

Incremental Learning of Human Behaviors using Hierarchical Hidden Markov Models

Dana Kulić and Yoshihiko Nakamura

Abstract—This paper proposes a novel approach for extracting a model of movement primitives and their sequential relationships during online observation of human motion. In the proposed approach, movement primitives, modeled as hidden Markov models, are autonomously segmented and learned incrementally during observation. At the same time, a higher abstraction level hidden Markov model is also learned, encapsulating the relationship between the movement primitives. For the higher level model, each hidden state represents a motion primitive, and the observation function is based on the likelihood that the observed data is generated by the motion primitive model. An approach for incremental training of the higher order model during online observation is developed. The approach is validated on a dataset of continuous movement data.

I. INTRODUCTION

The study and analysis of human movement is an important area of research, with potential for a great many applications, including activity recognition, understanding and predicting movement and intent, and learning from observation. In particular, for humanoid robots, the ability to understand and imitate human behavior will be a key skill to allow such robots to operate and become useful in the human environment.

It is postulated that human movement consists of atomic units called *primitives*, which are sequenced together and combined to form more complex behavior [1]. In order to learn and imitate human behavior, humanoid robots must learn both the movement primitives and the higher level order governing the sequencing of primitives. In addition, in order to be applicable in arbitrary human domains, where the task and the demonstrator may be changing, the learning system should be capable of continuous learning of both primitives and their sequencing, during online observation of human motion. In previous work [2], [3], we have been developing algorithms for incremental learning of movement primitives from continuous observation, based on stochastic modeling. In this paper, we propose an approach for simultaneously also learning the higher level order in the sequencing, by extending the stochastic model to higher levels of abstraction.

This work was partially supported by the Japanese Society for the Promotion of Science Category S Grant-in-Aid for Scientific Research 20220001 and by the Natural Sciences and Engineering Research Council of Canada.

D. Kulić is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada dkulic@ece.uwaterloo.ca

Y. Nakamura is with the Department of Mechano-Informatics, University of Tokyo, Tokyo, Japan nakamura@ynl.t.u-tokyo.ac.jp

A. Related Work

Robot learning from observation is an active area of research [4], [5]. The decomposition of human movement into motion primitives has been a popular approach, with several different frameworks proposed for modeling primitives, including neural networks [6], [7], stochastic models such as Hidden Markov Models (HMMs) or Gaussian Mixture Models (GMMs) [8], [9], phase oscillators for rhythmic motions [10], dynamical models [11], polynomial functions [12] or weighted graph structures [13]. In addition, learning at different hierarchy levels has also been studied. Several researchers consider the imitation of the motion primitives themselves [9], [10], [8], while other works consider how a system may learn a sequence of primitives which are defined a-priori [14], [15]. Takano and Nakamura [16] and Kulić et al. [17] consider the case where both the motion primitives and their sequencing is learned simultaneously. However, when motion primitives are being learned, as noted by Breazeal and Scasellati [4], the majority of algorithms discussed in the literature assume that the motions to be learned are segmented and clustered a-priori, and that the model training takes place off-line.

Taylor et al. [18] describe an approach for modeling human motion using a conditional restricted Boltzmann machine (CRBM). The learned model can generate continuous motion sequences, as well as learn the transitions between motions. Once the low level model consisting of individual motion patterns has been trained, additional higher order layers can be added to model the higher order structure of motion patterns. Jenkins and Matarić [19] describe a system for modeling motion in a lower dimensional subspace. In their algorithm, continuous time series data is first segmented using the *kinematic centroid segmentation* algorithm, and is then embedded in a lower dimensional space using the spatio-temporal Isomap (ST-Isomap) algorithm [20]. Once the data has been reduced, it is clustered into groupings using the “sweep-and-prune” technique. Once a model of the primitive behaviors is formed, a higher level re-processing of the data can be performed to discover meta-behaviors, i.e., probabilistic transition probabilities between the behaviors. While both these systems autonomously segment and cluster data, the algorithms cannot operate incrementally, as the entire range of motions is required to form the lower-dimensional space embedding and perform model training.

