Predicting Human Reaching Motion in Collaborative Tasks Using Inverse Optimal Control and Iterative Re-planning

Jim Mainprice[1], Rafi Hayne[2], Dmitry Berenson[2]
{*jim.mainprice@tuebingen.mpg.de , rhhayne@wpi.edu, dberenson@cs.wpi.edu*}
[1] *Max-Planck-Institute for Intelligent Systems, Autonomous Motion Department, Paul-Ehrlich-Str. 15, 72076 Tbingen, Germany*
[2]*Robotics Engineering Program, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609.*

***Abstract*— To enable safe and efficient human-robot collaboration in shared workspaces, it is important for the robot to predict how a human will move when performing a task. While predicting human motion for tasks not known a priori is very challenging, we argue that single-arm reaching motions for known tasks in collaborative settings (which are especially relevant for manufacturing) are indeed predictable. Two hypotheses underlie our approach for predicting such motions: First, that the trajectory the human performs is optimal with respect to an unknown cost function, and second, that human adaptation to their partner's motion can be captured well through iterative re-planning with the above cost function. The key to our approach is thus to learn a cost function which "explains" the motion of the human. To do this, we gather example trajectories from two participants performing a collaborative assembly task using motion capture. We then use Inverse Optimal Control to learn a cost function from these trajectories. Finally, we predict a human's motion for a given task by iteratively re-planning a trajectory for a 23 DoFs human kinematic model using the STOMP algorithm with the learned cost function in the presence of a moving collaborator. Our results suggest that our method outperforms baseline methods and generalizes well for tasks similar to those that were demonstrated.**

I. Introduction

Human-robot collaboration has become a popular research area in recent years due to the difficulty of automating tasks such as electronics or aircraft assembly. In such cases the human and the robot workers must adapt to each other's decisions and motions. In this paper we address an important step toward more fluid human-robot collaboration: the ability to predict human motion in collaborative settings.

A great deal of work in the fields of neuroscience [1], [2], [3] and biomechanics [4] has sought to model the principles underlying human motion. However, human motion in environments with obstacles has been difficult to characterize. Furthermore, human motion in collaborative tasks where two humans share a workspace is difficult to model due to unclear social, interference, and comfort criteria. While some of these principles have been studied in the context of human *navigation* [5], to our knowledge there is no framework for predicting human motion in collaborative *manipulation* tasks.

This paper describes our efforts toward creating such a framework. Predicting human motion in an open-ended collaborative context (where the tasks are not known a-priori) would require a very general model of how humans move in close proximity to one another, which we believe to be very difficult to obtain. Instead, we propose a method for predicting human motion for single-arm reaching in a collaborative context where the task is known. Though the range of applications of our methods is restricted, we argue that this is an important category of motions to be able to predict, since many pick-and-place tasks in manufacturing fall into this category. Being able to predict these motions well can move us closer to enabling safe and efficient human-robot collaborative manipulation in manufacturing.

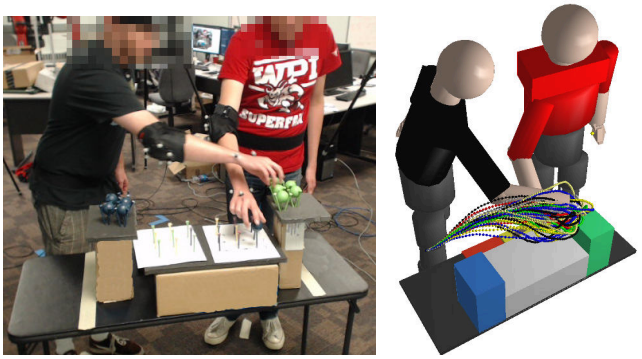


Fig. 1. Shared workspace assembly experiment (left) and sampling of collision free manipulation motions for Inverse Optimal Control (right).

Our approach to predicting human motion in these settings is based on studying how two humans collaborate in a shared workspace (as in Figure 1). We acknowledge that a human's motion in human-robot collaboration may differ from his/her motion when collaborating with a human partner; e.g. the human may be afraid of the robot and stay farther away from it than he/she would from another person. However we believe that studying how two humans collaborate gives us an important baseline against which human-robot collaborations can be judged; if we can predict what a natural motion for a human is in a given collaborative context, we can judge when the human deviates significantly from that motion in response to a robot's actions. Also, the method we present here for learning a cost function that describes human motion can be used to learn the cost function for a human-robot pair if the features of the cost function can be adjusted to account for the robot's kinematics (which would be straightforward for an anthropomorphic robot).

This work is supported in part by the Office of Naval Research under Grant N00014-13-1-0735 and by the National Science Foundation under Grant IIS-1317462.

Our approach is based on two hypotheses about collaborative human motion: 1) The trajectory the human performs is optimal with respect to an unknown cost function, and 2) Human adaptation to their partner's motion can be captured well through iterative re-planning of a trajectory which is locally-optimal with respect to the same cost function. Our method thus seeks to learn a cost function for which the human's motion is locally-optimal from training data.

