Proceeding of the IEEE
International Conference on Information and Automation
Shenzhen, China June 2011

Human-robot Collaborative Manipulation through Imitation and Reinforcement Learning*

Ye Gu, Anand Thobbi and Weihua Sheng
Department of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK 74074, USA
{*ye.gu, thobbi, weihua.sheng*}*@okstate.edu*

***Abstract*— This paper proposes a two-phase learning framework for human-robot collaborative manipulation tasks. A table-lifting task performed jointly by a human and a humanoid robot is considered. In order to perform the task, the robot should learn to hold the table at a suitable position and then perform the lifting task cooperatively with the human. Accordingly, learning is split into two phases. The first phase enables the robot to reach out and hold one end of the table. A Programming by Demonstration (PbD) algorithm based on GMM/GMR is used to accomplish this. In the second phase the robot switches its role to an agent learning to collaborate with the human on the task. A guided reinforcement learning algorithm is developed. Using the proposed framework, the robot can successfully learn to reach and hold the table and keep the table horizontal during lifting it up with human in a reasonable amount of time.**

***Index Terms*— Humanoids, Human-Robot Collaboration, Cooperative Manipulation, Imitation learning, Reinforcement learning.**

I. Introduction

Human-robot collaboration (HRC) [1] is a research field with a wide range of applications and high economic impact. HRC can be realized through joint actions carried out by each of the collaborating individuals. To work cooperatively on something the partners need to agree on a common goal and a joint intention to reach that goal. Previous research has indicated that one of the key objectives for human-robot collaboration is to reduce the amount of time it takes a robot to accomplish a task [2].

Imitation learning and reinforcement learning are important learning modalities for HRC. This paper propose a two-phase framework which combines imitation learning and reinforcement learning to enable the robot to learn dynamic tasks in a reasonable amount of time. Imitation learning, also referred to as Programming by Demonstration (PbD), is a powerful mechanism to reduce the search-space complexity for learning [3]. PbD mainly consists of three steps: representation, generalization and reproduction. The representation phase is for transferring skills across various agents and situations. 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 been proposed for endowing robots with imitation learning abilities [4], [5].

If PbD is used alone, large number of demonstrations performed by human are needed, although an optimal solution cannot be guaranteed [6]. For the table-lifting task, a generic feedback controller can be used. In [7], Gribovskaya *et al.* present a framework which combines programming by demonstration and adaptive control for physical human-robot interaction (pHRI). Initially, the robot has to undergo a two-phase learning procedure. PbD is used to learn human's motion model and the corresponding robot-action. An adaptive controller is then applied to adjust for the physical inconsistencies present in the robot.

Reinforcement learning can also be combined with PbD to solve the same problem. Using reinforcement learning an agent can learn behavior through trial-and-error by interacting with the environment [8], [9]. Outcome of the performed action is used as a reinforcement for updating the agent's state-action policy [10]. We chose to use a controller learned from reinforcement learning for the following reasons:

- It is possible to learn a good controller in a short time.
- It compensates for the time needed to manually tune the parameters of a feedback controller.
- Objective of the task is very 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. Also, such high level objectives are much easier to specify using reinforcement learning.

We use the Q-learning algorithm with a guided exploration strategy to learn the optimal state-action policy [11].

The proposed framework reduces the complexity of the table lifting task by combining PbD with reinforcement learning. In the first phase, PbD is used for reaching out to the table and in the second phase, reinforcement learning is used for keeping the table horizontal.

The paper is organized as follows; in Section II, we present the experimental platform. In Section III, we discuss the methodology of the proposed framework. In Section IV experimental results are provided and discussed. Section V concludes the paper with proposed future works.

