Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Representability of Human Motions by Factorial Hidden Markov Models

Dana Kulić, Wataru Takano and Yoshihiko Nakamura

Abstract— **This paper describes an improved methodology for human motion recognition and imitation based on Factorial Hidden Markov Models (FHMM). Unlike conventional Hidden Markov Models (HMMs), FHMMs use a distributed state representation, which allows for more efficient representation of each time sequence. Once the FHMMs are trained with exemplar motion data, they can be used to generate sample trajectories for motion production, and produce significantly more accurate trajectories compared to single Hidden Markov chain models. Due to the additional information encoded in FHMMs models, FHMM models have a higher Kullback-Leibler distance compared to single Markov chain models, making it easier to distinguish between similar models. The efficacy of using FHMMs is tested on a database of human motions obtained through motion capture. The results show that FHMMs provide better generalization to new data when compared to conventional HMMs during motion recognition, as well as providing a better fit for generated data.**

I. INTRODUCTION

In order for robots to operate successfully in human environments, they will need to be able to perform a wide variety of both precise and gross full body motions. In particular, in the case of humanoid robots, the ability to learn primitive and complex motion patterns by observing and imitating humans is highly desirable. Robot imitation from observation has received considerable attention in the research literature [1], [2].

Our motion model is inspired by the mirror neuron system, found in humans and other primates [3], [4]. The mirror neuron system is believed to be a direct-matching mechanism, whereby observed actions are understood when the visual representation of the observed action is mapped onto the *observer's* motor representation of the same action. The same neuronal representation is used for both motion recognition and motion generation. This hypothesis is supported by both experiments with monkeys and humans, where observation of a motor action generates a neural response in the motor areas of the brain corresponding to the observed action. The neurons which respond in this manner have been named *mirror neurons*. In monkeys, mirror neurons activate only when goal directed actions are observed (for example, grasping an object, or biting into food), but not when the demonstrator mimics the action without the object being present. However, mirror neurons also activate when the monkey cannot observe the action visually, but other means of inferring the action are available (eg. sound or previous knowledge), indicating that, in monkeys, the mirror neurons are used for action understanding, and not primarily for imitation learning. For humans, on the other hand, mirror neurons fire for both goal directed actions and for non-goal directed movements [4]. In addition, brain imaging studies indicate that human mirror-neurons code for both the action *and* for the movements forming an action. These two important differences seem to indicate that in humans, mirror neurons are used both for action understanding and imitation learning.

In previous research [5], [6], the mirror neuron model has been applied to humanoid robots, where observed human motion primitives have been encoded using Hidden Markov Models, and used for humanoid robot motion generation. The developed framework can be used to code for both non-goal directed movements, as well as goal-oriented movements, by incorporating additional sensor data about the goal, and/or changing to a goal-based co-ordinate frame [7].

Hidden Markov Models (HMMs) have been frequently used for modeling human motions. HMMs efficiently abstract time series data, and can be used for both subsequent motion recognition and generation. For example, Billard et al. [8] use HMM models for motion recognition and generation of humanoid motion patterns. The Bayesian Information Criterion (BIC) is used to select the optimal number of states for the HMM, by selecting the fewest number of states which adequately perform recognition of the training data. Following HMM training, spline fitting of the state output observation vector means is performed to generate the trajectory. In the experiments, three different tasks are demonstrated using the HOAP-2 humanoid robot. The robot is trained using kinesthetic training.

Takano et al. [9], [10] describe a hierarchical system of HMMs for learning and abstracting both human motion patterns and human to human interaction patterns during combat. The lower layer of HMMs abstract the primitive motion patterns, such as kick, punch, etc., based on observations of joint angles and velocities obtained from a motion capture system. A sampling based algorithm for generating motion based on the trained HMM is also presented.

When using HMMs for both motion recognition and motion generation, there is a tradeoff between recognition and generation performance, in particular when selecting the number of states of the model. A small number of states will give good generalization and recognition performance, while a large number of states will give better generation performance, at the risk of over-fitting and poor generalization. In addition, due to the binary state representation in the hidden Markov chain, the representational capacity of an HMM is

The authors are with the Department of Mechano-Informatics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, Japan {dana,takano,nakamura}@ynl.t.u-tokyo.ac.jp

1-4244-0912-8/07/$25.00 ©2007 IEEE.

limited [11].

