

Human-Robot Shared Workspace Collaboration via Hindsight Optimization

Stefania Pellegrinelli^{1,2}, Henny Admoni², Shervin Javdani² and Siddhartha Srinivasa²

Abstract—Our human-robot collaboration research aims to improve the fluency and efficiency of interactions between humans and robots when executing a set of tasks in a shared workspace. During human-robot collaboration, a robot and a user must often complete a disjoint set of tasks that use an overlapping set of objects, without using the same object simultaneously. A key challenge is deciding what task the robot should perform next in order to facilitate fluent and efficient collaboration. Most prior work does so by first predicting the human’s intended goal, and then selecting actions given that goal. However, it is often difficult, and sometimes impossible, to infer the human’s exact goal in real time, and this serial predict-then-act method is not adaptive to changes in human goals. In this paper, we present a system for inferring a probability distribution over human goals, and producing assistance actions given that distribution in real time. The aim is to minimize the disruption caused by the nature of human-robot shared workspace. We extend recent work utilizing Partially Observable Markov Decision Processes (POMDPs) for shared autonomy in order to provide assistance without knowing the exact goal. We evaluate our system in a study with 28 participants, and show that our POMDP model outperforms state of the art predict-then-act models by producing fewer human-robot collisions and less human idling time.

I. INTRODUCTION

Human-robot collaboration (HRC) studies interactions between humans and robots when executing tasks in a shared workspace [1]–[3]. Given a set of tasks (the actions to be performed, e.g., to open, to grasp, to move) and goals (the objects on which the task is performed, e.g., a door, a glass, a cup), the human can select tasks to perform and goals on which to perform them. Meanwhile, the robot can perform an uncorrelated task on a secondary goal [4] or can support the human during his/her task execution by performing another task on the same goal (e.g., to grasp a bottle if the human wants to open it) or on another goal (e.g., to open the fridge door if the human wants to grab a bottle from it) [5], [6] (Figure 1). Since one of the aims in HRC is to optimize human satisfaction and comfort [7], [8], in the majority of approaches, the human has priority over the robot on task selection. Thus, the robot must adapt to the human’s task selections. Robot adaptation is even more complex when possible constraints on the execution order of the tasks exist (e.g. in order to grasp the bottle in the fridge, the fridge door must be open) and when the robot is sharing workspace, goals and/or tasks with more than one human.

¹ITIA-CNR, Institute of Industrial Technologies and Automation, National Research Council of Italy, 20133, Milano, Italy
stefania.pellegrinelli@itia.cnr.it

²Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 {henny, sjavdani, siddh}@cmu.edu

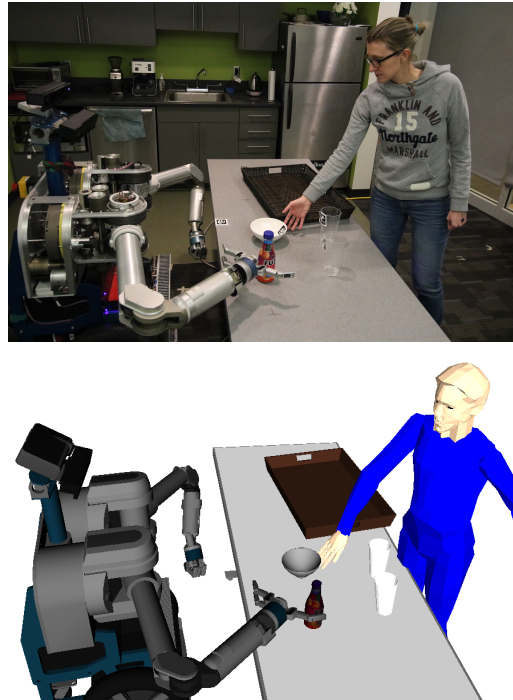


Fig. 1: An example of HRC for clearing a table. All objects on the table have to be moved on the tray.

Robot adaptation to human behavior has been widely investigated during the last decade in terms of how robot motion can be efficiently planned. Some papers focused on the generation of robot trajectories through the use of traditional motion planners [9], [10], such as Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [11] or Constrained Bi-directional Rapidly-Exploring Random Tree (CBiRRT) [9], while other papers worked on the predictability and legibility of the robot trajectories [12], i.e., the robot motion has to match human expectation and the robot goal can be easily inferred by the human.

All of these approaches rely on the hypothesis that, once the human goal is predicted with a certain probability, the robot trajectory can be calculated and executed. This framework has been referred to as *predict-then-act* [13]. This approach presents some disadvantages due to the lack of reactivity in the algorithms for the robot motion planning: (i) the approach is not robust to errors in the goal prediction; (ii) the approach can take into account the change of the human goal only through re-planning the robot’s trajectory while the previous trajectory is in execution [14], thus implying the need to stop the robot and then to restart it

or to connect two trajectories on the fly; (iii) this approach is often suboptimal, as the robot must act on a single goal even when it is not confident about its goal prediction, or perform no motion until it is confident.

An interesting reactive strategy that addresses these limitations is presented in [13] and [15]. However, both these papers focus on shared autonomy during robotic teleoperation. Shared autonomy involves a human teleoperating a robot, thus making the robot's goals and tasks the same as the human's goals and tasks. In contrast, HRC attempts to produce robot behaviors that are complementary to human behaviors. The key difference between HRC and shared autonomy is that in shared autonomy, human movements become a direct input for the robot, and the robot reactively adapts its motion by taking into account its continuous prediction of the human goal derived from this direct input; in HRC, the robot's action is independent of direct human control. Moreover, in shared autonomy, the robot and the human do not necessarily share the same physical workspace, even if they operate on it together. In contrast, a shared workspace is one of the basic assumptions of HRC.

