An Experience-Driven Robotic Assistant
Acquiring Human Knowledge to Improve Haptic Cooperation

José Ramón Medina   Martin Lawitzky   Alexander Mörtl   Dongheui Lee   Sandra Hirche

Institute of Automatic Control Engineering

Technische Universität München

80290 Munich, Germany

Email: {medina, ml, moertl, dhlee, hirche}@tum.de

Abstract—**Physical cooperation with humans greatly enhances the capabilities of robotic systems when leaving standardized industrial settings. Our novel cognition-enabled control framework presented in this paper enables a robotic assistant to enrich its own experience by acquisition of human task knowledge during joint manipulation. Our robot incrementally learns semantic task structures during joint task execution using hierarchically clustered Hidden Markov Models. A semantic labeling of recognized task segments is acquired from the human partner through speech. After a small number of repetitions, the robot uses an anticipated task progress to generate a feed-forward set point for an admittance feedback control scheme. This paper describes the framework and its implementation on a mobile bi-manual platform. The evolution of the robot's task knowledge is presented and discussed. Finally, the cooperation quality is measured in terms of the robot's task contribution.**

I. INTRODUCTION

As robots are entering new domains starting to provide close physical assistance to human workers, a strong need for the ability to learn semantic task knowledge from human co-workers arises. Any approach of pre-programming all possible interaction behaviors for all possible combinations of task goals is infeasible for fairly unstructured settings of human manual work. Instead, in our opinion, a cognition-enabled robotic co-worker is expected to implement a *learning-by-doing* strategy for physical interaction tasks. This implies that the robot starts as a rather passive *pack mule* being guided by a human partner. In order to exploit the naturally given cognitive capabilities of the human co-worker, a cognition-enabled robot observes the human task contribution in terms of physical signals and learns how to recreate the complementary patterns. Furthermore, the authors are convinced that a cognition-enabled robotic assistant should enrich its own experience by acquisition of meaningful semantic labels from dialogue with human co-workers as a basis for further linguo-haptic interaction improvement. Neuroscientific findings second this opinion as loop closure around sensory-motor observation, imitation and control through explicit communication is observed in action reproduction from human-human settings [1].

Exemplarily, in this work, we address the problem of joint bulky object transportation, as illustrated in Fig. 1, however, the conceptual approach is not limited to this application. This task is specifically challenging due to the tight physical

Fig. 1. Experimental scenario: Human and robot jointly carrying a bulky bumper to its mounting location during car restoration.

coupling between human and machine which inseparably serves as a channel for energy but also information exchange.

Caster-like robot partner behavior reactively compensating the object dynamics is well-suited for human-robot joint bulky load transport which is nicely shown in [2] and [3]. However, such a follower strategy implements merely a trolley for heavier loads rather than an actual cooperation partner and, while simple tasks can successfully be fulfilled, more complex tasks including environmental constraints typically require an active contribution to the task by the robot [4]. Active robotic assistance also reduces the effort applied by the human partner [5], [6], [7]. In order to plan the next robot action for assistance to the human, the next human action needs to be predicted. For simple motor tasks, findings from human motor behavior are considered for movement prediction, for example the well-known minimum jerk velocity profile for point to point movements [5], [8]. However, for more complex tasks there are not any such analytical models currently available. In consequence, learning from observation approaches have become a favorable method to address these challenges. We have investigated the theoretical background on task dynamics in joint manipulation and incremental learning for physical human-robot interaction [4], [9]. Remaining open questions include the implementation of an actively contributing robotic partner in joint human-robot

manipulation and generalization of the learning approach to higher-dimensional observations. The idea of a robot asking for semantic information has been extensively explored in the *Autonomous City Explorer* project. A robot traveling through the city of Munich extracted semantic information from communication with passers-by [10]. Transferring this concept to physical human-robot interaction in terms of joint human-robot manual work is part of this paper's inspiration.

II. PROBLEM STATEMENT AND CONCEPTUAL APPROACH

The envisaged task is a joint load transport from a commonly known starting pose to a goal pose, first only known to the robot's human partner. The research questions we address in this paper are: How can a full-scale mobile robot learn from haptic interaction, enrich its experience by asking questions and assist in an appropriate way based on its knowledge? Secondly: How does an assistive behavior emerge in our robot implementing our proposed approach over time?

We confine our problem to the following conditions:

- One human moves a bulky object together with a robot from a starting pose to a final pose along an intuitive trajectory avoiding collisions with the environment.
- Both participants *tightly grasp* the same *rigid* object with commonly known shape and dynamics.
- Haptic interaction through the object and speech are possible communication channels between the human and the robotic partner.
- Environmental constraints are such that a feasible path to the goal exists.

