Layered HMM for Motion Intention Recognition

Daniel Aarno and Danica Kragić*
Centre for Autonomous Systems - Computational Vision and Active Perception Laboratory
School of Computer Science and Communication
Royal Institute of Technology (KTH), SWEDEN
Email: {bishop, danik}@csc.kth.se

***Abstract*— Acquiring, representing and modeling human skills is one of the key research areas in teleoperation, programming-by-demonstration and human-machine collaborative settings. One of the common approaches is to divide the task that the operator is executing into several subtasks in order to provide manageable modeling.**

In this paper we consider the use of a Layered Hidden Markov Model (LHMM) to model human skills. We evaluate a gestem classifier that classifies motions into basic action-primitives, or gestems. The gestem classifiers are then used in a LHMM to model a simulated teleoperated task. We investigate the online and offline classification performance with respect to noise, number of gestems, type of HMM and the available number of training sequences. We also apply the LHMM to data recorded during the execution of a trajectory-tracking task in 2D and 3D with a robotic manipulator in order to give qualitative as well as quantitative results for the proposed approach. The results indicate that the LHMM is suitable for modeling teleoperative trajectory-tracking tasks and that the difference in classification performance between one and multi dimensional HMMs for gestem classification is small. It can also be seen that the LHMM is robust w.r.t misclassifications in the underlying gestem classifiers.

I. Introduction

In today's manufacturing industry, large portions of the operation has been automated. However, many processes are too difficult to automate and must rely on humans' supervisory control and decision making; in areas such as the identification of defective parts and process variations [1]. Similar issues also arise in medical applications [2]. When such skills are required, humans still have to perform straining tasks. Therefore, Human-Machine Collaborative Systems (HMCS) has been used to prevent ergonomic injuries and operator wear, by allowing cooperation between a human and a robotic system in a flexible way.

Learning human skills, using them in HMCS settings or transferring them to robots directly has been a core objective for more than three decades in the area of artificial intelligence, robotics and intelligent control. Application areas range from teleoperation to programming-by-demonstration (PbD), human-machine collaborative settings, automated visual surveillance and multi-modal human-computer interaction [3]–[12]. It has been widely recognized that the underlying system used for learning, representing, modeling and transferring skills have to deal with highly nonlinear relationships between the stimuli and responses (sensor/actuator systems). Such a system is strongly dependent on the varying state of the environment, since skills are stochastic properties that can only be measured in the statistical sense.

Learning human skills has been viewed as the problem of extracting specific skill characteristics given training data. In robotics community, Hidden Markov Models (HMMs) have been a popular method used for interpreting a human operator's intention during execution of a teleoperated or human-machine collaborative task, [3]–[8]. HMMs have also been frequently and successfully used for speech recognition, [13], handwritten character recognition, [14] and gesture recognition for interpreting sign language, [11].

The question studied in this paper is, given that the intention of an operator of a teleoperated system can be recognized online in real-time, whether it is possible to improve the task execution by allowing the system to adapt to the operator's need by applying the correct control mode in the transfer step and the similar manner as proposed in [4], [7], [8]. To be able to give the correct aid to the operator it is necessary for the HMCS to be able to successfully interpret the operator's intent both online and in real-time. The aim of this paper is therefore to investigate which parameters determine the success of the HMM approach to motion intention recognition as well as present the Layered Hidden Markov Model (LHMM) approach we are currently investigating.

This paper is organized as follows: section II describes related work in the area of task modeling and intention recognition and introduces the concept of the layered hidden Markov model. Section III describes simulations performed in order to evaluate the proposed LHMM method. In section IV the results of the simulations are verified on a real system. Finally the paper is summarized and conclusions are presented in section V.

II. Theoretical Background and Related Work

Hidden Markov models have been used to interpret human intention in a number of different areas. Li *et al.* used virtual fixtures for tracking a sine curve in two dimensions, [4]. A HMM approach was used to estimate whether the user was i) doing nothing, ii) following the curve, or iii) not following the curve. Based on this estimate, the virtual fixture was automatically switched on or off, enabling the user to avoid local obstacles.