Takano et al. [16] describe a hierarchical system of Hidden Markov Models (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. Once the primitive motion patterns (the proto symbols) are learned, a proto symbol space is generated using dissimilarity measures between the HMMs. The upper layer of HMMs encode the interaction patterns between two participants during combat. However, this approach also requires an offline training component to construct the proto symbol space, requiring that the motion primitives be learned offline and that their number be specified a-priori. A similar approach for modeling physical human-robot interaction is proposed by Lee et al. [21].

Kulić et al. [17] propose an approach for simultaneously learning an arbitrary number of motion primitives and their sequential relationship during online observation. In their approach, observed motion time series data stream is first stochastically segmented into potential motion primitive segments. The segmented motions are then passed to an incremental clustering algorithm which forms a tree representation of the learned motions, and abstracts each motion type into a generative model. Concurrently, a graph model is built representing the sequential relationship between the motion primitives. However, due to the deterministic graph model, this approach can be sensitive to motion primitive recognition errors.

In the domain of mobile robot trajectory learning, including vehicle or pedestrian trajectories, Vasquez et al. [22] propose an approach for incrementally learning a set of vehicle or pedestrian trajectories using Growing Hidden Markov Models (GHMMs). In the proposed approach, the system incrementally learns (following the observation of the location of the vehicle at each time step) a topological map via the Instantaneous Topological Map (ITM) algorithm [23]. In this algorithm, for each observed location, nearest nodes and edges are updated, including centroid adaptation for the closest node, as well as edge and node addition/deletion. Following the update of the topological map, the GHMM structure is updated by adding a state in the GHMM corresponding to each new node in the topological map, the state connectivity equal to the connectivity in the topological map, and initializing the observation model to a Gaussian centered on the centroid location computed by the topological map with a fixed size Gaussian. The initial state distribution and the state transition model are then learned incrementally using an incremental variant of the Baum-Welch algorithm.

B. Proposed Approach

The aim of our research is to develop robots that can learn motion primitives and higher level behaviors on-line while observing and interacting with a human partner over extended periods of time. In previous work, we have been developing algorithms for autonomous segmentation and extraction of movement primitives from continuous observation of human motion [2], [3]. Using continuous time-series data as the input, we first segment the data into potential motion primitives, using a modified version of the Kohlmorgen

and Lemm [24] algorithm for unsupervised segmentation. Next, the extracted segments are input into an automated clustering and hierarchical organization algorithm [2]. The resulting clusters are then used to form a generative model of each abstracted motion primitive, which can be used for subsequent recognition of previously extracted motion primitives as well as for motion generation on a humanoid robot. In previous work [17], we also proposed a method for learning the sequencing of the motion primitives based on a motion primitive graph. However, since the motion primitive graph encodes movement primitive transitions in a deterministic fashion, it can be sensitive to errors in motion primitive recognition, leading to the formation of spurious edges. In this paper, we propose an alternate, probabilistic approach for learning the higher level ordering between movement primitives, based on a higher abstraction HMM modeling the transitions between motion primitives. In the proposed approach, each hidden state of the higher order model corresponds to a motion primitive. The system is illustrated in Figure 1. We propose a novel approach for modeling the observation function at each state and incrementally learning the transition model during online observation of human motion. Section 2 summarizes the previously proposed segmentation and clustering algorithms used to extract the motion primitives [2], [3], while Section 3 describes the higher order model and the proposed incremental learning algorithm. In Section 4, the results of experiments verifying the complete algorithm on a continuous stream of human motion capture data are reported. Section 5 concludes the paper and provides directions for future work.

