

Goal Set Inverse Optimal Control and Iterative Re-planning for Predicting Human Reaching Motions in Shared Workspaces

Jim Mainprice¹, Rafi Hayne², Dmitry Berenson²

{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 Tübingen, 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 pairs of 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 reaching motions from the human’s current configuration to a task-space goal region by iteratively re-planning a trajectory using the learned cost function. Our planning algorithm is based on the trajectory optimizer STOMP [1], it plans for a 23 DoF human kinematic model and accounts for the presence of a moving collaborator and obstacles in the environment. Our results suggest that in most cases, our method outperforms baseline methods when predicting motions. We also show that our method outperforms baselines for predicting human motion when a human and a robot share the workspace.

I. INTRODUCTION

Human-robot collaboration is increasingly studied in an industrial context because many tasks (such as electronics or aircraft assembly) are stressful for humans but have proven difficult to automate. In such cases the human and the robot workers must adapt to each others’ decisions and motions. In this paper we address an important step toward more fluid human-robot collaboration: the ability to predict human motion in a shared workspace.

A great deal of work in the fields of neuroscience [2], [3], [4] and biomechanics [5] 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 [6], to our knowledge very few works address the problem of predicting human collaborative *manipulation*

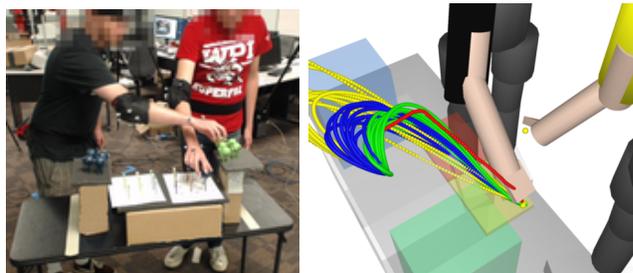


Fig. 1. Shared workspace assembly experiment (left) and Trajectories predicted by Goal Set STOMP, using IOC in green, and manually tuned cost functions in yellow and blue (right).

tasks [7], and no framework exists for predicting such human motion among obstacles.

This paper presents such a framework for reaching motions, which is based on studying how two humans collaborate in a shared workspace (as in Figure 1). 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 collaboration.

Studying how two humans collaborate also 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. We can also gauge how well a human is acclimated to a robot collaborator. This paper focuses on the method to obtain an accurate prediction for the above purpose, and though we envision eventually using this prediction in the robot’s motion planner, this application is not within the scope of this paper.

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

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.

smoothness and distance relationships between the humans are used as input for the Path Integral Inverse Reinforcement Learning (PIIRL) algorithm [8]. 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 a trajectory optimization algorithm based on STOMP [1]. We make two changes to the algorithm, which are crucial for our domain: 1) We adapt it for iterative motion re-planning in a dynamic environment; and 2) We allow the algorithm to search over a task-space goal region instead of specifying a goal configuration. The second change is especially important in predicting human motion, as we do not know the goal configuration *a priori*.

In our experiments we gathered the training data from pairs 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 most cases. 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. Finally, we show that our method can be used to predict human motion when a human and a robot share the workspace better than baseline methods.

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. Section V presents results that illustrate the ability of our method to predict collaborative reaching motions. We then present results from the human-robot workspace sharing scenario in Section VI.

A preliminary version of this work appeared in [9]. The version presented here contains changes to the trajectory optimization and inverse optimal control algorithms that allow predicting the human’s motion with a task-space goal set, which is essential for real-world applications where the human’s goal configuration is unknown. We also present results from an expanded human subjects study, results on generalizing learned weight vectors to new goal regions and among participants, and the results of a human-robot workspace sharing experiment.

II. RELATED WORK

A. Probabilistic Graphical Models

Graphical models have often been used for predicting human motion. Motion prediction based on Gaussian Mixture Models (GMMs), commonly used in gesture recognition [10], has been shown to perform well for high-dimensional movement recognition. Hidden Markov Models (HMMs), another popular stochastic modeling technique for human motion recognition and prediction [11], were used in [12], where Kulić et al. describe an approach for online incremental learning of full body motion primitives from observation

of human motion, allowing the same model to be used for both motion recognition and motion generation. Finally, Conditional Random Fields (CRFs) were used in [13], where Koppula and Saxena predict 3D trajectories of the human hand based on affordances. This work was recently extended in [14] to predict high-dimensional trajectories.

While these graphical model representations (i.e., GMMs, HMMs, and CRFs) allow one to efficiently encode relationships such as those between activities, objects and motions, they do not capture obstacles constraints well, an issue we address in this work. We also show that our method outperforms the GMM approach we have employed in [15] for the collaborative reaching motions we are considering.

