An Integrated Framework for Human–Robot Collaborative Manipulation

Weihua Sheng, *Senior Member, IEEE*, Anand Thobbi, and Ye Gu

***Abstract*—This paper presents an integrated learning framework that enables humanoid robots to perform human–robot collaborative manipulation tasks. Specifically, a table-lifting task performed jointly by a human and a humanoid robot is chosen for validation purpose. The proposed framework is split into two phases: 1) phase I—learning to grasp the table and 2) phase II—learning to perform the manipulation task. An imitation learning approach is proposed for phase I. In phase II, the behavior of the robot is controlled by a combination of two types of controllers: 1) reactive and 2) proactive. The reactive controller lets the robot take a reactive control action to make the table horizontal. The proactive controller lets the robot take proactive actions based on human motion prediction. A measure of confidence of the prediction is also generated by the motion predictor. This confidence measure determines the leader/follower behavior of the robot. Hence, the robot can autonomously switch between the behaviors during the task. Finally, the performance of the human–robot team carrying out the collaborative manipulation task is experimentally evaluated on a platform consisting of a Nao humanoid robot and a Vicon motion capture system. Results show that the proposed framework can enable the robot to carry out the collaborative manipulation task successfully.**

***Index Terms*—Humanoid robots, robot programming, imitation learning, reinforcement learning.**

I. Introduction

A. Motivation

FOR SERVICE robots to be useful, one of the fundamental abilities they should possess is to work collaboratively with humans. A common example where collaboration would be required is in a cooperative manipulation task, where a human–robot team has to manipulate an object in a coordinated way. Human–robot collaboration is a research field with a wide-range of applications and high-economic impact [1]. Technology that enables robots to work collaboratively with humans can be widely applied in the industry as well as in day-to-day scenarios. This field has seen a renewed interest in recent years because of the possibility of humanoid robots residing and working along with us in the near future.

The focus of this paper is to develop an integrated framework that can address the issues involved in a cooperative manipulation task, specifically a table-lifting task. It is a collaborative task which requires quick response and leader/follower role switching. Therefore, it can be used to evaluate our framework. An example of such human–robot table lifting is shown in Fig. 1. The robot will learn two separate skills: 1) reaching the table and 2) lifting the table with a human while keeping it horizontal. Therefore, a two-phase learning framework is proposed which employs imitation learning for the first phase and a combination of reinforcement learning and prediction-based control for the second phase.

For the first phase, the robot learns how to grab a table through trajectory-level demonstrations by a human subject. It is very crucial to choose the key variables which can characterize the demonstrated skills. In our task, we encode the human's left-hand movement in the task space. To extract the principle task space constraints from multiple demonstrations, the Gaussian mixture model (GMM) and Gaussian mixture regression (GMR) method [2] is applied.

For the second phase, the robot learns how to lift a table with a human. The strategy to be learned is dependent on the human's intention. To fulfill this task, instead of following certain generalized trajectories, the robot should be able to understand and predict the human's actions. Therefore, we formalized this problem as a high-level task. We propose a novel solution to address the problem of switching the robot's leader/follower behaviors automatically during phase II. The robot does not need to know the motion path of the object being transported beforehand. This is practically desirable, since the motion trajectory of the object may change, depending upon the environment, obstacles, or the physical limitations of the human and the robot.

B. Related Work

Imitation learning, also referred to as programming by demonstration, is a powerful learning mechanism that enables robots to acquire skills from humans quickly and conveniently. It mainly consists of three steps: 1) representation; 2) generalization; and 3) reproduction [3]. The representation phase encodes the demonstrated skill. The aim of the generalization phase is to extract the relevant characteristics of the demonstrated trajectories. At last, in the reproduction phase, the generalized trajectories are adjusted to a new situation, which are then enacted by the robot. Several frameworks have

Manuscript received March 24, 2013; revised February 28, 2014; accepted March 30, 2014. Date of publication October 31, 2014; date of current version September 14, 2015. This work was supported in part by the National Science Foundation under Grant CISE/CNS 0916864, Grant CISE/CNS MRI 0923238, Grant CISE/IIS 1231671, and Grant IIS 1427345, in part by the National Natural Science Foundation of China under Grant 61328302, and in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT1408. This paper was recommended by Associate Editor M. M. Carvalho.

The authors are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: weihua.sheng@okstate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2014.2363664

been proposed for endowing robots with imitation learning abilities [4], [5]. In this paper, we propose to use imitation learning for the humanoid robot to learn to grasp the table by observing multiple human demonstrations.

For collaborative human–robot tasks, traditional frameworks place the intellectual responsibility of planning and guiding the cooperative task entirely on the human. The robot is confined to work as a mere follower. Such robot followers are preprogrammed with simple reactive behaviors. For example, a popular approach to accomplish a cooperative object manipulation task is using impedance control [6]–[8]. However, adopting such a strategy requires the human to spend extra energy in dragging the robot, apart from the energy spent in moving the load itself. Furthermore, a goal such as keeping the table horizontal throughout the table-lifting task is difficult to achieve using this technique unless the robot can perform fast and accurate maneuvers. Predicting human behavior can help improve the task performance during collaborative tasks. Below we survey the works in this area.

1) Motion-Based Prediction: Using human motion prediction to improve human–robot collaboration has been investigated by some researchers. Maeda *et al.* [9] used a human motion prediction technique which enables the robot partner to work proactively with the human. Human motion prediction was made possible by following the assumption that the fellow human's motion satisfies the minimum jerk model in the cooperative manipulation setting [10]. Based on the estimation of the minimum jerk model parameters, the robot could predict the velocity profile of the human's motion, which could then be used to take a proactive action. This strategy was shown to reduce the human's effort for the cooperative manipulation task. Recently, we have seen a resurgence in the studies of physical human–robot interaction which make use of motion prediction strategies. Corteville *et al.* [11] presented a robot assistant which could predict the human's motion using an extended Kalman filter (EKF). The EKF was designed according to the minimum jerk model. The amount of assistance provided by the robot throughout the entire task had to be decided beforehand. Evrard and Kheddar [12] proposed a solution to change the role of the robot during the task execution using a homotopy switching model, although manually. Automatic adjustment of the homotopy variable which decides the role of the robot was left as an open question, for which the proposed work offers a solution. Another shortcoming in these approaches is the assumption that the robot should know the destination of the object being transported so that a plan of motion could be generated. If the destination is changed mid-way, a new subtask has to be generated on the fly which is nontrivial and is a separate work in itself [13]. Apart from cooperative tasks, human motion prediction has also been applied extensively in robotic teleoperation tasks [14]–[16].

Recent works show that the minimum jerk model may not be suitable for cooperative manipulation tasks [17]. The minimum jerk model assumption fails when there are large perturbations in the motion trajectory, or if the human decides to change the course of the trajectory during the task execution. In such cases, the robot might fail to comply with the human. Also, in order to apply the minimum jerk model successfully, the final position of the object must be known to both the human and the robot which is not practical. It is interesting to note that two humans can excel in a table-lifting task even if one does not know the final position of the table.

2) Intention-Based prediction: On the other hand, learning algorithms such as hidden Markov model (HMMs)/GMR, GMM/GMR have been used for intention prediction by several researchers. Wang *et al.* [18] developed a robot controller for human–robot handshaking. A position-based admittance controller is implemented. By using haptic data as inputs, an HMM-based high-level controller is used to estimate human intentions and modify the reference trajectory accordingly. Similarly, Calinon *et al.* [19] proposed a statistical model which can efficiently encapsulate typical communication patterns across different users. They encoded probabilistically the correlations between the dynamical signals (forces) and the kinematic parameters of the task in a continuous manner. Then the robot can autonomously select a controller to reproduce the collaborative skill with an appropriate behavior. Evrard *et al.* [20] presented the application of a statistical framework that allows to endow a humanoid robot with the ability to perform a collaborative manipulation task with a human operator. A set of demonstrations is performed using a bilateral coupling teleoperation setup; then the statistical model is trained in a pure leader/follower role distribution mode between the human and robot alternatively. The task is reproduced using GMR. The tasks learned above are all equipped with force and position sensors to build prediction models. With the richness of the information, the models built can efficiently distinguish the proactive/reactive behaviors. However, these supervised learning-based approaches require multiple demonstrations and complex training processes. In this paper, we get rid of the training procedure and only position information is used to verify our idea. However, we believe that with more sophisticated sensors and devices, the performance of our algorithm can be further improved.