The goal of this paper is to adapt the techniques currently used for teleoperation to human-robot shared workspace collaboration. A key aspect of such collaboration is minimizing the robot's disruptions of a human's tasks when the two are acting independently. In this domain, the human and the robot have to perform *different tasks on a set of goals*. The robot should not hinder the human's actions, and therefore should select only goals that the human has not currently selected. Thus, the idea is to provide a *real-time reactive approach* for HRC, overcoming the limitation of previous frameworks. To simplify the problem, we assume there are no precedence constraints among the tasks, i.e., that partial-order plans do not exist.

The approach is based on two distinct integrated modules allowing human-robot real-time collaboration (Figure 2). The first module continuously tracks and analyzes the movements of the human arm and updates a probability distribution on a predefined set of goals. This probability distribution represents the probability that the user will converge toward an obstacle and is used as an input for the second module, a Partially Observable Markov Decision Process (POMDP) following prior work [13]. The POMDP generates commands for the robot arm that reaches for the closest goal which is not the human goal. This approach allows for collaborative behavior in which human and robot have different tasks to perform on the same set of goals.

To evaluate our approach, we implemented our system and tested it in a real-time human-robot shared workspace collaboration. The collaboration (a gift-wrapping task) was designed so that human and robot each had distinct actions (the human tied bows out of ribbon around a set of boxes, while the robot stamped the boxes), but shared a set of goals on which to perform those tasks (the boxes). Our implemented system runs autonomously in real time by using hindsight optimization to solve the POMDP, enabling the robot to continually monitor and seamlessly replan its

motions according to changing human goals. We compare our system against the current state of the art predict-then-act method described above, as well as a baseline control algorithm that does not account for human goals, but simply executes a fixed set of robot behaviors.

Though the current work focuses on one-robot-one-human manipulation tasks without sequence constraints, extensions beyond these limitations are straightforward. The approach can be applied to navigation tasks [16], thus extending the work of Ziebart et al. [14] from human-robot avoidance to goal-directed collaboration. Moreover, the approach can be adapted to collaboration with two or more humans by modifying our goal prediction strategy to predict the goals of each human, and our belief to be over the set of user goals. Our action selection through a POMDP would otherwise be unchanged. Similarly, task precedences can be included by modifying the probability distribution on the basis of precedence constraints among the tasks. Specifically, trees of probability can be built and used to define a single probability distribution.

The paper, supported by a video, is structured as follows: Section II presents the state of the art in HRC; Section III provides a detailed description of the developed approach; Section IV details our evaluation and results; and Section V presents our conclusions and future work.

II. RELATED WORK

A. Human Goal Prediction

In many scenarios, effective HRC requires knowing which goal the human wants to achieve. Prior works suggest that it is ineffective to require the human to explicitly specify their goal (e.g. through buttons) [17]–[19]. Instead, we should infer the human's goal based on their actions and a model of how they would act for different goals.

In human-robot pedestrian settings, Ziebart et al. [20] propose maximum entropy inverse optimal control (Max-Ent IOC), which models the human as an intent driven agent stochastically optimizing some cost function. Best and Fitch [21] propose a Bayesian framework to predict the human's goal in complex 2D environments by using Probabilistic Roadmaps [22] and Monte-Carlo sampling.

In human-robot collaboration settings, many papers aim at identifying human intention by analyzing human kinematics during the execution of a specific set of tasks. Wang et al. [23] learn a generative predictor by extending Gaussian Process Dynamical Models (GPDMs) with a latent variable for intention. Koppula and Saxena [5] extend conditional random fields (CRFs) with object affordances to predict potential human motions for different tasks. Jiang and Saxena [24] extend this framework to take into account more complex human movements in static environments. All of these approaches try to identify and predict human trajectories. The current paper takes a different approach by identifying a probability distribution over the goals.

As our algorithm requires real-time goal prediction in continuous state and action spaces, we refer to the framework of Dragan and Srinivasa [15] used in shared autonomy

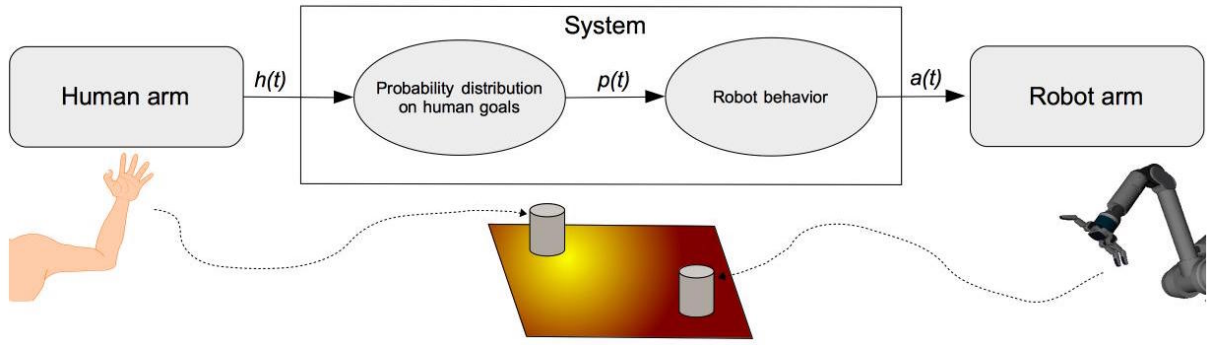


Fig. 2: Framework of the proposed approach. The system tracks the human arm position $h(t)$ and infers a distribution over the human’s goals $p(t)$ based on the human arm movements. Given this distribution, a POMPD identifies the best robot behavior, and selects robot action $a(t)$ to execute.

settings. Dragan and Srinivasa [15] extend the MaxEnt IOC framework [20] to continuous state and action spaces. They apply this framework with a fixed distance-based cost function to predict a human’s goal in real time by tracking their arm movements. On the basis of this prediction, we calculate a probability distribution on the set of human goals at each required time step. The details of the developed approach are described Section III-A.

