2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Online Intention Recognition for Computer-Assisted Teleoperation

Nikolay Stefanov, Angelika Peer, and Martin Buss
Institute of Automatic Control Engineering
Technische Universität München
D-80290 München, Germany
nikolay.stefanov@tum.de, angelika.peer@tum.de, m.buss@ieee.org

***Abstract*— An online intention recognition algorithm for computer-assisted teleoperation is introduced. The algorithm is able to distinguish between phases of a typical object manipulation task. It adopts a new advanced feature extraction algorithm which extracts features from haptic data and uses a Hidden Markov Model for stochastic classification. The method is implemented and validated on a real hardware setup. The obtained results reveal a robust and fast intention recognition.**

I. Introduction

When designing a high-quality telepresence and teleaction system, robust stability and ideal transparency are some of the main objectives to be realized. It is, however, a well-known fact that these objectives are contradictory and thus a compromise has to be accepted. Indeed, many control algorithms in literature, see [1] for an overview, still suffer either from a lack of transparency or a limited robust stability. Thus, some authors introduced computer assistance shared control algorithms that support the operator during the execution of a task with the aim of optimizing task performance and feeling of telepresence [2]–[4].

For computer-assisted teleoperation not only intelligent assistant functions, but also appropriate mechanisms for human intention recognition are required. Only a reliable intention recognition guarantees that the proper assistance is applied in a certain situation. In this paper we propose a method for human intention recognition based on the analysis of haptic interaction signals.

A. Related Work

Only few people deal with human intention recognition by analyzing haptic data. Most of them use a stochastic classifier for intention recognition. This implies that time series data cannot be handled directly, but needs to be pre-processed. When continuous classifiers are used, probability density functions (PDF) like Gaussian Mixture Models are calculated. Discrete stochastic classifiers require a discrete set of observations. In this case, features encoded in the signal must be extracted and stored in discrete classes before passing them to the stochastic classification.

This work is supported in part by the German Research Foundation (DFG) within the collaborative research center SFB453 "High-Fidelity Telepresence and Teleaction".

Calinon et al. [5], e.g., adopt a statistical approach based on Gaussian Mixture Regression and a continuous Hidden Markov Model (HMM) to learn typical human-robot interaction patterns. They consider the adaptation between human and robot to be a continuous process emitting typical communication patterns that can be encoded by an HMM when processing haptic data in the form of position, velocity, and force as input signals.

Takeda et al. [6] suggest a system for dance step estimation, which predicts the next step by observing haptic data. Again a stochastic model is used to classify time series data. The principle of the step estimator can be summarized as follows: First, features are extracted from time series data by adopting a moving window strategy and extracting window-related features. Then these features are passed as an observation sequence to an HMM that calculates the probabilities of future steps. The step with the largest probability is chosen to be the next step. The two major drawbacks of this method are the usage of windowing and the adoption of a fixed sampling rate for feature extraction. The former leads to information loss, while the latter introduces redundant information when no changes in the input signals occur.

In the literature, the importance of the feature extraction algorithm is often underestimated. Some of the widely used methods simply calculate an average over a specified number of samples [6], search for peaks, or use gradients of the signal as a feature [7]. There are only a few authors who consider advanced methods for feature extraction by transforming raw signal data into a more abstract domain [8], [9]. In this work, we propose a new method for feature extraction which results in a highly compressed feature vector which is a prerequisite for fast and accurate classification.