The proposed work uses a prediction-evaluation method to estimate the confidence of prediction and uses it to adjust the behavior of the robot. Our hypothesis stems from the observation that, in a human–human team performing a collaborative task, each human constantly predicts the other's motion. Based on how well the other person conforms to his predictions, the human can decide whether to lead him or follow him. We apply the same strategy to the humanoid robot. Another way of looking at this solution is, suppose if the robot is able to predict the human's motion accurately, it means that the robot has acquired an accurate model of the human's behavior. Hence, it can start behaving as a leader and proactively take the next action based on its prediction. However, if the robot has not been able to predict the human's motion correctly, it is better for the robot to reactively comply with the human. This intuition sets the basis for adjusting the leader/follower behavior of the robot continuously and dynamically.

The rest of this paper is organized as follows. In Section II, we introduce the experimental setup and present the problem statement. Section III presents the methodology of the proposed framework. The results are presented in Section IV.

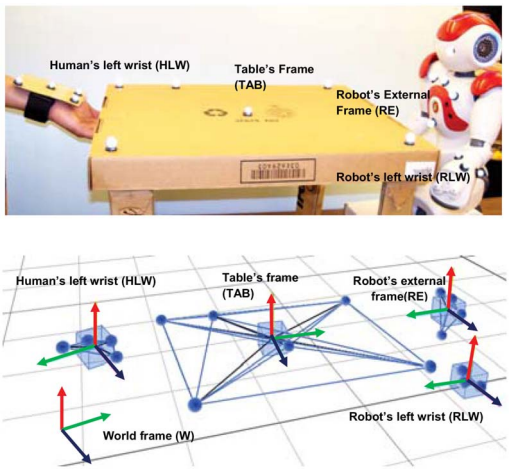


Fig. 1. Experimental setup and the various coordinate frames involved.

Section V discusses the experimental results. Section VI concludes this paper and discusses the future works.

II. Problem Statement

In this section, we first introduce the experimental setup of human–robot collaborative manipulation. Based on that, we formulate the problem to solve.

A. Experimental Setup

The table-lifting task requires a human and a humanoid robot to lift up a dummy table to a random height and bring it down. Fig. 1 shows the experimental setup. The hardware mainly consists of a motion capture system and a humanoid robot.

1) Motion Capture System: The Vicon MX motion capture system [21] is used for obtaining the positional information of the table and evaluating the performance. The frame rate of the motion capture system is 100 Hz. The motion capture system has an accuracy of 0.7 mm. Fig. 1 also shows the table with the markers. The absolute position of the table along the vertical axis is calculated by averaging the positions of the markers placed at the end points. Z_1 and Z_2 are the instantaneous 1-D coordinates of the human-end and the robot-end of the table, respectively.

2) Humanoid Robot: We use the Nao humanoid robot manufactured by Aldebaran robotics [22]. The robot has 25 degrees of freedom. It is equipped with a variety of sensors such as cameras, microphones, sonar distance sensors, inertial sensors, tactile sensors, and force sensors on the feet. An inverse kinematic procedure provided by the software development kit (SDK) is used to control the robot's end effector (hand-tip) position with respect to the previous position. Since the robot control is based only on controlling the end-effector position, no other parameters can be specified. Certain offset has to be added in order to compensate for the small but definite weight of the table. The controlling frequency is 10 Hz.

B. Problem Statement

The task we consider is to let the robot learn how to collaborate with a human subject to life up and put down a table while keeping the table horizontal. In our task, only the positional information of the table is used for characterizing the task. The motion capture system provides precise position and motion information about the table. Unlike the similar tasks in [20] fulfilled with force sensors, we only use position information to characterize the task. The focus is to present the proactive/reactive behavior switching controller. However, it is possible to apply our proposed approach to more complex tasks with sophisticated sensors. Motion of the robot hand during the lifting task is constrained to 1-D up-down motion. However, the proposed system can be easily extended to handle multiple dimensions.

TABLE I
Coordinate Frames Involved

Rigid Body	Notation
Human left wrist	HLW
Robot left wrist	RLW
Table	TAB
Robot's external frame (torso)	RE
World frame	W

In the first phase, the human demonstrates the table-reaching action multiple times, which is captured by the motion capture system. The goal is to extract the constraints from the demonstration, and map them to the robot embodiment. In the second phase, the robot observes the horizontal position of the table and decides on an appropriate action.

The motion capture system allows to create rigid body objects. Each rigid body defines its own coordinate frame having its translation and rotation with respect to the global frame. The rigid body frames defined are listed in Table I. The convention we follow in this paper are: the *x*-*y*-*z* coordinates (position) of a rigid body "A" with respect to rigid body "B" is denoted as ${}^{A}P_{B}$ and the rotation of body A with respect to body B is denoted by ${}^{A}R_{B}$. However, the end effector position of the robot is controlled with respect to a frame located somewhere inside its torso. We call this the robot's "internal" frame which is denoted by *RI*. The table's frame is used to observe the human's hand motion with respect to the table during the demonstration. The human demonstrations have to be mapped to the robot by removing the embodiment difference that exists between the human and the robot. This mapping is simplified by considering all the trajectories in the task space as opposed to the joint space.

Calibration is needed to derive the transformation for converting the trajectories in robot's external frame to the robot's internal frame. The robot's end-effector has to be controlled with respect to its internal frame of reference, while the mapped data obtained from demonstrations is the trajectory of the markers placed on the robot's hands with respect to the markers placed on the robot's torso. Hence, calibration is needed to establish a correspondence between the external marker frame with the internal robot frame. The robot's SDK can provide the position of the robot's end effector with respect to its internal frame of reference. Hence given the corresponding motion capture data and encoder data, a transformation can be derived.

III. Methodology

In this section, we discuss the detailed methodology for the proposed two-phase framework.

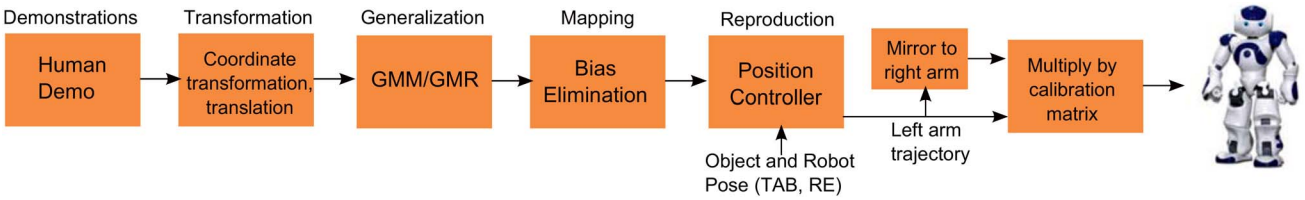


Fig. 2. Framework for phase I of the table lifting task.

A. Phase I—Imitation Learning

In the first phase, the robot learns to grasp the table by observing human demonstrations. For imitation learning, we adopt the probabilistic learning framework proposed by Calinon and Billard [2]. From the human demonstrations, the critical task constraints have to be extracted. Firstly, a GMM is used to encode the set of demonstrated trajectories (representation problem). Then, a GMR is applied to the GMM model to retrieve a smooth generalized version of these trajectories and associated variances (generalization problem) [23]. These generalized trajectories are then mapped to the robot's embodiment (human–robot mapping problem).

In a new scenario, such as with a new position and orientation of the robot and the table, a position controller is derived from the generalized trajectories. The block diagram for phase I is shown in Fig. 2. The details of the block diagram are described below.