B. Human-Robot Collaboration

Ziebart et al. [25] present a method for navigating a mobile robot in a room while avoiding humans. They utilize MaxEnt IOC to predict the human trajectory based on the goals they may be walking towards. With these predictions, they create a time-varying cost map on this shared space, with high cost where humans are likely to be. Finally, they integrate this cost map into robot planning with D* [26]. Bandyopadhyay et al. [27] similarly encode pedestrian motion models with intent into a robot planner through a MOMDP, which they use to select mobile robot motions. Karami et al. [28], [29] utilize a PODMP to infer user goals and select the next robot task in shared mission domains.

Nguyen et al. [30] and Macindoe et al. [31] apply POMDP models to creating agents in cooperative games. Like our method, the autonomous agents simultaneously infer human intentions, and take corresponding assistance actions based on a POMDP model. In contrast to these works, our state and action spaces are continuous.

For anthropomorphic robots, Lasota and Shah [32] propose modeling the human action and decision making process as a stochastic transition function with an MDP. During motion planning, they model the human arm as a dynamic obstacle, enabling them to avoid it. However, this approach requires training with the robot on specific tasks to identify the optimal policy. Mainprice et al. [4] avoid this requirement by predicting human motion through a Conditional Random Field (CRF), and similarly integrating these predictions into robot motion planning.

In contrast with the approaches described in this section, we propose a real-time reactive approach for human-robot cooperation where we continuously infer a distribution over

the human’s intention, and act based on that distribution. Human intention is modeled as a latent state in a POMDP.

III. APPROACH

We assume a set of human goals $g^h \in G^h$ and robot goals $g^r \in G^r$. Both the human and robot want to achieve all goals eventually. However, there are constraints as to how these goals can be achieved, e.g. human and robot cannot simultaneously use the same object. We assume these constraints are given by a *goal restriction set* $\mathcal{R} = \{(g^h, g^r) : \text{Cannot achieve } g^h \text{ and } g^r \text{ simultaneously}\}$. In order to efficiently collaborate with the human, our objective is to simultaneously predict the human’s intended goal, and achieve a robot goal not in the restricted set.

In the following, we present our method for achieving a robot goal while inferring the current user goal. To achieve all goals, we repeatedly execute this method, removing already achieved goals from G^r . See Section IV for details.

A. Human Goal Prediction

In order to select effective robot actions, we first infer a distribution over the human’s goal $P(g^h)$. Specifically, our aim is to infer a distribution over goals given the human trajectory ξ , e.g. a sequence of arm poses. Let p represent a state of the human, i.e. the pose of the human’s hand, p_0 the initial state, and p_g the state for achieving goal g^h . Let $\xi_{p_i \rightarrow p_j}$ correspond to a trajectory starting at p_i , and ending at p_j . Given the current trajectory $\xi_{p_0 \rightarrow p}$ from start state p_0 to current state p , we wish to infer a distribution over goals $P(g^h | \xi_{p_0 \rightarrow p})$. Using Bayes’ rule, this corresponds to:

$$P(g^h | \xi_{p_0 \rightarrow p}) \propto P(\xi_{p_0 \rightarrow p} | g^h)P(g^h),$$

where $P(g^h)$ is the prior probability of goal g^h . Without prior knowledge, we set this to the uniform distribution.

To compute $P(\xi_{p_0 \rightarrow p} | g^h)$, we follow the formulation of Maximum Entropy Inverse Optimal Control (MaxEnt IOC) [20], which shows that minimizing the worst-case predictive loss results in a model where the probability decreases exponentially with cost, $P(\xi | g^h) \propto \exp(-C_g^h(\xi))$.

Given this model, $P(\xi_{p_0 \rightarrow p} | g^h)$ can be computed marginalizing over the paths from the start to the goal (Dragan and Srinivasa [15]):

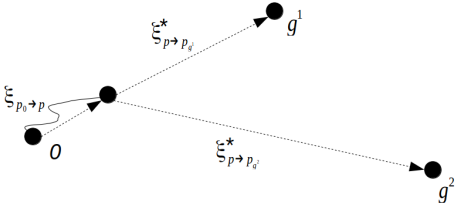


Fig. 3: Example of our goal prediction method, where the cost of a trajectory is the euclidean distance it traverses. Here, the currently executed human trajectory $\xi_{p_0 \rightarrow p}$ is more efficient for achieving g_1 than g_2 , as it deviates less from $\xi_{p_0 \rightarrow p_{g_1}}^*$ than $\xi_{p_0 \rightarrow p_{g_2}}^*$. Therefore, $P(g_1 | \xi_{p_0 \rightarrow p})$ is greater than $P(g_2 | \xi_{p_0 \rightarrow p})$.