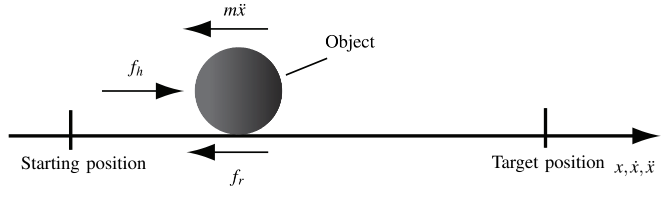

Fig. 1. Illustration of scenario used for intention recognition

As our aim is to develop a robust intention recognition algorithm for computer-assisted teleoperation based on online segmentation and classification, we briefly introduce the task our methods will be applied on. Instead of using a teleoperation system with a real slave device, we used a haptic interface to interact with a virtual, haptically rendered object. In doing so, we are able to avoid artifacts originating from the communication channel and can easily perform experiments with various objects and environment dynamics. The human can move the object by means of the haptic interface and is instructed to perform point-to-point movements by transferring the object from a starting to a target position, see Fig. 1. In doing so, the operator applies a force f_h to overcome the inertia of the object and operates against friction f_r. We focus on the recognition of two typical phases: i) the transportation phase (transportation of the object from the starting to the target position) and ii) the positioning phase (positioning the object at the target location). The block diagram of the envisioned intention recognition algorithm is illustrated in Fig. 2. It consists of the two steps: feature extraction and stochastic classification.

Section II focuses on the feature extraction from time-series data. This section is followed by a description of the stochastic classification algorithm as well as its training and evaluation, see Section III. The proposed online intention recognition algorithm is finally evaluated in a real hardware experiment which is described in Section IV. At the end, the presented work is summarized and future work is outlined.

II. Feature Extraction and Data Reduction

The proposed task recognition algorithm is supposed to distinguish between the transportation and positioning phase of a point-to-point movement. The following questions will be treated in this paper based on the assumption that the recorded time-series data for each of the two phases build distinguishable patterns as shown in Fig. 3:

1) Can the specific patterns for each phase be identified by analyzing haptic time series data?
2) Are there features that can describe the patterns accurately?

This section introduces a method for feature extraction from time series data, which is used in the next section for pattern recognition.

A. Feature Extraction

An interesting offline approach to extract features is introduced by Lin et al. [8]. They move a window over priorly captured time-series data and concatenate all raw data measurements enclosed by the time window. In doing so, they first concatenate normalized raw motion data from multiple heterogeneous signals to form a single vector that contains all measurements of a certain time instant. Then, these vectors are again concatenated to form a large super-vector containing all measurement data enclosed by the window. The dimension of this super-vector is reduced using Linear Discriminant Analysis (LDA) which projects the input vector into a lower-dimensional space. Finally, the features obtained after LDA are evaluated with a Bayes classifier and compared to manually segmented data. We adopt a similar approach, but instead of concatenating raw time-series data, we concatenate features priorly extracted from time-series data as introduced in [10]. The proposed method works online and results in a one-dimensional feature vector, which simplifies the structure of the stochastic model used for intention recognition.

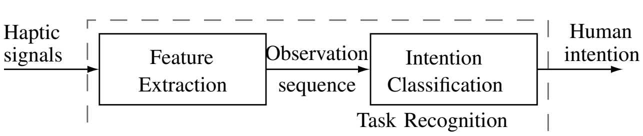

Fig. 2. Schematic view of the task recognition algorithm.

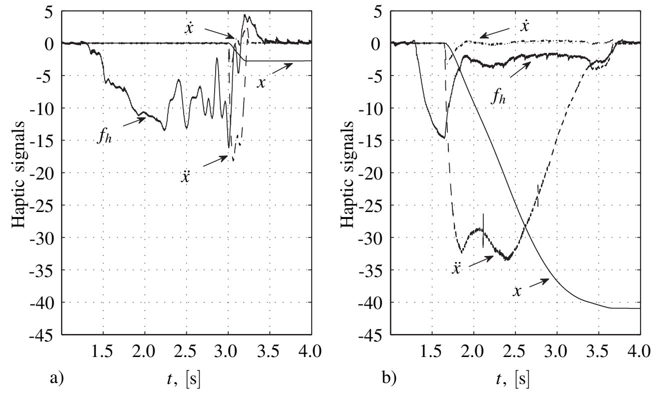

Fig. 3. Haptic time-series data: a) positioning phase, b) transportation phase.