B. Optimal Control

Optimal control has been investigated for decades and recently Ganesh and Burdet [16] used a manipulation task to show that the Central Nervous System (CNS) uses a motion planning phase with multiple plans, and a memory mechanism. Many experiments investigating reaching under various conditions [2], [3] suggest that at a high level the human motor-behavior can be modeled by the minimization of a cost function used to weigh different movement options for a task, as well as to select a particular solution. Stochastic Optimal Control [17] provides a theoretical framework for these models while taking into account motor noise inherent to sensorimotor control [4]. In this spirit, a detailed subject-customized bio-mechanical model has been used in [18] to efficiently reconstruct a subject’s motion dynamics from motion capture data in real-time using a whole-body control approach.

These works suggest that an optimality criterion can model human motor behavior, the aim of our work is to find such a criterion to predict human reaching motions in shared workspaces, without resorting to musculoskeletal modeling of the human such as [18].

C. Inverse Optimal Control

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 Process. IRL was introduced by Ng et al. in [19], who proposed two algorithms for discrete and continuous states spaces. Later, apprenticeship learning [20] introduced the idea of maximizing the margin between the cost of the demonstration and other solutions. Apprenticeship learning consists of iteratively solving the forward problem by modifying the weights at each iteration.

In [21], Ziebart et al. proposed an approach to IRL based on the maximum entropy principle. The recent methods based on this formulation [22], [23], [24] do not require solving the forward problem and allow handling high-dimensional continuous state spaces. Instead of solving the forward problem they either sample trajectories or solve for the local optimality of the demonstrations using the

demonstrations’ feature-derivatives with respect to states and actions. Sampling-based IOC approaches generally allow solving model-free problems. They are particularly efficient when using motion primitives such as in [22]. Motion primitives reduce the action space dimensionality, and allow learning of closed loop behavior [25]. Our approach derives from the sampling-based algorithm introduced in [8]. This method only requires local optimality of the demonstrated trajectories and can be used in a model-based fashion.

Only a few studies have employed IOC approaches to determine objective functions for the optimal control problem for human motion generation [26], [27], [28].

These objectives are usually of a kinematic, dynamic or geodesic nature. There is a question as to whether IOC can answer fundamental scientific questions about human movement. Since IOC can only inform us about the comparative influence of the selected basis functions, if the true criterion is not modeled, the result is not informative. However, our goal is not to find the true cost function used by a human, rather it is to create a practical model of human reaching that can be used for prediction in collaborative tasks.

D. Trajectory Optimization

We rely on recent developments in trajectory optimization for motion planning [29], [1] to compute low-cost motion predictions. Our trajectory optimizer is based on the Stochastic Trajectory Optimizer for Motion Planning (STOMP) algorithm, which has proven effective for the type of manipulation motion planning we consider [1]. Recently, STOMP was adapted to run faster than real-time [30]. 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). The obtained trajectory library is used as demonstrations for the IOC algorithm to learn a cost function. Finally, we use the learned cost function inside an iterative motion re-planner to predict how the human will move in close proximity to another human or robot.

A. Inverse Optimal Control algorithm

Intuitively, solving IOC consists of finding a cost function under which the demonstrated behavior is optimal. In most of the state-of-the-art techniques [19], [20], [21], [1], the cost function for a given trajectory ξ has been parameterized by a linear combination of user defined features $C(\xi) = w^T \Phi(\xi)$, where w is the parameter of interest and Φ the multi-valued feature function.

A human reaching motion trajectory can be represented as a time-parameterized curve in some human configuration space. These curves can be discretized in sequences of waypoints (i.e., configurations) at evenly spaced time intervals,

leading to the following definition for a trajectory:

$$\xi = [q_1 \cdots q_N]^T$$

where q_i are row vectors of configurations. The dimensionality of the corresponding vector space is $M * N$, where M is the number of DoF of the kinematic model, and N the number of waypoints.

Human reaching motions are inherently high-dimensional (in this work we consider $M = 23$ DoF and $N = 100$ waypoints for a duration of approximately one second). Computing globally-optimal solutions for motion planning problems of this nature is known to be intractable as the obstacles geometry generally introduces multiple local minima and the problem can not assumed to be convex, thus guaranteeing a globally-optimal IOC solution is also intractable. Furthermore, when we observe human motion using a motion capture system we do not have access to the dynamics of the human motor behavior. Hence in this work we focus on learning kinematic motion policies, i.e., we do not model forces or torques and we only parameterize the cost function with kinematic quantities. This simplification allows us to use a model-based approach (i.e., we assume that the transitions between states are all known and deterministic).