To gather training data, we record the motion of two humans performing a collaborative task using a motion capture system and then manually segment that recording into individual reaching motions. These reaching motions, along with a set of feature functions encoding trajectory smoothness and distance relationships between the humans are used as input for the Path Integral Inverse Reinforcement Learning (PIIRL) algorithm [6]. PIIRL produces a weighting for the feature functions that captures their relative importance. The learned cost function is then a weighted sum of the feature functions using the learned weights. To predict human motion we input the learned cost function into the STOMP algorithm [7], which we adapt for iterative motion re-planning in a dynamic environment.

In our experiments we gathered the training data from a pair of participants in a structured assembly task (see Figure 1). We found that we are able to capture a cost function for collaborative reaching motions that outperforms baseline methods in terms of generalizing to unseen reaching examples. We also found that re-planning was more effective than single-shot planning for capturing a human's adaptation to their partner's motion in cases where the motion of the two participants interfered significantly. While these results are preliminary and the method needs to be evaluated with a broader human-subjects study, the initial results are compelling since we are able to predict human motion well for these tasks given a training set of only seven trajectories.

The remainder of this paper is structured as follows: In the next section we give a description of related work. In Section III we describe the approach that enables us to recover the cost function from training data. In Section IV, we present the experimental setup used to gather collaborative reaching motions. In Section V, we present results that illustrate the ability of our method to predict collaborative reaching motions.

II. Related Work

Our work contributes to the field of autonomous robot manipulation in the presence of humans, by creating a method to predict human motion which could be used onboard a robot. In our prior work [8], we have incorporated early prediction of human motion with an iterative motion re-planning approach to generate efficient robot motions. However the prediction based on Gaussian Mixture Models (GMM), which is a commonly used technique in gesture recognition, is limited to a set of known tasks in a structured environment. Similarly to our approach Koppula et al. have integrated prediction of 3D trajectories of the human hand [9] in the robot planning using Conditional Random Fields (CRFs) to model affordances of objects in the scene. This work has been recently extended in [10] to predict high-dimensional trajectories but does not account for dynamic environments nor collaborative tasks as we aim to do in this work. Hidden Markov Models (HMMs) are another popular stochastic modeling technique for human motion prediction. In [11], Kuliç et al. describe an approach for on-line, incremental learning of full body motion primitives from observation of human motion encoded using HMMs, so that the same model can be used for both motion recognition and motion generation. These graphical model representations (i.e., GMMs, CRFs and HMMs), allow to encode efficiently relationships such as those between activities, objects and motions, but do not capture well obstacles constraints, which we address in this work.

The underlying principles of human motion have been investigated in terms of muscle activation and neural activity [1], [12], [4], but there is no study providing a model of human motion handling obstacles in collaborative manipulation tasks. A detailed subject-customized bio-mechanical model has been used in [13] to efficiently reconstruct a subject motion dynamics from motion capture data in realtime using a whole-body control approach. Many experiments investigating reaching under various conditions [1], [12] suggest that human motor-behavior is determined by the minimization of a cost function used to weight different movement options to a task, and to select a particular solution. Stochastic Optimal Control provides a framework to model such motor-behavior, while taking into account motor noise inherent to sensorimotor control [3]. In [14], Rigoux and Guigon describe a model derived from the maximization of the discounted weighted difference between expected rewards and foreseeable motor efforts, relevant to address the neural bases of decision making and motor control. Recently in [15], Ganesh and Burdet showed on a manipulation task, that the central nervous system uses a motion planning phase with multiple plans, and a memory mechanism. While this suggests that motion planning plays an important role in explaining human motion, to the best of our knowledge, Inverse Optimal Control of human reaching-motion has only been proposed in [16], where the authors present a method for transferring reaching behaviors from humans to robots. The authors were able to capture the complex, non-linear dynamics of the human musculoskeletal system, nonetheless the system was demonstrated on the control of a ball hitting task and did not consider workspace obstacles.

The Inverse Optimal Control (IOC) problem, occasionally named Inverse Reinforcement Learning (IRL), is the problem of finding the cost or reward function that an agent optimizes when computing a trajectory or policy given a set of demonstrated solutions. It is usually framed in the context of a Markov Decision Processes. IRL was introduced by Ng et al. in [17], who proposed two algorithms for discrete and continuous states spaces. Later apprenticeship learning [18] introduced the notion of margin maximization between the cost of the demonstration and other solutions. Apprenticeship learning consists of solving iteratively the

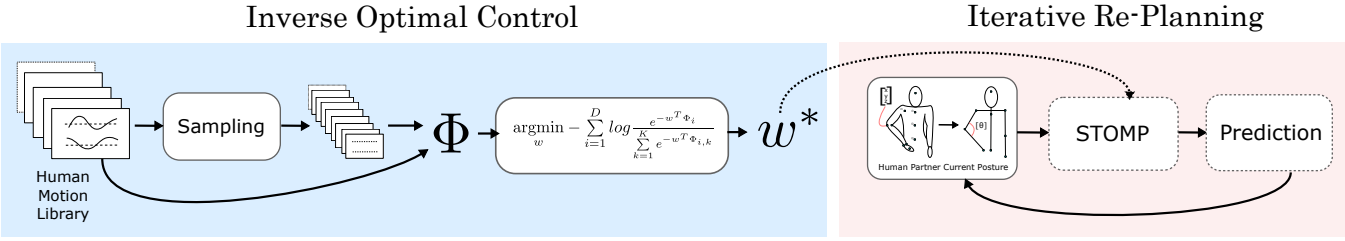


Fig. 2. Data flow through the system. The gathered human-motion library is used to generate sample trajectories. Features (Φ) are then computed for the demonstrated and sampled trajectories. The PIIRL algorithm is then applied to generate a weight vector w^*. Prediction of collaborative human reaching motions can then be performed by an iterative re-planning algorithm based on STOMP, relying on the learned weight vector w^* and a kinematic model of the human.