To extract features from continuous data, we split each signal into several classes according to its magnitude. In doing so, several thresholds for each measurement are introduced. The range between two thresholds is called a class and the number of classes is denoted with C. For each signal 3 thresholds are used to produce $C = 4$ classes as indicated in Fig. 4. Each data point is automatically assigned to one of the classes. Its belonging to a certain class can be described by a binary number N consisting of k bits. In the case of four classes, $k = 2$ bits are necessary for a unique representation.

Because of noise, chattering can occur if the signal gradient is low when passing between two classes. To avoid this effect the noise of each signal is measured in steady-state and a value corresponding to 1.5 times the variance is set as a hysteresis factor ε. This avoids low-pass filtering, which would introduce a delay in the observed signals.

The following S signals are considered (according to the

TABLE I

Thresholds and Hysteresis Applied on Measurement Data

Signal	ε	Val. 1	Val. 2	Val. 3
x	0.0005	0.01	0.10	0.50
$\dot{x}$	0.0500	0.20	0.60	1.00
$\ddot{x}$	0.5000	1.00	3.00	6.00
f_h	0.3000	3.00	10.00	16.00

TABLE II
LOG FROM FEATURE EXTRACTION PROCEDURE

Number	Time [ms]	Val	Signal	Class	Feature vector (bin)
0	0000				00|00|00|00
1	1226	-3.307092	force	1	01|00|00|00
2	1474	-10.587643	force	2	10|00|00|00
3	1640	-9.660919	force	1	01|00|00|00
4	1813	-10.340996	force	2	10|00|00|00
5	1858	-1.621133	acceleration	1	10|01|00|00
6	1874	-9.635919	force	1	01|01|00|00
7	1923	-2.691566	force	0	00|01|00|00
8	1930	-0.451002	acceleration	0	00|00|00|00
9	1971	3.319868	force	1	01|00|00|00
10	1976	1.513964	acceleration	1	01|01|00|00
11	1995	2.639007	force	0	00|01|00|00
12	2050	0.497919	acceleration	0	00|00|00|00

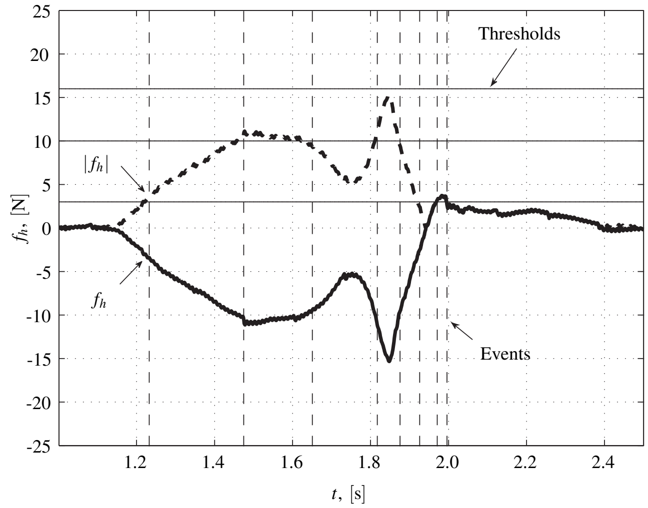

Fig. 4. Illustration of feature extraction procedure for the positioning phase.

order they are recorded): position x, velocity $\dot{x}$, acceleration $\ddot{x}$ and applied human force f_h. Thresholds for each signal are estimated by considering a large number of measured training data. The selected values for thresholds and hysteresis for each signal are shown in Table I.

B. Concatenating Features

The above-described algorithm performs an amplitude discretization of each signal. Our next goal is to obtain a one-dimensional feature vector. Instead of using separate variables for each signal, the corresponding features N are concatenated and stored by means of a single binary variable with length L bits. The entire length $L = S \cdot k$ depends on the number of observed signals S and the number of bits k needed to encode all classes C of a signal. In the case of $S = 4$ and $k = 2$, the feature vector is of length $L = 8$ bits. Consequently, the feature vector encodes $m = 2^L = 256$ different symbols. An example log from a feature extraction corresponding to time-series data shown in Fig. 4 is reported in Table II. The feature vector is finally passed to the HMM as an output sequence O.