The contribution of this paper is an experimental proof-of-concept study towards an experience-driven physical robotic assistant including knowledge acquisition, semantic labeling and motion re-creation and control.
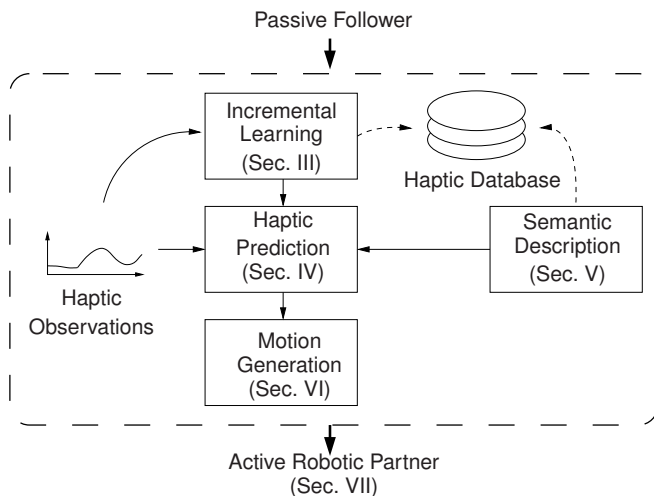

Fig. 2.   System Architecture

As depicted in Figure II, our approach consists of four major interacting modules. When the system is started, the robot acts as a passive follower, implementing a second order admittance control law in inertial coordinates and is pulled and pushed from start to goal by the human

partner through the object. Any force and motion input is unsupervisedly segmented, clustered and added as haptic interaction primitive to an experience database. Additionally, any force and motion input is compared to existing database entries, generating recognition matches. Patterns that have been observed multiple times are classified to be of interest and worth asking the human partner for a semantic description. This semantic knowledge is later used to ask the human partner about the desired trajectory depending on the current recognition quality. Whenever the recognition is successful either directly from database matches or with additional certainty from direct human partner feedback, a motion pattern is generated according to the corresponding database entry. This motion is translated into a virtual force input, acting on the virtual admittance mentioned above.

The remainder of this paper is organized as follows: In the next Section II an overview on the system architecture is given. Section III gives a brief theoretical background to the algorithm for segmentation and learning from force and motion data, followed by prediction in Section IV. The learned models are enriched with human semantic knowledge, in Section V. The feedback control algorithms for task execution are explained in VI. Implementation and results from our experimental evaluation are presented in Section VII.

By convention, bold characters are used for vectors and matrices. Variables representing functions of time are not necessarily marked as such for compactness of the mathematical descriptions.

III. AN HMM APPROACH TO LEARNING FORCE AND MOTION SIGNALS

The autonomous acquisition of haptic motion patterns requires unsupervised segmentation, clustering, and an incremental learning mechanism allowing generalization, and behavior recreation using a regression algorithm.

A. Automated Segmentation, Learning and Clustering

Making the robot more assistive as it gathers new observations requires as a first step an online autonomous incremental learning framework. The basic structure for the segmentation and incremental learning process is based on the algorithms in [11], which is summarized in this subsection and illustrated in Fig. 3 a).

In order to extract behavior patterns from observations autonomously, the observed force and motion signals are first segmented into potential primitives. The stochastic approach assumes that data belonging to the same primitive will have the same underlying distribution.

Once a segmentation point is detected, each segmented time series is encoded into a left-to-right HMM λ as explained in the following Section III-B. In order to group and structure similar observations, a hierarchical tree of behavior primitives is built as follows: the newly constructed HMM is compared to the existing nodes in the *primitive tree*[1] and

---

[1]Throughout this paper, 'primitive tree' denotes the a hierarchical tree of behavior primitives built by the clustering. The term 'node' denotes a node in the 'primitive tree'.
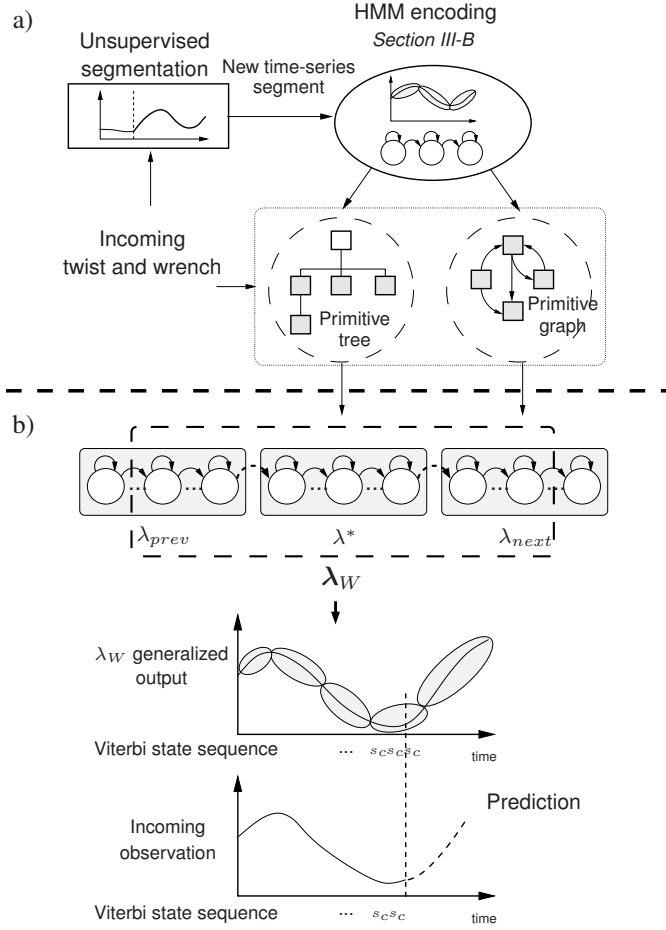
Fig. 3. Overview of the learning and prediction procedure. λ denotes the most likely HMM given the incoming observations, λ_{prev} denotes the HMM representing the previous primitive and λ_{next} refers to the HMM representing the next most likely primitive that follows the current one. λ_W is the window HMM built based in λ, λ_{prev} and λ_{next} and s_c is the currently estimated HMM state of λ_W given the incoming observations.