*This work is partially supported by NSF Grant #2003168 to CISE/CNS #0916864 and CISE/CNS MRI #0923238 to W. Sheng

978-1-61284-4577-0270-9/11/$26.00 ©2011 IEEE

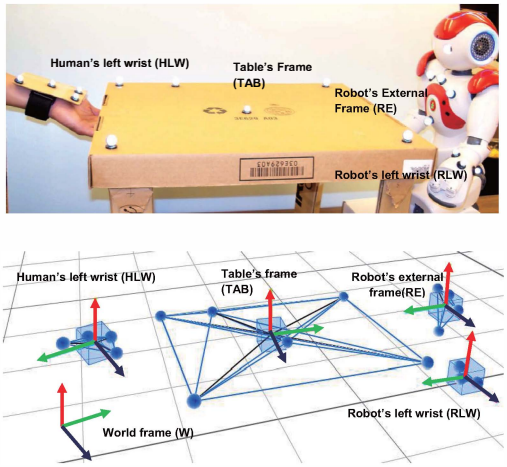

Fig. 1: Experimental Setting.

II. EXPERIMENTAL PLATFORM

This section presents the experimental platform developed to evaluate the proposed framework. The hardware includes the motion capture system, the humanoid robot and the dummy table. Fig. 1 illustrates the experimental setting.

The goal is to extract the task constraints from multiple human demonstrations andn further map these constraints to the robot's control frame. The demonstrations collected are in the task-space. Markers are attached and rigid-bodies are created to collected trajectories of interest. The position vector and the rotation matrix of each rigid-body is defined in table I

The convention we follow in this paper are: the X-Y-Z co-ordinate vector (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 ${}^{A}R_{B}$

Totally four frames are used; the world frame W, the table's frame TAB and the robot's internal RI and external frame RE. The world frame is the frame of the Vicon system in which the position vectors and rotation matrices are measured. The table's frame is used to observe the human's hand motion with respect to table during the demonstration. Human's demonstrations are mapped onto the robot's external frame which is converted to the robot's internal control frame by calibration.

A. Motion Capture System

The motion capture system used for our experiments is the Vicon MX motion capture system [12]. It is one of the most advanced optical motion capture systems available commercially. The system consists of 12 Vicon T-40 cameras. Each camera can capture a 10 bit grayscale image at a resolution of 4 megapixels. We can capture data at speeds upto 100 frames per second. However this speed is limited by the speed of robot to perform commanded actions. Atmost, the robot can acquire 10 frames per second. The system is equipped with sophisticated dynamic reconstruction algorithms for real time tracking. A Gigabit Ethernet port is provided for connecting the cameras to the system. With the given system, we can track any optical marker within a tolerance of 0.7 mm. We can create rigid bodies which are nothing but markers attached to a solid body in a specific pattern. The Vicon Tracker software is used for capturing the rigid-body data. The algorithms used in Tracker are optimized for tracking rigid bodies.

TABLE I: Co-ordinate Frames Involved

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

B. Humanoid Robot

The Nao humanoid robot [13] shown in Fig. 1 is used for the experiment. It is an autonomous, programmable and medium-sized humanoid robot which has 21 degrees of freedom (DOF), developed by Aldebaran Robotics. The robot can be controlled remotely using telnet-like commands on a wireless network. The SDK provided by the company includes an inverse kinematics procedure to control end-effector positions with respect to a frame of reference located in its torso.

C. Calibration

The robot's end-effector has to be controlled with respect to its internal frame of reference. But the data obtained from motion capture, is for the markers placed on the robot's body. Hence, a correspondence between the externally attached markers and the controllable points on the robot has to be found out. The robot's SDK can provide the position of the robot's end effector with respect to its internal frame of reference. Hence we can obtain a model for transformation given the motion capture data and the corresponding robot's data. We model this transformation as a homogenous transformation which takes care of scaling, translation and rotation. The homogenous transformation takes the form of a 4×4 matrix.

For calibration, the robot waves its hand in random trajectories trying to cover all the possible joint configurations of its arms. While it is doing so, positions are collected simultaneously from motion capture (denoted by A) and forward kinematics applied to robot's internal joint encoders (denoted by B). The linear least squares formula used to calculate this homogenous transformation (H) can be given as

$$H = (A^T A)^{-1} A^T B \tag{1}$$

III. METHODOLOGY

This section discusses the methodology for the proposed framework.