To solve the IOC problem locally we use the sampling-based *Path Integral Inverse Reinforcement Learning* (PIRL) algorithm [8], which can deal with high-dimensional continuous state-action spaces, and only requires local optimality of the demonstrated trajectories.¹ Compared to other IOC methods, PIIRL uses a prior on the kinematic property of the motion behavior to be learned by sampling from a fixed distribution. This allows the algorithm to be less myopic when generating cost functions. Other sampling-based IOC techniques typically use importance sampling [22]. PIIRL has been shown to outperform other state-of-the-art methods in [8] for kinesthetically taught manipulation motions, which are similar to the motions we consider in this work.

The problem considered by PIIRL is to recover a cost function composed of a control cost, and a general cost (i.e., configuration dependent) term that can be combined with a terminal cost, which we do not use in this work. Instead we use goal set trajectory sampling as described in Section III-D.

The cumulative cost $C(\xi)$, is a linear combination of user defined features $\Phi(\xi) = [G(\xi), A(\xi)]^T$, where A is the term enforcing smoothness (i.e., control cost) and G the general term of the form:

$$G(\xi) = \int_{t=0}^T \phi(q_t) dt \simeq \sum_{i=1}^N \phi(q_i) \Delta t,$$

where q_i is the configuration at index i along the trajectory and N the number of waypoints. Defining the cost function as a linear combination of features makes the problem of learning the weight vector w tractable.

¹PIIRL can also be used for model-free IOC, which typically requires rollouts on the physical system and is thus inapplicable when learning from motion capture data.

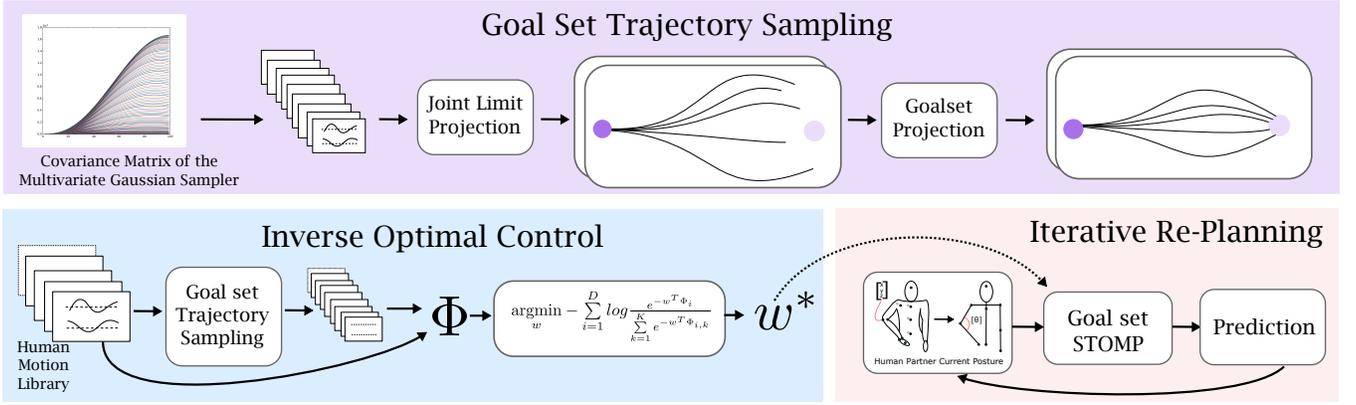


Fig. 2. Data flow through the system. The gathered human motion library is used to generate sample trajectories. Features (Φ) are then computed for the demonstrated and sampled trajectories. The PIIRL algorithm is then applied to generate a weight vector w^* . Prediction of collaborative human reaching 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. We use goal set trajectory optimization in both IOC and Iterative Re-Planning.

Thus each feature function penalizes motions that do not respect an associated property, see Section III-F for a description of the features we use to predict human motion.

PIIRL samples trajectories with low smoothness features around each demonstration in order to estimate the partition function [21]. The sampling distribution is defined using Multivariate Gaussians $\mathcal{N}(\xi_d, \Sigma = \sigma \mathbf{R}^{-1})$, centered at each demonstration ξ_d , where $\mathbf{R} = K^T K$, and K is a matrix of finite differences that computes time derivatives of configurations along the trajectory ($K \xi_d$). We set K to sample trajectories with low sums of accelerations, which, for the one dimensional case, has a band diagonal structure of the following form:

$$K = \begin{bmatrix} 6 & -4 & 1 & \dots & 0 & 0 & 0 \\ -4 & 6 & -4 & \dots & 0 & 0 & 0 \\ 1 & -4 & 6 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 6 & -4 & 1 \\ 0 & 0 & 0 & \dots & -4 & 6 & -4 \\ 0 & 0 & 0 & \dots & 1 & -4 & 1 \end{bmatrix}.$$

To learn cost functions that allow planning towards a task-space goal set, i.e., where the end configuration q_N is not specified, we introduced a modified distribution discussed in Section III-D and call the resulting algorithm Goalset-PIIRL. The weights are then obtained by solving the following convex minimization problem:

$$w^* = \underset{w}{\operatorname{argmin}} - \sum_{i=1}^D \log \frac{e^{-w^T \Phi_i}}{\sum_{s=1}^S e^{-w^T \Phi_{i,s}}}, \quad (1)$$

where D is the number of demonstrations and S the number of trajectory samples per demonstration. Φ_i are the features computed for demonstration i and $\Phi_{i,s}$ for the trajectory samples around that demonstration.