1) Coordinate Transformation: Various coordinate transformations are required for converting the captured trajectories to the trajectories of actual interest. The obtainable trajectories from the motion capture are ${}^{\mathrm{HLW}}P_W$, ${}^{\mathrm{RE}}P_W$, and ${}^{\mathrm{TAB}}P_W$ which are with respect to the world frame. The human's wrist trajectory with respect to the table (denoted by ${}^{\mathrm{HLW}}P_{\mathrm{TAB}}$) is of interest for learning. So, we need to transform the trajectory ${}^{\mathrm{HLW}}P_W$ from the world frame to the table's frame given the pose of the table (TAB). The transformation is essentially a translation followed by a rotation. It is given by

$$ {}^{\mathrm{HLW}}P_{\mathrm{TAB}} = {}^{\mathrm{TAB}}R_W \left({}^{\mathrm{HLW}}P_W - {}^{\mathrm{TAB}}P_W \right). \tag{1} $$

2) Generalization: Let $\{\varepsilon_j\}_{j=1}^{M}$ denote the M demonstrations. Each demonstration is normalized to 100 time steps using interpolation. Each datapoint $\varepsilon_j = \{t_j, \varepsilon_j^S\}$ consists of a time step t_j and a coordinate of position ε_j^S which is a point in the trajectory of the human's left wrist with respect to the table, ${}^{\mathrm{HLW}}P_{\mathrm{TAB}}$. The data is first normalized before applying GMM/GMR. The normalization is to make each demonstration has the same data length. The normalized dataset is first modeled by a GMM which has three components and four states. These parameters are experimentally decided using a trail-and-error method. Based on the GMM, a generalized version of the trajectories is computed by applying GMR.

3) Correspondence Problem: The task constraints are derived from ${}^{\mathrm{HLW}}P_{\mathrm{TAB}}$ which are obtained from the human demonstrations. The constraints derived for ${}^{\mathrm{HLW}}P_{\mathrm{TAB}}$ have to be mapped to the robot's end effector with respect to the table denoted by ${}^{\mathrm{RLW}}P_{\mathrm{TAB}}$. All the trajectories considered for learning are in the task space as opposed to joint space. Hence, inherently the embodiment mapping problem is simplified. The problem is further simplified by mapping only the positional constraints. Hence, only the bias difference shown in Fig. 3 between the human's wrist and the robot's end effector has to be taken care of. A method is proposed to calculate this dimension difference. The human and robot grasp a fixed object in space. The coordinates with respect to the fixed object are obtained. The difference between these two coordinates is the bias compensation. Hence, a human to robot mapping can be achieved.

Fig. 3. Embodiment difference between the human's hand and the robot's hand.

4) Reproduction: In the reproduction phase, a new trajectory for the robot's end effector ${}^{\mathrm{RLW}}P_{\mathrm{RE}}$ has to be produced based on the generalized version of ${}^{\mathrm{RLW}}P_{\mathrm{TAB}}$. Given the pose of the table (TAB) during reproduction phase, ${}^{\mathrm{RLW}}P_{\mathrm{RE}}$ can be derived as follows:

We have ${}^{\mathrm{RLW}}P_{\mathrm{TAB}}$ which is

$$ {}^{\mathrm{RLW}}P_{\mathrm{TAB}} = {}^{\mathrm{TAB}}R_W \left({}^{\mathrm{RLW}}P_W - {}^{\mathrm{TAB}}P_W \right). \tag{2} $$

This transformation is achieved by first translating and then rotating RLW to the TAB frame of reference.

${}^{\mathrm{RLW}}P_W$ can be obtained as

$$ {}^{\mathrm{RLW}}P_W = {}^{\mathrm{TAB}}R_W{}^{-1}\ {}^{\mathrm{RLW}}P_{\mathrm{TAB}} + {}^{\mathrm{TAB}}P_W. \tag{3} $$

Finally, we can derive ${}^{\mathrm{RLW}}P_{\mathrm{RE}}$ as

$$ {}^{\mathrm{RLW}}P_{\mathrm{RE}} = {}^{\mathrm{RE}}R_W \left({}^{\mathrm{RLW}}P_W - {}^{\mathrm{RE}}P_W \right). \tag{4} $$

${}^{\mathrm{RLW}}P_{\mathrm{RE}}$ is then converted to the robot's internal control frame using the homogeneous transformation obtained by calibration.

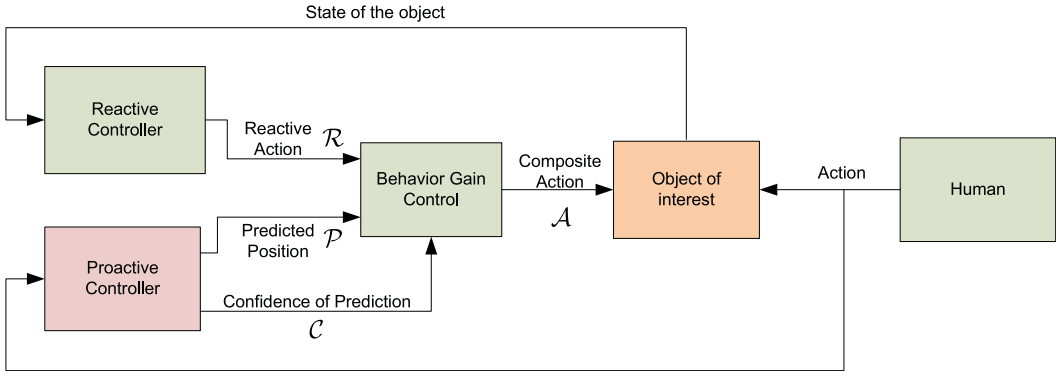


Fig. 4. Framework for phase II of the table lifting task.

5) Mirroring the Trajectory: In the imitation learning phase, only left hand demonstrations are provided to the robot. This trajectory is mirrored across the vertical robot axis to obtain the corresponding trajectory for the robot's right end effector.

B. Phase II—Collaboration

Once the robot holds the table successfully, it switches to phase II. The phase II framework is shown in Fig. 4, which consists of the reactive controller, the proactive controller, and the behavior gain control blocks. As the name suggests, the reactive controller generates a reactive robot behavior based on the current state of the environment. The proactive controller consists of a Kalman filter (KF)-based human motion predictor and an evaluation-based confidence generator. Based on the observed human actions, the predictor estimates the position of the human in the next time-step, which decides the robot's proactive action. Additionally, it generates the confidence of prediction, which is the key in adjusting the behavior of the robot. Based on the confidence value, the behavior gain control block mixes the reactive and proactive actions to generate a composite action which is taken by the robot. According to our hypothesis, the weight allotted by the gain control block to the proactive behavior varies directly according to the confidence value.

1) Reactive Controller: The reactive controller generates a reactive response by the robot to the observed state of the object. In the table-lifting task, the input of this controller is the difference between human-end and robot-end positions of the table which is encoded into the corresponding state. The output is a suitable action to perform so that a certain objective is achieved. For our experiments, the objective is to keep the table horizontal throughout the task. This can be accomplished using any generic feedback controller. However, we chose to use a controller learned from reinforcement learning for the following reasons.

a) It is possible to learn a good controller in a short time with limited learning space.
b) It compensates for the time needed to manually tune the parameters of a feedback controller.
c) The task is simple in the current experiment. However, in the future, we will consider complex tasks such as keeping a bowl in the center of the table while performing the table lifting task. Complex tasks like these have a long term reward to maintain for which reinforcement learning is most suited.

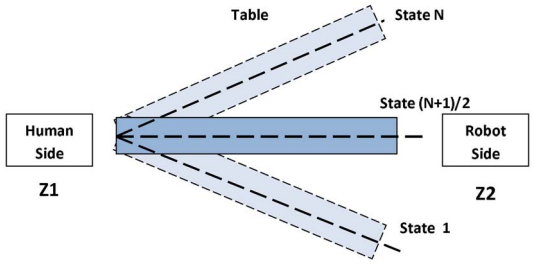


Fig. 5. State representation for reinforcement learning.