C. Feature Compression

Typical haptic rendering algorithms require a sampling rate of about 1 kHz [11]. Extracting a feature vector for every time instant would consequently produce a large amount of redundant data, which needs to be reduced in dimension before passing it to the stochastic classification algorithm. The intention of a human changes with much lower frequency, which would suggest downsampling measured data, but this leads to information loss. Thus, we propose another possibility for data reduction: we add an entry to the feature vector only when at least one of the signals passes a threshold. This leads to an event-oriented feature extraction. Table II visualizes the adopted procedure.

III. INTENTION MODELING

In this work, we adopt a Hidden Markov Model (HMM) for human intention recognition.

A. Brief Review of HMMs

A Hidden Markov Model models stochastic processes with unobservable (hidden) underlying states. While the observer is not aware of the actual state of the system, the probabilities to move from one state to another, the so-called transition probabilities [12], and the probability that a certain symbol is emitted by the system while being in one of its states, the emission (or output) probability, are known. An HMM is correctly defined if the following tuple $\lambda = (Q, V, \pi, A, B)$ is specified:

- $Q = \{q_1, q_2, \ldots, q_n\}$, set of n (hidden) states;
- $V = \{v_1, v_2, \ldots, v_m\}$, set of m output symbols;
- $\pi(i)$, initial probability of being in state q_i at time $t = 0$;
- A, $n \times n$ matrix of transition probabilities $P(q_j(t+1)|q_i(t)) = \{\alpha_{ij}\}$ of being in state i at time t and passing to state j at time $t+1$;
- B, $n \times m$ matrix of emission probabilities $\{b_{jk}\}$ with $b_{jk} = P(v_k \text{ for } t|q_j)$ of producing the observation v_k at time t while being in state q_i.

Three canonical problems are typically solved using HMMs [13]:

1) Given the model λ and an observation sequence $O = \{o_1, o_2, \ldots, o_\tau\}$ with $o_i \in V$, compute the probability that O is produced by λ.
2) Given the model λ and an observation sequence $O = \{o_1, o_2, \ldots, o_\tau\}$ with $o_i \in V$, find the state sequence $S = \{s_1, s_2, \ldots, s_\tau\}$ with $s_i \in Q$, which most likely has generated the emitted sequence.
3) Given a set $\{O_k\}$ of $k = 1, 2, \ldots$ observation sequences, adjust the model parameters λ to maximize $P(O|\lambda)$.

Most authors who analyze motion data work on the second problem, see e.g. [7]. Solutions for the second problem produce good results if a low-level interpretation is desired, e.g. when predicting the next most probable state. If a more abstract interpretation is desired as in the case of whole task recognition or if human behavior is totally stochastic, solving the first canonical problem as proposed in [14]

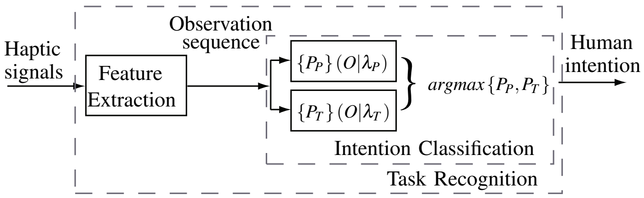

Fig. 5. Schematic representation of the task recognition algorithm.