forward problem, modifying the weights at each iteration. In [19], Zeibart et al. proposed an approach to IRL based on the maximum entropy principle. The methods based on this principle [20], [21], [22] do not require solving the forward problem and allow handling high-dimensional continuous state spaces. Instead of a solution to the forward problem they proceed by sampling trajectories. Our approach is based on the algorithm introduced in [6], which is based on a similar principle and only requires local optimality of the demonstrated trajectories.

We rely on recent developments in trajectory optimization for motion planning [23], [7] to compute low-cost motion predictions. Our framework uses the Stochastic Trajectory Optimizer for Motion Planning (STOMP) algorithm, which has proven effective for the type of manipulation motion planning we consider [7]. Recently, STOMP was adapted to run faster than real-time [24], and we plan to employ this new method in future work.

III. Approach

Our approach to predicting human motion in collaborative manipulation tasks consists of two phases (see Figure 2). First we gather a library of collaborative motions. We then segment the motions into elementary reaching motions (i.e., from a resting configuration to a grasping configuration). These motions are then used as demonstrations by the IOC algorithm to learn a cost function where these demonstrations are optimal. Finlay, we use the learned cost function inside an interative motion-planner to predict how the human will move.

A. Inverse Optimal Control algorithm

Human upper-body motions can be represented as time-parametrized curves τ in the human's configuration space. Because these motions are inherently high-dimensional (in this work we consider 23 DoFs), global optimality is intractable. Hence we use the *Path Integral Inverse Reinforcement Learning* (PIIRL) algorithm [6], which can deal with high-dimensional continuous state-action spaces, and only requires local optimality of the demonstrated trajectories.

The original IOC problem solved by PIIRL aims to recover a cost function composed of a control cost term and a general cost term (i.e., configuration dependent) that can be combined with a terminal cost term, which we do not use here. In our formulation of the problem, we consider linearly-parametrized cost functions where each elementary feature function is user defined. Thus each feature function penalizes motions that do not respect an associated property, see Section III-D for a description of the features we use.

The cumulative cost $C(\tau)$, and feature count $\Phi(\tau)$ of a trajectory are defined as follows:

$$C(\tau) = w^T \Phi(\tau) \, , \Phi(\tau) = \begin{bmatrix} G(\tau) \\ A(\tau) \end{bmatrix},$$

where Φ is a multi-valued feature function defined by the user, w are weights associated with the features, which the algorithm attempts to learn. A is a term enforcing smoothness (i.e., control cost) and G a general term of the form:

$$G(\tau) = \int_{t=0}^{T} \phi(q_t) \, \mathrm{d}t \simeq \sum_{i=1}^{N} \phi(q_i) \delta t,$$

where q_i is the configuration at index i along the trajectory and N the number of waypoints.

PIIRL samples trajectories with low smoothness features around each demonstration by sampling from a Multivariate Gaussian distribution with covariance R^{-1} centred at the demonstration (for a definition see [7]). Note that in the sampling phase, trajectories that collide with the environment are discarded by performing collision detection.

The weights are then obtained by solving the following convex minimization problem:

$$w^* = \underset{w}{\text{argmin}} - \sum_{i=1}^{D} log \frac{e^{-w^T \Phi_i}}{\sum_{k=1}^{K} e^{-w^T \Phi_{i,k}}},$$

where D is the number of demonstrations and K the number of samples per demonstration.

In the original version of PIIRL, a penalty on the L_1 norm of the weight vector w was added to the loss function to achieve learning with a large set of features. In this case the loss function is still convex but non differentiable due to the regularization term. In order to handle this non linearity, the Orthant-Wise Limited-memory Quasi-Newton [25] algorithm was used, which introduces additional projection steps and constrains the search to one orthant at a time. Using a regularization term adds a supplementary parameter to the algorithm that can be tuned through cross validation, however we found the results to be sparse enough without this regularization term (see Section V).

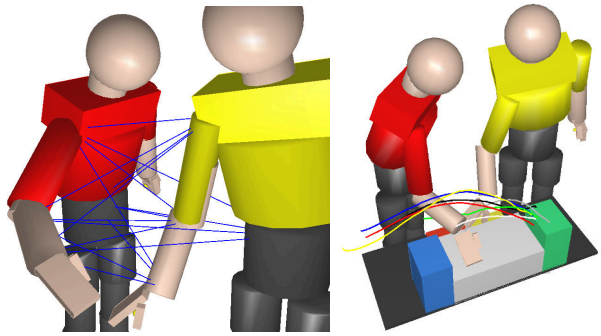


Fig. 3. Each line corresponds to a distance used in the feature vector (left). 3D model of the experiment used for collision checking with hand trajectories of the seven demonstrations used in the result section (right).