Yu *et al.* used HMMs to classify an operator's motion intention to three classes, path following, target alignment and

* This work has been supported by the Swedish Research Council.

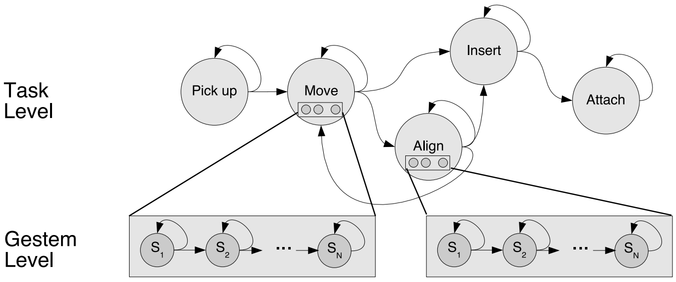

Fig. 1. Layered hidden Markov models

obstacle avoidance, [8]. Each class is associated with a virtual fixture that assists the operator.

In our previous work a combination of K-means clustering, SVMs and HMMs was used to automatically extract a set of virtual fixtures given sensor traces of an operator performing a task, segment the task into a number of subtasks, corresponding to a particular fixture and provide online assistance by applying the correct fixture during subsequent task executions, [7]. The output of the HMM was used to adjust the compliance of the virtual fixture so that the fixture was harder when the system was more certain about the current state. This allowed the system to handle task-deviations (i.e. none of the subtasks were executed) by lowering the stiffness of the fixture. However, the subtasks used in [7] was limited to straight lines.

In [3], Hundtofte *et al.* used HMMs at the *gestem* level as opposed to the task level. This means that basic interaction primitives are modeled by a HMM and the task is represented as a network of such HMMs. In our current work we combine gestem classification with task-level modeling by the suggested LHMM approach so to handle more complicated types of tasks. This can be viewed as an extension of the work presented in [7] where the SVM classifiers are replaced by the more expressive HMM classifiers. The work presented above has been concerned with modeling either the motion primitives or the teleoperated/HMCS task. In our current work we are integrating the gestem classification with the higher level task modeling.

Hierarchical Hidden Markov Models (HHMMs) and layered hidden Markov models have been used to model various phenomena that exhibit stochastic structures at several different levels in areas such as speech and text recognition, modeling of group actions in meetings and extracting context from video, [12], [15]–[18]. Zhang *et al.* used a two-layer HMM to model individual and group actions during meetings in [16]. An I-HMM was used to model individual actions. The recognized individual actions was then passed along to the G-HMM that was used to classify group actions. In [12], a LHMM was used to recognize different types of activity in an office environment. In [17] Xie *et al.* used a HHMM to automatically segment a soccer game into two classes, *pause* and *play* in an unsupervised setting.

In this work, we are considering a LHMM approach to user intention recognition in teleoperated, HMCS or PbD settings where the sensory information is sampled motion data from any pose measuring sensor. The LHMM is preferable over the HHMM since in the HHMM the states are contain another HMM and thus represents a time sequence of the raw signals. On the other hand, in the LHMM there are several HMMs running at parallel at any given level of the hierarchy, where each HMM corresponds to a different "concept".

Hidden Markov models, [13] can be used on two levels for modeling human actions. A HMM can be used to recognize the operator's motion primitives, or *gestems* as in [3] or to model the mental stages of the operator performing a teleoperation task as in [19]. A gestem-level HMM is used to recognize a primitive motion sequence and a task-level HMM is used to recognize a complete task. A layered hidden Markov model, [12] consists of N levels of HMMs where the HMMs on level $N+1$ corresponds to observation symbols or probability generators at level N, see Fig. 1.

It should be noted here that a LHMM could in practice be transformed into a single layered HMM where all the different models are concatenated together. Some of the advantages that may be expected from using the LHMM over a large single layer HMM is that the LHMM is less likely to suffer from over-fitting since the individual subcomponents are trained independently on smaller amounts of data. A consequence of this is that a significantly smaller amount of training data is required for the LHMM to achieve a performance comparable of the HMM. Another advantage is that the layers at the bottom of the LHMM, which are more sensitive to changes in the environment such as the type of sensors, sampling rate etc, can be retrained separately without altering the higher layers of the LHMM.