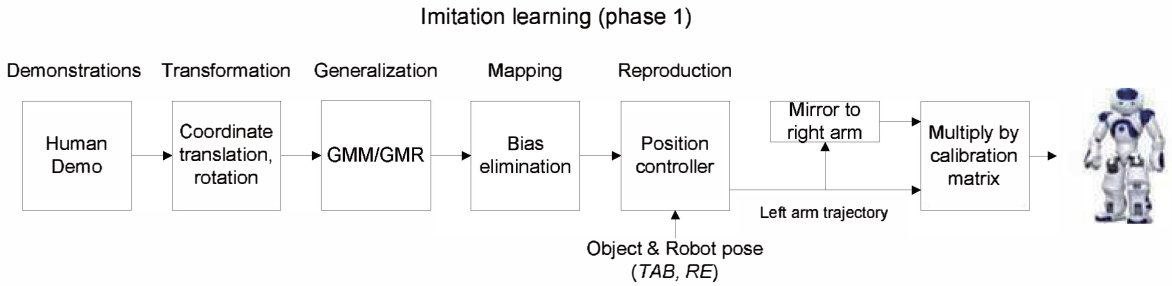

Fig. 2: The block diagram of the imitation learning phase.

A. Imitation Learning Phase

The first phase is imitation learning. Firstly, the trajectories of interest have to be derived from the human demonstrations, then we have to extract critical constraints from the trajectories. Gaussian Mixture Model (GMM) is used to encode the set of demonstrated trajectories (representation phase). Gaussian Mixture Regression (GMR) [14] is then applied to retrieve a smooth generalized version of these trajectories and associated variances (generalization phase). After mapping the constraints to the robot's perspective, the robot can generate his own trajectories based on the constraints, in the reproduction phase a position controller is derived from the generalized trajectories for the new position and orientation of the robot and the table. The block diagram of the proposed learning algorithm is shown in Fig. 2. The details of the block diagram are described next.

1) Coordinate transformation: This section deals with the required coordinate transformations. ${}^{HLW}P_W$, ${}^{RE}P_W$ and ${}^{TAB}P_W$ are all in the Vicon's world frame. The human's wrist trajectory with respect to the table (denoted by ${}^{HLW}P_{TAB}$ is of interest for learning. So we need to transform the trajectory from the world frame to the table's frame. This transformation includes translation and rotation which is given by

$$ {}^{HLW}P_{TAB} = {}^{TAB}R_W({}^{HLW}P_W - {}^{TAB}P_W) \tag{2}$$

2) Generalization: For imitation learning we adopt the probabilistic learning framework proposed by Calinon *et al.* [15]. Let $\{\varepsilon_j\}_{j=1}^{N}$ denote the N demonstrations. Each demonstration is normalized to 100 time steps. Each datapoint $\varepsilon_j = \left\{t_j, \varepsilon_j^S\right\}$ consists of a time step t_j and a coordinate of position ε_j^S which is a point in trajectory of the human's left wrist with respect to the table, ${}^{HLW}P_{TAB}$. The dataset is first modeled by a Gaussian Mixture Model(GMM) of K components [16], each data point is defined by its probability density function

$$p(\epsilon_j) = \sum_{k=1}^{K} \pi_k N(\epsilon_j; \mu_k, \Sigma_k) \tag{3}$$

where, π_k are prior probabilities and $N(\epsilon_j; \mu_k, \Sigma_k)$ are Gaussian distributions defined by centers μ_k and covariance matrices Σ_k, whose temporal and spatial components can be represented separately as

$$\mu_k = (\mu_k^T, \mu_k^S), \;\; \Sigma_k = \begin{pmatrix} \Sigma_j^{TT} & \Sigma_j^{TS} \\ \Sigma_j^{ST} & \Sigma_j^{SS} \end{pmatrix} \tag{4}$$