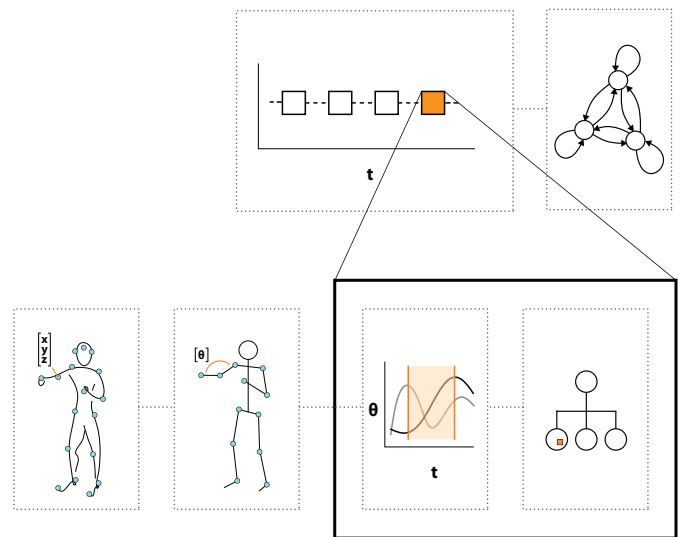


Fig. 1. Overview of the system. The bottom row show the lower level processing, detailed in Section II. $[x, y, z]$ marker position data is first converted to joint angle data for the humanoid kinematic model using on-line inverse kinematics. The joint angle data is next segmented using on-line stochastic segmentation. Segments are incrementally clustered and organized in a tree-structure. The top row shows the upper level processing, detailed in Section III. At the higher level of the hierarchy, sequences of motion primitives are observed and used to incrementally train a higher order model representing the temporal relationships between the primitives.

II. SEGMENTATION AND MOTION PRIMITIVE LEARNING

In the proposed approach [2], [3], the on-line learning system autonomously segments, clusters and learns the sequencing of full-body motion primitives from on-line observation of full body human motions. First, the incoming continuous time series data is autonomously segmented into potential motion primitive segments. The Kohlmorgen and Lemm segmentation algorithm [24], [25] is used to perform the segmentation. This algorithm finds optimal segment points by defining a Hidden Markov Model over a set of sliding windows defined over recently observed data. A state transition model is defined such that the cost is lowest to remain in the same state (i.e, there is an increased cost to switch states), and an observation function based on the difference between the current data and the data in the state window. The optimal sequence of segments is found by formulating an optimization based on the tradeoff between data similarity and the cost of switching to a new state. The optimization problem is solved via an online version of the Viterbi algorithm.

Once the incoming time series data has been segmented into potential primitives, each segment is sequentially passed to the clustering module. In the proposed clustering approach [2], a hierarchical tree structure is incrementally formed representing the motions learned by the system. Each node in the tree represents a motion primitive, which can be used to recognize a similar motion, and also to generate a model of the motion. Each motion primitive is modeled as an HMM or a Factorial HMM, which abstracts the modeled data as stochastic dynamic process. In the case of an HMM, the dynamics of the motion are modeled by a single 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 is associated with a continuous output distribution model $B[N, K]$, where K is the number of outputs in the observation vector. For continuous data such as motion data in this case, a Gaussian or a mixture of Gaussians output observation model is used.

$$b_i(\mathbf{O}) = \sum_{c=1}^C w_{ic} C_{ic} \exp\left(-\frac{1}{2}(\mathbf{O} - \boldsymbol{\mu}_{ic})^T \Sigma_{ic}^{-1} (\mathbf{O} - \boldsymbol{\mu}_{ic})\right) \quad (1)$$

$$C_{ic} = \frac{1}{(2\pi)^{K/2} |\Sigma_{ic}|^{1/2}} \quad (2)$$

where \mathbf{O} represents the observation vector at a given time step, $b_i(\mathbf{O})$ represents the likelihood that the given observation vector was generated by hidden state i , C represents the number of Gaussian distributions in the mixture of Gaussians, and $\boldsymbol{\mu}_{ic}$ and Σ_{ic} represent the mean and covariance matrix of Gaussian distribution c in state i .

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 algorithm initially begins with one node (the root node). Each time a motion is observed from the teacher, it is encoded into an HMM or FHMM and compared to existing nodes via a tree search algorithm, and placed into the closest node. Each time a new observation is added to a node, local clustering is performed within the observations of the node, using a standard clustering technique [26] based on the Kullback-Leibler distance between motion models. If a cluster with sufficiently similar data is found, a child node is formed with this data subset, representing a newly learned motion primitive. A node model is then trained with the data from the clustered motions. The model can be used for subsequent recognition and generation. Therefore the algorithm incrementally learns and organizes the motion primitive space, based on the observations seen thus far.