$$P(\xi_{p_0 \rightarrow p} | g^h) = \frac{\exp(-C_g^h(\xi_{p_0 \rightarrow p})) \int_{\xi_{p \rightarrow p_g}} \exp(-C_g^h(\xi_{p \rightarrow p_g}))}{\int_{\xi_{p_0 \rightarrow p_g}} \exp(-C_g^h(\xi_{p_0 \rightarrow p_g}))},$$

where $\int_{\xi_{p_i \rightarrow p_j}}$ integrates over all trajectories between p_i and p_j . However, computing $P(\xi_{p_0 \rightarrow p} | g^h)$ in such a way is generally intractable. Instead, we follow Dragan and Srinivasa [15] who use Laplace's method, obtaining the following formulation:

$$P(\xi_{p_0 \rightarrow p} | g^h) \approx \frac{\exp(-C_g^h(\xi_{p_0 \rightarrow p}) - C_g^h(\xi_{p \rightarrow p_g}^*))}{\exp(-C_g^h(\xi_{p_0 \rightarrow p_g}^*))}$$

Where $\xi_{p_i \rightarrow p_j}^* = \arg \min_{\xi_{p_i \rightarrow p_j}} C^h(\xi_{p_i \rightarrow p_j})$, i.e. the optimal trajectory from p_i to p_j .

Intuitively, this equation relates goal probability to how efficiently we are achieving that goal relative to the best we could have done. See Figure 3 for an illustration.

B. Robot Behavior

Assuming the human has selected their goal g^h , our task is to accomplish any robot goal $g^r \in G^r$ that is not within the goal restriction set \mathcal{R} . One possible method is to utilize our predictor to infer the humans's goal, commit to a robot goal not in the restricted set, then execute motions. However, correctly inferring and committing to a specific human goal can be difficult, e.g. cluttered scenes, reducing the effectiveness of collaboration. Instead, we follow Javdani et al. [13] and encode our problem as a Partially Observable Markov Decision Process (POMDP), enabling us to minimize the expected robot cost for the (unknown) human goal.

Formally, let $x \in X$ be the continuous robot state (e.g. position, velocity), and let $a \in A$ be the continuous action (e.g. velocity, torque). We model the robot as a deterministic system with transition function $T : X \times A \rightarrow X$.

In order to formalize our POMDP, we define a system state as both the robot state and human goal, $S = X \times G^h$. As we do not know the humans' goal beforehand, we keep track of a distribution over the system state, known as the *belief* b . We assume the robot state x is known, and this distribution is only over the human goal g^h .

To infer the human goal, we receive observations $o \in O$ which provides us information through a measurement model Ω . In our setting, observations correspond to human states p , which updates our distribution through the MaxEnt IOC.

We encode the task through a cost function $C^r : S \times A \rightarrow \mathbb{R}$, which the robot incurs at each time step. The goal of the POMDP is to minimize the total expected cost, $\mathbb{E}[\sum_t C^r(s_t, a_t)]$ [13].

Hindsight Optimization: Solving this POMDP, i.e. finding the optimal cost-minimizing action for any belief b , is computationally intractable due to the continuous state and action spaces. Additionally, this would rely on accurate human models to model observations. Instead, we follow Javdani et al. [13] and utilize the QMDP approximation [33], also referred to as hindsight optimization [34], [35] to select actions. The idea is to estimate the cost-to-go of the belief by assuming full observability will be obtained at the next time step. The result is a system that never takes actions for information gathering, but can plan efficiently through deterministic subproblems. We believe this is a suitable approximation for our POMDP, as we assume human motions are independent of robot actions, and we therefore cannot explicitly gather information anyway.

For any known system state $s = (x, g^h)$, assume we have defined the *value function* $V(s)$, which specifies the cost-to-go if we acted optimally from s . For any action, this also defines the *action-value function* $Q(s, a) = C^r(s, a) + V(s')$, which corresponds to incurring the cost of taking action a in state s , transitioning to state s' , and following the optimal policy thereafter. The QMDP approximation [33] selects actions by:

$$\arg \min_a \sum_s b(s) Q(s, a)$$

As our actions are continuous, we may not be able to compute $\arg \min_a$ exactly. Instead, we follow Javdani et al. [13] and utilize a first-order approximation to select an approximately optimal action.

Defining the Cost Function: A key challenge is defining a cost function $C^r(s, a)$ which, when optimized, produces behavior for effective collaboration. Our objective is to produce behavior that quickly achieves any goal $g^r \in G^r$ not in the restricted set \mathcal{R} for g^h in system state $s = (x, g^h)$.

Javdani et al. [13] provide a method for defining cost functions that achieve any target in a set, e.g. any grasp pose for an object. Let $C_g^r(x, a)$ be a cost function such that, when optimized, efficiently achieves the robot goal g^r . In our setting, we wish to achieve any goal not in the restricted set, leading to:

$$C^r(s, a) = C^r((x, g^h), a) = \min_{g^r \text{ s.t. } (g^r, g^h) \notin \mathcal{R}} C_g^r(x, a)$$

Importantly, Javdani et al. [13] provide a method for computing the corresponding value function V . Let V_{g^r} be the corresponding value function for robot goal g^r . For deterministic MDPs, this leads to:

$$V(s) = V((x, g^h)) = \arg \min_{g^r \text{ s.t. } (g^r, g^h) \notin \mathcal{R}} V_{g^r}(x)$$

IV. EXPERIMENTS

We implemented the system on a real-world shared workspace collaboration. First, we show that an implementation of our algorithm successfully operates in real time with naive users. Second, we investigate the effectiveness of the POMDP representation with hindsight optimization by comparing our approach to a system that monitors user goals, but separates the goal prediction and assistance phases. To achieve this second goal, we measured the fluency and efficiency of the human-robot collaboration across three conditions: (i) when using our *POMDP* model with hindsight optimization; (ii) when using a state of the art *predict-then-act* controller that must first predict goals, then act when goal prediction reaches a certain confidence; and (iii) when using a *control* algorithm that executes a fixed sequence without accounting for human goals.