Based on the GMM, a generalized version of the trajectories is computed by applying Gaussian Mixture Regression (GMR). The procedure is as follows. For each component k, the expected distribution of likelihood of ε_j^S given a time step t_j and gaussian mixture component k is defined by

$$p(\epsilon_j^S | t_j, k) = N(\epsilon_j^S;\ \hat{\epsilon}_k^S, \hat{\Sigma}_k^{SS}) \tag{5}$$

$$\hat{\epsilon}_k^S = \mu_k^S + \Sigma_k^{ST}(\Sigma_k^{TT})^{-1}(t_j - \mu_k^T) \tag{6}$$

$$\hat{\Sigma}_k^{SS} = \Sigma_k^{SS} - \Sigma_k^{ST}(\Sigma_k^{TT})^{-1}\Sigma_k^{TS} \tag{7}$$

By taking the complete GMM into account, the expected distribution is defined by

$$p(\epsilon_j^S | t_j) = \sum_{k=1}^{K} \beta_{k,j} N(\epsilon_j^S;\ \hat{\epsilon}_k^S, \hat{\Sigma}_k^{SS}) \tag{8}$$

where $\beta_{k,j}$ is the probability of the component k responsible for t_j. By using the linear transformation property of Gaussian distribution, and estimation of the conditional expectation of ϵ_j^S given t_j is thus defined by $p(\epsilon_j^S\ |t_j) \propto N(\hat{\epsilon}_j^S, \hat{\Sigma}_j^{SS})$, where the parameters of the Gaussian distribution are defined by

$$\hat{\epsilon}_j^S = \sum_{k=1}^{K} \beta_{k,j}\hat{\epsilon}_k^S, \;\; \hat{\Sigma}_j^{SS} = \sum_{k=1}^{K} \beta_{k,j}^2 \hat{\Sigma}_k^{SS} \tag{9}$$

By evaluating $\left\{\hat{\epsilon}_j^S, \hat{\Sigma}_j^{SS}\right\}$ at different time steps t_j, a generalized form of the trajectories $\hat{\epsilon} = \left\{t_j, \hat{\epsilon}_j^S\right\}$ and associated covariance matrices $\hat{\Sigma} = \left\{\hat{\Sigma}_j^{SS}\right\}$ representing the constraints along the task can be computed [17].

3) Correspondence problem: In the next step, the constraints derived from ${}^{HLW}P_{TAB}$ will be applied to the robot. Since the human's dimension is different from the robot, the constraints derived for ${}^{HLW}P_{TAB}$ have to be mapped to the robot's end effector with respect to the table ${}^{RLW}P_{TAB}$. This problem can be simplified if we consider only the position of the human's wrist with respect to the table. Then, it only needs to compensate for the dimension difference shown in Fig. 3 between the human's wrist and the robot's end effector. The dimension difference is nothing but a constant bias. A simple method is proposed to calculate this dimension difference. We put markers on a fixed object, then we let human with markers on the wrist and robot with markers on the end effector touch the same point on the box respectively. The coordinates with respect to the fixed object are obtained. The difference of these two coordinates is the dimension difference.

Fig. 3: Dimension difference between human'hand and robot's hand.

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

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

$$ {}^{RLW}P_{TAB} = {}^{TAB}R_W({}^{RLW}P_W - {}^{TAB}P_W) \tag{10} $$

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

$$ {}^{RLW}P_W = {}^{TAB}R_W{}^{-1}\,{}^{RLW}P_{TAB} + {}^{TAB}P_W \tag{11} $$

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

$$ {}^{RLW}P_{RE} = {}^{RE}R_W({}^{RLW}P_W - {}^{RE}P_W) \tag{12} $$

${}^{RLW}P_{RE}$ is thus, the trajectory of the robot's left end effector with respect to its torso, which will be enacted by the robot.

5) Mirroring the trajectory: In the imitation learning phase, only left hand demonstrations are provided to the robot. This trajectory is mirrored to obtain the corresponding trajectory of robot's right end effector. The mirroring process is keeping all the X and Z coordinates in the trajectory of the left arm as they are and reversing all the Y coordinates.