III. Experimental Evaluation with Synthetic Data

To be able to better analyze and reproduce the results we first carry out experiments on synthetic data. A reference task consists of a sequence of motion primitives randomly generated from two groups of motion primitives. The first group contains straight lines of varying directions and lengths and the second group is made up of circle segments with varying starting and ending angles as well as orientation and radius. Fig. 2 shows typical simulated operator trajectories. These trajectory types may seem simple, but they were choosen because we believe that there exists several relevant tasks in areas such as medical surgery or automotive assembly that can be decomposed into a sequence of linear and circular motions, [5], [7] In section IV we perform similar experiments with a robotic manipulator to verify the simulated results. There, the data is taken from a trajectory-tracking task where the end-effector of the manipulator is force controlled by a human operator.

The simulated operator trajectories are created in the following way. Given a reference trajectory T_r, a target point $\mathbf{p}$ is selected on T_r so that the distance to $\mathbf{p}$ from the current position $\mathbf{q}$ is larger than some threshold ξ. A direction of motion $\mathbf{d}$ is then computed as the average between the direction towards $\mathbf{p}$ from $\mathbf{q}$ and the current direction of motion.

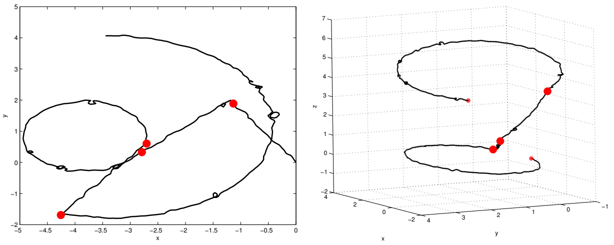

Fig. 2. Typical simulated operator trajectories in 2D (left) and 3D (right). The red dots marks the change from one primitive to the next.

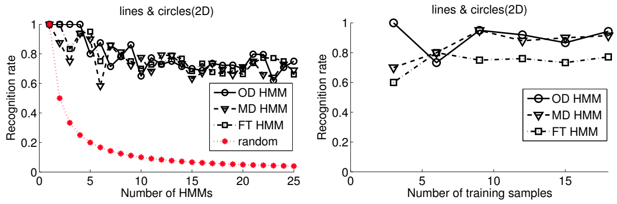

Fig. 3. **Left**: typical deterioration of classifier performance as function of the number of HMMs. **Right**: The effect of the number of training samples on the HMM classification.

A random error $\mathbf{e}_d$ is then added to $\mathbf{d}$ where each element of $\mathbf{e}_d$ is generated independently according to (1), where Γ is generated from a normal distribution $(\mu = 0, \sigma = 1)$ and κ determines the noise level. Finally the current position $\mathbf{q}$ is updated by taking a step of size $\delta\cdot(1+2\kappa\cdot\Gamma)$ in the direction of $\mathbf{d}$ where δ determines the step-size, which was set to 0.05 in all experiments.

$$\mathbf{e}_d(i) = \kappa\cdot\Gamma \tag{1}$$

The value of κ was set to 0.2 for all experiments if not otherwise stated.

In the experimental evaluation three different types of HMMs are considered. The reason for evaluating different types of HMMs is that previous work has proposed the use of different kinds of HMMs and we are interested in investigating if there is an apparent advantage to any of them.

One dimensional HMM: The simplest HMM is the one dimensional (OD) HMM. The observation symbols are then taken from a finite set $O = \{o_1, o_2, ..., o_K\}$ of K discrete symbols. The B matrix is then used to lookup the probability of observing the k:th symbol in state i, that is $b_j(k) = P(o_k|i = j)$.