To compare these three algorithms, we measure user performance on objective and subjective metrics detailed below. We hypothesize that:

- H1 Task fluency will be improved with our POMDP algorithm compared with the predict-then-act and control systems.
- H2 Task efficiency will be improved with our POMDP algorithm compared with the predict-then-act and control systems.
- H3 People will subjectively prefer the POMDP algorithm to the predict-then-act or control systems.

A. Design

We developed a gift-wrapping task (Figure 4). A row of four boxes was arranged on a table between the human and the robot; each box had a ribbon underneath it. The robot’s task was to stamp the top of each box with a marker it held in its hand. The human’s task was to tie a bow from the ribbon around each box. By nature of the task, the goals had to be selected serially, though ordering was unspecified. Though participants were not explicitly instructed to avoid the robot, tying the bow while the robot was stamping the box was challenging because the robot’s hand interfered, which provided a natural disincentive toward selecting the same goal simultaneously. To compare the three systems, we used a within subjects design. Each participant completed the gift-wrapping task three times, each time with a different robot control algorithm. The order of the algorithms was counterbalanced.

B. Metrics

Task fluency involves seamless coordination of action. One measure for task fluency is the minimum distance between the human and robot end effectors during a trial. This was measured automatically by a Kinect mounted on the robot’s head, operating at 30Hz. Our second fluency measure is the proportion of trial time spent in collision. Collisions occur when the distance between the robot’s end effector and the human’s hand goes below a certain threshold. We determined that 8cm was a reasonable collision threshold based on observations before beginning the study.



Fig. 4: Participants performed a collaborative gift-wrapping task with HERB to evaluate our reactive goal prediction system against a non-adaptive control and a state of the art predict-then-act system.

Task efficiency relates to the speed with which the task is completed. Objective measures for task efficiency were total task duration for robot and for human, the amount of human idle time during the trial, and the proportion of trial time spent idling. Idling is defined as time a participant spends with their hands still (i.e., not completing the task). For example, idling occurs when the human has to wait for the robot to stamp a box before they can tie the ribbon on it. We only considered idling time while the robot was executing its tasks, so idle behaviors that occurred after the robot was finished stamping the boxes—which could not have been caused by the robot’s behavior—were not taken into account.

We also measured subjective *human satisfaction* with each algorithm through a seven-point Likert scale survey evaluating perceived safety (four questions) and sense of collaboration (four questions).

C. Implementation

We implemented the three control algorithms on HERB [36], a bi-manual mobile manipulator with two Barrett WAM arms. A Kinect was used for skeleton tracking and object detection. Motion planning was performed using CHOMP, except for our algorithm in which motion planning works according to Section III-B.

The stamping marker was pre-loaded in HERB’s hand. A stamping action began at a home position, the robot extended its arm toward a box, stamped the box with the marker, and retracted its arm back to the home position.

To implement the non-adaptive control algorithm, the system simply calculated a random ordering of the four boxes, then performed a stamping action for each box. To implement the predict-then-act algorithm, the system ran the human goal prediction algorithm from Section III-A until a certain confidence was reached (50%), then selected a goal that was not within the restricted set \mathcal{R} and performed a stamping action on that goal. There was no additional human goal monitoring once the goal action was selected. In contrast, our POMDP implementation performed as described in Section III, accounting continually for adapting human goals and seamlessly re-planning when the human’s goal changed.

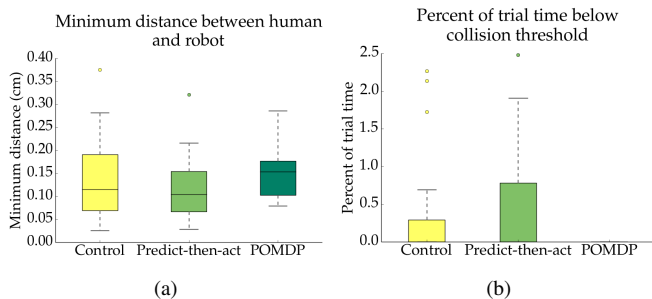


Fig. 5: Distance metrics: no difference between algorithms for minimum distance during interaction, but the POMDP algorithm yields significantly less time in collision between human and robot.

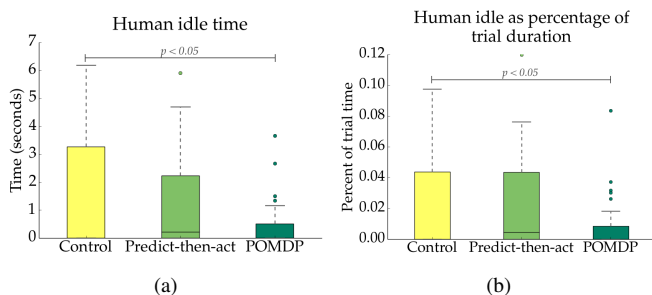


Fig. 7: Idle time metrics: POMDP yielded significantly less idle time, both for (a) absolute time and (b) percentage of trial time, than the control algorithm.