III. HIERARCHICAL MODELING

As movement primitives are being extracted from the demonstration, we also seek to learn the higher order structure of the motion primitives, and how motion primitives may be sequenced. We model the higher level structure in the form of a second, higher-level hidden Markov Model. At the lowest level, each HMM represents a single motion primitive, i.e., there is a separate model of each movement pattern. In the higher order HMM, a single model represents the grouping of movement patterns, each motion primitive is represented by a single hidden state in the model. The higher order HMM consists of the set (A^h, B^h, π^h) , where A^h is the state transition model, B^h is the observation model, and π^h is the initial probability distribution. The state transition model $A^h[N^h, N^h]$ represents the sequencing relationship between the primitives. We use an ergodic model [27], where any node may transition to any other node, and initial probability distribution $\pi^h[N^h]$ gives the likelihood that a movement sequence will begin with a given motion primitive. For each state, the output observation model $B^h[N^h]$ is defined as the likelihood that an observed data sequence segment could have been generated by the model of the motion primitive.

$$b_i^h(O^s) = P(O^s | \lambda_i), \quad (3)$$

where b_i^h is the output observation model for state i , λ_i is the lower level HMM or FHMM model of the motion primitive corresponding to state i , and O^s is an observation sequence segment,

$$O^s = [\mathbf{O}_{t=1} \ \mathbf{O}_{t=2} \ \dots \ \mathbf{O}_{t=T^s}] \quad (4)$$

The probability $P(O^s | \lambda_i)$ that the observation sequence segment O^s could have been generated by the motion primitive model λ_i is computed via the forward procedure [27].

Given a set of motion primitives $[\lambda_1 \ \lambda_2 \ \dots \ \lambda_{N^h}]$ and a segmented time series data of human movement $[O_1^s \ O_2^s \ \dots \ O_T^s]$,

the state transition model and initial probability distribution can be learned via the Baum-Welch algorithm [27].

$$N_{ij} = \sum_{t=1}^{T-1} \alpha_t^h(i) a_{ij}^h b_j^h(O_{t+1}^s) \beta_{t+1}^h(j) \quad (5)$$

$$D_i = \sum_{t=1}^{T-1} \sum_{j=1}^{N^h} \alpha_t^h(i) a_{ij}^h b_j^h(O_{t+1}^s) \beta_{t+1}^h(j) \quad (6)$$

$$\tilde{a}_{ij}^h = \frac{N_{ij}}{D_i} \quad (7)$$

$$\tilde{\pi}_i^h = \sum_{j=1}^{N^h} \alpha_t^h(i) a_{ij}^h b_j^h(O_{t+1}^s) \beta_{t+1}^h(j) \quad (8)$$

D_i is the expected number of transitions from state i , N_{ij} is the expected number of transitions from state i to state j , a_{ij}^h is the state transition probability from state i to state j , π_i^h is the probability that a sequence of movement primitives will begin with state i , and α^h and β^h are the forward and backwards variables [27],

$$\alpha_t^h(i) = P(O_1^s O_2^s \dots O_t^s, q_t = S_i | \lambda^h) \quad (9)$$

$$\beta_t^h(i) = P(O_{t+1}^s O_{t+2}^s \dots O_T^s | q_t = S_i, \lambda^h) \quad (10)$$

However, this learning approach requires that the number of motion primitives be specified a-priori and that the training takes place off line. To allow incremental learning during observation, the system must be able to handle the addition of new hidden states each time a new motion primitive is learned, and have the ability to learn the transition rules incrementally, from partial observation sequences. To enable this functionality, we implement an incremental learning rule as follows:

$$\tilde{a}_{ij}^h = \frac{\bar{a}_{ij}^h + (N_{ts} - 1) a_{ij}^h}{N_{ts}} \quad (11)$$

$$\tilde{\pi}_i^h = \frac{\bar{\pi}_i^h + (N_{ts} - 1) \pi_i^h}{N_{ts}} \quad (12)$$

where N_{ts} is the number of training sequences applied to date, \bar{x} is the parameter obtained via the Baum-Welch algorithm and \tilde{x} is the parameter computed using the incremental learning rule. The incremental learning rule is applied each time a partial observation sequence is available. Following an incremental update, each row of the state transition matrix and the initial probability distribution are normalized so that the sum over all states is equal to one.