In this paper, we develop a methodology for modeling human motion data using Factorial Hidden Markov Models. This representation allows for more accurate modeling of the motion dynamics by increasing the number of states used to represent the motion, while at the same time reducing the likelihood of overfitting encountered with conventional HMMs. Section II describes the differences between FHMMs and conventional HMMs. In Section III, the use of FHMMs for modeling human motion is described. Section IV outlines the experimental results, while the conclusions and directions for future work are outlined in Section V.

II. FACTORIAL HIDDEN MARKOV MODELS

A Hidden Markov Model (HMM) abstracts the modeled data as a stochastic dynamic process. The dynamics of the process are modeled by a hidden discrete state variable, which varies according to a stochastic state transition model $A[N, N]$, where N is the number of states in the model. Each state value is associated with a continuous output distribution model $B[N, K]$, where K is the number of outputs. Typically, for continuous data, a Gaussian or a mixture of Gaussians output observation model is used. HMMs are commonly used for encoding and abstracting noisy time series data, such as speech [12] and human motion patterns [8], [6]. Efficient algorithms have been developed for model training (the Baum-Welch algorithm), pattern recognition (the forward algorithm) and hidden state sequence estimation (the Viterbi algorithm) [12]. The Baum-Welch algorithm is a type of iterative Expectation-Maximization (EM) algorithm. In the Expectation step, given a set of model parameters and a data sequence, the posterior probabilities over the hidden states are calculated. Then, in the Maximization step, a new set of model parameters are calculated which maximize the log likelihood of the observations. The EM algorithm has time complexity $O(TN^2)$, where T is the length of the observation sequence, and N is the number of states. Once trained, the HMM can also be used to generate a representative output sequence by sampling the state transition model to generate a state sequence, and then sampling the output distribution model of the current state at each time step to generate the output time series sequence. A schematic of an HMM is shown in Fig. 1.

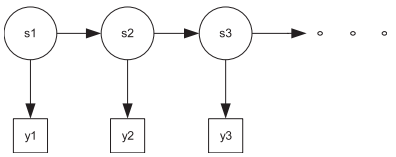

Fig. 1. Hidden Markov Model

A Factorial Hidden Markov Model (FHMM) [11] is a generalization of the HMM model, where there may be multiple dynamic processes interacting to generate a single output. In an FHMM, multiple independent dynamic chains contribute to the observed output. Each dynamic chain m is represented by its own state transition model $A_m[N_m, N_m]$ and output model $B_m[N_m, K]$, where M is the number of dynamic chains, N_m is the number of states in dynamic chain m, and K is the number of outputs. At each time step, the outputs from all the dynamic chains are summed, and output through an expectation function to produce the observed output. The expectation function is a multivariate Gaussian function with the chain output as the means, and a covariance matrix representing the signal noise. For example, FHMMs have been used to model speech signals from multiple speakers [13], and the dynamics of a robot and changing environment for simultaneous localization and mapping [14]. Fig. 2 shows a schematic of an FHMM.