is inserted as a member of the closest node. If sufficiently similar members are found in the node, a new child group of this node is formed. Note that a node is also represented by an HMM, which is trained with the generated outputs from its members and the distance between the HMMs is estimated using a symmetric version of the Kullback-Leibler divergence. The tree represents the robot's knowledge in terms of a compact database of continuously incoming data and each node represents a *behavior primitive*.

In parallel to the clustering process, the temporal relation between behavior primitives is learned building a directed *primitive graph* where the observed primitives are represented by its nodes and an edge between two nodes represents the probability to transit from one node to another. This additionally learned structure allows trajectory predictions not only within a primitive but also during primitive transitions.

B. Encoding Force and motions into HMMs

HMMs provide a compact and probabilistic spatio-temporal representation of the training data. Each HMM λ is given by a set of parameters $\{\pi, a, w, \mu, \Sigma\}$, where π represents the initial state probabilities, a represents the

probability of the transitions between the states, and w, μ and Σ are the weight, the mean and the covariance of the mixture components of the states.

While HMM's discretized state space provides a good recognition performance, it also leads to limitations when generating a continuous trajectory as required for haptic signal predictions. In order to improve this performance, the spatio-temporal correlation is learned during the training and the responsibility over time of each state is calculated.

Multiple observations are used as training data. Each observation $o = \{{}^s o(t), {}^t o(t)\}$ consists of the spatial data ${}^s o(t)$ which is the observed twist trajectory and wrench at time t, and its associated time sequence ${}^t o(t) = t$.

The above mentioned standard HMM parameters are trained using the Baum-Welch algorithm [12] for the spatial data. Additionally, the temporal data is used to calculate the time mean ${}^t \mu_{ik}$, the variance of the time ${}^t \Sigma_{ik}$ and the covariance between temporal and spatial data ${}^{ts} \Sigma_{ik}$ for each state i and each mixture component k.

C. Decoding from HMMs

Using the additional temporal information acquired in the training, the responsibility over time ${}^t \gamma_{ik}(t)$, which represents the probability of being at state i at time t with the k-th mixture component based on the time information, is calculated as

$$
{}^t \gamma_{ik}(t) = \frac{\mathcal{N}(t|{}^t \mu_{ik}, {}^t \Sigma_{ik})}{\sum\limits_{s=1}^{N} \sum\limits_{j=1}^{K} \mathcal{N}(t|{}^t \mu_{sj}, {}^t \Sigma_{sj})} \tag{1}
$$

where a Gaussian for each state i and each mixture component k is centered on the mean of the time ${}^t \mu_{ik}$ and with a variance ${}^t \Sigma_{ik}$. Note that N is the number of states and K is the number of mixture components of each state.

The spatial data is generated by the Gaussian Mixture Regression (GMR) algorithm weighted according to the responsibility over time. For each time step t the conditional expectation of the spatial data $\hat{o}(t)$ is given by

$$
\hat{o}(t) = \sum_{i=1}^{N} \sum_{k=1}^{K} {}^t \gamma_{ik}(t) \left({}^s \mu_{ik} + \frac{{}^{ts} \Sigma_{ik}}{{}^t \Sigma_{ik}} (t - {}^t \mu_i) \right).
$$

IV. FORCE AND MOTION PREDICTION

Relying on the knowledge acquired in Sec. III represented by the primitive graph and the primitive tree, a predicted set point is generated. An overview of the prediction procedure is shown in Fig. 3 b).

Given the incoming twist and wrench, the most likely node λ^* and its next most likely primitive λ_{next} are selected from the primitive tree and the primitive graph respectively. A window HMM λ_W is then defined over the last estimated λ_{prev}, the current most likely λ^* and the next most likely primitive λ_{next}. Note that a window HMM over sequentially executed left-to-right HMMs is built connecting the last state of the first HMM to the first state of the second one.

Using the resulting window model, the Viterbi algorithm can be applied to estimate the current HMM state s_c. To improve the accuracy of the prediction and estimate how far the execution of this state is, the generalized output of the window model can be used as a reference. Comparing the duration of estimated the state s_c in the incoming observations and the generalized output, the current state can be approximated as a time index of the latter.