In this work the symbols are generated by k-means clustering of all the training directions. The number of cluster centers is 25 in all experiments. This number was chosen by an offline examination of the data. We note here that the number of cluster centers is not crucial for the performanec, but using too few clusters will make it hard to distinguish between different motion directions while using too many will make generalization difficult.

Multi dimensional HMM: The multi dimensional (MD) HMM assumes independence between the different dimensions of the input data. Thus there will be a B matrix for each dimension of the input data. This means that for a D dimensional HMM the observation symbols are also D dimensional where each dimension d contains values from a finite enumerated set, [19].

In this work, each dimension is split into 10 equally sized bins and the input directions are projected into these bins generating the observation symbols. As with the number of cluster centers the exact number of bins is not important but it has to be selected to facilitate discrimination and generalization.

Multi dimensional HMM with FT: The third type of HMM considered in this paper is similar to the MD HMM except that instead of mapping the raw motion directions to symbols, each dimension of the raw input directions are pre-processed by applying the Fourier transform to small overlapping windows, similar to that reported in [8]. In this work a Hamming window of size 6 was used with 50% overlap.

A. Experimental Evaluation

For the LHMM to be successful there must be a robust underlying gestem classifier. Furthermore the LHMM and gestem classifiers must be able to produce good results online with only partial observation sequences. The experimental evaluation in this work consists of evaluating the HMM gestem classifier for the three HMM types described in the begining of this section with respect to the number of gestems, the influence of the number of training samples, the effect of noise on classification performance and the online behavior.

1) The Gestem Classifier: The first experiment evaluated the offline performance of the HMM gestem classifiers with respect to the number of different gestems. The input data was generated as described in Section III, thus some gestems can be very similar. If the gestems are not generated at random but chosen from some set of gestems that are constructed to be easy to distinguish between (such as the letters of the alphabet) the performance could be expected to be better than that reported here. As it can be seen in Fig. 3 (left), the recognition performance drops almost linearly from 100% to about 70% for 25 gestems. The type of HMM or gestem type (circles, lines or mixture) appears to have no statistical significance on the recognition performance. However, for three dimensional data the classification performance is a bit better but that can be explained with the fact that the individual gestems are less likely to be similar.

It is well known that HMMs can be successfully trained with only a small amount of training data. Especially if there are few outliers such as in our training data where the motion is perfect except for the introduced white noise. It can be seen in Fig. 3 (right)that the recognition rate is quite high even for only two training runs. This is a good feature of the HMM gestem classifier since in many settings extensive training is not possible. When the type of noise changes and outliers are introduced the necessary number of training sequences will increase in order to be able to capture the larger variations that occurs. However, preliminary results indicate that in practice the necessary number of training sequences is actually quite

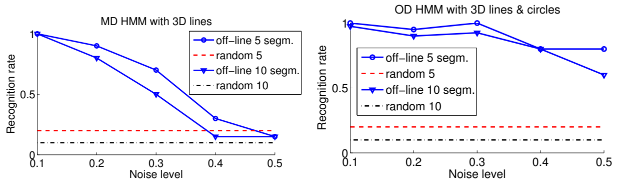

Fig. 4. **Left:** Typical effect of noise on HMM classification. **Right:** Effect of noise on HMM classification with distinct gestems.

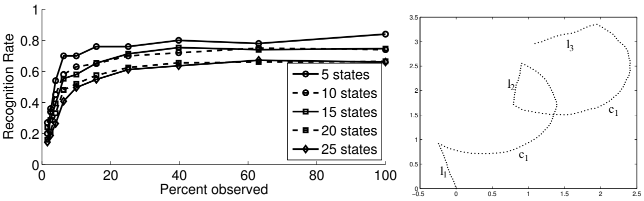

Fig. 5. **Left:** Online recognition performance of MD HMM with 3D data consisting of lines and circle segments. **Right:** Example trajectory of a task with 5 states and 4 gestems.

low as long as the training sequences are representative for what will occur during execution.