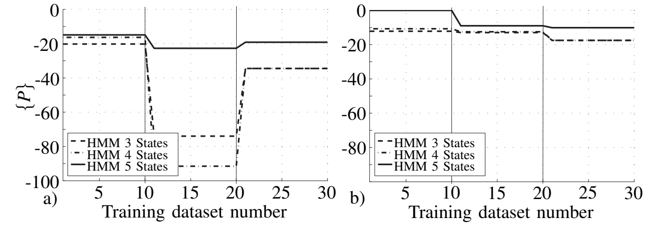

Fig. 6. HMM threefold-validation to determine required number of states: Results for a) positioning phase, b) transportation phase

is more suitable. As the goal in the current study is to distinguish between two predefined phases (positioning and transportation) of a given task that can be executed in a random order, a similar approach is applied here. The difference from the aforementioned work is that force and position data are used directly and there is no need for online impedance estimation, which speeds up the classification.

The following procedure is used for intention recognition: Two HMMs are trained offline for each phase of the task. As the observation sequence is used for both of them, the probability that a certain observation can be explained by one of the HMMs is dependent on the HMM parameters only. As each HMM is trained for a concrete phase of the task, the current phase is considered to be represented by the HMM with the highest probability (see Fig. 5).

B. Training Procedure

To distinguish between the two phases of the task, HMMs for each of the phases need to be trained: λ_P for positioning and λ_T for transportation. First, features are extracted from the measured haptic data as described in Section II. Then, the resulting feature vectors are passed as an observation sequences to the training algorithm. Since the sequences are one-dimensional, the HMM Toolbox of Matlab is used for training by adopting the Baum-Welch algorithm [15], [16]. Note, that the length of the observation sequence is different for each recorded data set. As the states do not have any physical interpretation, there is no deterministic method to identify the number of required states. Thus, the following approach is adopted: An initial rough estimate is obtained by training a couple of HMMs with different numbers of states and observing the rank of their transition matrices A. If the transition matrix A loses its full rank, the HMM contains redundant states. The HMM with the highest numbers of states, but still with a full rank transition matrix is used to define a region of minimal required states. In our example, 30 training datasets are used for each of the phases to be identified. To prove consistency of the results, a threefold cross-validation is performed [17]. For this purpose, the training datasets are split into equal groups, each group containing 10 datasets. Two of them are used for training and one for validation. Following the aforementioned procedure, for each phase, 9 HMMs (HMMs with 3, 4, and 5 states for each of the 3 groups of datasets) are trained using a recursive training procedure with $\varepsilon = 10^{-5}$ as tolerance.

Validation of the HMMs is performed by computing the forward probability, i.e. the probability that a HMM has produced a given output sequence. Since the resulting probabilities are very small (because of the large number m of possible observations), the logarithmic probability $\{P\}$ is used for comparison. Fig. 6 shows the average probability for each group of datasets when being evaluated using HMMs with 3, 4, and 5 states. Except for the second group of the positioning task, where some outliers caused a drop in the recognition ratio for HMMs with 3 and 4 states, no considerable differences in the recognition ratio were observed when increasing the number from 3 to 4 or 5 states. Moreover, classifiers of higher degree become too general, which worsens the overall classification results [18]. Because of this and to reduce computation power, HMMs with 3 states were selected for both phases to be recognized.

C. HMM Evaluation

Stochastic classification of the two tasks is realized as shown in Fig. 5. For evaluation, 10 sequences representing each the transportation and positioning phase are evaluated by using the trained HMMs. The probability $\{P_P\}(O|\lambda_P)$ that the sequence O is emitted by model λ_P trained with positioning data, as well as the probability $\{P_T\}(O|\lambda_T)$ that the sequence is emitted by λ_T trained with transportation data are calculated using the forward algorithm.

As the task was performed with different speeds, datasets are not necessarily of the same length. Typical datasets for positioning result in a length of 10-12 symbols, and datasets for transportation in 8-10 observation symbols, see Fig. 4 and Table II for an example. Thus, each time the Feature Extraction block provides a new symbol, 3 different observation sequences of length 8, 10 and 12 are formed by adding the new symbol to the already existing ones. Then the resulting sequences are passed to the Stochastic Classifier and the probabilities $\{P_P\}(O|\lambda_P)$ and $\{P_P\}(O|\lambda_P)$ are determined. The pair with the biggest difference is finally used to identify the actual human intention.

