7	6/7	6/7	6/7	6/7
(b)	12/4	9/4	8/4	8/4	r	9/4	r	7/4	7/4	7/4	7/5	r
	12/4	8/4	8/4	8/4	8/4	8/4	8/4	9/5	9/4	9/5	9/5	9/5
(c)	11/2	6/4	6/4	r	6/5	7/5	r	r	7/5	7/5	7/5	r
	11/2	6/4	6/4	6/5	6/5	6/5	6/6	6/6	6/6	6/6	6/6	6/6
(d)	13/4	9/4	9/5	r	r	10/5	10/5	r	10/5	10/5	10/5	r
	13/4	9/4	8/5	9/6	8/7	8/7	8/7	8/8	8/7	8/7	8/7	9/7
(e)	12/3	8/5	7/6	r	r	7/6	r	r	r	r	r	6/6
	12/3	8/5	7/6	7/6	7/7	7/7	7/7	6/7	6/7	6/7	6/7	6/7
(f)	10/4	6/5	6/5	7/5	6/5	r	6/5	r	r	r	6/5	6/5
	10/4	6/5	6/5	7/5	7/5	6/6	6/6	6/6	6/6	6/6	6/6	6/6

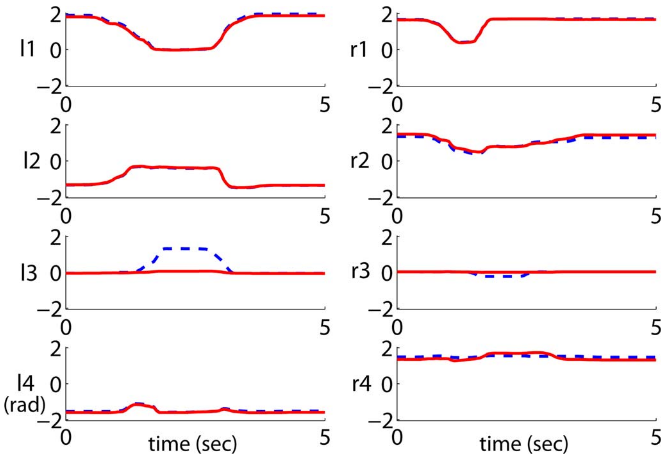


Fig. 9. Example of trajectory rejected by "acceptability by latent space." Blue dotted line: Teaching trajectory. Red line: Rejected trajectory which is recontructed through projections.

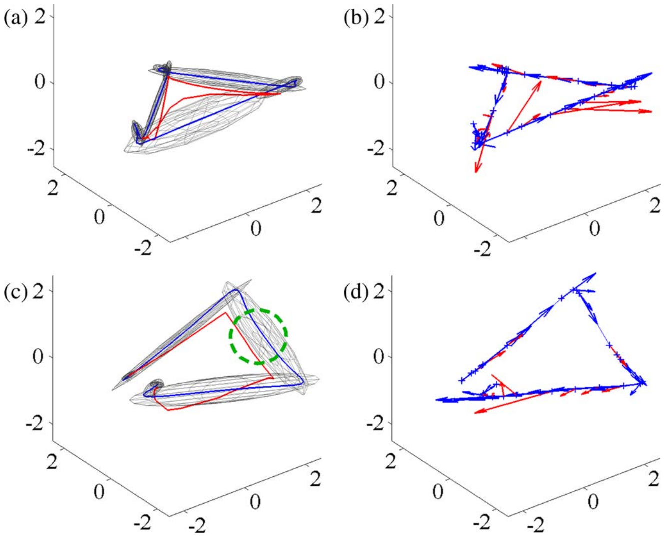


Fig. 10. Example of trajectory rejected by (a) and (b) pattern similarity and (c) and (d) spatial inconsistency. (Blue) Accepted and (red) rejected trajectories with arrows representing velocity directions. (Ellipsoids) Gaussian components in GMM.