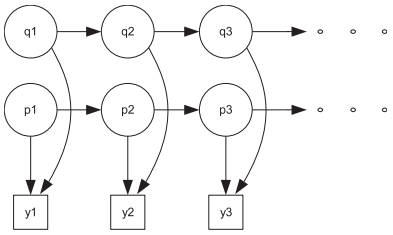

Fig. 2. Factorial Hidden Markov Model

An FHMM can be trained by an adaptation of Baum-Welch algorithm [11]. However, this (exact) algorithm has a time complexity of $O(TMN^{M+1})$ where T is the length of the data sequence. This means that the time complexity increases exponentially with the number of chains. This is due to the fact that, even though the dynamic chains are independent of each other, they become dependent given an observation sequence. Therefore, the E-step of the EM algorithm becomes intractable for a large number of chains. Faster algorithms, for which the time complexity is quadratic as a function of the number of chains, have been developed for FHMM training, which implement an approximate rather than the exact E step. These are based on variational methods [11], or generalized backfitting [15]. During the E-step, the expectation of the vector state occupation probabilities $\gamma_t(i)$, the expectation of the matrix of state occupation probabilities at two consecutive time steps $\xi_t(i, j)$, and the expectation of the joint probability between the state vectors of two chains $\eta_t(m, n)$ are computed. During the M-step, the covariance matrix C, the output model $B_m[N_m, K]$, the state transition model $A_m[N_m, N_m]$, and the initial state distribution probabilities $\pi_m(i)$ are updated. The pseudo-code for a generic FHMM training algorithm as developed by Ghahramani and Jordan [11] is summarized in Fig. 3.

Once the FHMM is trained, pattern recognition can be implemented by an adaptation of the forward-backward algorithm [11].

To generate a representative sequence from an FHMM, at each time step, the value of the hidden state for each chain is determined using the state transition model A_m, and the output contribution for each chain determined based on the output model for the current hidden state. The output is then formed by sampling from an output distribution whose expectation is the sum of all the respective chain outputs.

```
1: procedure FHMMTRAIN
2:     Initialize A_m[N_m, N_m], B_m[N_m, K], π_m(i) and C
3:     for cycle ← 1, maxIterations do
4:         E-Step
5:         for n ← 1, numSequences do
6:             Forward Algorithm
7:             Backward Algorithm
8:             Estimate γ_t(i)
9:             Estimate ξ_t(i, j)
10:            Estimate η_t(m, n)
11:        end for
12:        M-Step
13:        Calculate B_m[N_m, K]
14:        Calculate C
15:        Calculate A_m[N_m, N_m]
16:        Calculate π_m(i)
17:    end for
18: end procedure
```

Fig. 3. Generic FHMM Training Algorithm Pseudocode

III. HUMAN MOTION PATTERN REPRESENTATION USING FHMMS

HMMs have been widely used to model human motion data for recognition and generation [8], [6]. However, when using the same HMM structure for both recognition and generation, there is an inherent tradeoff when selecting the HMM model, (i.e. the number of model states). An HMM model with a low number of states will be better at generalizing across variable data, and better at correctly recognizing new data. In general, an HMM with a low number of states (5 - 20) provides excellent recognition performance [8], [6]. When automatically selecting the number of states based on Bayesian [8] or Akaike [16] information criteria, under 10 states are usually selected for typical human motion patterns such as walking, kicking, punching, etc. However, in this case, recognition criteria only are used to select the appropriate number of states. At the same time, a model with a low number of states will not be able to faithfully reproduce the observed motion through generation. On the other hand, a large state model will be better at reproducing the observed motion, but will be prone to over-fitting. Factorial HMMs, which use a distributed rather than a single multinomial state representation, provide a more efficient approach for combining good generalization for recognition purposes with sufficient detail for better generation.

Once a group of motion patterns has been generated, they can be compared by using a probabilistic distance measure [12]:

$$D(\lambda_1, \lambda_2) = \frac{1}{T}[logP(O^{(2)}|\lambda_1) - logP(O^{(2)}|\lambda_2)] \quad (1)$$

where λ_1, λ_2 are two HMM models, $O^{(2)}$ is an observation sequence *generated* by λ_2 and T is the length of the observation sequence. Since this measure is not symmetric, the average of the two intra HMM distances is used to form a symmetric measure.

The distance measure is based on the relative log likelihood that a generated sequence is generated by one model, as compared to a second model. It represents a Kullback-Leibler (KL) distance between the two models. The formulation of the distance based on the model probability means that this measure can similarly be applied to Factorial HMM models, by using the modified forward procedure [11] to calculate the log likelihood, as well as used to compare FHMM and HMM models.

The distance measure quantifies the level of difficulty in discriminating between two models λ_1, λ_2. The distance measure can also be used to construct a motion pattern vector space by using multidimensional scaling [9]. In this method, a vector space is constructed such that the error between the actual distances and the distances in the vector space is minimized. By encoding more information about each pattern, FHMMs can improve the ability to discriminate between motion patterns, which can be especially useful when there are many similar motion patterns. Using the more detailed FHMM models increases the intra-model distances, as depicted conceptually in Fig. 4. In addition, if the FHMM and HMM models of the same motion remain sufficiently similar, FHMM models may be combined with HMM models, by using FHMM models only in dense areas of the motion model space where better discriminative ability is required (shown in Fig. 5).