The offline evaluation results are shown in Fig. 7. On the left the evaluation of 10 positioning sequences is illustrated. The stars indicate the probability a sequence is produced by the HMM for positioning λ_P, while the circles denote the results for the same sequence when evaluated with a HMM for transportation λ_T. On the right the corresponding results for 10 transportation sequences are shown. Missing points on both plots denote $\{P\}(O|\lambda) \to -\infty$ or $P(O|\lambda) = 0$. As can be seen the proposed method achieves an almost 100% classification.

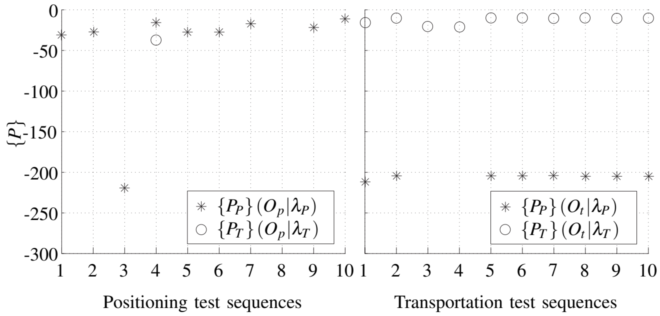

Fig. 7. Results of the cross validation test.

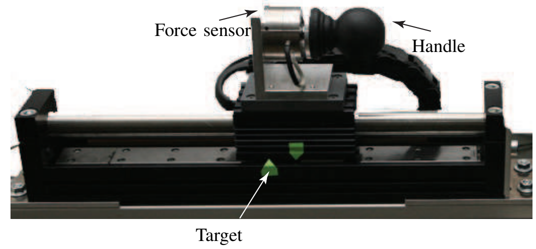

Fig. 8. 1 DOF linear haptic interface used in experiments.

IV. Online Experimental Evaluation

The intention recognition algorithm proposed in this paper is designed to distinguish between phases of a typical manipulation task. Object handling in a constraint environment including phases of transportation and positioning as already illustrated in Fig. 1 is chosen as evaluation scenario. In this section the experimental setup used for evaluation of the proposed intention recognition algorithm is introduced and results obtained when applying it are presented.

A. Hardware Setup and Scenario

To simulate the virtual object a haptic rendering algorithm is used which implements a rigid object of mass $m = 3$ kg that can slide horizontally over a rigid surface. In order to increase the realism of the simulation, the following friction model is implemented:

$$\begin{aligned} f_f &= f_{st} + f_k + f_v \quad \text{with} \\ f_{st} &= \begin{cases} -f_h & \text{for } f_h < \mu_s f_N \\ 0 & \text{for } f_h \geq \mu_s f_N \end{cases} \\ f_k &= \mu_k sgn(f_h) \\ f_v &= \mu_v \dot{x} \end{aligned}$$

where f_{st} is the static, f_k the kinetic and f_v the viscous friction and $\mu_s = 0.5$, $\mu_k = 0.2$ and $\mu_v = 0$ are the corresponding friction coefficients. The human interaction force is denoted with f_h, the normal force with f_N (in the considered 1 DOF case, the normal force has the meaning of a threshold). No visual feedback of the object is provided. The velocity signal is obtained by derivation of the position signal, acceleration is measured by an additional acceleration sensor. The resulting model simulates a 3 kg heavy steel object that can be moved on a wooden surface (see [19]).

Experiments are performed using a 1 DOF haptic interface controlled by position-based admittance control, see Fig. 8. The interface uses linear actuator which is equipped with a force sensor and hand knob. Its control is implemented in Matlab/Simulink and executed on the Linux Real Time Application Interface RTAI. The haptic rendering runs on another computer, and communication is realized by a UDP connection in a local area network.