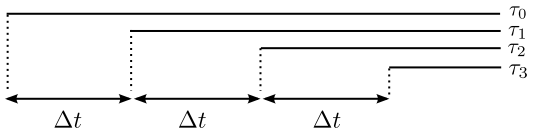


Fig. 4. Division of a demonstration τ_0 into smaller segments

B. Iterative re-planning

Iterative re-planning consists of planning iteratively while considering the current environment as static. It is a common approach to accounting for dynamic obstacles in robot motion planning [26], [24]. Typical approaches either maintain a tree or graph of collision-free motions, which is updated at each replanning step, or deform the current trajectory locally given the updated positions of obstacles in the world. Our approach aims to recover a cost function that can be used for such a framework. Thus, once the library of collaborative motion trajectories is gathered, it is segmented manually into elementary manipulation motions, which are then cut into smaller segments by advancing Δt along each demonstration τ_0 as depicted in Figure 4. The newly generated sub-segments are added to the demonstration trajectory set. For each segment the initial velocity $\dot{q_0}$, acceleration $\ddot{q_0}$ and jerk $\dddot{q_0}$, as well as the configuration of the other human and the positions of obstacles are used to compute the features for that segment and for its corresponding sample trajectories.

When planning with the human model, we make use of the STOMP algorithm [7], which is a trajectory optimizer that iteratively deforms an initial solution by stochastically estimating the gradient in trajectory space. It internally represents the trajectory by an m by n matrix, where m is the number of DoFs and n the number of waypoints. At each iteration, trajectories are sampled from a Multivariate Gaussian distribution with covariance R^{-1} (see [7]), the general and control costs of the sampled trajectories are combined to generate the update. Thus it does not require the analytical gradient of the cost function to be known, and generally converges to a local minimum within 100 iterations.

The original STOMP algorithm presented in [7] optimizes a combination of obstacle and smoothness cost. The first is estimated by summing a penetration cost for a set of bounding spheres in the obstacles at every waypoint using a signed Euclidean Distance Transform (EDT). Note that the weight of this cost is manually tuned in our result section. The second is estimated by summing the squared accelerations along the trajectory using finite differencing. In order to account for the other human, we sum a third cost criterion defined in the next section. The smoothness cost is defined differently from [6], it is also described in the next section. In order to account for smoothness between each replanning step, a buffer of configuration waypoints from the previous replanning step is used to compute velocity, acceleration and jerk at the initial configuration.

C. Human kinematic model description

We model the human kinematics following the recommendation for joints coordinates in [4]. The model is composed of translational and rotational joints. In our experiments we only account for upper body and right arm motions, which totals 23 DoFs. Three translations and three rotations are used for the pelvis, three rotations for the torso joint, three translations followed by three rotations for the shoulder joint, one translation followed by three rotations for the elbow, one translation followed by three rotations for the wrist joint.

When predicting motions using STOMP the bounds of the translational joints are set using the minimal and maximal values observed in the motion capture data. These translations are used to compensate for errors in the computation of joint centers arising from marker placement errors. They are also useful for addressing the approximations we make in modeling the human kinematics.

D. Feature functions

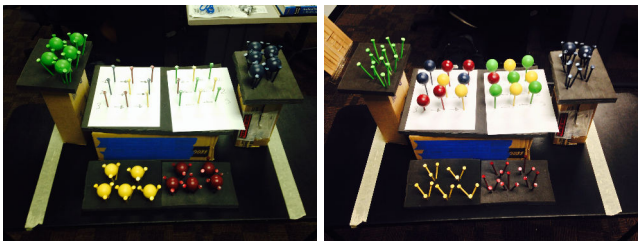
We consider variants of feature functions that have been introduced in previous work to account for human-robot interaction constraints [27], [28], [5]. We make use of two types of features inspired by the *proxemics* theory [29] and experiments in neuroscience [1]:

1) Distances between human links: The goal of these features is to avoid collision risks. However, in situations requiring close interaction (e.g., reaching over the other person to access an object), two people may come close to one another. To model this avoidance behavior we consider 16 pairwise distances (see Figure 3) along the arm and pelvis between the two humans (i,e,. wrist, elbow, shoulder, pelvis).

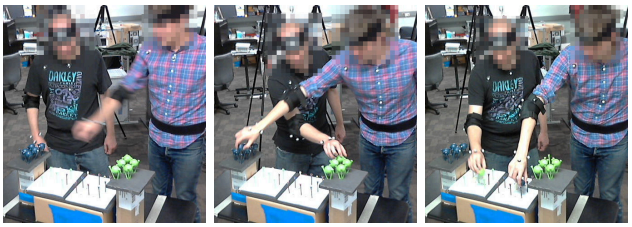
2) Smoothness: These features ensure that the trajectory remains smooth. We measure configuration and task space length, squared velocities, squared accelerations and squared jerks along the trajectory using finite differencing.