D. Procedure

28 participants (14 female, 14 male; mean age 24, SD 6) performed the gift-wrapping task. Each participant was compensated \$5 for their time.

After providing consent, participants were introduced to the task by a researcher. They then performed the three gift-wrapping trials sequentially. Immediately after each trial, before continuing to the next one, participants completed an eight question Likert-scale survey to evaluate their collaboration with HERB on that trial. At the end of the study, participants provided verbal feedback about the three algorithms. All trials and feedback were video recorded.

E. Results

Two participants were excluded from all analyses for noncompliance during the study (not following directions). Additionally, for the fluency objective measures, five other participants were excluded due to Kinect tracking errors that affected the automatic calculation of minimum distance and time under collision threshold. Other analyses were based on video data and were not affected by Kinect tracking errors.

To evaluate H1 (fluency), we conducted a repeated measures ANOVA testing the effects of algorithm type (POMDP, predict-then-act, and control) on our two measures of human-robot distance: the minimum distance between participant and robot end effectors during each trial, and the proportion of trial time spent with end effector distance below the 8cm

collision threshold (Figure 5). The minimum distance metric was not significant ($F(2, 40) = 1.405, p = 0.257$). However, proportion of trial time spent in collision was significantly affected by algorithm type ($F(2, 40) = 3.364, p = 0.045$). Interestingly, the POMDP algorithm never entered under the collision threshold. Post-hoc pairwise comparisons with a Bonferroni correction reveal that the POMDP algorithm yielded significantly ($p = 0.033$) less time in collision than the predict-then-act algorithm (POMDP $M = 0.0\%$, $SD = 0$; predict-then-act $M = 0.44\%$, $SD = 0.7$).

Therefore, H1 is partially supported: the POMDP algorithm actually yielded no collisions during the trials, whereas the predict-then-act algorithm yielded collisions during 0.4% of the trial time on average. This confirms the intuition behind the differences in the two algorithms: the POMDP continually monitors human goals, and thus never collides with the human, whereas the predict-then-act commits to an action once a confidence level has been reached, and is not adaptable to changing human goals.

To evaluate H2 (efficiency), we conducted a similar repeated measures ANOVA for the effect of algorithm type on task durations for robot and human (Figure 6), as well as human time spent idling (Figure 7). Human task duration was highly variable and no significant effect for algorithm was found ($F(2, 50) = 2.259, p = 0.115$). On the other hand, robot task duration was significantly affected by algorithm condition ($F(2, 50) = 79.653, p < 0.005$). Post-hoc pairwise comparisons with a Bonferroni correction reveal that differences between all conditions are significant at the $p < 0.005$ level. Unsurprisingly, robot task completion time was shortest in the control condition, in which the robot simply executed its actions without monitoring human goals ($M = 46.4s, SD = 3.5s$). It was significantly longer with the predict-then-act algorithm, which had to wait until prediction reached a confidence threshold to begin its action ($M = 56.7s, SD = 6.0$). Robot task time was still longer for the POMDP algorithm, which continually monitored human goals and smoothly replanned motions when required, slowing down the overall trajectory execution ($M = 64.6s, SD = 5.3$).

Total task duration (the maximum of human and robot time) also showed a statistically significant difference ($F(2, 50) = 4.887, p = 0.012$). Post-hoc tests with Bonferroni correction show that control ($M = 58.6s, SD = 14.1$) performed significantly ($p < 0.05$) faster than the other two algorithms (predict-then-act: $M = 60.6s, SD = 7.1$; POMDP: $M = 65.9s, SD = 6.3$). However, there is no statistically significant difference between predict-then-act and POMDP algorithms. In other words, though algorithm execution time slows down robot actions, people's faster performance with the POMDP somewhat redeems this by eliminating differences between POMDP and predict-then-act algorithms.

Total idle time was also significantly affected by algorithm type ($F(2, 50) = 3.809, p = 0.029$). Post-hoc pairwise comparisons with Bonferroni correction reveal that the POMDP algorithm yielded significantly less idle time than the control

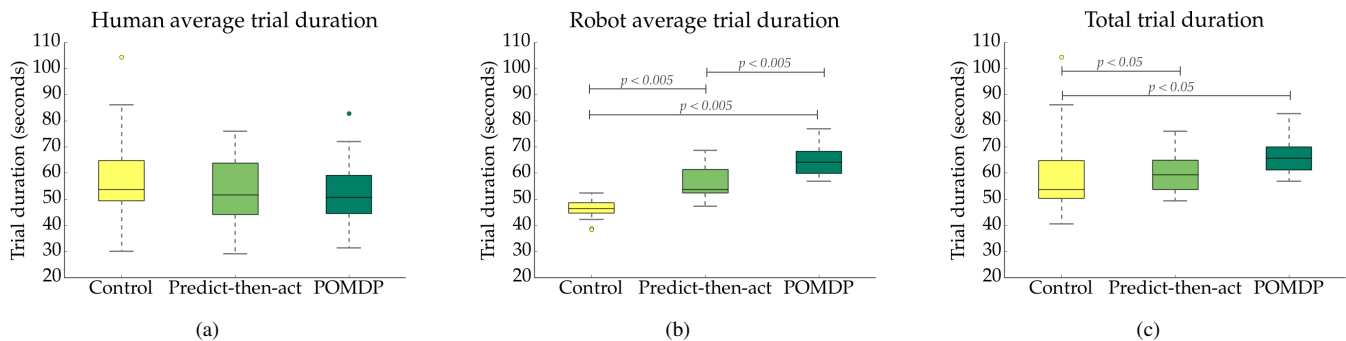


Fig. 6: Duration metrics: human trial time was approximately the same across all algorithms, but robot time increased with the computational requirements of the algorithm. Total time thus also increased with algorithmic complexity.

condition (POMDP $M = 0.46s$, $SD = 0.93$, control $M = 1.62s$, $SD = 2.1$). In other words, the POMDP algorithm performed significantly better than the non-adaptive control in reducing human idling time, while the predict-then-act method did not.