A partial observation sequence, used to incrementally train the model, is generated each time a sequence of two or more known motion primitives are recognized during observation. A partial observation sequence consists of all consecutive instances of known motion primitives, starting with the first recognized pair, and terminating when the first unknown motion primitive is detected. A sequence is also terminated

if an invalid motion is detected (i.e., a period of time during which no significant movement takes place).

However, when incremental learning is used with partial observation sequences, it is likely that some sequences will not contain exemplars of all the states. Applying the incremental learning rule to all states will result in spurious updates not supported by the actual training data. To avoid this problem, states are updated only when transitions from a state are observed in the incremental training sequence, as measured by D_{ij} (Equation 6). A separate counter N_{ts}^i is maintained for each state, so that incremental training updates are only made when the exemplar sequence contains the corresponding state.

When a new motion primitive is learned, a new hidden state is added to the higher order model. The state transition model and the initial probability model are modified to include a small probability of transitions to the new state and a small probability of starting from a new state. The state transition values from previous states to the new state are initialized to random and normalized.

Our approach is most similar to the growing HMM model proposed by Vasquez et al. [22]. Vasquez et al. develop an algorithm for learning trajectories in Cartesian space, so they use a topological map and the Instantaneous Topological Map (ITM) algorithm to initially learn the trajectories. In this algorithm, for each observed location, nearest nodes and edges are updated, including centroid adaptation for the closest node, as well as edge and node addition/deletion. Following the update of the topological map, the GHMM structure is updated by adding a state in the GHMM corresponding to each new node in the topological map, the state connectivity equal to the connectivity in the topological map, and initializing the observation model to a Gaussian centered on the centroid location computed by the topological map with a fixed size Gaussian. The initial state distribution and the state transition model are then learned incrementally. Unlike the Vasquez et al. approach, the trajectories being learned in this paper consist of high DoF full body trajectories. We use the lower level learning HMMs, as detailed in Section II, to form new nodes and the observation model for each node. Once a new node is added to the higher order HMM, similar to Vasquez et al., learning of the state transition model is carried out incrementally. However, unlike Vasquez et al., the proposed approach is also able to make use of very short training sequences, by only updating the affected portion of the state transition model.

Unlike previously proposed deterministic models of primitive sequencing [17], the approach proposed herein is more robust to recognition errors, especially when similar motion primitives are present. Similar to a deterministic model, the stochastic model can also be used to generate sequences of movement primitives for a humanoid robot, to detect and monitor human activity, and to predict future movement of the observed human, based on the sequence of primitives executed thus far.

IV. EXPERIMENTS

A human motion data set was used to test the proposed algorithms on a continuous sequence of a variety of whole body motions. A total of 34 reflective markers are attached to the demonstrator's body, located on key body parts such as the hands, feet, elbows and knees, and the marker $[x,y,z]$ position is captured and computed by the motion capture system online. The demonstrator performs arm raises, bow and squat motions.

The Motion Analysis motion capture system captures the Cartesian position of markers located on the body with a sampling rate of 5ms. An online inverse kinematics algorithm [28] is used to perform inverse kinematics computations to convert the data to joint angle positions in real time. A simplified inverse kinematics model of the demonstrator is used, consisting of 34 DoFs for the joint movement and the 6 DoF base body joint.

The video of the demonstrator motions was also analyzed manually. In the manual analysis, segment points between motion primitives were determined and each motion primitive labeled.

The continuous time series data consisting of the demonstrator base body and joint angles was used to incrementally learn the motion primitives and the higher order model, using the approach proposed in Sections II and III. A data flow diagram showing an overview of the entire process is shown in Figure 1. In addition, the entire sequence of segmented motion primitives was used to train the higher order model off line for comparison purposes.