The HMM is able to handle a large amount of noise as long as the noise is consistent during training and classification. To evaluate what amount of noise the gestem classifiers can handle, we tested the classification performance with several synthetic runs generated by varying the value of κ in (1) from 0.1 to 0.5. As can be seen in Fig. 4 (left) an acceptable value of κ is somewhere between 0.2 and 0.3. The noise sensitivity is highly affected by the similarity of the gestems. If the gestems are similar, the performance decreases almost linearly with increased noise. If the gestems contains few common symbols, the classification performance remains relatively unaffected until the noise starts to dominate (i.e is large compared to the nominal motion). An example of this can be seen in Fig. 4 (right).

So far, all the experiments have been conducted offline where the whole gestem was available. In order to work in the intended setting, the LHMM and gestem classifiers must be made to work online with only partially observed gestems. The next experiment evaluated the gestem classifiers online performance. Fig. 5 (left) shows that the gestem classifiers can produce good results after observing only a small fraction (10%-20%) of the gestem. The results here will depend strongly on the similarity between the first parts of the gestems and the success will thus vary depending on the type of task. Another important aspect for online classification is the exact time at which the HMM recursion starts. In this case the exact times where known due to the fact that the test data was syntheticly generated. If the change time for switching between gestems are off there is a risk to observe very unlikely observation symbols and thus the correct HMM can be severely penalized in the beginning of the classification and in worst case never recover. There are ways around this problem, for example using the continuous HMM presented in [4]. In this work, we have used an alternative approach based on a CUSUM test [**?**] of the change in likelihood of the most probable model.

2) The LHMM: Fig. 5 (right) shows a 2D trajectory that contains four gestems, $G = \{l_1, l_2, l_3, c_1\}$. The "mental model" of this task is that the gestems should be performed in a sequential-left-to-right (SLR) fashion with the c_1 gestem appearing twice so the task should go through the five different states $S_1, \ldots, S_5$ and thus execute the gestems in the following order: l_1, c_1, l_2, c_1, l_3. The gestem is exactly the same in S_2 and S_4 so one cannot differentiate between these states by simply monitoring the output from the four gestem classifiers.

A task level HMM is now trained on the output of the gestem classifiers. That is, the trajectory is classified by the gestem classifiers (online) and the sequence of winning gestems are used as input to the task-level HMM which is trained in order to extract the task-model. Fig. 6 (left, bottom plot) shows a typical classification sequence obtained by the gestem classifiers. The dashed lines indicate the switch from one state to the next. Note that there are only four gestems recoginized in the bottom plot whereas there is five states in the top and middle plots since the gestem c_1 is associated with two states.

It can be seen from Fig. 6 (left) that even though the gestem classifiers are sometimes confusing gestem l_1 and c_1 the task-level HMM is still capable of determining the correct state. This is because the miss-classifications of the gestem classifiers are consistent with training data and thus the task-level HMM expects some miss-classifications. Furthermore the discriminant power of the LHMM is much better than that of the HMM, i.e. the difference between the most probable and the second most probable state is in general much larger for the LHMM.

IV. Experimental Evaluation with a Robot System

In order to verify the validity of the proposed approach and to show that the quantitative results obtained with the synthetic data are relevant, we have performed a number of qualitative experiments with a robot manipulator. The robot used is ActivMedia PowerBot and the manipulator used is a made from a number of PowerCube elements and passive links and it is mounted on a mobile base.

The manipulator has a JR3 force-torque sensor mounted between the end-effector and the last link, providing 6 DOF force-torque measurements at the end-effector. The force-torque sensor provides decoupled data at 8 kHz per channel, which is low-pass filtered with the bandwidth 30 Hz (-3 dB) by a DSP. The filtered force-torque data is used to control the position and orientation (pose) of the end-effector by driving the end-effector with a velocity proportional to the forces and torques. Due to the kinematics of the manipulator, large motions (of the joints) are sometimes required to realize small changes in orientation of the end-effector. This can make control of the manipulator more difficult than for a PUMA-like robot, that is, 6 rotary DOF with the 3 DOF of the wrist intersecting a single point. In all experiments the platform is

