Using Human Motion Estimation for Human-Robot Cooperative Manipulation

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

Abstract—Traditionally the leader or follower role of the robot in a human-robot collaborative task has to be predetermined. However, humans performing collaborative tasks can switch between or share the leader-follower roles effortlessly even in the absence of audio-visual cues. This is because humans are capable of developing a mutual understanding while performing the collaborative task. This paper proposes a framework to endow robots with a similar capability. Behavior of the robot is controlled by two types of controllers such as reactive and proactive controllers each giving the robot follower and leader characteristics respectively. Proactive actions are based on human motion prediction. We propose that the role of the robot can be governed by the confidence of prediction. Hence, the robot can determine its role during the task autonomously and dynamically. The framework is demonstrated and evaluated through a table-lifting task. Experimental results confirm that the proposed system improves the overall task performance.

I. INTRODUCTION

One of the fundamental abilities service robots should possess, is to work collaboratively with humans on manipulative tasks. Such technology can be widely applied in the industry as well as in common day-to-day scenarios. This field has seen a renewed interest in recent years because of the possibility of humanoid robots residing along with us in the near future.

Traditionally, the intellectual responsibility of planning and guiding the co-operative task is placed entirely on the human while the collaborating robot is assigned a mere follower role. Techniques such as impedance control [1], [2] requires the human to spend extra energy in moving the robot along with the load. Furthermore, a goal such as keeping the table exactly horizontal throughout the table-lifting task is very difficult to achieve using this technique alone.

Maeda et al. were among the earliest to provide a solution to this problem [3], by using a human motion prediction technique based on the minimum jerk model [4]. The robot partner could use the predicted velocity to take proactive actions. This strategy was shown to reduce the human’s effort. Corteville et al. presented a cooperative manipulation framework based on an extended Kalman filter (EKF) designed according to the minimum jerk model for human motion prediction [5]. However, the role of the robot had to be decided beforehand. In [6] the authors 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 [5] and [6] 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 midway, a new subtask has to be generated on the fly which is non-trivial and is a separate work in itself [7].

Recent works show that the minimum jerk model may not be suitable for cooperative manipulation tasks [8]. The minimum jerk model assumption fails when there are perturbations in the motion trajectory, or if the human decides to change the course of the trajectory during the task execution. Also, in order to apply the minimum jerk model successfully, the final position of the object must be known both to the human and robot which is inconvenient in real world scenarios. Other related work include [9] which proposes a task-model learning approach combined with an adaptive control system. After going through a two-step learning process, the robot can work collaboratively with the human while inferring his intent.

In this work, we propose a novel solution to address the problem of switching the robot’s role automatically during a cooperative manipulation task. The robot does not need to know the final position of the object. This is practically desirable, since the motion trajectory of the object may require to be changed during task execution, depending upon the environment, obstacles or the physical limitations of the human and/or the robot.

The proposed work uses a prediction-evaluation method to estimate the confidence of prediction and using it to adjust the role 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 reactivity comply.

In section II we present the experimental platform. Section
The table-lifting task consists of the human and the Nao humanoid robot [10] lifting up a dummy table to a random height and then keeping it down. Fig. 1 shows the experimental setup. Positional information of the table obtained from a motion capture system is used for characterizing the task. We do not use force sensors because the table does not have a significant weight. Motion of the robot hand is constrained to 1-D up-down motion. However, the proposed system can be easily extended to handle multiple dimensions. C++ is used at the front-end for communicating with the robot and MATLAB© is used at the back-end for processing data.

A. Motion Capture System

The Vicon MX motion capture system [11] is used for obtaining the positional information of the table, needed for carrying out the task as well as for evaluating the performance. Frame rate of the motion capture system is 100 Hz. The robot can acquire motion capture data only after it completes the commanded action. Hence, the frame-rate is limited by the speed of the robot to complete a commanded action which is typically 100 ms. Markers are attached on the table as shown in Fig. 1. The absolute position of the ends of the table along the vertical axis is calculated by averaging the positions of the markers placed at the end points. Let $Z_1$ and $Z_2$ be the instantaneous 1-D coordinates of the human-end and the robot-end of the table respectively.

B. Humanoid Robot

We use the medium sized Nao humanoid robot manufactured by Aldebaran Robotics. An inverse kinematic procedure provided by the SDK is used to control the end effector (hand-tip) position of the robot. Certain offset (+0.3 mm) had to be added in order to compensate for the small, but definite weight of the table.