The sequence begins with the demonstrator repeatedly performing the arms raise (AR) - arms lower (AL) motion 8 times each. The data is incrementally passed to the algorithm, where autonomous segmentation is performed, and segmented movements are clustered to learn the individual movement primitives. The two motion primitives AR and AL are extracted after 8 repetitions. At this point the primitives become available for recognition, but the demonstrator immediately switches to the bow down/ bow raise motion, before a sequence of known motion primitives can be recognized and observed. After 7 repetitions of the bow motions, two more motion primitives are extracted, the bow lower (BL) and the bow raise (BR) primitive, bringing the system up to four known motion primitives. The system then observes and recognizes a sequence consisting of 6 known motion primitives, the sequence bow down - bow raise repeated 3 times. This sequence is presented to the higher level model as the first incremental training sequence. Table I shows the state transition table following the first incremental training sequence. Since the training sequence only contains exemplars of the last two motion primitives, the expected number of transitions from the first two states (as computed by equation 6) approaches zero, so these two states are not updated. As no previous sequences with these two states have been used for training, the section of the state transition table for these two states is not yet trained. This is indicated by "NT" in Table I.

TABLE I
STATE TRANSITION TABLE FOLLOWING THE FIRST INCREMENTAL
TRAINING SEQUENCE

State	State			
	AR	AL	BL	BR
AR	NT	NT	NT	NT
AL	NT	NT	NT	NT
BL	0.001	0.001	0.000	0.998
BR	0.001	0.001	0.998	0.000

TABLE II
STATE TRANSITION TABLE FOLLOWING THE SECOND INCREMENTAL
TRAINING SEQUENCE

State	State			
	AR	AL	BL	BR
AR	0.000	0.998	0.001	0.001
AL	0.998	0.000	0.001	0.001
BL	0.001	0.001	0.000	0.998
BR	0.001	0.001	0.998	0.000

After a brief pause in movement, the demonstrator next returns to perform more arm raise motions. Since these movements have been previously learned, they are recognized by the system as known motion primitives. A sequence of 4 repetitions of these two motions is recognized, generating the next incremental training sequence. This sequence is presented to the higher level model as the second incremental training sequence. Table II shows the state transition table following the second incremental training sequence. In the second training sequence, the first two nodes are activated, allowing the system to train the corresponding section of the state transition table.

The demonstrator next begins performing a new movement, consisting of the squat lower and squat raise primitives. Following 6 and 8 repetitions respectively, the squat lower (SL) and the squat raise (SR) motion primitives are recognized. The demonstrator then continues performing the squat movements, resulting in a sequence of 13 recognized motion primitives consisting of SL and SR repetitions, which is presented to the higher level model as the third incremental training sequence. Table III shows the state transition table following the third incremental training sequence. Following the addition of two new motion primitives, the state transition table is expanded to add the new nodes. In the third training sequence, the two new nodes are activated, allowing the system to train the corresponding section of the state transition table.

For comparison, the entire data sequence is used to train the higher order model using off-line training via the standard Baum-Welch algorithm. Table IV shows the state transition model obtained with the off-line training method. The incrementally trained model was trained on a very small subset of the full movement sequence (only sequences of recognized motion primitives), while the off-line trained model was trained on the entire sequence, including both recognized and

TABLE III
STATE TRANSITION TABLE FOLLOWING THE THIRD INCREMENTAL
TRAINING SEQUENCE

State	State					
	AR	AL	BL	BR	SL	SR
AR	0.000	0.978	0.001	0.001	0.010	0.010
AL	0.978	0.000	0.001	0.001	0.010	0.010
BL	0.001	0.001	0.000	0.978	0.010	0.010
BR	0.001	0.001	0.978	0.000	0.010	0.010
SL	0.001	0.001	0.001	0.001	0.000	0.996
SR	0.001	0.001	0.001	0.001	0.996	0.000