Applying the regression explained in Section III-C and given a desired prediction time horizon of L samples in the future, the predicted data is calculated as follows

$$\hat{\boldsymbol{o}}(t_c + L) =$$
$$\sum_{i=1}^{N} \sum_{k=1}^{K} {}^t\gamma_{ik}\left({}^t o(t+L)\right)\left({}^s\boldsymbol{\mu}_{ik} + \frac{{}^{ts}\boldsymbol{\Sigma}_{ik}}{{}^t\Sigma_{ik}}\left({}^t o(t+L) - {}^t\mu_{ik}\right)\right),$$

$$\text{with } {}^t o(t_c + L) = \left(\frac{L + n_{s_c}}{v_f} + t_{s_c}\right) \quad , \qquad (2)$$

where t_c is the current state estimation on the generalized state sequence, t_{s_c} is the time index of s_c on the generalized state sequence, n_{s_c} is the length of s_c on the incoming observations state sequence. v_f is a velocity factor given by $\frac{|\dot{\boldsymbol{x}}_l|}{|\dot{\boldsymbol{x}}_c|}$, where $\dot{\boldsymbol{x}}$ is the currently estimated twist and $\dot{\boldsymbol{x}}_c$ the currently observed one. Note that $\dot{\boldsymbol{x}}_l$ is calculated from $\hat{\boldsymbol{o}}(t_c)$.

V. EXTRACTION OF TASK SEMANTICS

Continuous force and motion patterns as used in the segmentation and learning algorithms as described in Section III describe the physical development of a task. However, in order to exploit the cognitive capabilities of the human partner, an intuitive direct user interface on a more abstract level must be provided. The extracted haptic primitives as introduced in Section III can be parameterized to provide a viable level of abstraction with a spatio-temporal resolution suitable for natural language descriptions. Whenever an unlabeled primitive is recognized, the robot acquires a description from its human partner. This label can then be used to

- merge two or more primitives representing the same semantic meaning,
- generate queries on the next primitive to choose in cases of similar priors,
- reconfirm choices whenever the measured haptic input deviates from the expected.

A. Merging multiple nodes

User input can be used to cluster two or more nodes with the same semantic meaning into one. Therefore, the generalized outputs of the source nodes $\hat{\boldsymbol{o}}$ are used as training input for a new node getting the same label. The source nodes become children of the new node.

B. Query generation

In order to avoid false predictions on the upcoming primitive, a dialogue, based on the prior of the transition from λ^* to λ_{next}: $p(\lambda^* \rightarrow \lambda_{next})$, given by the primitive graph is executed. We distinguish three different cases, as depicted in Fig. 4:

a) $p(\lambda^* \rightarrow \lambda_{next}) = 1$. The upcoming primitive is deterministic. No query is generated.
b) $1 > p(\lambda^* \rightarrow \lambda_{next}) > \epsilon$. The robot is very certain about the upcoming primitive. A reconfirmation query is generated, suggesting λ_{next} as the next step.
c) $p(\lambda^* \rightarrow \lambda_{next}) \leq \epsilon$. No clear candidate for the upcoming primitive can be determined. A neutral query is phrased, asking for the next step λ_{next}.

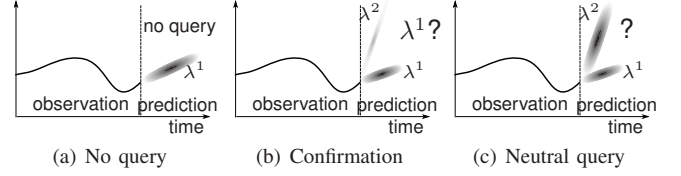(a) No query   (b) Confirmation   (c) Neutral query

Fig. 4. Query-generation mechanism

VI. ROBOT CONTROL ARCHITECTURE

A feedback control scheme is adopted to provide a prediction-based assistive robot behavior as well as a homogeneous reactive behavior allowing for human force inputs.

A. Overall Control Scheme

As depicted in Fig. 5, holonomic maneuverability is provided by an admittance control scheme of a mobile robot. The robot moves the object along a resulting pose trajectory $\boldsymbol{x}$ leading to an object wrench $\boldsymbol{u}_{obj}$ in the robot's contact point, serving as input to the admittance control law. Additionally, a human partner interacts with the object on a different contact point with wrench $\hat{\boldsymbol{u}}_h$. The object geometry transforms this wrench to $\boldsymbol{u}_h$ acting in the robot's contact point. Using the prediction $\boldsymbol{x}_{pred}$ explained in Section IV an assistive impedance control law generates an active robot force input $\boldsymbol{u}_r$.
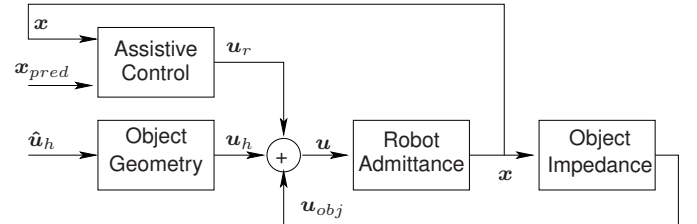

Fig. 5. Overall control scheme consisting of an impedance type assistive control and admittance type reactive control scheme.