IV. Experiments

The experiment we designed to gather training and test data consisted of two participants standing shoulder to shoulder parallel to a table; each working on an individual task within a shared workspace (Figure 5). In order to execute their task, the participants must place colored balls on pegs of the corresponding color, which were placed in a specified order (Figure 5(a)). Adhesive tape was placed



(a) Initial state (b) End state



(c) t = 0.0 sec (d) t = 0.5 sec (e) t = 1.0 sec

Fig. 5. Experiment design (top) and motion capture of the task (bottom).

on the pegs allowing quick and easy placement. The aim of our experiment is to simulate a packing task, for instance packing different chocolates into a sampler box.

A. Experiment flow

The participants look at the color of the first empty peg in their plan, pick up a ball from the corresponding color zone, and place the ball on top of the peg until all pegs in the plan are filled with balls (Figure 5(b)). Following a predetermined order of execution denies the participants the ability to switch tasks in mid-motion. This allows us to study the manipulation planning component of human motion in isolation. In future work, we will investigate our results with a task planner and allow the pegs to be filled in any order.

B. Recording method

In order to record these interactions, we used a Vicon motion capture system consisting of eight Bonita cameras. Subjects wore a suit consisting of three rigid plates and nine markers which had been placed according to standards in use in the field of biomechanics [4]. Our full marker set (seen in Figure 5(c)) consists of a waist-belt and headband attached to rigid objects, a marker on the back of the hand, two on each side of the wrist, an elbow pad, two markers on either side of the shoulder, and two markers straddling both the sternum and xyphoid process. This set of markers allows us to easily find the center of rotation of the wrist, elbow, shoulder and torso. From these joint centers, we obtain a 23 DoFs configuration of the right arm and torso for each participant using analytical inverse kinematics.

Recording fluid collaborative motion can be difficult when one participant occludes the other from the Vicon cameras. A joint defined by a pair of markers becomes occluded if one of the markers becomes occluded. Upon noticing frequent occlusions of the elbow in our tests, we switched from a marker pair to an elbow pad with multiple markers.

The tracker then labels each marker in a known calibration pose wherein the subject stands upright with their hands rested comfortably at their side. After each update from the Vicon system, marker indices are matched by closest distance to the previous frame. If a marker label cannot be found within a threshold distance, it is considered occluded, and the missing marker is filled in with data from the previous update.

V. Results

In this section we present results illustrating the capacity of the algorithm to recover a cost function using distance between links and smoothness features. We define an active human, whose motion trajectories are used as demonstrations, and a passive human, whose trajectories serve as contexts for the IOC and for the iterative re-planning prediction tests.

First we validate the approach by planning with a manually defined weight vector and measuring the cost difference between the initial trajectory and the recovered trajectory. This experiment gives us a rough estimate for tuning the number trajectory samples per demonstration in PIIRL. We then evaluate the capacity of the IOC algorithm to predict human motion by comparing the demonstrations to motions computed using our framework (i.e., with a weight vector generated by PIIRL on these demonstrations). We then show the ability of the predictions to generalize to new situations by performing leave-one-out testing over seven motions. We compare the results to two baseline methods. Finally we demonstrate the usefulness of the re-planning approach on a particularly difficult motion.

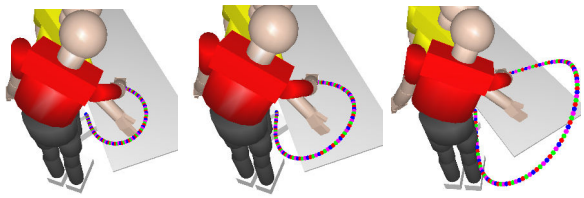


Fig. 6. Three trajectories computed using the STOMP motion planner. (a) Trajectory planned with a user-given weighing of features. (b) Trajectory planned with weights recovered by PIIRL. (c) Trajectory planned with a random weight vector.

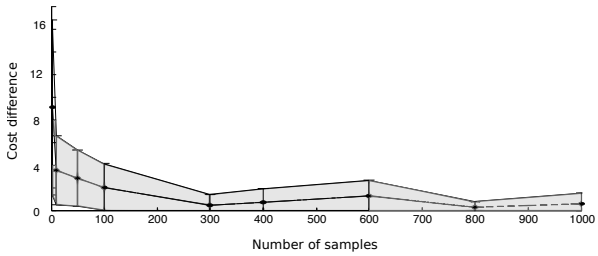


Fig. 7. Average difference in cost and standard deviation, function of the number of samples used by PIIRL. The difference in cost compares solutions planned on the example of Figure 6 using the recovered weight vector and the original weight vector.

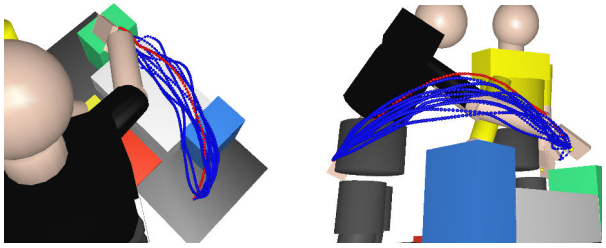


Fig. 8. Three trajectories computed using the STOMP motion planner. Demonstrations in red and 10 predicted motion trajectories given the start and goal configuration in blue.