Sampling from $\mathcal{N}(\xi_d, \Sigma = \sigma \mathbf{R}^{-1})$ allows the algorithm to converge with less samples than other methods (see [8]). Note that in the sampling phase we change the last entry of the block of K corresponding to each DoF to allow variation of the end configuration q_N in the trajectory samples as shown in the upper part of Figure 2. We then perform joint limit and goal set projection with respect to the

metric \mathbf{R} (see Section III-D). Trajectory samples colliding with the environment and the other human are discarded by performing collision detection.

In the original version of PIIRL, a penalty on the L_1 norm of the weight vector w is added to the loss function in equation 1 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 [31] algorithm is 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. In order to tune the regularizer we run a learning phase with a range of values and select the one that minimizes our validation criterion.

B. Iterative re-planning

Iterative re-planning consists of planning iteratively while considering the current environment as static. It is a common approach to account for dynamic obstacles in robot motion planning [32], [30]. 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 in elementary manipulation motions, which are then cut in smaller segments by advancing Δt along each demonstration ξ_0 as depicted in Figure 3. 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.

C. Goal Set STOMP

When planning with the human model, we use the STOMP algorithm [1], which is a trajectory optimizer that iteratively

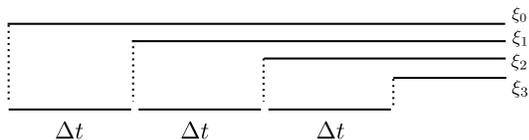


Fig. 3. Division of a demonstration ξ_0 into smaller segments

deforms an initial solution where the initial and goal configuration are fixed, by stochastically estimating the gradient in trajectory space. At each iteration, trajectories are sampled from a Multivariate Gaussian distribution $\mathcal{N}(\xi, \Sigma = \sigma^2 \mathbf{R}^{-1})$, and combined to generate the update. Thus, STOMP does not require the analytical gradient of the cost function to be known, and generally converges to a local minimum within 100 iterations. Our goal set version of the algorithm, which is similar in spirit to [33], relies on sampling trajectories with different goal configurations conditioned to meet the goal region (see Section III-D). In order to initialize the optimizer we use Jacobian-based inverse kinematics [34] to seed the algorithm with an initial goal configuration (analytical methods cannot be applied due to the redundancy of the human kinematics). This method finds an inverse kinematics solution that minimizes the configuration-space euclidean distance to the initial configuration.

The original STOMP algorithm presented in [1] optimizes a combination of obstacle and smoothness cost. The first is estimated by summing a penetration cost for a set of bounding spheres to the obstacles at every waypoint using a signed Euclidean Distance Transform (EDT) as defined in [29]; the second is estimated by summing the squared accelerations along the trajectory using finite differencing (Equation 2). In our version of the algorithm, we use a richer set of smoothness features, as we also account for task-space smoothness, described in III-F. In order to account for smoothness at each re-planning step, a buffer of configuration waypoints from the previous re-planning step is used to compute velocity, acceleration, and jerk at the initial configuration. Finally, to account for the posture and for the other human, we add a third cost criterion defined in section III-F. Note that the weight of the obstacle cost is manually tuned in our result section.

D. Goal Set Trajectory Sampling

Here we introduce goal set trajectory sampling used in our version of *Goalset-STOMP* and *Goalset-PIIRL*. In the standard version of both algorithms, trajectories are sampled from a Multivariate Gaussian distribution $\mathcal{N}(\xi, \Sigma = \sigma^2 \mathbf{R}^{-1})$, that generates trajectories with fixed end configuration. ξ is the trajectory we are considering, which is the current solution when planning, and demonstration when learning.