B. Reinforcement learning

For reinforcement learning, the state, action space and the rewards have to be defined. Instead of having a complex contact environment, the environment in our task is simplified. The inclination of the table object determines the state. The action space consists of a predetermined discrete set of commands which move the robot's hand-tip up or down by specified distances. The objective of the task is to keep the table horizontal during the task. Accordingly, the reward structure has been designed to give a positive reward if the robot decreases the slope of the table or a negative reward if the robot increases the incline of the table. The reward r is calculated as

$$ r = (|Z2 - Z1|)_t - (|Z2 - Z1)|)_{t+1} \tag{13} $$

where $Z1$ and $Z2$ represent the position of the human-end and the robot-end of the table respectively.

The state definition for N states is shown in Fig. 4. In our task, we chose $N = 5$.

The update for the Q-learning algorithm is given by

$$ \Delta Q(s_t, a_t) = \alpha[r + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{14} $$

where α is the learning rate, γ is the discount factor.

In order to speed up the reinforcement learning phase, a guided-exploration method is used. The guided learning algorithm is given below.

Algorithm 1 Guided Q 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(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 Eq. (14), based on reward r.
end while
```

$Visit(s_i, a_i)$ is a counter which counts the number of visits to the state-action tuple (s_i, a_i). Given a state, the action selection for exploration is done on the basis of number of visits to the particular state-action pair. The state-action pair explored least is given more priority.

IV. Experiments and Results

A. Calibration results

The calibration matrix is obtained by moving the robot arm randomly, during which the coordinates of the robot's

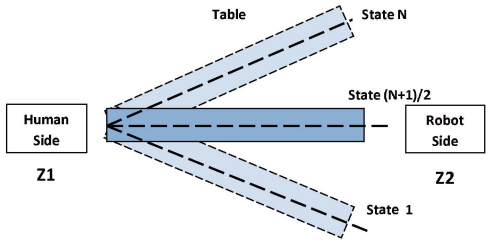

Fig. 4: State definition for the reinforcement learning.

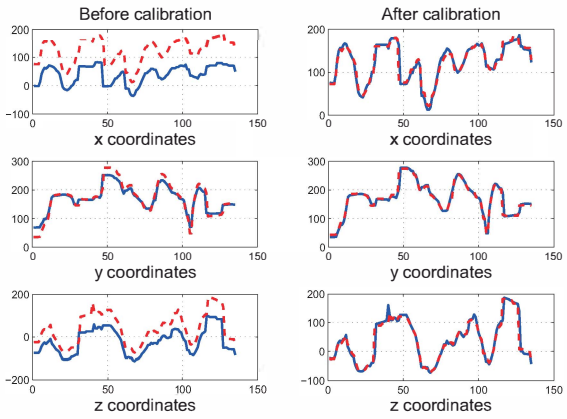

Fig. 5: Data comparison before and after calibration. Solid line represents data in the Vicon's frame and the dash line represents data in the robot's internal frame.

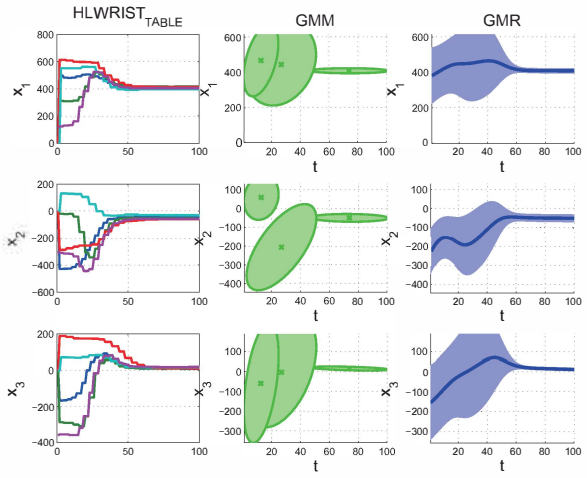

Fig. 6: Trajectory encoding and generalization.

left arm with respect to its own torso is collected both in the Vicon system co-ordinate frame and the robot's internal frame. Then, the homogeneous transformation is used to calculate the calibration matrix. Fig. 5 shows the coordinates differences before and after calibration. It can be observed that using the calibration matrix the trajectories can be converted from the motion capture frame to the robots internal frame.

B. Imitation learning results