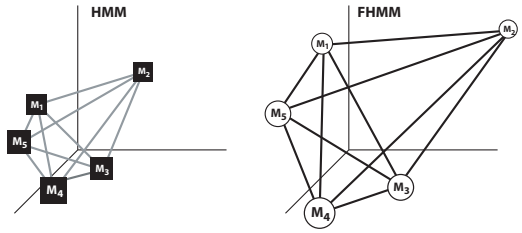

Fig. 4. Schematic comparing an HMM model vector space and an FHMM model vector space (the axes represent the principal directions of the vector space)

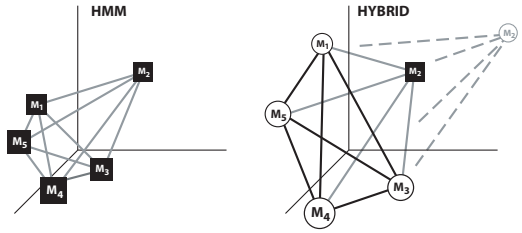

Fig. 5. Schematic comparing an HMM model vector space and a hybrid HMM-FHMM model vector space (the axes represent the principal directions of the vector space)

A. Deterministic Motion Generation

Once the FHMM model is trained, it can be used for new motion recognition or robot motion generation. If a linked FHMM model is used, recognition and generation are performed by executing the forward algorithm or the generation algorithm on the root chain and the target chain.

When the generated motion sequence is to be used for robot motion commands, we do not want to introduce the

noise characteristics abstracted by the HMM model. In this case, we use a greedy policy to estimate the optimum state sequence. First, for each chain m, the starting state q_0^m is selected by choosing the highest value from the initial state probability distribution. At each state, the state duration is calculated based on the state transition matrix,

$$\bar{d}_i^m = \frac{1}{1 - a_{ii}^m} \tag{2}$$

Following $\bar{d}_i^m$ samples in state i, the next state is selected by greedy policy from the state transition matrix, excluding the $1 - a_{ii}^m$ probability. If the model type is front-to-back, the algorithm iterates until the final state is reached, otherwise the state sequence is generated for the specified number of time steps. Once the state sequence has been generated for each chain, the output sequence is calculated by summing the contribution from each chain at each time step, based on that chain's current state value. The generation algorithm pseudo-code is shown in Fig. 6. Alternatively, if a motion most similar to a recently observed motion is required, the optimal state sequence could be generated by using the Viterbi algorithm [17].

After the trajectory is generated, some low-pass filtering or smoothing is required as a post-processing step to eliminate the artifacts caused by discrete state switching and generate a smooth trajectory for use as a command input.

procedure GENERATE
  for $m \leftarrow 1, M$ **do**
    Initialize
    $q_0^m \leftarrow argmax_i(\pi_m(i))$
    while $t < T, i \neq i_{terminal}$ **do**
      $\bar{d}_i^m \leftarrow \frac{1}{1-a_{ii}^m}$
      $q_{t:t+\bar{d}_i^m} \leftarrow q_t$
      $q_{t+1} \leftarrow argmax_i a_{ij}^m, i \neq j$
    end while
  end for
  for $t \leftarrow 0, T$ **do**
    $y_t \leftarrow \sum_{m=1}^{M} B^m(q_t^m)$
  end for
end procedure

Fig. 6. Greedy Generation Algorithm Pseudocode

IV. SIMULATION RESULTS

In the first set of simulations, the performance of the FHMMs is compared to conventional HMMs for recognition and generation. Both models are tested on a data containing a series of 9 different human movement observation sequences obtained through a motion capture system [18]. The data set contains joint angle data for a 20 degree of freedom humanoid model from multiple observations of walking (WA - 28 observations), cheering (CH - 15 observations), dancing (DA - 7 observations), kicking (KI - 19 observations), punching (PU - 14 observations), sumo leg raise motion (SL - 13 observations), squatting (SQ - 13 observations), throwing (TH - 13 observations) and bowing (BO - 15 observations). Fig. 7 shows selected frames of an animation for an example of a walking motion from the data set.