Idle time percentage (total idle time divided by human trial completion time) was also significant ($F(2, 50) = 3.258$, $p = 0.047$). As with total idle time, post-hoc pairwise tests with Bonferroni correction show that the POMDP yielded shorter idle percentages than the control (POMDP $M = 0.9\%$, $SD = 1.9$, control $M = 2.8\%$, $SD = 3.6$).

Therefore, H2 is partially supported: although total human task time was not significantly influenced by algorithm condition, the total robot task time, human idle time, and human idle percentage were all significantly affected by which algorithm was running on the robot. The robot task time was slower in the POMDP condition, but human idling was significantly reduced by the POMDP algorithm.

To evaluate H3 (subjective responses), we first conducted a Chronbach’s alpha test to assure that the eight survey questions were internally consistent. The four questions asked in the negative (e.g., “I’m dissatisfied with how HERB and I worked together”) were reverse coded so their scales matched the positive questions. The result of the test showed high consistency ($\alpha = 0.849$), so we proceeded with our analysis by averaging together the participant ratings across all eight questions.

During the experiment, participants sometimes saw collisions with the robot. We predict that collisions will be an important covariate on the subjective ratings of the three algorithms. In order to account for whether a collision occurred on each trial in our within-subjects design, we cannot conduct a simple repeated measures ANOVA. Instead, we conduct a linear mixed model analysis, with average rating as our dependent variable; algorithm (control, predict-then-act, and POMDP), collision (present or absent), and their interaction as fixed factors; and algorithm condition as a repeated measure and participant ID as a covariate to account for the fact that participant ratings were not independent across the three conditions. Table I shows details of the scores for each algorithm broken down by whether a collision occurred.

	No Collision		Collision	
	mean rating (SD)	N	mean rating (SD)	N
Control	5.625 (1.28)	14	4.448 (1.23)	12
Predict-then-act	5.389 (1.05)	18	4.875 (1.28)	8
POMDP	5.308 (0.94)	26	—	0

TABLE I: Subjective ratings for each algorithm condition, separated by whether a collision occurred during that trial.

We found that collision had a significant effect on ratings ($F(1, 47.933) = 6.055$, $p = 0.018$), but algorithm did not ($F(1, 47.933) = 0.312$, $p = 0.733$). No interaction was found. In other words, ratings were significantly affected by whether or not a participant saw a collision, but not by which algorithm they saw independent of that collision. Therefore, H3 is not directly supported. However, our analysis shows that collisions lead to poor ratings, and our results above show that the POMDP algorithm yields fewer collisions. Therefore, these results support our approach to improving HRC.

V. CONCLUSIONS

This paper presents a new approach for real-time human-robot collaboration. The human and the robot work to achieve a predefined set of goals. The method infers a probability distribution over the human’s goal based on their motions. This probability distribution is used by a POMDP to select actions for achieving a robot goal that does not interfere with the human’s inferred goal. Compared to existing methods, this approach enables us to take actions without committing to a human goal, and continuously replan trajectories as we receive new observations.

We performed a user study in order to assess the fluency and efficiency of our approach. We found that the proportion of time spent in collision, proportion of time spent idling, and total time spent idling were significantly reduced by the use of our algorithm. Our analysis of subjective responses indicated that collisions, which our algorithm had the fewest of, had a significant effect on ratings. However, independent of collisions, ratings were not significantly affected by the algorithm.

This work can be extended in several directions. First, we can incorporate a model of how the human infers the robot's goal into our framework. Using these models for motion generation had led to more fluent collaboration [37], and we can use similar models to predict the effect of robot motion on which goal the human chooses. Second, the applicability of the approach can be considerably extended introducing preconditions among tasks, i.e., partial-order plans based on precedence constraints among the tasks. The approach will be extended in this direction, preserving a satisfactory level of quality in human-robot collaboration. Finally, the prediction of human goals could be modified taking into account multiple human subjects working in the same workspace.

ACKNOWLEDGMENT

The research has been partially funded by European Project FourByThree (H2020-FoF-06-2014), DARPA SIMPLEX (through ARO contract #67904LSDRP), NSF CPS (#1544797), NSF NRI (#1227495), and NSF GRFP (#DGE-1252522).