In the imitation learning phase, multiple demonstrations are given by the human. In each demonstration, the human tried to approach the same position of the table with his or her left hand from an arbitrary initial position. An open source code from `http://www.calinon.ch/` has been used to run the GMM/GMR. The GMM/GMR results are shown in Fig. 6. Generalized trajectories and constraints are thus obtained. From the results, it can be seen that the constraint on the human-hand's initial position is very loose. In contrast, the constraint on the human hand's final position is very strict, which indicates the final position of the robot's end effector respect to the table is constant. After compensating for the size different between human's hand and robot's hand, the robot can generate its own trajectory given the contraints extracted. Every time the table is moved to a new position, new trajectory is reproduced by the position controller so that the robot can successfully approach the table. Finally the calibration matrix is used to convert the trajectories from the robot's external frame to the robot's internal frame. In the imitation learning phase, only left hand demonstrations 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 Fig. 7. The images (a)-(d) shows the robot replaying the generalized trajectories extracted from the demonstrations and the images (e)-(f) presents the robot reproducing the trajectories in a new situation (different pose of the table). It is observed that the trajectories generated are smooth, with which the robot success in approaching the table.

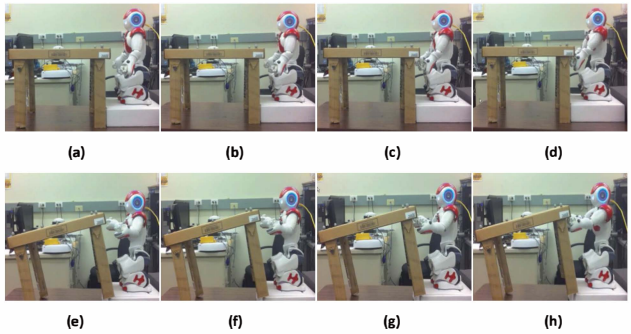

Fig. 7: (a)-(d) Replaying the generalized trajectory. (e)-(h) Reproducing the generalized trajectory in an unknown table pose.

C. Reinforcement learning results

For state-action space consisting of 5 states and 5 actions, the performance of random exploration is compared with that of guided exploration. For training, in each experiment, the number of iterations is fixed to 100. To test the speed of convergence, the experiment was performed 100 times. Fig. 8 show the speed of convergence for these two algorithms for a single experiment respectively. 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 40 trials.

After learning the optimal policy, we apply it to the robot. Fig. 9 shows the positions of the ends of the table for human and robot side, the moving range of the table is within 20 cm. Fig. 10 shows the whole process of the table lifting task. From these figures, we can see that the robot can follow the human's action and perform the table lifting task successfully.

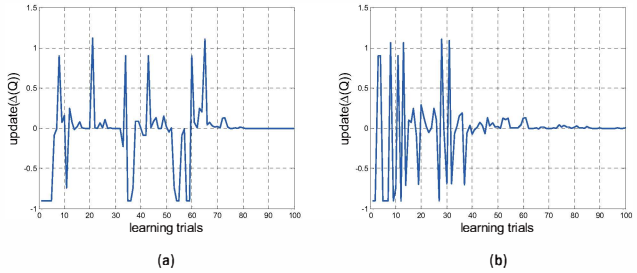

Fig. 8: (a) Random exploration learning performance. (b) Guided exploration learning performance.

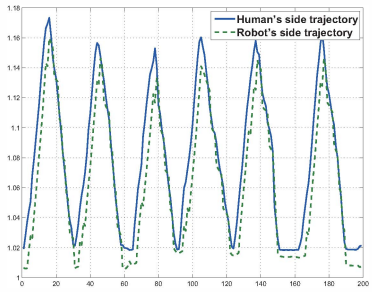

Fig. 9: Trajectory of the robot's movement and human's movement during lifting the table.

However we could also observe some jerks during the task which are due to the imperfections in the position controlled end effector of the robot. Also the movement of the robot's end effector has some delays, since it takes some time for robot to realize the human's action.

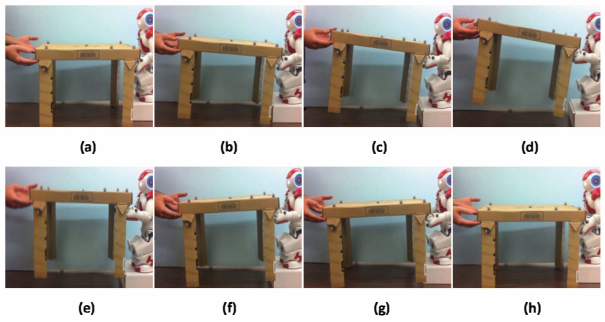

Fig. 10: Snapshots of human robot performming the table lifting task. (a)-(d) the robot lifting the table up. (e)-(h) the robot putting the table down.

V. CONCLUSIONS AND FUTURE WORKS