	Joint center distances				Task space			
Run	μ	σ	min	max	μ	σ	min	max
1	33.17	3.35	28.41	40.33	44.44	2.27	41.56	49.69
2	37.20	3.57	32.39	43.91	53.08	2.17	49.50	56.13
3	53.37	7.17	40.36	64.42	76.26	4.22	70.08	82.28
4	20.95	2.28	17.94	25.12	31.39	1.00	29.69	33.09
5	21.59	1.84	17.58	23.75	26.50	1.13	25.03	28.44
6	10.12	1.23	7.83	12.11	23.62	0.85	22.46	25.00
7	25.56	5.09	16.91	34.30	45.69	3.45	40.18	52.38
All	28.85	3.50	23.06	34.84	42.99	2.15	39.78	46.71

TABLE I

DTW PERFORMED BETWEEN THE SEVEN DEMONSTRATIONS AND THE TRAJECTORIES PLANNED ON A STATIC ENVIRONMENT, RESULTS ARE AVERAGED OVER 10 RUNS

A. Validation

We first planned a trajectory using the STOMP algorithm [7] in a static environment with no replanning and a user input weight vector (see Figure 6). We then used PIIRL presented in Section III-A to recover a weight vector using this planned trajectory as a demonstration. Figure 7 shows the difference in cost between the "demonstrated" trajectory and the trajectories planned using the recovered weights as the number of samples PIIRL considers increases. We use the original weight vector to assess the difference in cost to measure how close the trajectories obtained under the learned weights are to the locally optimal solution. The results are averaged over 10 runs. As one can see, the mean and standard deviation decrease as the number of samples increases, which indicates the capacity of the algorithm to recover cost functions for the type of reaching motions we consider. These results were used to select the number of sample trajectories, it is set to 700 in the rest of the results presented in this section.

B. Predicting human motion

In order to validate the capacity of the approach to predict human motion, we ran the experiment described in Section IV to gather collaborative motions. The motions were then segmented manually into seven elementary manipulation motions (i.e., from a resting posture to a grasping posture).

The seven motions start and end in the same area (see Figure 9). We then run PIIRL without the segmenting phase described in Section III-B, using these seven demonstrations to generate seven weights w^*, one for each demonstration. This set of weights is then used to plan for motion-trajectory

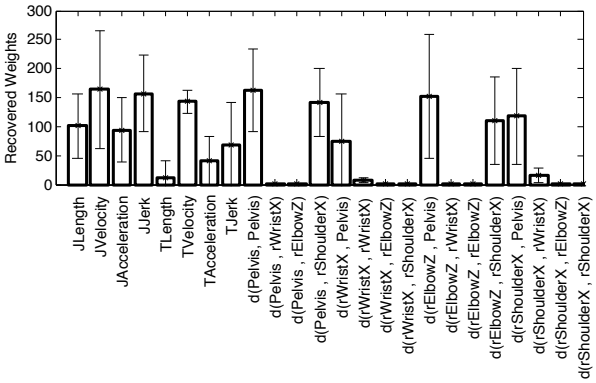


Fig. 10. Average weight vector and standard deviation from the leave-one-out testing performed over the seven motions of Figure 9. The distances are described with active human in the bottom.

predictions using our motion planning framework. The start and goal configurations are set to the ones from the manually segmented demonstrations-trajectories. However they could be set using biomechanics-based inverse-kinematics [30]. We report the results of Dynamic Time Warping (DTW) comparison between the resulting trajectory predictions and the demonstrations and in Table I.

DTW is an algorithm for measuring similarity between two temporal sequences that may vary in time or speed. It relies on a distance metric between waypoints. We use two metrics throughout this section: sum of joint center distances and task space distances. We do not report the configuration space metric as it does not give a fair estimate due to the high redundancy of our kinematic model, which represents the elbow and wrist as ball joints. The joints considered in the first metric are the pelvis, torso, shoulder, elbow and wrist. The task space metric combines Euclidean distance and angle between consecutive Quaternions. Figure 8, shows demonstrations 1 and 2 along with the corresponding motion trajectories predicted by the motion planner. Comparing trajectories is known to be a difficult problem. The values in Table I and visualization in Figure 8 provide a reference point for what DTW values we can expect for visually similar trajectories.

C. Leave-one-out testing

To evaluate the capacity of our predictions to generalize to new situations we have performed a leave-one-out testing over the seven motions. The demonstration trajectories were cut into smaller segments using the procedure described in Section III-B, with a $\Delta t = 0.1sec$. Resulting in 33 demonstrations used for IOC. The obtained average, and standard deviation values of the weights are shown in Figure 10.