TABLE IV
STATE TRANSITION TABLE GENERATED BY OFF-LINE TRAINING

State	State					
	AR	AL	BL	BR	SL	SR
AR	0.000	0.935	0.001	0.001	0.062	0.001
AL	0.935	0.000	0.001	0.001	0.062	0.001
BL	0.100	0.100	0.000	0.798	0.001	0.001
BR	0.001	0.001	0.996	0.000	0.001	0.001
SL	0.001	0.001	0.117	0.001	0.000	0.880
SR	0.001	0.001	0.001	0.001	0.996	0.000

unrecognized motion, as well as motion segments containing no motion (demonstrator standing around, etc.). Some increased noise is observed in the off-line trained model, due to the presence of invalid and unrecognized motions in the training set. The results show that the incremental training and the off-line training achieve nearly identical results, validating the incremental approach.

The incremental training approach was also analyzed for timing performance. Without any code optimization, a training cycle required on average 1.5 seconds to execute, making the approach promising for on-line processing. In general, when using full model training, the training time will be a function of the number of states in the model (i.e., the number of motion primitives).

V. DISCUSSION

The proposed approach enables incremental learning of the motion primitive sequencing, while also taking into account a probabilistic model of the lower level motion primitives. Compared to deterministic methods [17], the proposed approach generates transition probabilities based both on the observed sequence of primitives, and the certainty that the primitives have been correctly recognized, whereas the deterministic method considers only the observed sequence of primitives and assumes perfect recognition. Therefore, it is expected that the probabilistic approach will be less sensitive to errors in recognition, particularly when similar motions are being executed or when there is a lot of noise in the motion measurement data, as would be typical with more complex movements or when there are similarities between motion primitives. This would be particularly helpful when generating movement on a humanoid robot, as incorrectly learned sequencing rules could result in invalid motions

which cannot be performed on the robot.

While the current approach allows for incremental training, the training procedure still considers all the states, thus making the computational costs of the training procedure a function of the number of states. The proposed training procedure does allow for only partial updates to the full state transition table, based on the states which are visited during the partial training sequences being considered, but the states to be updated are identified during the training procedure itself, by initially considering the entire state space. One potential approach for reducing the computation time would be to first identify the states likely to be affected, and then to perform the training only on the identified state subspace.

A drawback of the incremental training procedure compared to off-line training is that, since data is being considered sequentially, only a subset of all the possible training data is being used for training the higher order model. In the proposed approach, low level nodes are first learned through on-line observation, high level training data is only generated after the low level nodes have been learned. It typically requires 6 - 8 observations of a primitive before the primitive is abstracted. Therefore, it is possible that the incremental procedure learns only a subset of the possible motion primitive sequencing transitions, especially if transitions shown in the early part of the demonstration are not subsequently repeated. This problem could be partially addressed by initializing the system with common motion primitives, so that motion primitives can be recognized immediately and training data generated.

In the experiments conducted thus far, only a single demonstrator has been used for motion teaching. Since the demonstrator data is first converted to joint angle data via a kinematic model, there is a convenient way to normalize data from multiple demonstrators into a coherent data set, simply by parameterizing the kinematic model according to each demonstrator, such as the link lengths and height. However, if object movement or relative positions between an object and the demonstrator are also to be included, this normalization approach may not be suitable, and alternate techniques may need to be considered. In addition, for movements incorporating object handling, it may be desirable to generalize movement primitives by including parametrization. This can be incorporated into the proposed framework through the use of parametric hidden Markov models [29]. When considering multiple users, there is no guarantee that different demonstrators would perform the same motion primitive in the same way, it is still possible that demonstrations from different demonstrators would result in different motion primitives due to individual variability. This question will be addressed in future work.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed a novel approach for learning the sequencing rules of motion primitives during on-line continuous demonstration via stochastic modeling. Motion primitives are modeled as Hidden Markov models and learned incrementally during on-line observation. The higher level

order between motion primitives is also modeled as a Hidden Markov model, where each hidden state in the higher order model corresponds to a motion primitive. An approach for incremental training of the higher order model was proposed and verified on a human motion database.