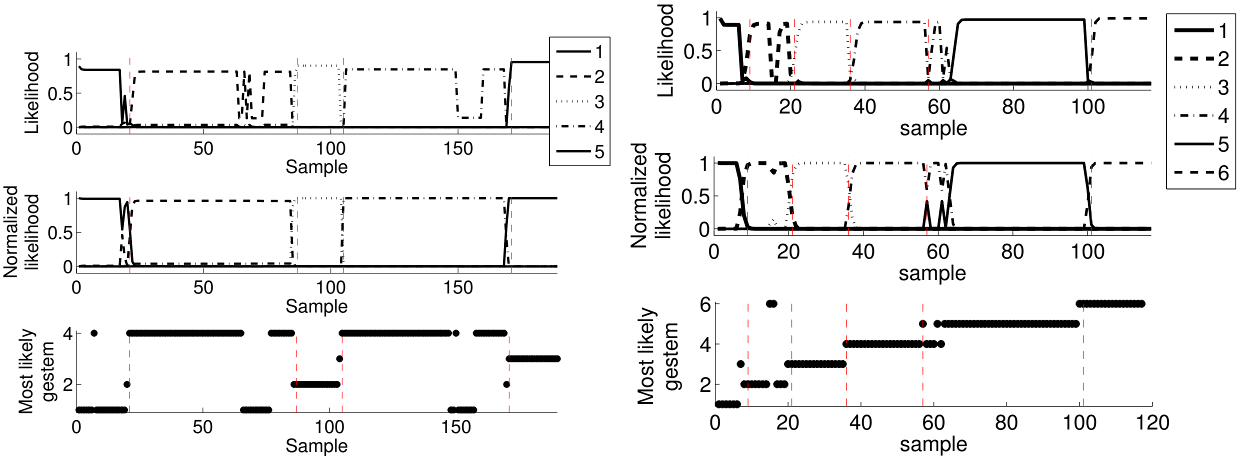

Fig. 6. The top plot shows the likelihood of each state and the plot in the middle shows the normalized likelihood. The bottom plot shows the (online) classification of the motion by the gestem classifiers and is the input to the task-level HMM. **Left:** Classification of the LHMM for the task shown in Fig. 5. **Right:** Classification of the LHMM for the 3D trajectory-tracking task executed on with the robot manipultaor as shown in Fig. 7 (right).

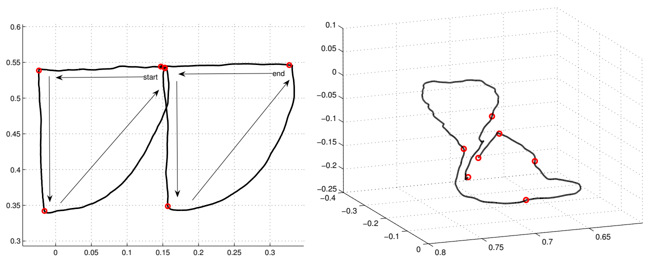

Fig. 7. Representative trajectories for the two trajectory-tracking tasks.

stationary and the operator guides the manipulator by applying forces to the end-effector.

Two trajectory-tracking tasks are used in the experiments. The first task consists of tracking a sequence of lines and circle segments on a planar 2D surface, very similar to the simulated trajectories used previously. The second task consists of tracking a trajectory on an object in 3D without touching the object. Two representative trajectories are shown in Fig. 7.

The trajectories are normalized so that samples are 1 cm apart, this is a reduction of data of about 90%. From the normalized data the sequence of motion directions is computed and k-means clustering is used to identify 10, for 2D data, or 25, for 3D data, cluster centers used as the symbols in a one-dimensional HMM. The sequence of motion directions is then transformed to a sequence of observation symbols. A total of five trajectories were recorded and three of them were used for training and two for testing. The reason for using only 5 trajectories is that one important aspect of the proposed system is to provide good results even with little training data, which should be possible given the previously presented results.

Fig. 6 (right) shows the results of the online classification of the gestems for the 3D trajectory in Fig. 7 (right).

As it can be seen the classification is good even though there were only three training trajectories available. One of the reasons for this is that it is the same person that performs the training and testing sequences. For operator independent training the number of required training samples is expected to be higher.