The obtained weights indicate the importance of smoothness features rather than distance features. The distances are measured between the active human (on the left) and passive human (on the right). All distances between the active human's arm and passive's pelvis are important, as well as between the active's elbow and passive's shoulder. The high

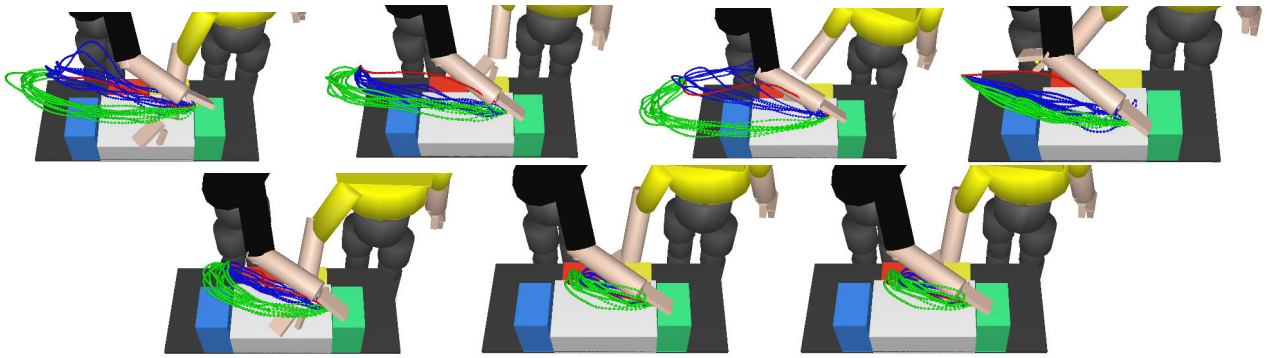


Fig. 9. Seven demonstrations (red) along with the ten trajectories planned with *baseline 0* (green) and with the weight vector obtained by IOC (blue).

	Re-planning								No Re-planning							
	Joint center distances				Task space				Joint center distances				Task space			
Method	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max
baseline 1	81.01	3.88	74.48	87.43	63.11	2.51	58.70	67.85	64.96	3.94	58.21	71.03	57.78	2.69	53.86	62.57
baseline 0	39.55	3.34	33.83	45.72	48.04	1.89	45.29	51.45	34.06	3.72	28.60	39.78	45.81	1.91	42.60	49.08
With IOC	35.34	4.78	27.36	43.22	43.56	4.21	37.03	50.46	30.98	4.27	24.12	39.04	42.18	3.17	37.06	46.90

TABLE II

DTW PERFORMED BETWEEN THE SEVEN DEMONSTRATIONS AND THE TRAJECTORIES PLANNED WITH AND WITHOUT RE-PLANNING, RESULTS ARE AVERAGED OVER 10 RUNS.

weight values corresponding to the distances of the active's pelvis and passive's body links do not impact the overall motion as participants do not move the pelvis as much as the arm during manipulation. Note that during the sampling and planning phase the pelvis translation bounds are set to the minimal and maximal values observed in the demonstrations, which constrains the pelvis motions to remain within these bounds.

For comparison, trajectories were also generated using two baseline methods:

- Conservative tuning (*baseline 1*): the squared velocities and the 16 link distances manually tuned to the same value.
- Aggressive tuning (*baseline 0*): the squared velocities without considering the distances between links.

Table II summarizes the DTW similarity values using the joint center distance and the task space metric, for all methods with and without replanning. DTW is computed between the respective demonstrations (i.e., from which the start and goal configurations are extracted) and the planned trajectories. In the "no re-planning" version STOMP only considers the initial configuration of the passive human, and thus the active human motion is not updated according to the passive human motion.

Trajectories planned with *baseline 0* and with the IOC recovered weights have lower DTW scores than the ones planned with *baseline 1*, this is consistent throughout both metrics and with or without replanning. Trajectories planned with *baseline 0* sometimes outperform trajectories planned with the IOC recovered weights, which can be explained by the sparsity of the distance weights recovered by PIIRL (*baseline 0* does not consider distances between links at all). Nevertheless, the average DTW values of the IOC recovered cost functions version are lower than the *baseline 0* version's, indicating the validity of using IOC compared to hand tuning the weight vector. This can be seen on Figure 9, which shows the demonstrations (red), the predictions obtained with the *baseline 0* method (green), and the predictions obtained with the IOC recovered weight vector (blue), here both types of predictions use iterative re-planning.

Note that the average DTW values for the leave-one-out test are close to the values obtained in the validation test (joint distance score: 30.98 compared to 28.85), which shows the capability of our method to generalize to new situations.

The "no re-planning" approach tends to outperform the "re-planning" approach throughout the different methods, but remains close as indicated by the task space values comparison when using the recovered cost function: 43.56 and 42.18 (stddev 4.21 and 3.17) with and without re-planning respectively.