A set of nine HMM and FHMM models each are trained on the data, one for each motion type. Each FHMM consists of a two chains of front-to-back (Bakis type) hidden Markov chains of 15 states each. The corresponding HMMs are also front-to-back type, and contain an equivalent (15x15) number of states. For both model types, the covariance matrix was constrained to be diagonal during training, and the minimum covariance was constrained to 0.001, to avoid numerical underflow/overlow during training. Each model was trained on 5 randomly selected exemplars of a motion type. The learning algorithm for each model was run for 100 iterations. Each model was then tested by computing the log-likelihood of observing the following observation sequences: an example from the training set, a randomly drawn example of the same motion outside the training set, and an example of a different motion. This testing procedure was repeated for 100 trials. At the start of each trial, the state transition parameters were initialized to random values. The mean and variance parameters in the output distribution model were initialized by calculating the mean values and covariance of the output vector over all the training data. The means were initialized by sampling from a Gaussian distribution with the data based means and covariance. The average log likelihood and standard deviation for each test case are shown in Tables I and II, for the FHMM models and HMM models, respectively. The training column indicates the average log likelihood and standard deviation over the 100 test cases for a randomly selected exemplar out of the training set, the novel column indicates the log likelihood distribution for an example of the same motion, which was outside of the training set, and the different column indicates the log likelihood distribution for an example of a different motion.

TABLE I
FHMM RECOGNITION RESULTS

Motion Type	Training	Novel	Different
Walk	2134± 165	1616± 205	-66589± 15158
Cheer	2468± 284	1220± 724	-60908± 19899
Dance	3016± 288	1675± 913	-60922± 15016
Kick	1344± 170	733 ± 651	-48060± 17208
Punch	2613± 295	1281± 868	-52786± 23188
Sumo Leg	2818± 350	791± 1418	-49262± 20734
Squat	2617± 229	2025± 302	-58060± 17283
Throw	3223± 254	1893± 641	-48517± 21690
Bow	2599± 295	1821± 681	-54118± 18877

As can be seen from the results in Tables I and II, HMMs perform well at recognizing data from the training set, but show a significant drop in performance when recognizing new data, indicating over-fitting. On the other hand, FHMMs demonstrate significantly better generalization, and show a much smaller drop in performance when recognizing new motions of the same type as compared to the HMM. The additional structure imposed by the FHMM assumption that the dynamic chains are independent (but become dependant given an observation sequence) reduces the likelihood of overfitting for the FHMM model. Both models are equally

good at rejecting data from a different motion sequence.

TABLE II
HMM RECOGNITION RESULTS

Motion Type	Training	Novel	Different
Walk	2818 ± 297	1150 ± 536	-48973± 25913
Cheer	3448 ± 497	-2461 ± 5236	-50147± 25914
Dance	4756 ± 687	-3234 ± 7996	-59168± 15563
Kick	2028 ± 353	-405 ± 2003	-19262± 17553
Punch	3828 ± 611	-8497 ± 18242	-53006± 19779
Sumo Leg	4929± 1068	-5727 ± 6611	-45079± 21941
Squat	3535 ± 457	1606 ± 738	-31098± 26645
Throw	4776± 1454	-15468± 26723	-53540± 20551
Bow	3884 ± 519	-1980 ± 5968	-52923± 24598

Fig. 9 shows an example of generation results for the left and right knee joints during a walking motion, before any post-processing of the trajectory has been applied. One of the walking motions used during training of the models is shown for comparison. Fig. 8 shows frames from an animation of a walking motion generated by the walk FHMM. As can be seen in Fig. 9, in motion of the left knee, the HMM fitted state outputs at two different peaks, likely due to the same peak executed at different speeds in the training set, indicating overfitting.