B. Results

The proposed intention recognition algorithm was applied to three random sequences (each one minute long) of transportation and positioning tasks performed online, see Fig. 9 for an example. The thick line refers to the recorded position signal while the thin dotted line indicates the estimated human intention. As can be observed the proposed algorithm enables a very fast recognition while simultaneously achieving a high recognition rate of 89%. As long as features with a reasonable compression rate are used, a very important property of HMMs holds, the invariance to local time warping [20]. This explains the high robustness of the proposed intention recognition algorithm.

Finally, we would like to highlight the importance of using force data for intention recognition when performing a haptic-related task. A great number of studies exist that focus on the classification of free space motion only and thus limit their analysis to motion data. When manipulation in a constraint environment is considered, not only the trajectory of motion, but also the impedance contains information about the human intention [14]. Although we do not estimate the human impedance explicitly, changes in the impedance are encoded in our feature vector. Hence, the presented approach gives the possibility for accurate intention recognition at an early stage even when the differences between positioning and transportation phase are not that expressive when observing motion data only. For illustration we adopted the same proposed algorithm using motion data only and used the same training and test data. The obtained results are illustrated in Fig. 10. One can see that the recognition rates (75%) are comparable to the ones obtained when also force is considered, but as expected the task cannot be recognized at an so early stage of execution than when using haptic data.

V. Conclusion

In this work, an intention recognition algorithm to be applied in computer-assisted teleoperation was introduced. A new method for feature extraction was proposed, and the used stochastic classifier was presented. The proposed classifier is trained offline and then evaluated on a real hardware setup by online classification of a randomly performed task consisting of transportation and positioning phases. The new proposed method is based on an event-based feature extraction and uses an HMM for stochastic classification. The obtained results indicate a very fast and accurate recognition of different phases of the task.

A major drawback of the proposed method, however, is the usage of a discrete stochastic classifier, which requires discrete observation signals. Thus, time-series data needs to

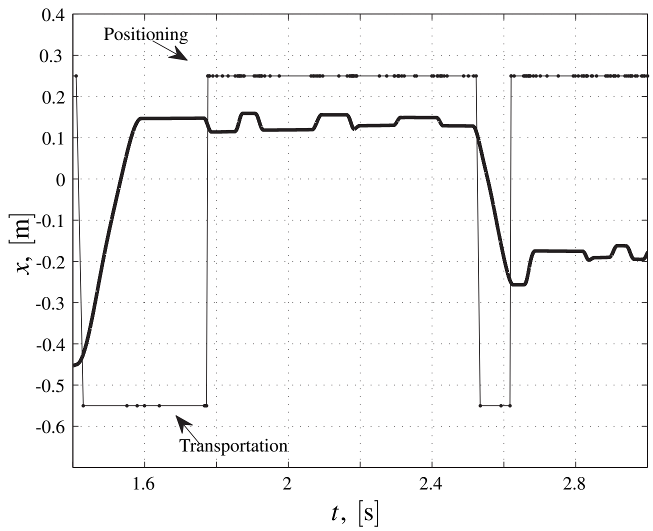

Fig. 9. Experimental results when using haptic data for intention recognition. Intentions are immediately recognized.

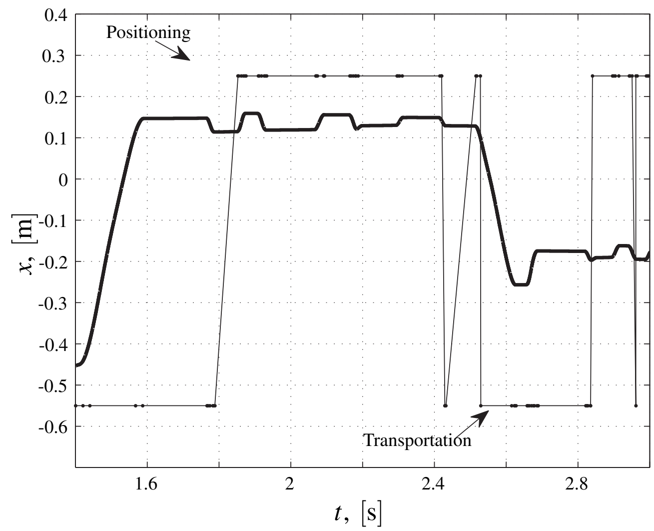

Fig. 10. Experimental results when using motion data only for intention recognition. Task recognition is delayed.