III. METHODOLOGY

Fig. 2 shows the conceptual block diagram for the proposed methodology. The framework mainly consists of the reactive controller, proactive controller and the behavior gain control blocks. The reactive controller generates a reactive behavior based upon the current state of the environment. The proactive controller consists of an EKF based human motion predictor and an evaluation-based confidence generator. Based upon the observed human actions, the predictor estimates the position of the human in the next time-step, which decides the robot’s proactive action. Based upon the confidence value, the behavior gain control provides a weighted sum of reactive and proactive actions to generate a composite action to be taken by the robot. According to our hypothesis, the weight allotted by the gain control block to the proactive behavior varies directly with the confidence value. In the remainder of this section, we discuss the details of the proposed framework.

A. Reactive Controller

In the table lifting task, the action suggested by the reactive controller would be so as to make the table horizontal. This can be accomplished using any generic feedback controller. However, we chose to use a controller learned from reinforcement learning for the following reasons:

- It compensates for the time needed to manually tune the parameters of a feedback controller. A good controller can be learnt in a short time.
- 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. Such tasks have a long term reward to maintain for which reinforcement learning is most suited. Also, such high level objectives are much easier to specify.

In this work we use the discrete Q-learning algorithm. The Q-table update equation is given by

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

where $r$ is the reward, $\alpha$ is the learning rate and $\gamma$ is the discount factor. The state of the environment is determined by the incline of the table at the given moment. Incline of the table is quantized into a discrete number of states. Fig. 3 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 remains comfortably stationary. To speed up the learning phase we use a simple guided reinforcement learning algorithm. Essentially, the action selection for exploration is done on the basis of the number of visits to the particular state-action pair, instead of random action selection as in \( \epsilon \)-greedy algorithms. The guided reinforcement learning algorithm is given below.

**Algorithm 1 Guided Reinforcement Learning**

1: Initialize \( \text{Visit}(s_i, a_i) = 0 \) \( \forall i \in N \)
2: Initialize Q-table \( Q(s_i, a_i) = 0 \) \( \forall i \in N \)
3: while Learning phase do
4: \( t = \text{timestep} \)
5: \( s_t = \text{getState()} \)
6: Select \( a_t \leftarrow \arg\min(a)\{\text{Visit}(s_t, a)\} \)
7: Take action \( a_t \)
8: \( \text{Visit}(s_t, a_t) \leftarrow \text{Visit}(s_t, a_t) + 1 \)
9: \( r = \text{getReward()} \)
10: Update \( Q(s_t, a_t) \) using (1)
11: end while

**B. Proactive Controller**

The role of the proactive controller is to generate a prediction of the human’s position in the next time-step, along-with a confidence measure for the prediction. For the prediction purpose, an extended Kalman filter is used. The state of the EKF \( x_k \) is given by

\[
x_k = \begin{pmatrix} s_k \\ v_k \\ a_k \end{pmatrix}
\]  

(2)

The measurement model is given by

\[
z_k = s_k + v
\]  

(3)

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

We use the assumption 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

\[
x_{k+1} = \begin{pmatrix} s_k + v_k t + \frac{1}{2} a_k t^2 \\ v_k + a_k t \\ a_k \end{pmatrix} + w
\]  

(4)

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

\[
\hat{s}_{k+1} = \hat{s}_k + \hat{v}_k t + \frac{1}{2} \hat{a}_k t^2
\]  

(5)

The variance of the measurement noise \( (R) \) is initialized to 0.7 which corresponds to the uncertainty in measurement by the motion capture system. As shown in Fig. 4, it is possible to get accurate predictions.

For obtaining the confidence of prediction, we derive inspiration from [12], wherein the authors proposed a technique, to obtain a confidence measure based on the statistical properties of the residuals between the predicted measurements and the observed measurements. In our technique, the EKF 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 \( \rho \). 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

\[
\mathcal{L}_k = \exp \left( -\frac{(z_k - \hat{s}_k)^2}{2\rho^2} \right)
\]  

(6)

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

![Predictions obtained from EKF](image)
Fig. 5. Confidence value with predictions

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

$$C_{k+1} = \frac{L_k + \phi L_{k-1} + \cdots + \phi^{k-1} L_1}{1 + \phi + \cdots + \phi^{k-1}}$$

(7)

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

C. Behavior Gain Control

At a given time step $k$, let the reactive controller suggest a next-step action $R_{k+1}$ and the proactive controller suggest an action $P_{k+1}$. Let the confidence of this prediction be $C_{k+1}$. The gain control block combines these together to form a composite action $A_{k+1}$ given by

$$A_{k+1} = C_{k+1}P_{k+1} + (1 - C_{k+1})R_{k+1}$$

(8)

This action is taken by the robot at time-step $k + 1$. The inspiration for this form has been taken from [6]. Note that because $0 < C \leq 1$, 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 $C_{k+1}$ is high, 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. EXPERIMENTAL RESULTS

In this section we present the experiments performed and the results obtained.