We tested the LHMM on the task shown in Fig. 7 (left) with the sequence of gestems $\{l_1,l_2,c_1,l_2,c_1,l_1\}$. Even though the accuracy of the underlying gestem classifiers are very high the use of the LHMM is still motivated by two facts. First, it can encode the sequence of the gestems and thus tell them apart even though the same gestem appears more than ones. Second the discriminate power is greater so we can have a more confident classification. We also tested the same LHMM on the sequence $\{l_1,l_2,c_1,l_1,c_1,l_2\}$ which is not seen during training. We see that the LHMM can still recognize the correct state sequence. However, there is a significant delay before the evidences (observations) are strong enough warrant a state change. This can be seen around sample 70 and 90 of Fig. 8 (right).

It can be seen from Fig. 6 that even though the gestem classifiers are sometimes detecting the wrong gestem the LHMM can still clearly recognize the correct state. It can also be seen from the unnormalized likelihood that the total probability drops rapidly when unexpected gestems are identified. This can be used to assign a measure of the certainty of the system and can be useful determining how much confidence to put into the classification during, for example, a fixturing of the motion, as was done in [7]. It should be noted that the LHMM was tested on manually segmented data and thus the classification is restarted at the “perfect” time which explains the zero delay for switching states and thus indicates the best possible results that can be achieved with the implemented system.

V. Summary and Conclusions

Task segmentation and modeling is one of the core research areas in the field of teleoperation, human-machine collaborative and programming-by-demonstration systems. In this

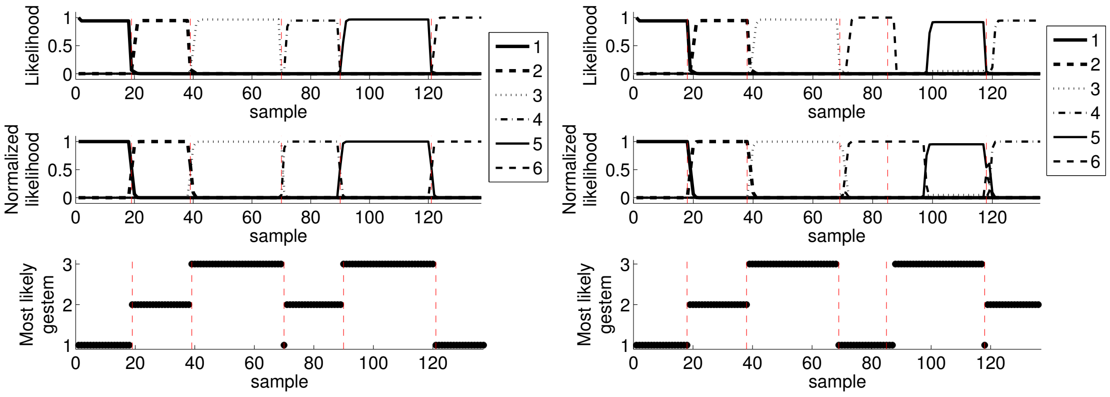

Fig. 8. Classification of the LHMM for the two dimensional trajectory-tracking task. The top plot shows the likelihood of each state and the plot in the middle shows the normalized likelihood. The bottom plot shows the (online) classification of the motion by the gestem classifiers and is the input to the task-level HMM. **Left:** Classification of the trained sequence $\{l_1, l_2, c_1, l_2, c_1, l_1\}$. **Right:** Classification of a sequence not seen during training, $\{l_1, l_2, c_1, l_1, c_1, l_2\}$.

paper, we have investigates which parameters determine the success of the HMM approach to motion intention recognition as well as presented a layered hidden Markov model approach for complex task modeling.