Using reinforcement learning, an agent can learn behaviors through trial-and-error by interacting with a dynamic environment. Outcome of the performed action is used as a reinforcement for updating the agent's state-action policy [24]. In this paper, the Q-learning algorithm with a guided exploration strategy is used to learn the optimal state-action policy [25]. The Q-table update equation is given by

$$\Delta Q(s_t, a_t) = \alpha \left[r + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \quad (5)$$

where r is the reward, α is the learning rate, and γ is the discount factor. For the task at hand, γ does not play a significant role, since there is no sense of a long term reward. The state of the environment is determined by the incline of the table at the given moment which is obtained from the motion capture system. Incline of the table is quantized into a number of discrete states. Fig. 5 shows a state space consisting of N states. The action space consists of a predetermined set of commands which move the robot's hand-tip up or down by specified distances. The robot has to undergo an online learning phase to learn the Q-table. During this phase, it is assumed that the human hands remain comfortably stationary. To speed up the learning phase we use a guided reinforcement learning algorithm based on counting the number of state-action visits.

Essentially, the action selection for exploration is done on the basis of the number of visits to the particular state-action

Algorithm 1 Guided Reinforcement Learning

```
Initialize Visit(s_i, a_i) = 0 ∀i ∈ N
Initialize Q-table Q(s_i, a_i) = 0 ∀i ∈ N
while Learning phase do
    t = timestep
    s_t = getState()
    Select a_t ← argmin(a)[Visit(s_t, a)]
    Take action a_t
    Visit(s_t, a_t) ← Visit(s_t, a_t) + 1
    r = getReward()
    Update Q(s_t, a_t) using (5)
end while
```

pair, instead of random action selection as in ϵ-greedy algorithms [26]. The reinforcement learning algorithm is shown in Algorithm 1.

2) Proactive Controller: The proactive controller is the most important block of the proposed system. The input is the displacement of the human-end position of the table with respect to the robot's end position of the table. The output of the proactive controller is the same as that of the reactive controller. The role of the proactive controller is to keep track of actions performed by the human and generate a prediction of the human's position in the next time-step, along-with a confidence measure for the prediction. For this purpose, a KF is used. The state is defined as

$$x_k = \begin{pmatrix} s_k \\ v_k \\ a_k \end{pmatrix}. \tag{6}$$

The measurement model is given by

$$z_k = s_k + n \tag{7}$$

where s_k is the displacement of the human's end of the table (equivalently his hand-tip), v_k is the velocity, a_k is the acceleration, and $n \sim N(0, R)$ is the measurement noise, all at the instant k.

We assume that the acceleration of the human hand changes slowly throughout the motion since humans naturally try to minimize jerk. Note that this is not the same as using the minimum jerk model.

Hence, the state update equation can be written as

$$x_{k+1} = \begin{pmatrix} s_k + v_k t + \frac{1}{2} a_k t^2 \\ v_k + a_k t \\ a_k \end{pmatrix} + w \tag{8}$$

where $w \sim N(0, Q)$ is the process noise.

Based on the state estimate $\hat{x}_k$, the human's position at the next time-step can be predicted as

$$\hat{s}_{k+1} = \hat{s}_k + \hat{v}_k t + \frac{1}{2} \hat{a}_k t^2. \tag{9}$$

The variance of the measurement noise (R) is initialized to 0.7 which corresponds to the uncertainty in measurement obtained by the Vicon system. Using this KF, it is possible to get reasonably accurate predictions of the human's motion.

To obtain the confidence of prediction, we are inspired by Simon *et al.* [27], wherein they proposed a technique to derive a confidence measure based on the statistical properties of the residuals between the predicted measurements and the observed measurements. In our technique, the KF provides a state estimate and an associated covariance matrix. Firstly, we marginalize the covariance matrix to include only the 1-D variance associated with the position prediction, say ρ. Let the predicted position be $\hat{s}_k$. Then, we evaluate the likelihood of the observed measurement, z_k using an unnormalized Gaussian distribution given by

$$\mathcal{L}_k = \exp\left(-\frac{\left(z_k - \hat{s}_k\right)^2}{2\rho^2} \right). \tag{10}$$

We choose an unnormalized Gaussian distribution to make $0 < \mathcal{L} \leq 1$. It can be seen that $\mathcal{L}$ would give us a direct measure of confidence about the prediction based on the evaluation of the previous prediction against the true measurement. However, considering only the last step measurement error is not sufficient. For the confidence measure, we introduce a function given by

$$\mathcal{C}_{k+1} = \frac{\mathcal{L}_k + \phi \mathcal{L}_{k-1} + \cdots + \phi^{k-1} \mathcal{L}_1}{1 + \phi + \cdots + \phi^{k-1}}. \tag{11}$$

The subscripts denote the time-steps at which they were obtained. Hence, $\mathcal{C}_{k+1}$ is the confidence of prediction for the next time-step, that considers all the likelihoods observed previously, weighted by the forgetting factor ϕ, where $0 < \phi \leq 1$. This function can be implemented recursively. Also, it can be seen that the denominator is for normalization.

3) Behavior Gain Control: At a given time step k, let the reactive controller suggest a next-step action $\mathcal{R}_{k+1}$ and the proactive controller suggest a next-step action $\mathcal{P}_{k+1}$. Let the confidence of this prediction be $\mathcal{C}_{k+1}$. The gain control block combines these together to form a composite action $\mathcal{A}_{k+1}$ given by

$$\mathcal{A}_{k+1} = \mathcal{C}_{k+1} \mathcal{P}_{k+1} + (1 - \mathcal{C}_{k+1}) \mathcal{R}_{k+1}. \tag{12}$$

This action is taken by the robot at time step $k+1$. The inspiration for this form has been taken from Evrard and Kheddar [12]. Note that because $0 < \mathcal{C} \leq 1$, the robot does not act as a pure leader or pure follower, but has characteristics of both in different amounts.

If the confidence of prediction $\mathcal{C}_{k+1}$ is high, a larger weight is allotted to the proactive action. Hence, the robot's action has leader-like characteristics. If the robot is not very confident about the prediction, larger weight is allotted to the reactive behavior and the robot's action seems follower-like. Since the system works in real time, the change of behavior is dynamic and automatic.

IV. Experiments and Results

In this section, we present the details of the experiments designed to evaluate the proposed framework along with the obtained results.

A. Calibration

The calibration matrix is obtained by moving the robot arm randomly covering as many configurations as possible, during

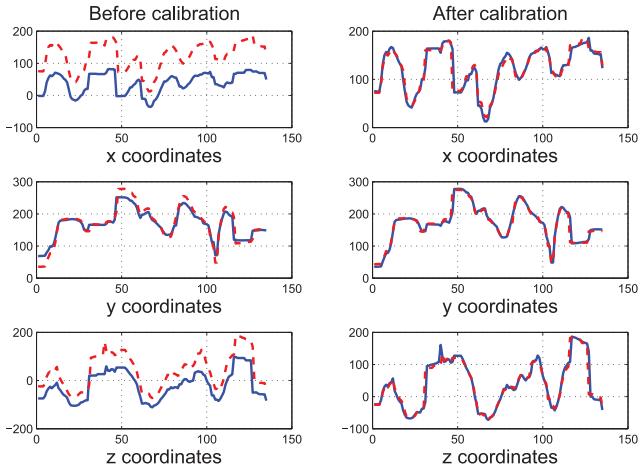


Fig. 6. Data comparison before and after calibration. Solid lines represent data in the Vicon's frame and the dashed lines represent data in the robot's internal frame.