This paper proposed 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.

However, in our task, the robot only behaves as a follower and simply reacts to the human's action. For future works, if the robot can predict human's motion, the performance can be improved. Also, in the reinforcement learning phase, the states and actions are discrete. Using continuous state-action representation can make the robot's action smoother.

ACKNOWLEDGMENTS

This project is partially supported by the NSF grant CISE/CNS 0916864 and CISE/CNS MRI 0923238.

REFERENCES

[1] M. B. Andrea Bauer, Dirk Wollherr, "Human-robot collaboration: A survey," *International Journal of Humanoid Robotics (IJHR)*, vol. 5, pp. 47 – 66, 2008.

[2] Y. E. Uri Kartoun, Helman Stern, "A human-robot collaborative reinforcement learning algorithm," *Journal of Intelligent Robotic Systems*, vol. 60, pp. 217–239, 2010.

[3] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation," *Robotics Automation Magazine, IEEE*, vol. 17, no. 2, pp. 44 –54, 2010.

[4] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109 – 116, 2004, robot Learning from Demonstration.

[5] D. C. Bentivegna, C. G. Atkeson, and G. Cheng, "Learning tasks from observation and practice," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 163 – 169, 2004.

[6] K. Hamahata, T. Taniguchi, K. Sakakibara, I. Nishikawa, K. Tabuchi, and T. Sawaragi, "Effective integration of imitation learning and reinforcement learning by generating internal reward," in *Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on*, vol. 3, 2008, pp. 121 –126.

[7] E. Gribovskaya, A. Kheddar, and A. Billard, "Motion Learning and Adaptive Impedance for Robot Control during Physical Interaction with Humans," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011.

[8] A. M. Leslie Pack Kaelbling, Michael Littman, "Reinforcement learning: A survey." *Journal of Artificial Intelligence Research*, vol. 4, pp. 237 – 285, 1996, special issue on Information technology.

[9] S. Vijayakumar, T. Shibata, and S. Schaal, "Reinforcement learning for humanoid robotics," in *Autonomous Robot*, 2003, p. 2002.

[10] R. S. Sutton and A. G. Barto, "Reinforcement learning i: Introduction," 1998.

[11] B.-N. Wang, Y. Gao, C. Zhao-Qian, and J.-Y. C. S.-F. Xie, "A two-layered multi-agent reinforcement learning model and algorithm," *Journal of Network and Computer Applications*, vol. 30, no. 4, pp. 1366 – 1376, 2007.

[12] Vicon mx motion capture system. [Online]. Available: http://www.vicon.com/products/

[13] Aldebaran humanoid robot. [Online]. Available: http://www.aldebaran-robotics.com/en

[14] D. Cohn, Z. Ghahramani, and M. Jordan, "Active learning with statistical models." *Artificial Intelligence Research*, no. 4, pp. 129 – 148, 1996, special issue on Information technology.

[15] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 367 –372.

[16] G. Schwarz, "Estimating the dimension of a model." *Annals of Statistics*, vol. 6, pp. 461 – 464, 1978, special issue on Information technology.

[17] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.