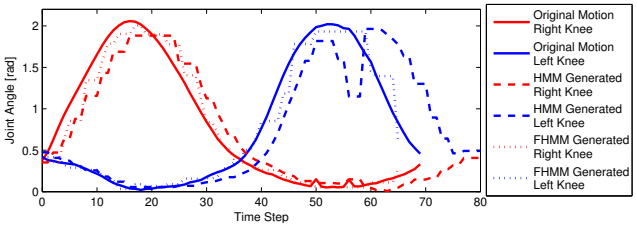

Fig. 9. Comparison of Generation Results of the HMM (15x15 states) and FHMM (two chains of 15 states) for the Knee Joints during a Walking Motion, prior to applying any post processing (i.e., low-pass filtering)

In the second set of experiments, the increase in accuracy and discrimination power of using an FHMM was compared against using a single chain HMM with a small number of states. The same set of FHMMs was used as in the first set of experiments above, while the HMMs consisted of front-to-back models with 15 states each. Fig. 10 shows a comparison between the FHMM and the low state number HMM before any trajectory post processing has been applied. As can be seen in Fig. 10, due to the higher number of states available to represent the motion, FHMMs achieve better spacial accuracy compared to a single chain HMM model.

Table III shows the intra model distances between each of the HMM models, and Table IV shows the equivalent intra-model distances between the FHMM models. The distances are calculated according to Eq. **??**. The distances are a measure of the dissimilarity of the models. The average intra model distance between HMM models is 377, while the average intra model distance between FHMM models is 467. The use of FHMMs increases the models discriminative ability by enlarging the distances between each model in the motion pattern vector space.

TABLE III
HMM INTRA DISTANCES

	WA	CH	DA	KI	PU	SL	SQ	TH	BO
WA	0								
CH	488	0							
DA	728	723	0						
KI	485	444	526	0					
PU	713	696	423	382	0				
SL	435	513	469	400	463	0			
SQ	554	619	534	594	576	295	0		
TH	702	648	454	421	78	447	662	0	
BO	562	477	448	469	552	336	538	561	0

TABLE IV
FHMM INTRA DISTANCES

	WA	CH	DA	KI	PU	SL	SQ	TH	BO
WA	0								
CH	659	0							
DA	740	734	0						
KI	638	658	604	0					
PU	741	740	718	636	0				
SL	528	699	689	720	674	0			
SQ	740	739	729	688	642	737	0		
TH	732	729	567	580	115	698	696	0	
BO	740	739	504	678	643	737	695	638	0

Table V compares the distances between the FHMM and HMM models, using the probabilistic distance measure. The diagonal terms indicate the model distance between the FHMM and HMM model for the same motion, while the off-diagoal terms indicate the model distance between the HMM and FHMM model of different motions. As expected, the inter-model distance for the same motion type is very small (average 10.2), which suggests it would be possible to use a mixture of HMM and FHMM models, depending on the complexity of the motion modeled, and the degree of similarity between the motions modeled. For example, in the data set tested here, it may be advantageous to use FHMMs for describing only the punch and throw motions, as they are quite similar to each other and the most difficult to distinguish [16].

TABLE V
INTER MODEL DISTANCES

HMM Mdls	FHMM Models								
	WA	CH	DA	KI	PU	SL	SQ	TH	BO
WA	8.3								
CH	647	11.2							
DA	734	731	8.3						
KI	612	636	553	8.8					
PU	736	734	652	442	11.8				
SL	509	680	675	657	645	16.5			
SQ	735	725	705	617	596	497	6.8		
TH	726	722	512	419	81	493	682	9.7	
BO	735	645	475	576	587	517	576	607	10.6

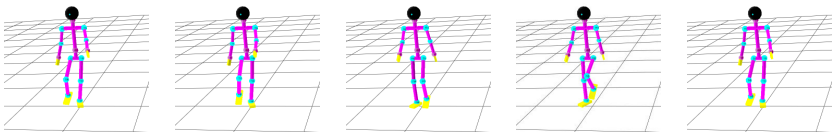

Fig. 7. Sample Walking Motion - Animated from joint angle data provided by the motion capture system

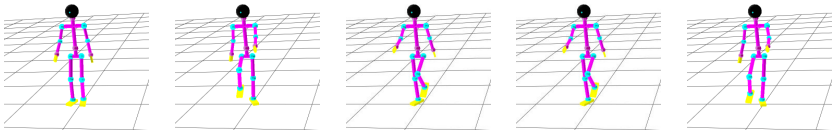

Fig. 8. FHMM Generated Walking Motion - Animated from data generated by the trained Walk motion model