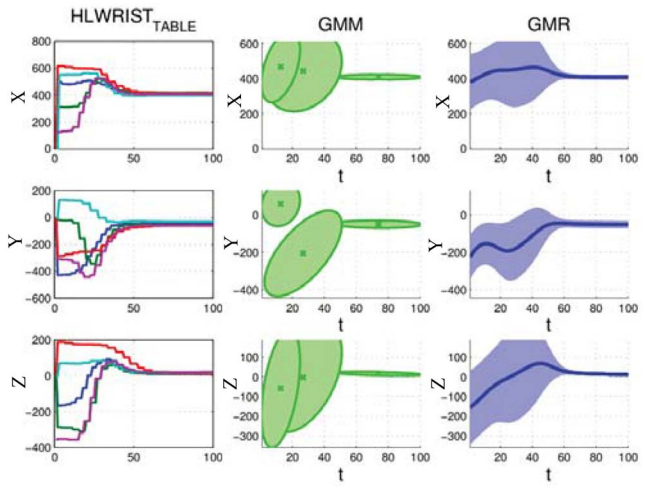


Fig. 7. Trajectory encoding and generalization. The figures on the left column represent the trajectories of each dimension of each demonstration. The figures in the middle column show the GMMs to model the trajectories of each dimension. The figures on the right column show the generalized trajectories of each dimension and the corresponding variances.

which the coordinates of the robot's left arm with respect to its own torso are collected both in the motion capture system frame and the robot's internal frame. Then, the homogeneous transformation is used to calculate the calibration matrix. The homogenous matrix basically gives a transformation for the robot hand motion from the motion capture system to fit the data obtained from the internal encoder data. Fig. 6 shows the coordinates in the two frames before and after calibration. It can be observed from the figure that by using the calibration matrix the trajectories can be successfully converted from the motion capture frame to the robots internal frame.

B. Imitation Learning

In the imitation learning phase, multiple demonstrations are performed by the human. In each demonstration, the human tries to approach the same position of the table with his left hand from an arbitrary initial position. We implemented the imitation learning algorithms using some open source code for GMM/GMR [28]. Fig. 7 shows the results for trajectory encoding and generalization. The parameters are chosen using a heuristic method. Generalized trajectories and constraints are thus obtained. From the trajectory encoding and generalization results, we can see that the variance of the trajectory is decreasing as the human hand approaches the table, which indicates that the final positions of the human's hand with respect to the table should be consistent and satisfy this positional constraint. After compensating for the bias difference between human's hand and robot's hand, the robot can generate its own trajectory given the extracted constraints. Given a new position of the table with respect to the robot, a new trajectory is reproduced by the position controller. The calibration matrix is then used to convert the trajectories from the robot's external frame to the robot's internal frame.

In the imitation learning phase, demonstrations of grasping the table with only the left hand are provided to the robot. This trajectory is mirrored to obtain the corresponding trajectory of robot's right end effector. The results are shown in Figs. 8 and 9. The former is the robot replaying the generalized trajectories extracted from the demonstrations and the latter is the robot reproducing the trajectories in a new situation (different position of the table).

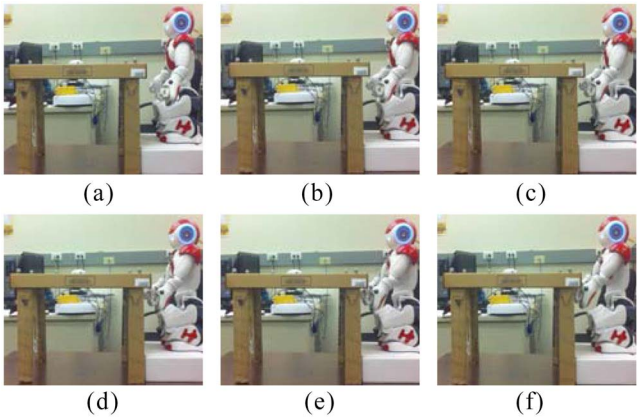


Fig. 8. (a)–(f) Replaying the generalized trajectory.

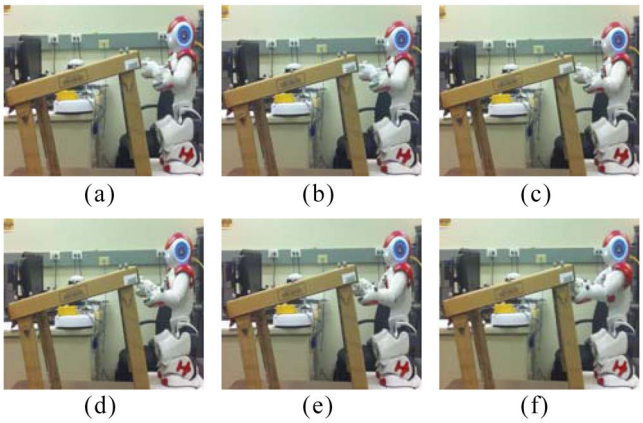


Fig. 9. (a)–(f) Reproducing the generalized trajectory in an unknown position.

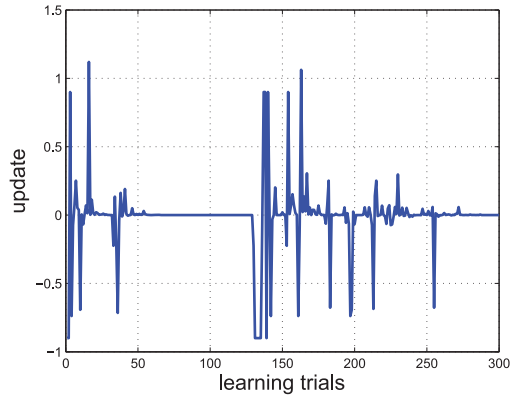


Fig. 10. Random exploration learning performance.

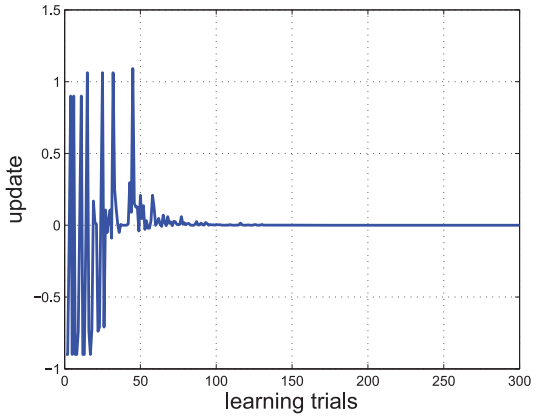


Fig. 11. Guided exploration learning performance.

C. Reinforcement Learning

For Q-learning, a state-action space consisting of five states and five actions was chosen. The reward r was decided as

$$r = (|Z_2 - Z_1|)_k - (|Z_2 - Z_1|)_{k+1} \tag{13}$$

where Z_1 and Z_2 represent the position of the human-end and the robot-end of the table, respectively.

Hence, if the slant of the table is decreased, the robot receives a positive reward. The action set consists of actions $\{+2, +1, 0, -1, -2\}$, which correspond to the direction and magnitude of the robot's motion by a defined position step. The position step was set to be 2 cm, since it is the smallest precise movement that can be performed by the robot's arm. Values of the reinforcement learning parameter used were: learning rate $\alpha = 0.9$ and discount factor $\gamma = 0.2$.

For state-action space consisting of five states and five actions, the performance of random exploration is compared with that of guided exploration. A simulator mimicking the real situation was set up to test the performance of the reinforcement learning algorithms initially. For training, in each simulation experiment, the number of iterations is fixed to 300. To test the speed of convergence, the experiment was performed 100 times. Figs. 10 and 11 show the speed of convergence for these two algorithms respectively for a single experiment. The figures show the Q-update for each trial as the learning progresses. It is observed that the guided exploration policy converges much faster and is more stable than the random exploration policy. On average, the random exploration took more than 100 trials to reach an optimal policy whereas the guided learning algorithm could reach the optimal policy within 50 trials.

To test the guided exploration strategy in the real-world scenario, ten trials were performed to test how quickly the algorithm could converge to an optimal policy. Median value for the number of iterations to converge was 36. The longest episode took 62 iterations before it could converge. Each learning trial took about 5 min to complete. Some of the snapshots of the table lifting task can be seen in Fig. 12.