be discretized first. Since probability density functions allow a better approximation, we expect an even better recognition rate when using continuous HMMs.

The application of the algorithm is not limited to teleoperation only, but can be similarly applied for intention recognition in human-robot interaction scenarios where the robot is supposed to physically interact with the human. Thus, applications range from rehabilitation (where the robot is supposed to guide the human on a certain path), industrial assembly (where the robot is supposed to assist the human in transporting and positioning heavy objects) to training in medicine or sports (where skill acquisition by adopting shared control algorithms is of concern).

Future work will be focused on improving robustness of the proposed approach with respect to changes in the environment or different object properties. We also seek after methods allowing fast adaptation to different human operators.

REFERENCES

[1] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, 2006.

[2] U. Unterhinninghofen, F. K. Freyberger, and M. Buss, "Study on computer assistance for telepresent reaching movements," in *EuroHaptics '08: Proceedings of the 6th international conference on Haptics*, (Berlin, Heidelberg), pp. 745–754, Springer-Verlag, 2008.

[3] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3639–3644, April 2007.

[4] D. Aarno and D. Kragic, "Motion intention recognition in robot assisted applications," *Robot. Auton. Syst.*, vol. 56, no. 8, pp. 692–705, 2008.

[5] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *Advanced Robotics, 2006 14th International Conference on*, 2009.

[6] T. Takeda, K. Kosuge, and Y. Hirata, "HMM-based dance step estimation for dance partner robot -ms dancer-," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3245–3250, Aug. 2005.

[7] D. Kulic and E. Croft, "Affective state estimation for Human - Robot interaction," *Robotics, IEEE Transactions on*, vol. 23, pp. 991–1000, Oct. 2007.

[8] H. Lin, I. Shafran, D. Yuh, and G. Hager, "Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions," *Computer aided surgery*, vol. 11, pp. 220–230, 2006-Sep.

[9] A. Fod, M. J. Mataric, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, pp. 39–54, 2002.

[10] R. T. Olszewski, *Generalized Feature Extraction for Structural Pattern Recognition in TimeSeries Data*. PhD thesis, School of Computer Science Carnegie Mellon University, 2001.

[11] B. Siciliano and O. Khatib, eds., *Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008.

[12] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience Publication, 2000.

[13] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, pp. 4–16, Jan 1986.

[14] Z. Wang, A. Peer, and M. Buss, "An HMM approach to realistic haptic human-robot interaction," in *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pp. 374–379, March 2009.

[15] G. J. Mclachlan and T. Krishnan, *The EM Algorithm and Extensions (Wiley Series in Probability and Statistics)*. Wiley-Interscience, October 2007.

[16] J. Salojärvi, K. Puolamäki, and S. Kaski, "Expectation maximization algorithms for conditional likelihoods," in *Proceedings of the 22nd International Conference on Machine Learning (ICML-2005*, pp. 753–760, ACM press, 2005.

[17] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 ed., 2005.

[18] S. Calinon and A. Billard, "Learning of Gestures by Imitation in a Humanoid Robot," in *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, pp. 153–177, Cambridge University Press, 2007. K. Dautenhahn and C.L. Nehaniv (Eds).

[19] K.-H. Grote and J. Feldhusen, eds., *DUBBEL – Taschenbuch für den Maschinenbau*. Berlin, Heidelberg: Springer, 22 ed., 2007.

[20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.