B. Interaction Control

The reactive robot behavior is realized implementing an admittance control law

$$\boldsymbol{u} = \boldsymbol{M}_r \ddot{\boldsymbol{x}} + \boldsymbol{D}_r \dot{\boldsymbol{x}} \qquad (3)$$

with a rendered virtual mass $\boldsymbol{M}_r$ and rendered virtual viscous friction $\boldsymbol{D}_r$. Note, that task-related constraints can be easily introduced by rendering a virtual stiffness $\boldsymbol{K}_r$ in the desired spatial directions.

The assistive robot behavior based on the motion prediction $\boldsymbol{x}_{pred}$ is rendered by an impedance control scheme

$$\boldsymbol{u}_r = \boldsymbol{K}_p(\boldsymbol{x}_{pred} - \boldsymbol{x}) + \boldsymbol{K}_d(\dot{\boldsymbol{x}}_{pred} - \dot{\boldsymbol{x}}) \qquad (4)$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ denote the proportional and derivative control gains respectively.

C. Manipulator-Base Coordination

The underlying control concept of the manipulator-base coordination adopted in this paper is depicted in Fig. 6. Similar to [13], the admittance control law is calculated in inertial coordinates so that repositioning of the mobile base does not affect the end-effector position. The actual end-effector pose $^R\boldsymbol{x}_m$ is used to derive a velocity command $^R\dot{\boldsymbol{x}}_b$ to the mobile base, following the base control law:

$$^R\dot{\boldsymbol{x}}_b = \begin{pmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{pmatrix} = \mathrm{diag}\left(K_{hdg} \quad K_{dst} \quad K_{tng} \right) \begin{pmatrix} e_{hdg} \\ e_{dst} \\ e_{tng} \end{pmatrix}$$

Three independent proportional control laws move the mobile base minimizing heading error e_{hdg}, distance error e_{dst} and tangential error e_{tng} of the base pose w.r.t. the end-effector pose $^R\boldsymbol{x}_m$, as illustrated in Fig. 7.
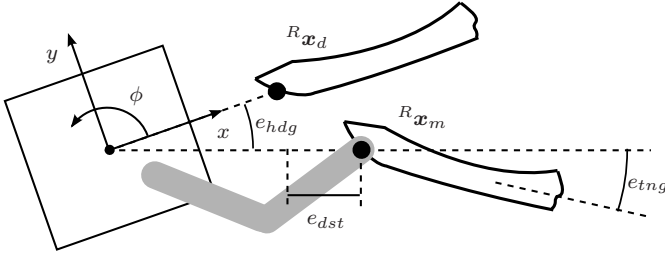

Fig. 7. Base pose control w.r.t. to a reference pose $^R\boldsymbol{x}_d$ of the manipulator's end-effector.

A reference pose of the end-effector $^R\boldsymbol{x}_d$ is chosen to meet certain requirements regarding task-related manipulability. The resulting motion command $^R\dot{\boldsymbol{x}}_b$ is then executed by an omni-directional velocity control law as proposed in [14].

VII. EXPERIMENTAL EVALUATION

In order to evaluate our approach and to get an impression on the behavior of a robot implementing our proposed scheme for learning, semantic labeling and control, we conducted a full-scale experiment in our laboratory. As an exemplary domain, we chose a classic-car restoration setting. The manipulated object, a 1.2 m long *Mini*'s steel bumper (b) weighing 1.9 kg is depicted in Fig. 8. The distance between the pre-defined grasp points of human and robot is 1.1 m. More information on the robot used can be found in [15].

A. The Experimental Robot Platform

The robot used in this experiment (see Fig. 8) stands on a four-wheeled omni-directional mobile platform (f) which offers roughly human-like maneuverability and smooth motion [16]. Two identical anthropomorphic 7-degrees-of-freedom (DoF) arms (c) are front-mounted on the top of the main chassis to provide a human-like working space [17]. In this experiment, only the right arm is used. Mounted onto a JR3 wrench sensor, the manipulator is equipped with a Schunk PG70 two-finger parallel gripper (a) which allows a

tight grasp of the object. Lithium-ion polymer batteries (e) power the system for long periods without recharging. For computational power, the robot carries three PCs (d). The first is a an *Intel Core i7 920* running at 2.66 GHz executing the online learning and prediction algorithm on multiple CPU cores utilizing the *OpenMP* library at an update rate of 20 Hz.
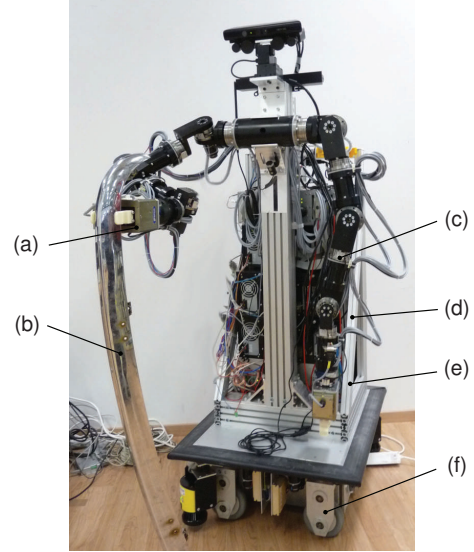

Fig. 8. The human-sized mobile robot used in the experiment.