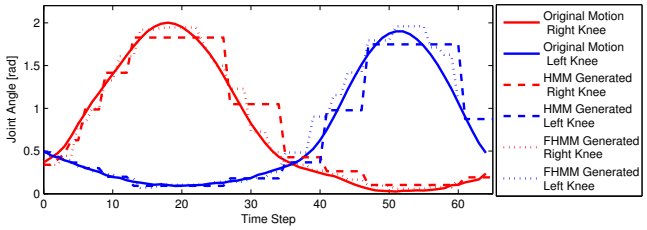

Fig. 10. Comparison of Generation Results of the HMM (15 states) and FHMM (two chains of 15 states) for the Knee Joints during a Walking Motion, prior to applying any post processing (i.e., low-pass filtering)

V. CONCLUSIONS AND FUTURE WORK

This paper presents a novel approach for abstracting human motion patterns using Factorial Hidden Markov Models. Experimental results show that FHMMs provide better generalization capability than single-chain HMMs, especially as the number of states increases. Using a larger number of states allows a model to generate a more faithful reproduction of the abstracted data. Similar to HMMs, FHMMs can be used to discriminate between motion types, based on the KL distance between motions. Compared to HMM models, using FHMMs increases the intra-model distance between groups, which could help make model discrimination easier, especially if the models are very similar. Distance measurements remain consistent between HMM and FHMM models, suggesting that it may be possible to use a combination of the two model types, depending on the accuracy and discrimination requirements for each motion. For example, HMM models could be used for areas of the motion space where there are few, well separated examples, while applying FHMMs in dense areas of the motion space, where a better model is required to distinguish between similar motions. Directions for future work include the adaptive selection of the appropriate model based on the similarity of observed motions.

VI. ACKNOWLEDGMENTS

This work is supported by the Japanese Society for the Promotion of Science grant 18.06754 and Category S Grant-in-Aid for Scientific Research 15100002.

REFERENCES

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

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

[3] G. Rizzolatti, L. Fogassi, and V. Gallese, "Neurophysical mechanisms underlying the understanding and imitation of action," *Nature Reviews: Neuroscience*, vol. 2, pp. 661–670, 2001.

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

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

[6] W. Takano, "Stochastic segmentation, proto-symbol coding and clustering of motion patterns and their application to signifiant communication between man and humanoid robot," Ph.D. dissertation, University of Tokyo, 2006.

[7] D. Lee and Y. Nakamura, "Mimesis scheme using a monocular vision system on a humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2162–2167.

[8] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, pp. 370–384, 2006.

[9] W. Takano, K. Yamane, T. Sugihara, K. Yamamoto, and Y. Nakamura, "Primitive communication based on motion recognition and generation with hierarchical mimesis model," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 3602–3608.

[10] W. Takano, K. Yamane, and Y. Nakamura, "Primitive communication of humanoid robot with human via hierarchical mimesis model on the proto symbol space," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, 2005, pp. 167–174.

[11] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[12] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[13] A. Betkowska, K. Shinoda, and S. Furui, "Fhmm for robust speech recognition in home environment," in *Symposium on Large Scale Knowledge Resources*, 2006, pp. 129–132.

[14] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Neural Information Processing Systems*, 1999.

[15] R. A. Jacobs, W. Jiang, and M. A. Tanner, "Factorial hidden markov models and the generalized backfitting algorithm," *Neural Computation*, vol. 14, pp. 2415–2437, 2002.

[16] D. Kulić, W. Takano, and Y. Nakamura, "Incremental on-line hierarchical clustering of whole body motion patterns," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2007.

[17] D. Lee and Y. Nakamura, "Mimesis from partial observations," in *International Conference on Intelligent Robots and Systems*, 2005, pp. 1911–1916.

[18] H. Kadone and Y. Nakamura, "Symbolic memory for humanoid robots using hierarchical bifurcations of attractors in nonmonotonic neural networks," in *International Conference on Intelligent Robots and Systems*, 2005, pp. 2900–2905.