D. Prediction Results

The previously described KF is used for predicting the human motion one time-step ahead. Each time step is typically 100 ms, which is the minimum time required for the robot's arm to move from one position to another. Fig. 13 shows the predicted and observed values of position, velocity, and acceleration.

The predicted position is calculated from (9). True velocity and acceleration are derived from the observed position, and are shown in the figure for comparison with the predicted values. It can also be observed from Fig. 13 that the predictions are inaccurate during the initial steps of the motion. After about 1-time steps the estimates improve. It can be seen that the difference between the predicted velocity and the calculated velocity is very small. Considering the noise caused by the calculation of the second derivative of the position, the change of the acceleration is reasonably small, which partly justifies our assumption that the acceleration remains constant.

Fig. 14 shows the role of the forgetting factor ϕ in determining the confidence ($\mathcal{C}$). Since it is not possible to reproduce the exact same trajectory during the task, the confidence trajectories shown in Fig. 14 are computed offline step-by-step using data collected from a table-lifting task. As seen from (11), a low value of ϕ means that the predictor allots a small weight to older likelihood estimates. Thus, $\mathcal{C}$ mostly depends upon the recently observed $\mathcal{L}$. Hence, if the likelihood values $\mathcal{L}$ changes quickly, it causes $\mathcal{C}$ to fluctuate heavily. Using a similar reasoning, a large value of ϕ causes the confidence measure to settle very slowly. Hence, the robot cannot adapt to the motion changes quickly and generates high-confidence values even for incorrect predictions. A good value for ϕ which gives a good tradeoff between smooth variation and adaptability for $\mathcal{C}$ was found to be 0.45.

Fig. 15 shows how the confidence of the prediction varies throughout the task, along with the position predictions and observations. It can be seen that initially, when the task has not begun and the table is still, the predictor accurately estimates the motion to be zero which causes the high-confidence value at the beginning. Once the trial starts, in the initial steps, the predictions are inaccurate because of the drastic change in the motion model. This causes the confidence value to drop down suddenly. The reactive controller of the robot becomes dominant in this region. As the predictor gains knowledge about the motion, the predictions keep improving. As the predictions

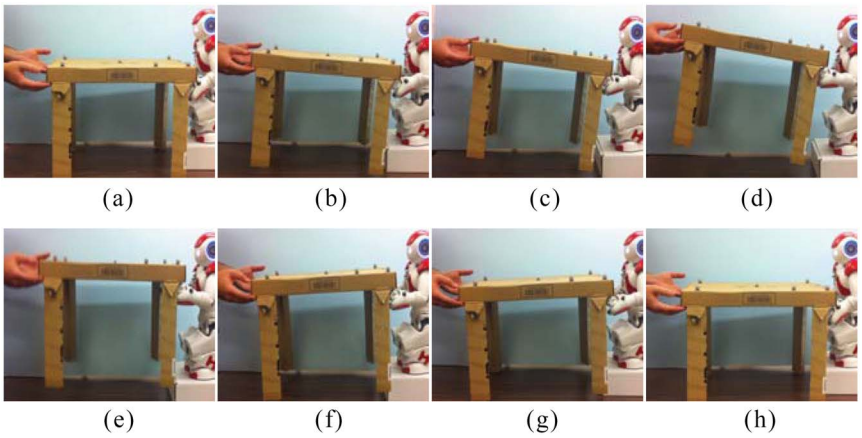


Fig. 12. Snapshots of human–robot team performing the table lifting task. (a)–(d) Lifting the table up. (e)–(h) Putting the table down.

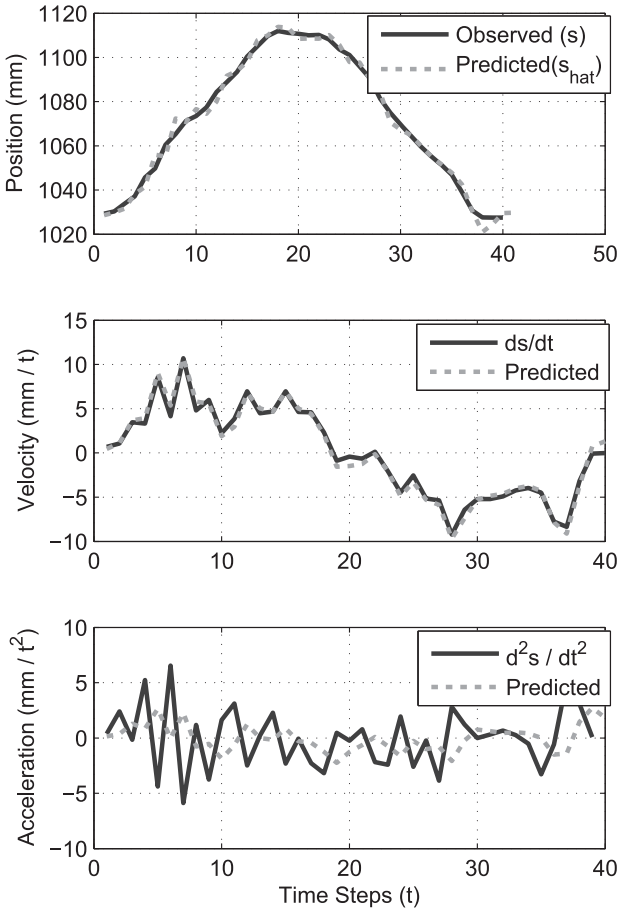


Fig. 13. Predictions obtained from KF (each time step is 100 ms).

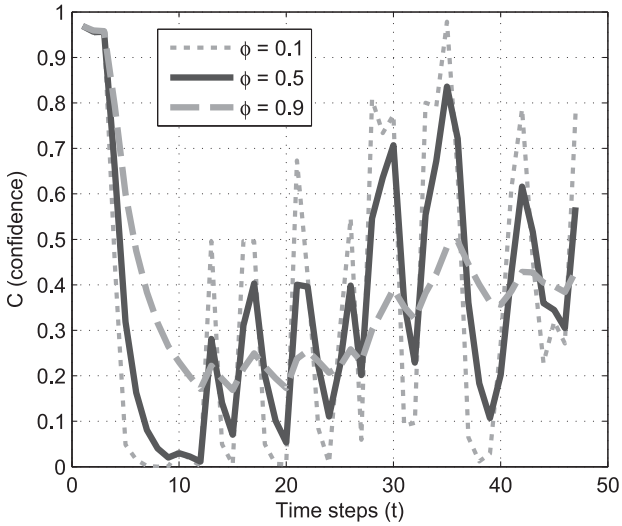


Fig. 14. Effect of the forgetting factor on confidence (each time step is 100 ms).

improve, the confidence values also improve. As a result, the proactive behavior becomes more dominant.

E. Handling Irregular Cases

One of the major improvements our system offers over most existing systems, is that, no assumption has been made regarding the trajectory of the entire motion. The human has the right to change the trajectory at any point of time, during the trial. Fig. 16 shows a case where the motion of the human is not typical. Instead of lifting up the table and putting it down continuously, the human chooses to take a pause while lifting the table up. Because of this, an abrupt change of motion can be seen around time-step 15. The confidence value drops to zero in 3–4 time steps. During this phase, the robot starts behaving as the follower and simply tries to make the table horizontal using the reactive controller. As the human continues to keep still, the predictor learns this model and predicts zero movement. Hence, although the confidence is high and the robot is the leader, there is no proactive action since the predicted change in position is zero. Again at time-step 35, the human starts moving the table upward. Again, the robot switches from leader to follower based on the confidence value. Once the motion has been stabilized the robot maintains a confidence value centered somewhere around 0.5.