D. Significant interference

To show the capacity of the re-planning approach to better predict human motion in more difficult situations we have selected a motion where the passive human interferes significantly with the active human while he/she is reaching. The weight vector is obtained by training with all seven

Type	μ	σ	min	max
	Joint center distances			
No re-planning	52.89	9.66	39.94	67.09
With re-planning	44.91	6.62	36.15	55.20
	Task space			
No re-planning	49.22	8.25	37.75	63.78
With re-planning	36.20	8.13	24.81	50.77

TABLE III

DTW PERFORMED BETWEEN THE DEMONSTRATION OF FIGURE 11 AND THE TRAJECTORIES PLANNED, RESULTS ARE AVERAGED OVER 10 RUNS

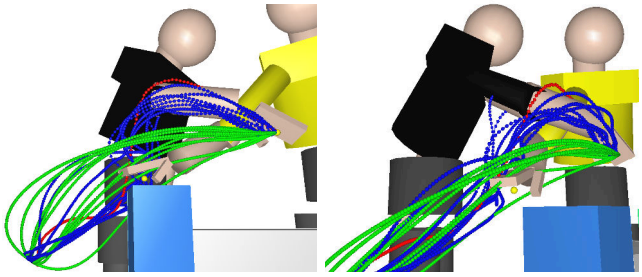


Fig. 11. A demonstration of the benefits of re-planning on a difficult example. Original motion (red) and predicted motions with (blue) and without (green) re-planning.

motions used in the leave-one-out phase, but does not include the "demonstration" from which we extract the start and end configurations for prediction. The motions obtained with and without re-planning are shown in Figure 11, and the DTW results are shown in Table III. In this case, using re-planning better predicts the active human motion because the trajectories generated with no re-planning collide with the arm of the passive human. This result is underscored by the smaller average DTW values found for the joint center distances and task space metric.

VI. Conclusion and Future Work

We have presented an important step toward predicting how humans move when collaborating on a manipulation task by applying inverse optimal control to data gathered from motion capture of collaborative manipulation in a shared workspace.

To demonstrate the feasibility and efficacy of our approach we have provided test results consisting of learning a cost function, and comparing the planned motions using the learned weights to the demonstrations using Dynamic Time Warping (DTW). The approach based on Inverse Optimal Control (IOC) allows us to find a cost function balancing different features that outperforms hand-tuning of the cost function in terms of task space and joint center distance DTW. The method presented in this paper could be extended to allow learning of a cost function for robot motion planning of human-robot collaborative manipulation tasks where the human and the robot manipulate objects simultaneously in close proximity.

Future work concerns enhancing the type of features to be taken into account to improve the prediction, and retargeting these features for motion planning on a PR2 robot.

References

[1] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[2] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, no. 6862, pp. 446–449, 2001.

[3] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature neuroscience*, vol. 5, no. 11, pp. 1226–1235, 2002.

[4] Wu and et al, "Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion – part ii: shoulder, elbow, wrist and hand," *Journal of biomechanics*, vol. 38, no. 5, pp. 981–992, 2005.

[5] 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.

[6] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning objective functions for manipulation," in *ICRA*, 2013.

[7] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *ICRA*, 2011.

[8] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IROS*, 2013.

[9] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," RSS, 2013.

[10] Y. Jiang and A. Saxena, "Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs," in *Robotics: Science and Systems, RSS*, 2014.

[11] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[12] E. Burdet and T. E. Milner, "Quantization of human motions and learning of accurate movements," *Biological cybernetics*, vol. 78, no. 4, pp. 307–318, 1998.

[13] E. Demircan, T. Besier, S. Menon, and O. Khatib, "Human motion reconstruction and synthesis of human skills," in *Advances in Robot Kinematics: Motion in Man and Machine*, pp. 283–292, Springer, 2010.

[14] L. Rigoux and E. Guigon, "A model of reward-and effort-based optimal decision making and motor control," *PLoS computational biology*, vol. 8, no. 10, p. e1002716, 2012.

[15] G. Ganesh and E. Burdet, "Motor planning explains human behaviour in tasks with multiple solutions," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 362–368, 2013.

[16] T. Mori, M. Howard, and S. Vijayakumar, "Model-free apprenticeship learning for transfer of human impedance behaviour," in *International Conference on Humanoid Robots*, 2011.

[17] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *ICML*, 2000.

[18] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML*, 2004.

[19] B. D. Ziebart and et al., "Maximum entropy inverse reinforcement learning.," in *AAAI*, pp. 1433–1438, 2008.

[20] A. Boularias, J. Kober, and J. R. Peters, "Relative entropy inverse reinforcement learning," in *International Conference on Artificial Intelligence and Statistics*, 2011.

[21] N. Aghasadeghi and T. Bretl, "Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals," in *IROS*, 2011.

[22] T. Park and S. Levine, "Inverse optimal control for humanoid locomotion," in *Robotics Science and Systems Workshop on Inverse Optimal Control and Robotic Learning from Demonstration*, 2013.

[23] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *ICRA*, 2009.

[24] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using gpus," in *ICRA*, 2013.

[25] G. Andrew and J. Gao, "Scalable training of l 1-regularized log-linear models," in *ICML*, 2007.

[26] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[27] J. Mainprice, E. Akin Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, "Planning human-aware motions using a sampling-based costmap planner," in *ICRA*, 2011.

[28] J. Mainprice, M. Gharbi, T. Siméon, and R. Alami, "Sharing effort in planning human-robot handover tasks," in *ROMAN*, 2012.

[29] E. T. Hall, "A system for the notation of proxemic behavior," *American anthropologist*, 1963.

[30] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," *Journal of Physiology-Paris*, vol. 103, no. 3, pp. 211–219, 2009.