In order to sample trajectories that meet the goal set constraint while allowing different postures at the goal configuration, we sample from a modified covariance matrix (see Figure 2 for its one dimensional version) and we project the samples to the goal set with respect to the metric \mathbf{R} . We define goal set constraints as having the final configuration

Algorithm 1: Goal set trajectory sampling

```

R  $\leftarrow K^T K$  ;
 $\xi_t \leftarrow \mathcal{N}(\xi, \Sigma = \sigma^2 \mathbf{R}^{-1})$  ;
for  $i = 1 \dots \text{MaxIterations}$  do
  if  $\|x(q_N) - x_0\| < \epsilon$  then
     $\perp$  return  $\xi_t$  ;
   $C \leftarrow [ 0 \dots 0 \ J(q_N) ]$  ;
   $\Delta\xi \leftarrow -\mathbf{R}^{-1} C^T (C \mathbf{R}^{-1} C^T)^{-1} (x(q_N) - x_0)$  ;
   $\xi_t \leftarrow \xi_t + \eta * \Delta\xi$  ;
return Failure ;

```

of the motion q_N place the human's hand at a given point.²

The modification to the covariance matrix can be obtained by changing the endpoint smoothness term computation in the finite differencing matrix K , which specifies the precision matrix of the Multivariate Gaussian sampler ($\mathbf{R} = K^T K$). Note that this modified matrix is also used in STOMP to project the noisy update [1], however the matrix computing the smoothness term is left unchanged to enforce high smoothness at the endpoint. \mathbf{R} defines a metric over trajectory space in the following way:

$$\begin{aligned} \sum_{j=1}^M \sum_{i=1}^N \|\ddot{q}_{ij}\|^2 &= (K\xi)^T (K\xi) = (\xi^T K^T) (K\xi) \\ &= \xi^T (K^T K) \xi = \xi^T \mathbf{R} \xi = \|\xi\|_{\mathbf{R}}^2 \end{aligned} \quad (2)$$

Under this metric distances between trajectories can be computed as $d(\xi_1, \xi_2) = \|\xi_1 - \xi_2\|_{\mathbf{R}}$. Thus to project a sample to the goal region, we must minimize the projection update with respect to that metric. This is equivalently denoted:

$$\begin{aligned} &\underset{\Delta\xi}{\text{minimize}} && \frac{1}{2} \|\Delta\xi\|_{\mathbf{R}}^2 \\ &\text{subject to} && h(\xi_t + \Delta\xi) = 0 \end{aligned}$$

where $\|\Delta\xi\|_{\mathbf{R}}^2 = (\xi - \xi_t)^T \mathbf{R} (\xi - \xi_t)$, and the constraint function h is:

$$h(\xi) = x(q_N) - x_0,$$

where x_0 is a task-space point that defines the goal set, q_N is the last configuration of trajectory ξ_t and $x(q)$ is the forward kinematics function for configuration q . The constraint function h can be approximated by the first order Taylor expansion:

$$h(\xi_t + \Delta\xi) \approx h(\xi_t) + \frac{\partial}{\partial \xi} h(\xi_t) \Delta\xi,$$

where $\frac{\partial}{\partial \xi} h(\xi_t) = C$ is a $P \times Q$ matrix where P is the dimension of the task-space, and $Q = M * N$, is the dimension of trajectory space. C contains zeros except for the last block which contains $\frac{\partial x(q_N)}{\partial q} = J(q_N)$, the kinematic

²We restrict our definition of goal sets to ones specified by a 3D point with free rotation, however note that a similar algorithm can be derived for a broader class of task space regions as defined in [35].

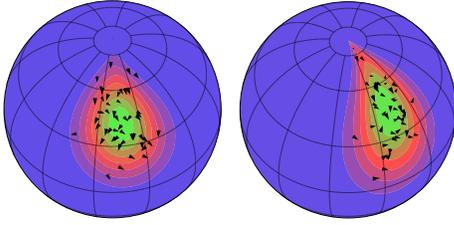


Fig. 4. Goal set regions explored using the *Goal set trajectory sampling algorithm 1* around two reaching motion trajectories. The color is proportional to the density of samples (green high, purple low). The standard deviation σ is set to the value used for Inverse Optimal Control in our experiments.

Jacobian of the arm. Thus for a small $\Delta\xi$, the Lagrangian with linearized goal constraint can be written:

$$g(\Delta\xi, \lambda) = \frac{1}{2}\Delta\xi^T \mathbf{R}\Delta\xi + \lambda^T [(x(q_N) - x_0) + C\Delta\xi],$$

which solves to:

$$\Delta\xi = -\mathbf{R}^{-1}C^T(C\mathbf{R}^{-1}C)^{-1}(x(q_N) - x_0),$$

The reader may refer to [34] for details on this result, which is obtained by setting the gradient of the Lagrangian with respect to $\Delta\xi$ and λ to 0.

Since the linearization of h is only valid for small $\Delta\xi$, we take small incremental steps scaled by η , which is set to 0.01 in our experiments. This leads to the method presented in Algorithm 1, where σ is the standard deviation.