B. Implementation

The entire manipulator control scheme is implemented in *MATLAB/Simulink* and executed on the *Linux Real-Time Application Interface (RTAI)* using *Matlab's Real-Time Workshop* on a second PC. The control algorithm runs at a frequency of 1 kHz. The third PC is identical to the first and ensures real-time control of the mobile platform and a synchronized data management utilizing the *KogmoRTDB* real-time database [18] available at [19] and the *PREEMPT_RT Linux* real-time kernel patch [20]. The *MARY Text-to-Speech System* is utilized for English-language speech synthesis.

For simplicity, the implementation of the impedance control law is reduced to the x/y-plane. The parameters from equations (4) and (3) were set to the following values:

$$\begin{aligned}
\boldsymbol{M}_r &= \mathrm{diag}\left(15\,\mathrm{kg} \quad 15\,\mathrm{kg} \quad 0.3\,\mathrm{kgm}^2 \right) \\
\boldsymbol{D}_r &= \mathrm{diag}\left(80\,\mathrm{Ns/m} \quad 80\,\mathrm{Ns/m} \quad 7\,\mathrm{Nms/rad} \right) \\
\boldsymbol{K}_p &= \mathrm{diag}\left(0 \quad 0 \quad 0 \right) \\
\boldsymbol{K}_d &= \mathrm{diag}\left(30\,\mathrm{Ns/m} \quad 30\,\mathrm{Ns/m} \quad 0.1\,\mathrm{Nms/rad} \right)
\end{aligned}$$

Note, that the zero-value for $\boldsymbol{K}_p$ leads to a drift-free behavior of the assistance controller to compensate for drift induced by the robot's odometry. The 12-dimensional input vector to the HMM training method is composed of the following dimensions in intertial coordinates:

- the 3-dimensional position of the end-effector $^0\boldsymbol{p}$
- the 3-dimensional angular velocity $^0\boldsymbol{\omega}$
- the 6-dimensional wrench $^0\boldsymbol{u}$ in inertial coordinates

As unified Gaussian computations on 6-D poses remain a computationally extensive problem [21], we decided to use angular velocities as unambiguous training input.
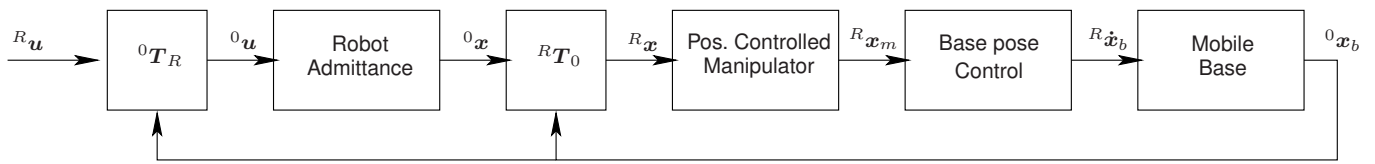
Fig. 6. Inertial admittance-type control scheme

The online learning algorithm is parameterized as follows: Each segment and each node is encoded in an HMM with 15 states and 1 Gaussian per state. In order to form a node in the primitive tree, a group of at least 2 members is needed and only the upper nodes of each branch are considered for the prediction. We use a window of 90 samples over the incoming observations for the primitive recognition and a window HMM with 30 states. The predicted data term used as reference for the impedance control law was 0.3 s.
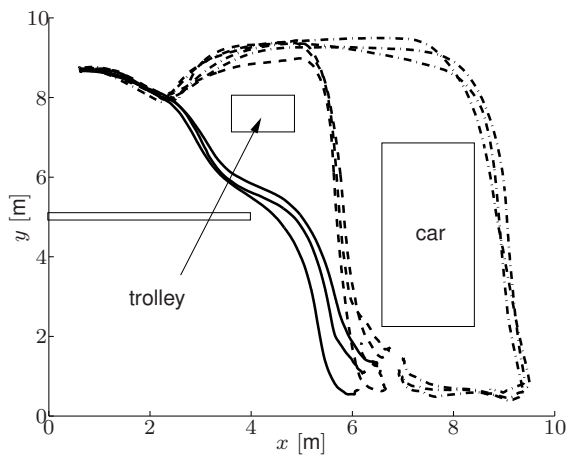
Fig. 9. 2-dimensional position component of 12-dimensional training data from three trials per each of three semantically different paths

C. Results

Fig. 9 depicts the 2-dimensional position component of the actual first nine trials taken with our system in this scenario. The human partner has chosen three semantically different paths from the starting pose to the goal pose.

Note, that the odometry drift leads to diverging paths. In Fig. 10 the 2-dimensional component of the 12-dimensional generalized output of the learning algorithm after the nine trials from Fig. 9 is shown. Additionally, the labels acquired from the human partner after successful re-detection of primitives are shown. Note, that the primitives are labeled, not the furcations. The primitive labeled *left of the car* does not terminate near the actual goal pose. The training data did not yield the necessary characteristics for successful segmentation as the human partner had difficulties to avoid collisions between the robot and the car in narrow space.

The labeled primitive graph from Fig. 11 shows the semantic map derived from the learning and labeling procedure. This graph is used and further extended in every further trial.