REFERENCES

- [1] R. Alami, R. Chatila, A. Clodic, S. Fleury, M. Herrb, V. Montreuil, and E. a. Sisbot, "Towards human-aware cognitive robots," in *National Conference on Artificial Intelligence - AAAI*, 2006, p. 8.
- [2] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, 2005.
- [3] N. Pedrocchi, M. Malosio, and L. Molinari Tosatti, "Safe Obstacle Avoidance for Industrial Robot Working Without Fences," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3435–3440.
- [4] J. Mainprice and D. Berenson, "Human-Robot Collaborative Manipulation Planning Using Early Prediction of Human Motion," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 299–306, 2013.
- [5] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *IEEE International Conference on Intelligent Robots and Systems*, p. 2071, 2013.
- [6] K. Hawkins, N. Vo, S. Bansal, and A. Bobick, "Probabilistic Human Action Prediction and Wait-sensitive Planning for Responsive Human-robot Collaboration," in *IEEE 13th International conference on humanoid robots*, Atlanta, GA, 2013, pp. 499–506.
- [7] T. Arai, R. Kato, and M. Fujita, "Assessment of operator stress induced by robot collaboration in assembly," *CIRP Annals - Manufacturing Technology*, vol. 59, no. 1, pp. 5–8, 2010.
- [8] D. Kulic and E. A. Croft, "Estimating intent for human-robot interaction," in *IEEE Int. Conference on Advanced Robotics*, 2003.
- [9] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 625–632.
- [10] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 4569–4574.
- [11] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," *2009 IEEE International Conference on Robotics and Automation*, 2009.
- [12] A. D. Dragan and K. C. T. Lee, "Legibility and Predictability of Robot Motion," in *International Conference on Human-Robot Interaction (HRI)*, vol. 1, 2013.
- [13] S. Javdani, S. Srinivasa, and J. A. D. Bagnell, "Shared Autonomy via Hindsight Optimization," *Robotics: Science and Systems*, 2015.
- [14] B. D. Ziebart, A. Maas, J. A. D. Bagnell, and A. K. Dey, "Human Behavior Modeling with Maximum Entropy Inverse Optimal Control," *AAAI Spring Symposium*, pp. 92–97, 2009.
- [15] A. Dragan and S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.
- [16] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.
- [17] D. Vanhooydonck, E. Demeester, M. Nuttin, and H. V. Brussel, "Shared control for intelligent wheelchairs: an implicit estimation of the user intention," in *Proceedings of the ASER International Workshop on Advances in Service Robotics*, 2003.
- [18] M. A. Goodrich and D. R. O. Jr., "Seven principles of efficient human robot interaction," in *IEEE Transactions on Systems, Man, and Cybernetics*, 2003.
- [19] S. Green, M. Billingham, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *International Journal of Advanced Robotic Systems*, vol. 5, 2007.
- [20] B. Ziebart, A. Maas, J. A. D. Bagnell, and A. Dey, "Maximum Entropy Inverse Reinforcement Learning," in *Proceedings of AAAI*, 2008, pp. 1433–1438.
- [21] G. Best and R. Fitch, "Bayesian Intention Inference for Trajectory Prediction with an Unknown Goal Destination," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Sept. 28 - Oct. 2, 2015, pp. 5817–5823.
- [22] L. E. Kavragi and J.-C. Latombe, "Probabilistic roadmap for robot path planning," in *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, J. Wiley, Ed., 1998, pp. 33–53.
- [23] Z. Wang, K. Mülling, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human-robot interaction," *The International Journal of Robotics Research*, 2013.
- [24] Y. Jiang and A. Saxena, "Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent CRFs," *Robotics: Science and Systems*, pp. 1–8, 2014.
- [25] B. Ziebart, N. Ratliff, G. Gallagher, and K. Peterson, "Planning-based prediction for pedestrians," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.
- [26] D. Ferguson and A. Stentz, *Robotics Research: Results of the 12th International Symposium ISRR*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ch. Field D*: An Interpolation-Based Path Planner and Replanner, pp. 239–253.
- [27] T. Bandyopadhyay, K. Won, E. Frazzoli, and D. Hsu, "Intention-Aware Motion Planning," *Proc. of the 10th Workshop on the Algorithmic Foundations of Robotics*, pp. 475–491, 2012.
- [28] A. B. Karami, L. Jeanpierre, and A. I. Mouaddib, "Partially observable markov decision process for managing robot collaboration with human," *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, pp. 518–521, 2009.
- [29] —, "Human-robot collaboration for a shared mission," *Human-Robot Interaction (HRI)*, pp. 155–156, 2010.
- [30] T.-H. D. Nguyen, D. Hsu, W.-S. Lee, T.-Y. Leong, L. P. Kaelbling, T. Lozano-Pérez, and A. H. Grant, "Capir: Collaborative action planning with intention recognition," in *Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.
- [31] O. Macindoe, L. P. Kaelbling, and T. Lozano-Pérez, "Pomcop: Belief space planning for sidekicks in cooperative games," in *Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012.
- [32] P. A. Lasota and J. A. Shah, "Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 57, no. 1, pp. 21–33, 2015.
- [33] M. Littman, A. Cassandra, and L. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning*, 1995.
- [34] E. K. P. Chong, R. L. Givan, and H. S. Chang, "A framework for simulation-based network control via hindsight optimization," in *IEEE Conference on Decision and Control*, 2000.
- [35] S. W. Yoon, A. Fern, R. Givan, and S. Kambhampati, "Probabilistic planning via determinization in hindsight," in *AAAI Conference on Artificial Intelligence*, 2008.
- [36] S. Srinivasa, D. Berenson, M. Cakmak, A. C. Romea, M. Dogar, A. Dragan, R. A. Knepper, T. D. Niemueller, K. Strabala, J. M. Vandeweghe, and J. Ziegler, "Herb 2.0: Lessons learned from developing a mobile manipulator for the home," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 1–19, July 2012.
- [37] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa, "Effects of Robot Motion on Human-Robot Collaboration," *Human Robot Interaction*, 2015.