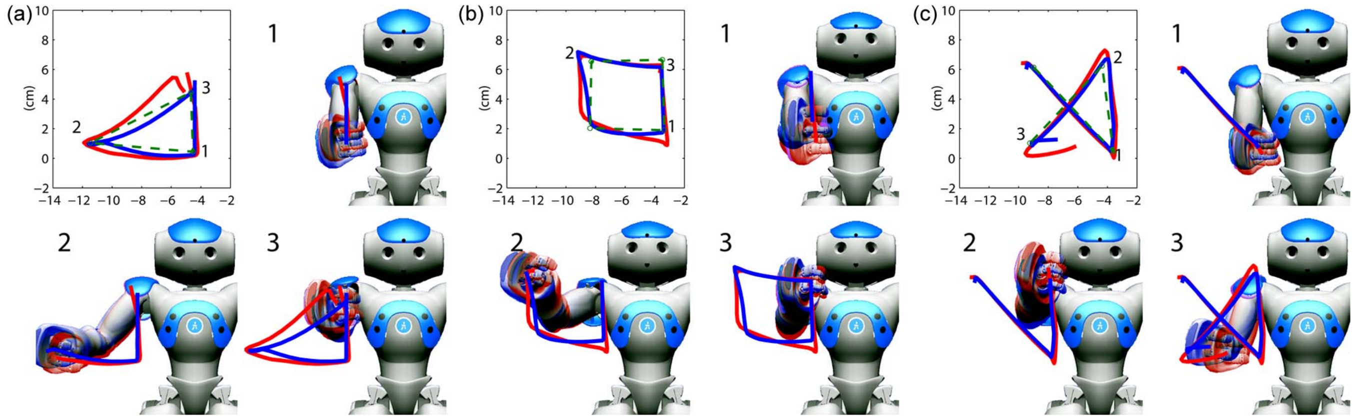


Fig. 11. Reproduction of drawing gestures: (a) triangular shape, (b) rectangular shape, and (c) cross-shape.

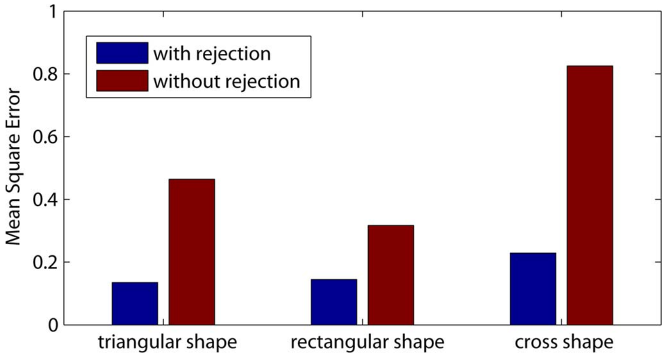


Fig. 12. Comparison of the RMSDs between the learning with and without rejection for the three drawing gestures.

Gaussian components in the most recently learned motion model. In Fig. 10(a) and (b), the velocity directions of the segments from the rejected trajectory differ from the accepted case. Fig. 10(c) and (d) shows an example of trajectory rejection by spatial inconsistency. The rejected trajectory has a similar pattern to the accepted trajectory, but it is discordantly located in the dotted circle region.

D. Learning Drawing Motions With/Without Rejection

To evaluate the effectiveness of the data evaluation procedure, we tested drawing gestures to focus on end point motions. The robot learned three drawing gestures: triangular, rectangular, and cross-shaped. Each motion was learned with and without rejection. For each drawing task, 12 kinesthetic teaching sessions were provided. Fig. 11 shows the three reproduced drawing behaviors. Insets 1, 2, and 3 illustrate sequential snapshots taken during each drawing task. In each inset, the robot's hand position produced from the learning with no rejection is overlapped with that from the learning with rejection (blue line: with rejection; red line: without rejection; dotted line: desired trajectory). It is visually evident that the blue trajectories tend to draw the shapes more accurately in all three cases. Fig. 12 verifies it by computing the root-mean-square deviation (RMSD) between the desired and the drawn trajectories. Even without the data evaluation procedure, the proposed learning algorithm can extract the desired behaviors to some extent even from teachings including unsatisfactory motions due to the probabilistic model's properties [8]. However, the explicit removal of bad teaching examples helps enhance motion accuracy.

IV. Conclusion

This paper has presented an incremental online learning algorithm for a robot to learn a behavior from teaching data. The learned behavior is expected to include the essential characteristics that the teaching data intended to express. A motion model is constructed, and the robot stores the model parameters to produce the behavior at any time necessary. Our algorithm requires no presumption on its model parameters and can construct the model incrementally through teaching trials. Our algorithm allows a robot to judge the properness of a teaching trial by itself and therefore is better suited for interactive scenarios between a teacher and a robot. A robot evaluates whether a newly encoded teaching trial can be well covered by the constructed latent space. Then, the robot scores the similarity of the teaching trial to its current learned motion model through a proposed metric. The similarity measure aims to detect if the teaching trial includes the critical characteristics of a targeted behavior but not to examine trajectory fitting. The experimental results verify the feasibility and power of our algorithm.