An effect of successful prediction is visible in Fig. 12. As a comparison of the entire trajectory from start to goal is difficult, due to the significant trial-to-trial variance, we
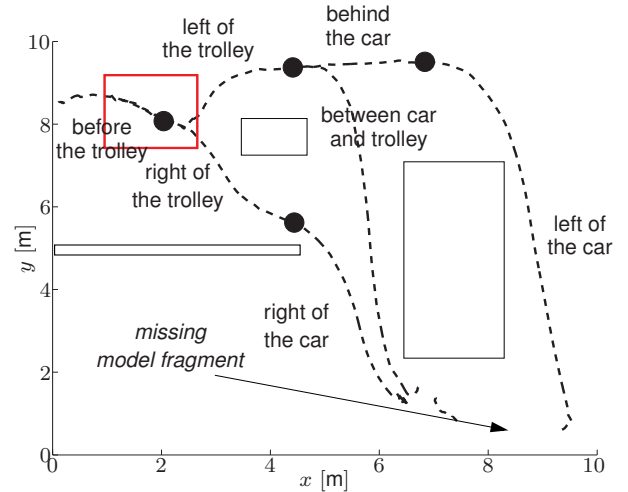

Fig. 10. 2-dimensional position component of 12-dimensional generalized output and acquired semantic labels for the graph nodes. The red box represents the furcation area examined in Fig. 12.
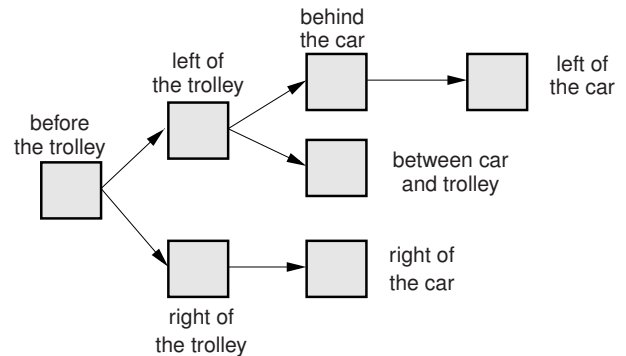

Fig. 11. Resulting primitive graph. Note, that nodes represent motion segments, not single locations.

decided to present data of the first furcation as marked in Fig. 10. The green line shows the baseline implementation: A passively following robot. Significant force is required to pull the robot into the y-direction, perpendicular to the primary direction of motion. The red line shows, how completely successful prediction due to user feedback leads to significantly reduced forces. The case of a false prediction was provoked to generate the force trajectory shown in black. However, after the positive slope along a distance of approx. $0.5\,\mathrm{m}$, the prediction is corrected and the force returns to a comparable absolute value as in the correctly predicted case.

For this proof-of-concept implementation, we consider the required exerted wrench as a suitable measure for the performance evaluation of the assistance. Fig. 13 shows the assistance improvement due to successful prediction in relation to the passive case (1. trial), and mispredicions occuring due to missing user feedback.
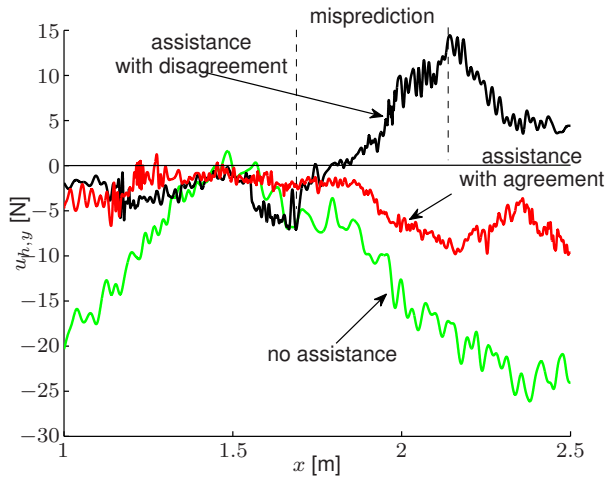
Fig. 12. Wrench component $u_{h,y}$ plotted for three different conditions of assistance. The window for position x is centered around the node transition 'before the trolley' - 'left/right of the trolley'.
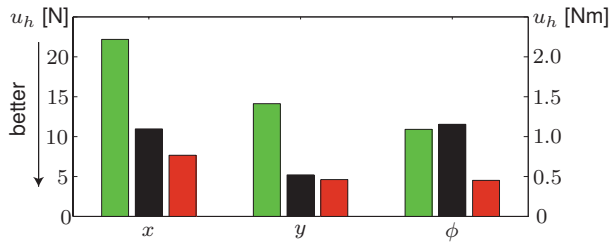

Fig. 13. Average exerted force and torque during bifurcation situations for the *no assistance*, *false* and *successful prediction* from left to right.