A. Learning the Reactive Controller based on Q - Learning

For Q-learning, a state-action space consisting of 5 states and 5 actions was arbitrarily chosen. The reward $r$ was decided as

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

(9)

where $Z_1$ and $Z_2$ represent the position of the human-end and the robot-end of the table respectively. Hence, if the incline 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. Values of the reinforcement learning parameter used were: learning rate $\alpha = 0.9$ and discount factor $\gamma = 0.2$.

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 minutes to complete.

B. Prediction

Fig. 4 shows the predicted and observed values of position, velocity and acceleration. Each time step is 100 ms. The predicted position is calculated from (5). True velocity and
acceleration are derived from the observed position, and are shown in the figure for comparison with the predicted values. It can be observed from Fig. 4 that the estimates keep improving with time. The calculated acceleration nearly remains centered at 0 with small changes. This partly justifies our assumption that the acceleration remains nearly constant.

C. Confidence Measure

Fig. 5 shows the variation confidence \( C \) through the task. It can be seen that initially, when the table is still, the predictor accurately estimates the motion to be zero resulting in high confidence value. Once the task starts, the predictions are inaccurate initially because of the drastic change in the motion model. This causes confidence value to drop down. The reactive controller of the robot is dominant in this region. Eventually, as the predictions improve, the confidence value increases. As a result the proactive behavior becomes more dominant.

Fig. 7 shows the role of the forgetting factor \( \phi \) in determining the confidence. Since it is not possible to reproduce the exact same trajectory during the task, the confidence trajectories shown in Fig. 7 are computed offline step-by-step using data collected from one trial. As seen from (7), a low value of \( \phi \) means that the predictor allots a small weight to older likelihood estimates. Thus, \( C \) mostly depends upon the recently observed \( L \). Hence, if the likelihood values \( L \) change quickly, it causes \( C \) fluctuate heavily. Using a similar reasoning, a large value of \( \phi \) causes the confidence measure to settle very slowly. Hence the robot cannot adapt to the motion changes quickly and can generate high confidence values even for incorrect predictions. A good value for \( \phi \) which gives a good tradeoff between smooth variation and adaptability was found to be 0.45.

D. 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. Fig. 6 shows a non-typical case where the human chooses to take a pause during the task. 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 and also keeps still. At time-step 35, the human again starts moving the table upwards. Again, the robot switches from leader to follower. Once the motion has been stabilized the robot maintains a confidence value centered somewhere around 0.5.

E. Overall System Performance

In this experiment we evaluate the improvement offered by our system. 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 error given by \( \sum_t |Z_{1t} - Z_{2t}| \).

Fig. 8 shows the trajectories of both ends of the table for cases where the 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. 8:

- The RMSE for case I is always smaller.
- The motion observed for case I is smoother.
- The absolute error is lower in case I.

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 keep it down for 10 trials. Totally, for both cases, 100 trials were acquired. The table I shows the average RMSE for the 10 trials observed for each subject, for each case. It can be seen that, for all the subjects, the RMSE is lower for the ‘with predictions’ case. Hence a definite improvement can be observed.

V. DISCUSSIONS AND CONCLUSIONS

A. Discussions

For comparison with the human-robot team, similar experiments were performed for human-human teams. The average RMSE observed for a human-human team was 6.531 mm. The motion of the robot is jerky when its reactive behavior is dominant, 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. Thus, the results are not only influenced by the robot’s performance alone, but also by the human performance.

In Fig. 8 we could observe in case I, the trajectory is much smoother when the human is placing the table down as compared to moving upwards. This is because, inherently, the robots motion while lifting the table against gravity is jerky because of the internal control characteristics. This induces some jerks in the human motion also since they are coupled by the table. Because of this, the prediction suffers, which causes lower confidence levels. 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 almost 10 motion capture samples per second. With a faster robot, more samples could be obtained per second which would improve the quality of predictions. Finally, our work 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.

To extend this work to higher dimensions, a few changes might be necessary. The new task objective would be to keep the table level in all three dimensions. A multidimensional EKF has to be considered. The reactive controller could be replaced by a simple PID controller since reinforcement learning might take a longer time to converge for the enlarged state-action space. The humanoid robot might also be required to walk around to satisfy the new task objective.

B. Conclusions

This work develops a framework that utilizes human motion prediction to adjust the leader/follower role of the collaborating robot in a co-operative manipulation task. Behavior of the robot is governed by a weighted sum of the outputs from the reactive and proactive controllers. The proactive controller is based on an EKF 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.

For future works, we propose to utilize longer term predictions. 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.

REFERENCES