In future work, we will investigate the behavior of motion primitive learning in the presence of multiple demonstrators. We will also investigate the use of the higher order model for improving the segmentation results and for predicting human motion. We will also apply and validate the proposed approach with other types of human motion, including task and goal based motion such as interaction with the environment.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the assistance of Mr. Hirotaka Imagawa with the collection of the motion capture data.

REFERENCES

- [1] S. Schaal, "Dynamic movement primitives - a framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*, H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witte, Eds. Springer Tokyo, 2006, pp. 261–280.
- [2] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.
- [3] —, "On-line segmentation and clustering from continuous observation of whole body motions," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.
- [4] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in Cognitive Sciences*, vol. 6, no. 11, pp. 481–487, 2002.
- [5] V. Krueger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: A review on action recognition and mapping," *Advanced Robotics*, vol. 21, no. 13, pp. 1473–1501, 2007.
- [6] W. Ertlhagen, A. Mukovskiy, E. Bicho, G. Panin, C. Kiss, A. Knoll, H. van Schie, and H. Bekkering, "Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning," *Robotics and Autonomous Systems*, vol. 54, pp. 353–360, 2006.
- [7] M. Ito and J. Tani, "On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system," *Adaptive Behavior*, vol. 12, no. 2, pp. 93–115, 2004.
- [8] 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.
- [9] 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.
- [10] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, pp. 79–91, 2004.
- [11] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002, pp. 1398–1403.
- [12] M. Okada, K. Tatani, and Y. Nakamura, "Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002, pp. 1410–1415.
- [13] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg, "Learning from and about others: towards using imitation to bootstrap the social understanding of others by robots," *Artificial Life*, vol. 11, no. 1–2, pp. 1–32, 2005.
- [14] M. N. Nicolescu and M. J. Matarić, "Learning and interacting in human-robot domains," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 5, pp. 419–430, 2001.
- [15] P. F. Dominey, G. Metta, F. Nori, and L. Natale, "Anticipation and initiative in human-humanoid interaction," in *Proceedings of the IEEE International Conference on Humanoid Robots*, 2008, pp. 693–699.
- [16] 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.
- [17] D. Kulić, D. Lee, Ch. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," in *Proceedings of the IEEE International Conference on Humanoid Robots*, 2008, pp. 326–332.
- [18] G. W. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Proceedings of the Conference on Neural Information Processing Systems*, 2006, pp. 1345–1352.
- [19] O. C. Jenkins and M. Matarić, "Performance-derived behavior vocabularies: Data-driven acquisition of skills from motion," *International Journal of Humanoid Robotics*, vol. 1, no. 2, pp. 237–288, 2004.
- [20] —, "A spatio-temporal extension to isomap nonlinear dimension reduction," in *Proceedings of the International Conference on Machine Learning*, 2004, pp. 441–448.
- [21] Ch. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *Proceedings of the IEEE International Conference on Humanoid Robots*, 2008, pp. 399–405.
- [22] D. Vasquez, T. Fraichard, and C. Laugier, "Growing hidden markov models: An incremental tool for learning and predicting human and vehicle motion," *The International Journal of Robotics Research*, vol. 28, no. 11–12, pp. 1486–1506, 2009.
- [23] J. Jockusch and H. Ritter, "An instantaneous topological map for correlated stimuli," in *Proceedings of the International Joint Conference on Neural Networks*, 1999, p. 529534.
- [24] J. Kohlmorgen and S. Lemm, "A dynamic hmm for on-line segmentation of sequential data," in *NIPS 2001: Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, 2002, pp. 793–800.
- [25] B. Janus and Y. Nakamura, "Unsupervised probabilistic segmentation of motion data for mimesis modeling," in *Proceedings of the IEEE International Conference on Advanced Robotics*, 2005, pp. 411–417.
- [26] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [27] 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.
- [28] K. Yamane and Y. Nakamura, "Natural motion animation through constraining and deconstraining at will," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 3, pp. 352–360, 2003.
- [29] D. Herzog, A. Ude, and V. Krueger, "Motion imitation and recognition using parametric hidden markov models," in *Proceedings of the IEEE International Conference on Humanoid Robots*, 2008, pp. 339–346.