VIII. CONCLUSION AND FUTURE WORK

In this paper, a novel approach for a robotic system, unsupervisedly gaining knowledge on joint manipulation and acquiring semantic labels for its own experience was presented. The goal of this approach is an improvement of the robot's joint manipulation skills, exploiting semantic task knowledge. The architecture is implemented on a full-scale robot and evaluated in a classic-car restoration scenario where a human carried a car's bumper multiple times from a starting position to a final position on three semantically different ways. The results from our proof-of-concept implementation show great promise for the applicability of our approach to a wider selection of more complex tasks. Even the fundamental investigations on the combination of control and HMM-based prediction control, psychological studies on the acceptance and perceived cooperativity and extensive evaluations in simpler scenarios are necessary. The segmentation algorithm in its current implementation requires manual tuning depending on the signal dimensions involved which leaves room for improvement. Another challenging goal is a sophisticated flexible dialogue allowing for a more natural speech feedback from and to the user.

REFERENCES

[1] D. Wolpert, K. Doya, and M. Kawato, "A unifying computational framework for motor control and social interaction," *Phil. Trans. R. Soc. Lond. Series B*, no. 358, pp. 593–602, 2003.

[2] Y. Hirata and K. Kosuge, "Distributed Robot Helpers Handling a Single Object in Cooperation with Human," in *Proc. IEEE ICRA*, 2000, pp. 458–463.

[3] K. Kosuge and Y. Hirata, "Human-Robot Interaction," in *Proc. IEEE ROBIO*, 2004, pp. 8–11.

[4] M. Lawitzky, A. Mörtl, and S. Hirche, "Load Sharing in Human-Robot Cooperative Manipulation," in *Proc. IEEE Ro-Man*, 2010, pp. 185–191.

[5] B. Corteville, E. Aertbelien, H. Bruyninckx, J. D. Schutter, and H. V. Brussel, "Human-inspired Robot Assistant for Fast Point-to-point Movements," in *Proc. IEEE ICRA*, 2007, pp. 3639–3644.

[6] N. Jarrasse, J. Paik, V. Pasqui, and G. Morel, "How can human motion prediction increase transparency?" in *Proc. IEEE ICRA*, 2008, pp. 2134–2139.

[7] P. Evrard and A. Kheddar, "Homotopy Switching Model for Dyad Haptic Interaction in Physical Collaborative Tasks," in *Proc. EHS EuroHaptics*, 2009, pp. 45–50.

[8] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, pp. 1688–1703, 1985.

[9] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Auton. Robot.*, pp. 1–17, 2011.

[10] A. Bauer, D. Wollher, and M. Buss, "Information Retrieval System for Human-Robot Communication Asking for Directions," in *Proc. IEEE ICRA*, 2009, pp. 4150–4155.

[11] D. Kulic, D. Lee, C. Ott, and Y. Nakamura, "Incremental learning of full body motion primitives for humanoid robots," in *IEEE Humanoids*, 2008, pp. 326–332.

[12] J. A. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," International Computer Scienece Institute, Tech. Rep., 1998.

[13] U. Unterhinninghofen, T. Schauß, and M. Buss, "Control of a Mobile Haptic Interface," in *Proc. IEEE ICRA*, 2008, pp. 2085–2090.

[14] N. Nitzsche, U. Hanebeck, and G. Schmidt, "Design Issues of Mobile Haptic Interfaces," *J. Robotic Syst.*, vol. 20, no. 9, pp. 549–556, 2003.

[15] D. Brscic, M. Eggers, F. Rohrmüller, O. Kourakos, S. Sosnowski, D. Althoff, M. Lawitzky, A. Mörtl, M. Rambow, V. Koropouli, J. Medina, X. Zang, W. Wang, D. Wollherr, K. Kühnlenz, C. Mayer, T. Kruse, A. Kirsch, J. Blume, A. Bannat, T. Rehrl, F. Wallhoff, T. Lorenz, P. Basili, C. Lenz, T. Röder, G. Panin, W. Maier, S. Hirche, M. Buss, M. Beetz, B. Radig, A. Schubö, S. Glasauer, A. Knoll, and E. Steinbach, "Multi Joint Action in CoTeSys - setup and challenges," Technische Universität München and Ludwig-Maximilians-Universität München, Tech. Rep., 2010.

[16] U. Hanebeck, N. Saldic, and G. Schmidt, "A Modular Wheel System for Mobile Robot Applications," in *Proc. IEEE/RSJ IROS*, 1999, pp. 17–22.

[17] B. Stanczyk and M. Buss, "Development of a Telerobotic System for Exploration of Hazardous Environments," in *Proc. IEEE/RSJ IROS*, 2004, pp. 2532–2537.

[18] M. Goebl and G. Färber, "A Real-Time-capable Hard- and Software Architecture for Joint Image and Knowledge Processing in Cognitive Automobiles," in *Proc. IEEE IV*, 2007, pp. 734–740.

[19] M. Goebl. (2011) KogMo-RTDB - Real-time Database for Cognitive Automobiles. [Online]. Available: http://www.kogmo-rtdb.de

[20] The Preempt RT community. (2011) Real-Time Linux Wiki. [Online]. Available: http://rt.wiki.kernel.org

[21] W. Feiten, P. Atwal, R. Eidenberger, and T. Grundmann, "6D Pose Uncertainty in Robotic Perception," in *Proc. GWR*, 2009, pp. 1–10.