Experimental evaluation shows that LHMMs are suitable for modeling and real-time recognition of teleoperative, HMCS and PbD tasks. The evaluation has also shown that both one and multi dimensional HMMs are suitable for modeling gestems and they are even able to handle gestems that are quite similar in nature as long as the SNR is low. The HMMs are able to suppress relatively large amounts of noise as long as the noise is white. However, preliminary results indicate that the HMMs are more sensitive to other types of disturbances. Based on the the experimental evaluation, it is clear that the LHMM has a strong potential to model complex tasks since it is able to perform well even with miss-classifications in the underlying layers. This means that as long as the gestem classifiers produce consistent misclassification during training and testing the layered structure of the LHMM is able to handle this. The LHMM also has a much greater discriminating power than the standard HMM approach.

In our future work we will implement a system similar to that in [7] using a LHMM to solve a larger set of more complicated tasks. A Further extension will be to add more sensing modalities to the system, allowing more general intention modeling.

References

[1] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. Moore, R. B. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.

[2] R. H. Taylor and D. Stoianovici, "Medical Robotics in Computer-Integrated Surgery," in *IEEE Transactions on Robotics and Automation*, 2003.

[3] C. S. Hundtofte, G. D. Hager, and A. M. Okamura, "Building a Task Language for Segmentation and Recognition of User Input to Cooperative Manipulation Systems," in *10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2002.

[4] M. Li and A. Okamura, "Recognition of Operator Motions for Real-Time Assistance Using Virtual Fixtures," in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003.

[5] D. Kragić, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human-Machine Collaborative Systems for Microsurgical Applications," in *International Symposium on Robotics Research*, 2003.

[6] A. Castellani, D. Botturi, M. Bicego, and P. Fiorini, "Hybrid HMM/SVM Model for the Analysis and Segmentation of Teleoperation Tasks," in *IEEE Int. Conf. on Robotics and Automation*, 2004.

[7] D. Aarno, S. Ekvall, and D. Kragić, "Adaptive Virtual Fixtures for Machine Assisted Teleoperation Tasks," in *IEEE Int. Conf. on Robotics and Automation*, 2005.

[8] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation Assistance Based on Motion Intention Recognition," in *IEEE Int. Conf. on Robotics and Automation*, 2005.

[9] R. Zöllner, O. Rogalla, R. Dillmann, and M. Zöllner, "Understanding Users Intention: Programming Fine Manipulation Tasks by Demonstration," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.

[10] M. Kaiser and R. Dillmann, "Building Elementary Robot Skills from Human Demonstration," in *IEEE Int. Conf. on Robotics and Automation*, 1996.

[11] R.-H. Liang and M. Ouhyoung, "A Real-Time Continuous Gesture Recognition System for Sign Language," in *Int. Conf. on Automatic Face and Gesture Recognition*, 1998.

[12] N. Oliver, A. Garg, and E. Horvitz, "Layered Representations for Learning and Inferring Office Activity from Multiple Sensory Channels," *Comput. Vis. Image Underst.*, vol. 96, no. 2, pp. 163–180, 2004.

[13] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. of the IEEE, vol 77, no. 2*, 1989.

[14] A. Brakensiek, A. Kosmala, D. Willett, W. Wang, and G. Rigoll, "Performance Evaluation of a New Hybrid Modeling Technique for Handwriting Recognition Using Identical On-Line and Off-Line Data," in *Int. Conf. on Document Analysis and Recognition*, 1999.

[15] S. Fine, Y. Singer, and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications," *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, 1998.

[16] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, and G. Lathoud, "Modeling Individual and Group Actions in Meetings: a Two-Layer HMM Framework," in *2nd IEEE Workshop on Event Mining: Detection and Recognition of Events in Video, In Association with CVPR*, 2004. IDIAP-RR 04-09.

[17] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Unsupervised Discovery of Multilevel Statistical Video Structures Using Hierarchical Hidden Markov Models," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, (Baltimore, MD), July 2003.

[18] A. Dielmann and S. Renals, "Dynamic Bayesian Networks for Meeting Structuring," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2004.

[19] B. Hannaford and P. Lee, "Multi-Dimensional Hidden Markov Model of Telemanipulation Tasks With Varying Outcomes," in *IEEE Int. Conf. on Systems, Man and Cybernetics*, 1990.