F. Overall System Performance

In this experiment, we evaluate the improvement offered by our system for the table-lifting task. If Z_{1t} is the position of human side of the table and Z_{2t} is the position of robot side at any instant t, then the objective is to minimize the absolute

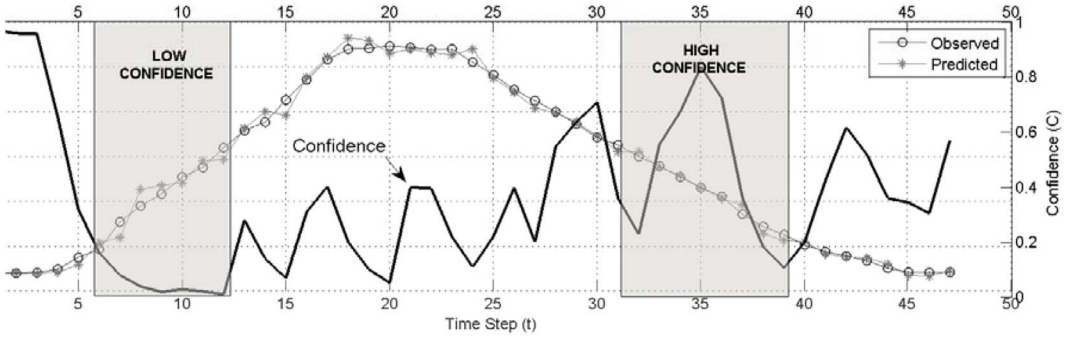


Fig. 15. Confidence value with predictions. The human subject smoothly lifts the table up and down with the robot (each time step is 100 ms).

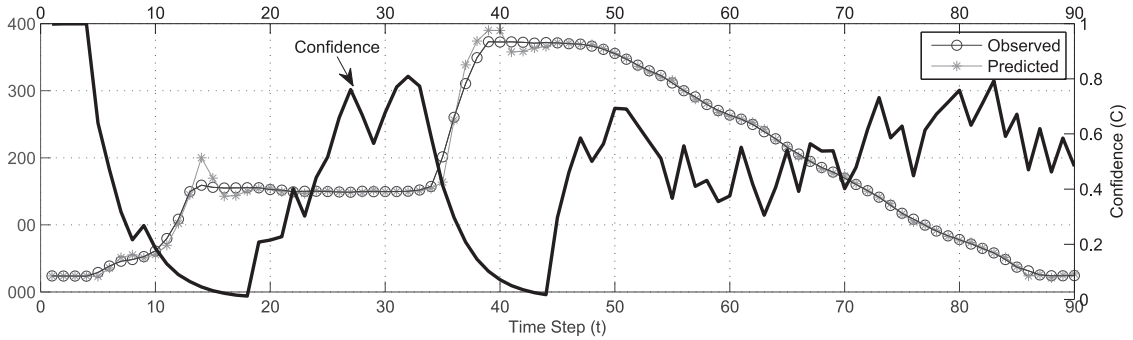


Fig. 16. Irregular case. The human subject pauses for a while during lifting the table up, then puts the table down smoothly with the robot (each time step is 100 ms).

error given by

$$\text{AbsoluteError} = \sum_t |Z_{1t} - Z_{2t}|. \tag{14}$$

We use the motion capture system to record the trajectories of the human and robot table ends. Fig. 17 shows these trajectories for cases where the proposed system was used (case I: with predictions) and the case where only the reactive controller was used (case II: without predictions) The figure also shows the absolute error calculated for the two cases. We use the root mean square error (RMSE) to characterize the performance.

The following observations can be made from Fig. 17.

1) The RMSE for case I is less than the RMSE for case II.
2) The motion observed for case I is smoother than that of case II.
3) The absolute error is lower in case I.

Quantitative results are provided in Table II for multiple users. Five human subjects were asked to participate in the table-lifting task with the robot, one at a time. Each person was asked to lift up the table to a random height and bring it down for ten trials. Totally, for both cases, 100 trials were acquired. Table II shows the average RMSE for the ten trials observed for each subject, for each case. It can be seen that, for all the users, the RMSE is lower when the proposed approach is used as opposed to a simple reactive approach. Hence, a definite improvement can be observed.

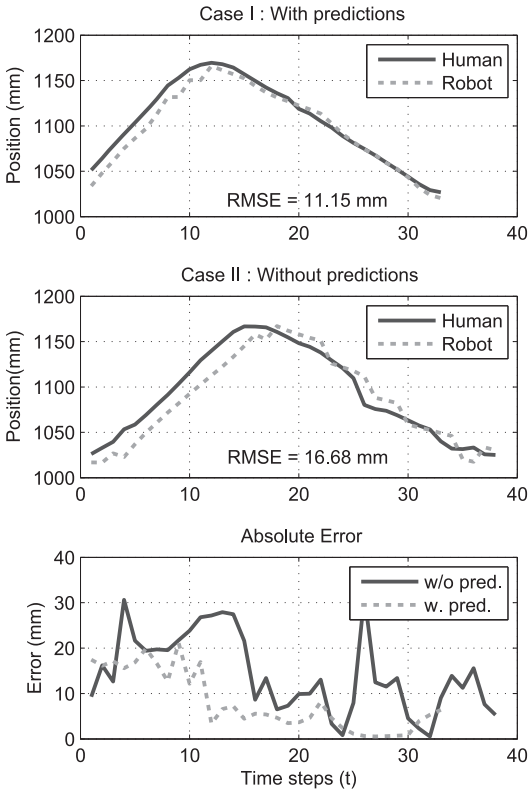


Fig. 17. Performance comparison between with and without predictions (each time step is 100 ms).

V. Discussion

To see how well the human–robot team mimics real-human collaboration, we conducted an experiment on human–human table lifting. As shown in Fig. 18, the RMSE for the human–human case is about 6.53 mm. We also observed that the motion of the robot is jerky when its reactive behavior is

TABLE II
AVERAGE RMSE

Subject	Case 1 Avg. RMSE w/o Prediction (mm)	Case 2 Avg. RMSE with Prediction (mm)
1	19.139	12.967
2	23.567	16.591
3	24.872	18.418
4	20.085	15.391
5	22.432	17.684

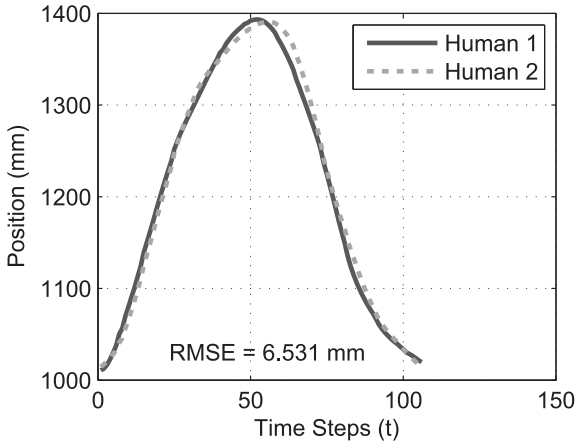


Fig. 18. Human–human team lifting the table. First lifting the table up then bringing the table down (each time step is 100 ms).

dominant. This is mainly because of the fixed step sizes. The design of our system is such that the prediction accuracy influences the confidence of prediction. Because of this, many interesting possibilities follow. Better predictions result in better confidence values which allows for proactive robot behavior. Hence, if the human keeps moving smoothly as the robot expects him to move, the motion of the robot is also smooth. This in turn causes smoother motion of the table as a whole and hence smoother motion of the human, thus resulting in better predictions. However, if the motion of the human is jerky, then the robot is unable to estimate the motion accurately, and hence does not allow for leader behavior. The predictions are not fully utilized in such cases and reflects poor performance. Thus, the results are not only influenced by the robot's performance alone, but also by the human performance. Especially, subject 1 had been working with the system for a longer time than others. Hence, the results for subject 1 were better compared to other human subjects.

In Fig. 17, we could observe in case I, the trajectory is much smoother when the human is placing the table down as compared to moving upward. This is because, inherently, the robot motion while lifting the table against gravity is jerky due to the internal control characteristics (We can only specify the speed while the acceleration is not under our control). This induces some jerks in the human motion also since they are coupled through the table. Because of this, the prediction suffers, which causes lower confidence levels. But while moving downward, the robot is able to move very smoothly which allows the human to move smoothly and hence the system is utilized to its full potential resulting in better performance. It can also be speculated that sophisticated velocity or torque controlled robots would yield smoother motions and offer better improvements in performance using the proposed technique.