The matrix $(C\mathbf{R}^{-1}C^T)$ can be singular, thus in our implementation, we add a regularization term to the diagonal. Note that in our experiments, we project the samples to the joint limits while maintaining smoothness by using a QP solver. We then project the trajectory to the goal set. In order for the projection to stay within the joint limits, we modify the Jacobian matrix by zeroing out the columns that would make the update exceed the joint limits.

Figure 4 shows goal set regions sampled using Algorithm 1. 1000 samples are used for displaying a Gaussian kernel density estimate of the roll and pitch (azimuthal and polar angles respectively) of the frame attached to the hand at q_N . These two angles specify the alignment between the metacarpals long axes and the goal set center point, while the yaw angle corresponds to the orientation of the hand around that axis. A random subset of 50 samples depict yaw angles using black arrows. Note that the trajectory samples are collision free and respect joint limits.

The size of the goal set regions depends on the standard deviation σ . Larger values of the standard deviation correspond to more exploration around the goal region.

E. Human kinematic model description

We model human kinematics following the recommendation for joints coordinates in [5]. The model is composed of prismatic and hinge joints. In our experiments we only account for upper body and right arm motions, which total 23 DoF. 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,

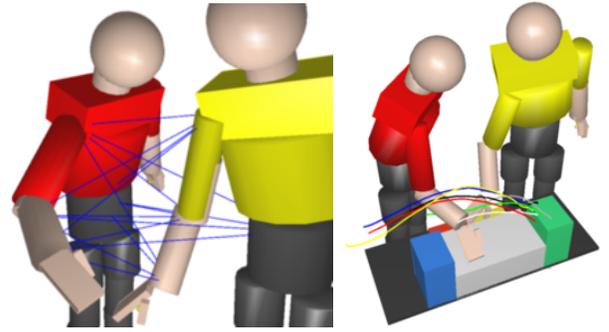


Fig. 5. Each line corresponds to a distance used in the feature vector (left). 3D model of the experiment used for collision checking with hand trajectories of the seven demonstrations used in the leave-one-out test of the result section (right).

one translation followed by three rotations for the elbow, one translation followed by three rotations for the wrist joint.

When predicting motions, the bounds of the joints are set using the minimal and maximal values observed in the motion capture data with additional offset to allow the learning and optimization to exceed these bounds. The prismatic joints in our kinematic model 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 human kinematics.

F. Feature functions

We consider variants of feature functions that have been introduced in previous work to account for human-robot interaction constraints [36], [37], [6]. We use three types of features inspired by *proxemics* theory [38] and experiments in neuroscience [2]:

1) *Distances between human links*: The goal of these features is to avoid collision. 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 5) 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 the sum of configuration and task-space length, squared velocities, squared accelerations and squared jerks along the trajectory using finite differencing.

3) *Distance to a resting posture*: These features ensure that the trajectory remains close to a resting posture by applying a weighted configuration space distance to a predefined resting posture of the form $\sum_{i=1}^N \|q_i - q_{rest}\|_W \Delta t$, where W is a diagonal matrix of learned weights.

IV. HUMAN COLLABORATION EXPERIMENT SETUP

The aim of our experiment was to gather training and test data in a workspace sharing setting. We chose to simulate a packing task, for instance packing different chocolates into a sampler box. The experiment we designed consisted of two participants standing shoulder to shoulder parallel to a table; each working on an individual task within a shared

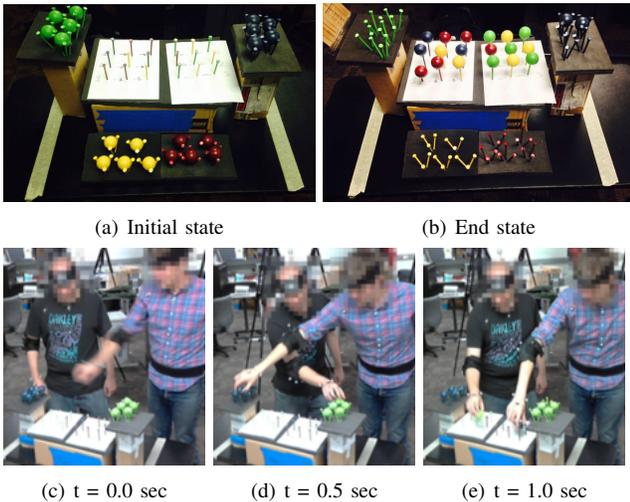


Fig. 6. Experiment design (top) and motion capture of the task (bottom).

workspace (Figure 6). The task was for the participants to place colored balls on pegs of the corresponding color in a specified order (Figure 6(a)). Adhesive tape was placed on the pegs allowing quick and easy placement.