Recently, the learning by imitation schemes have been extended to include interactions with users under consideration on environments [22]–[24] and to refine or reuse policies by teacher feedback [25], [26]. They focus on interactive learning. Our work also bears similarity since learning is implemented interactively through adequate teaching trial selection by a robot. Our current learning algorithm does not consider constraints such as joint torque limit, robot dynamics, and contact task. The constraints may be taken care of by a controller if we regard our algorithm as a desired motion generator. However, a more intelligent motion command generation would take into account the issues. Hence, a natural extension would be to consider the underlying dynamics.

References

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, May 2009.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York: Springer-Verlag, 2008, pp. 1371–1394.

[3] H. Friedrich, S. Mnch, R. Fillmann, S. Bocionek, and M. Sassin, "Robot programming by demonstration (RPD): Supporting the induction by human interaction," *Mach. Learn.*, vol. 23, no. 2/3, pp. 163–189, May/Jun. 1996.

[4] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends Cognit. Sci.*, vol. 3, no. 6, pp. 233–242, Jun. 1999.

[5] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends Cognit. Sci.*, vol. 6, no. 11, pp. 481–487, Nov. 2002.

[6] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 144–151.

[7] D. B. Grimes, D. R. Rashid, and R. P. N. Rao, "Learning nonparametric models for probabilistic imitation," in *Proc. Adv. Neural Inf. Process Syst.*, 2006, vol. 19, pp. 521–528.

[8] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 286–298, Apr. 2007.

[9] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 105–112.

[10] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proc. IEEE Int. Conf. HRI*, Arlington, VA, 2007, pp. 255–262.

[11] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamic system modulation for robot learning via kinesthetic demonstrations," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1463–1467, Dec. 2008.

[12] S. Calinon and A. Billard, "Active teaching in robot programming by demonstration," in *Proc. IEEE Int. Conf. Robot Human Interact. Commun.*, 2007, pp. 702–707.

[13] M. Fujimoto and Y. A. Riki, "Robust speech recognition in additive and channel noise environments using GMM and EM algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, pp. 941–944.

[14] O. Arandjelovic and R. Cipolla, "Incremental learning of temporally-coherent Gaussian mixture models," in *Proc. BMBC*, 2006.

[15] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, Mar. 2005.

[16] K. Wang and T. Gasser, "Alignment of curves by dynamic time warping," *Ann. Stat.*, vol. 25, no. 3, pp. 1251–1276, Jun. 1997.

[17] H. Zhao, P. C. Yuen, and J. T. Kwok, "A novel incremental principal component analysis and its application for face recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 4, pp. 873–86, Aug. 2006.

[18] M. Artac, M. Jogan, and A. Leonardis, "Incremental PCA or on-line visual learning and recognition," in *Proc. Int. Conf. Pattern Recog.*, Quebec City, QC, Canada, 2002, vol. 3, pp. 781–784.

[19] J. Barbic, N. S. Pollard, J. K. Hodgins, C. Faloutsos, J. Y. Pan, and A. Safonova, "Segmenting motion capture data into distinct behaviors," in *Proc. Int. Conf. Graph. Interface*, 2004, pp. 185–194.

[20] O. Cappé and E. Moulines, "On-line expectation-maximization algorithm for latent data models," *J. Roy. Stat. Soc. Ser. B*, vol. 71, no. 3, pp. 593–613, Jun. 2009.

[21] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *Proc. Int. Conf. Robots Autom.*, Kobe, Japan, 2009, pp. 769–774.

[22] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using Gaussian mixture models," in *Proc. Int. Conf. Auton. Agent Multiagent Syst.*, May 2007, p. 223.

[23] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *J. Artif. Intell. Res.*, vol. 34, no. 1, pp. 1–25, Jan. 2009.

[24] D. Grollman and O. Jenkins, "Dogged learning for robots," in *Proc. Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 2483–2488.

[25] B. Argall, B. Browning, and M. Veloso, "Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot," *Robot. Auton. Syst.*, vol. 59, no. 3/4, pp. 243–255, Mar. 2011.

[26] E. Sauser, B. Argall, G. Metta, and A. Billard, "Iterative learning of grasp adaptation through human corrections," *Robot. Auton. Syst.*, vol. 60, no. 1, pp. 55–71, Jan. 2012.