Due to the limitation in the control speed of robot, we could obtain at most ten motion capture samples per second. With a faster robot, more samples could be obtained per second which would improve the quality of predictions.

VI. CONCLUSION

This paper contributes an integrated framework for conducting human–robot collaborative manipulation tasks. We developed a two-phase learning framework, which combines imitation learning and reinforcement learning. Using imitation learning the robot could reach out and hold the end of the table. Through reinforcement learning, the robot can learn to collaborate with human for the table-lifting task. With the guided exploration strategy for Q-learning, the learning speed is improved. Using the entire framework, the robot could learn to perform the collaborative table-lifting task quickly and successfully. An extension to the basic reactive control to reactive/proactive control is also proposed in phase II. It utilizes human motion prediction to adjust the leader/follower behavior of the collaborating robot. The proactive controller is based on Kalman filtering for human-motion prediction. A novel technique to derive a measure of confidence of the prediction has also been proposed. Experimental results were presented to provide conclusive evidence that the proposed approach offers a definite improvement over simple reactive approaches. Additionally, the system does not make any assumptions about the motion trajectory of the object which is practically desirable. This framework can be extended to other human–robot collaborative tasks which involve leader/follower behavior switching, such as collaboratively moving an object from one place to another.

For future work, we propose to utilize longer-term predictions. Various techniques exist that can provide longer-term predictions about the final time and position of the human movement. However, these techniques rely heavily on the success of the minimum jerk model. A combination of such techniques with the proposed framework should provide a better solution. Apart from the long-term prediction part, a general case where the human action does not necessarily translate directly to robot action can be considered. Complex objectives in the cooperative task could also be added. For example, we will study how to scale the proposed framework to handle multiple tasks, such as lifting up a table while balancing something on it. Finally, our framework can also be easily extended to proactive teleoperation. The teleoperated robot can choose to take a proactive action based on the confidence values which could reduce the effect of time delays observed in teleoperation and increase transparency.

REFERENCES

[1] M. B. A. Bauer and D. Wollherr, "Human-robot collaboration: A survey," *Int. J. Humanoid Robo.*, vol. 5, no. 1, pp. 47–66, Mar. 2008.

[2] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *Proc. IEEE/RSJ Int. Conf. Robot. Autom.*, Nice, France, 2008, pp. 367–372.
[3] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robot. Autom. Mag.*, vol. 17, no. 2, pp. 44–54, Jun. 2010.
[4] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robot. Auton. Syst.*, vol. 47, nos. 2–3, pp. 109–116, 2004.
[5] D. C. Bentivegna, C. G. Atkeson, and G. Cheng, "Learning tasks from observation and practice," *Robot. Auton. Syst.*, vol. 47, nos. 2–3, pp. 163–169, 2004.
[6] N. Hogan, "Impedance control: An approach to manipulation," in *Proc. Amer. Control Conf.*, San Diego, CA, USA, 1984, pp. 304–313.
[7] M. Rahman, R. Ikeura, and K. Mizutani, "Investigating the impedance characteristic of human arm for development of robots to co-operate with human operators," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 2. Tokyo, Japan, 1999, pp. 676–681.
[8] K. Kosuge, M. Sato, and N. Kazamura, "Mobile robot helper," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1. San Francisco, CA, USA, 2000, pp. 583–588.
[9] Y. Maeda, T. Hara, and T. Arai, "Human-robot cooperative manipulation with motion estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 4. Maui, HI, USA, 2001, pp. 2240–2245.
[10] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, Jul. 1985.
[11] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel, "Human-inspired robot assistant for fast point-to-point movements," in *Proc. IEEE Int. Conf. Robot. Autom.*, Roma, Italy, 2007, pp. 3639–3644.
[12] P. Evrard and A. Kheddar, "Homotopy switching model for dyad haptic interaction in physical collaborative tasks," in *Proc. EuroHaptics Conf. Symp. Haptic Interf. Virtual Environ. Teleoperator Syst.*, Salt Lake City, UT, USA, 2009, pp. 45–50.
[13] C. Smith and P. Jensfelt, "A predictor for operator input for time-delayed teleoperation," *Mechatronics*, vol. 20, no. 7, pp. 778–786, 2010.
[14] C. Passenberg, A. Peer, and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics*, vol. 20, no. 7, pp. 787–801, 2010.
[15] N. Jarrasse, J. Paik, V. Pasqui, and G. Morel, "How can human motion prediction increase transparency?" in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, May 2008, pp. 2134–2139.
[16] S. Clarke, G. Schillhuber, M. Zaeh, and H. Ulbrich, "Prediction-based methods for teleoperation across delayed networks," *Multimedia Syst.*, vol. 13, pp. 253–261, Jan. 2008.
[17] S. Miossec and A. Kheddar, "Human motion in cooperative tasks: Moving object case study," in *Proc. IEEE Int. Conf. Robot. Autom.*, Bangkok, Thailand, 2009, pp. 1509–1514.
[18] Z. Wang, A. Peer, and M. Buss, "An HMM approach to realistic haptic human-robot interaction," in *Proc. World Haptics 3rd Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environ. Teleoperator Syst.*, Salt Lake City, UT, USA, Mar. 2009, pp. 374–379.
[19] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Munich, Germany, Jun. 2009, pp. 1–6.
[20] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: Object-lifting case study with a humanoid," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Paris, France, Dec. 2009, pp. 399–404.
[21] (Oct. 2014). *Vicon Mx Motion Capture System*. [Online]. Available: http://www.vicon.com
[22] (Oct. 2014). *Aldebaran Humanoid Robot*. [Online]. Available: http://www.aldebaran- robotics.com/en
[23] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 129–145, Jan. 1996.
[24] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.
[25] B.-N. Wang, Y. Gao, Z.-Q. Chen, J.-Y. Xie, and S.-F. Chen, "A two-layered multi-agent reinforcement learning model and algorithm," *J. Network Comput. Appl.*, vol. 30, no. 4, pp. 1366–1376, 2007.
[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[27] D. Simon, D. L. Simon, and D. Viassolo, "Kalman filter constraint switching for Turbofan engine health estimation. Discussion," *Eur. J. Control*, vol. 12, no. 3, pp. 331–345, 2006.
[28] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. Lausanne, Switzerland: EPFL/CRC Press, 2009.

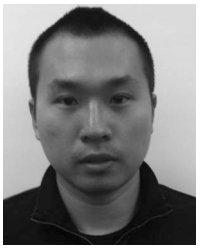
Weihua Sheng (M'02–SM'08) received the B.S. and M.S degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2002.

He is an Associate Professor at the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, USA. From 1997 to 1998, he was a Research Engineer at the Research and Development Center, Huawei Technologies, Shenzhen, China. From 2002 to 2006, he was with the Electrical and Computer Engineering Department, Kettering University (formerly General Motor Institute), Flint, MI, USA. His current research interests include wearable computing, mobile robotics, human-robot interaction, and intelligent transportation systems. He has authored over 130 papers in major journals and international conferences and holds one U.S. patent. He has participated in organizing various IEEE international conferences and workshops in the area of intelligent robots and systems.

Dr. Sheng was the recipient of six Best Paper Awards in international conferences. His research was supported by National Science Foundation, Department of Defense, Defense Department's EPSCoR program, and Department of Transportation. He is currently an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.

Anand Thobbi received the B.E. degree in electronics and telecommunications from the University of Pune, Pune, India, and the M.S. degree in electrical and computer engineering from Oklahoma State University, Stillwater, OK, USA, in 2009 and 2011, respectively.

He is a Software Engineer at the International Electronic Machines Corporation, Troy, NY, USA.



Ye Gu received the B.E. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, and the M.S. degree in mechanical engineering from the University of Minnesota, Duluth, Duluth, MN, USA, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree from the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, USA.