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, continuing until all pegs in the plan are filled with balls (Figure 6(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 (seen in Figure 6(c)) based on standards in use in biomechanics literature [5]. The suit consisted of a waist-belt and headband attached to rigid plates, 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 xiphoid 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 DoF configuration of the right arm and torso for each participant using analytical inverse kinematics.

C. Instructions to participants

To collect data on human interactions in a shared workspace, we conducted a human subjects study utilizing the experiment and recording methods presented in the previous sections. The study consisted of 10 pairs of participants with each pair performing the experiment 6 times for a total of 60 runs of the experiment. The participants consisted of 4 women and 16 men with an average age of 21.

Upon entering the experiment area, participants were read a script which briefly explained they were to perform a collaborative manipulation task. Next, the specific task to be performed was explained verbally while simultaneously being performed in front of the participants. Subjects were shown a resting position in which they were to hold their left arm behind their back with their right arm comfortably relaxed by their side. Finally, the script explained that if a part of the task was accidentally performed out of order, or if a ball fell from one of its pegs, the participant should continue performing their task instead of attempting to rectify the error. This ensured the integrity of the remainder of the task.

V. HUMAN COLLABORATION EXPERIMENT RESULTS

In this section we present results illustrating the capability of our framework to recover a cost function using the link distances, distance to a resting posture and smoothness features presented in Section III-F. All algorithms were implemented in C++ using the motion planning software Move3D [39]. The IOC optimization was performed in Matlab.

a) Prerequisites: In [9], we provide a controlled study of the approach by performing motion planning using the original STOMP algorithm on a human model with a manually defined weight vector and measure the cost difference between the initial trajectory and the recovered trajectory. This experiment gave us an estimate for the required number of trajectory samples per demonstration required by PIIRL to converge. In order to tune the regularizer, we ran IOC on the user-study dataset over a range of 10 values of the parameter and selected 0.01, which induced the best validation scores.

b) Summary: We evaluate the quality of predictions of our goal set learning and motion algorithms against baseline tuning of the cost function by performing leave-out-testing on one class of reaching motions (i.e., reaching to a specific goal region). We then assess the method's ability to generalize across users and goal regions (three users and four goal regions) by training over a large set of motions, and we provide a comparison to Gaussian Mixture Model (GMM) based prediction on this dataset.

c) Active and Passive Humans: We define an active human, whose motion trajectories are used as demonstrations and are later predicted, and a passive human, who may interfere with the active human. In the prediction phase the passive human model configuration is set from the corresponding time index of the passive human recorded trajectories. The motion planner uses the passive human configuration to generate the signed-distance-field and compute link distances between the two humans.

1) Validation scores: To compute the similarity between the observed trajectories and the predicted trajectories we use Dynamic Time Warping (DTW), which is an algorithm for measuring similarity between two temporal sequences that may vary in time or speed. DTW relies on a distance metric between pairs of configurations.

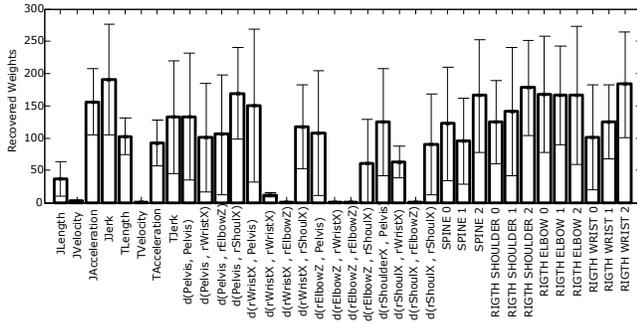


Fig. 7. Mean weight vector and standard deviation from the leave-one-out testing performed over the seven motions of Figure 8. The distances are described with active human in the bottom.

We use two configuration 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 joints using spherical joints.

- *Joint center distance*: The joints considered in the first metric are the pelvis, torso, shoulder, elbow and wrist.
- *Task-space metric*: The task-space metric combines Euclidean distance and angle between consecutive Quaternions as follows:

$$d(T_1, T_2) = \|p_1 - p_2\| + 0.1 * \cos^{-1}(|\langle v_1, v_2 \rangle|)$$

where p_1 and p_2 are the origins of frames T_1 and T_2 defined in some common coordinate system, and v_1 and v_2 the Quaternions.

A. Evaluation

1) *Leave-one-out testing*: To evaluate the capability of our predictions to generalize to new situations we have performed a leave-one-out test over the seven motions of Figure 5. The demonstration trajectories were processed in smaller segments using the procedure described in Section III-B, with $\Delta t = 0.1$ sec., resulting in 33 demonstrations used for IOC.

Each feature function is normalized to the range of the features in the samples, thus one can look at the relative influence of the parameters by looking at the weight values. The obtained mean, and standard deviation of the weights are shown in Figure 7. This shows the relative importance of smoothness features rather than distance features, and the overall importance of the postural features. All distances between the active human’s arm and passive’s pelvis are important, however distances involving wrist to wrist have no influence, which is expected due to the close proximity of the two humans when sharing the workspace. The high 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 their pelvis as much as their arm during manipulation. All the DoF are set to the bounds observed in the dataset, which constrains the pelvis motions to remain within these bounds. Regarding the smoothness criteria, acceleration and jerk appear as dominant

features, while velocity features do not appear to play any role. Unexpectedly, length of the joint space motion only plays a minor role.

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

- Conservative tuning (*baseline 1*): the weights for the squared accelerations and 16 link distances manually set to the same value.
- Aggressive tuning (*baseline 0*): the weights for the squared accelerations set to the same value and the distance weights set to 0.

For both baseline we do not use the postural term. The leave-one-out section of Table I 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 initial configuration and task-space goal point are extracted) and the predicted trajectories.

In the “no re-planning” version Goalset-STOMP only considers the initial configuration of the passive human, while when re-planning, the passive human configuration is updated at each replanning step with the configuration at which it would be at that time step.

Figure 8 shows the trajectories predicted for the seven motions with each tuning method (i.e., *baseline 0*, *baseline 1* and IOC) but without re-planning. It also shows the trajectory executed by the human (i.e., the demonstration from which the initial configuration and task-space goal are extracted to initialize the prediction).

Trajectories planned with *baseline 0* and with the IOC recovered weights have lower DTW scores than the ones planned with *baseline 1*. These results are consistent throughout both metrics, with or without replanning. The “no replanning” approach tends to outperform the “re-planning” approach slightly. This is due to the absence of motions which involved significant interference in this dataset. We report results on an example where this is not the case in Section V-A.3.

2) *Generalization Among Participants*: Out of the 10 pairs of participants, we selected three pairs for the quality of data obtained (i.e., absence of marker loss and occlusions of the motion capture system during the 6 runs). We manually segmented the data to obtain individual reaching motions either from a resting posture to a grasping configuration or between two grasping configurations. We selected 73 training and 20 testing motions that contained diverse start configurations and goal regions and were of good recording quality. We then learned a weight vectors using our framework by augmenting the training set as described in Section III, leading to 461 demonstrations.

The results are reported in the lowest section of Table I and close-ups of some of the trajectories obtained without replanning, are presented in Figure 9. The IOC is able to outperform the baseline methods in terms of task-space but not in terms of joint center distances in the case of *baseline 0*. The standard deviations are similar to the ones found in the leave-one-out tests, but the scores are significantly lower in

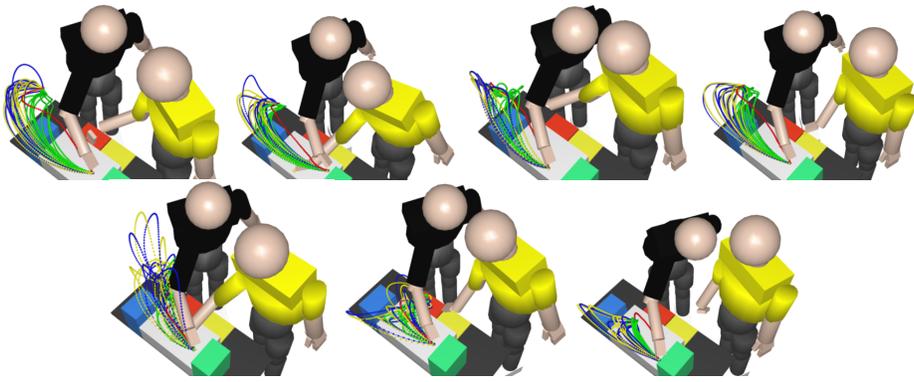


Fig. 8. Seven demonstrations (red) along with the ten trajectories resulting from *baseline 1* (yellow), *baseline 0* (blue) and our IOC framework (green). The demonstrations start and goal configurations are used to initialize the algorithm. The hand of the active human (black shirt) and passive human (yellow shirt) are set to the final configuration in the image.

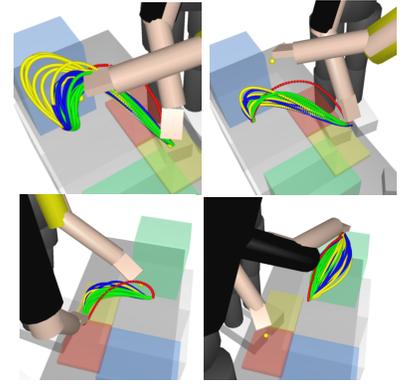


Fig. 9. The IOC weight vector is learned from the dataset comprising multiple participants. Observed motion (red), *baseline 1* (yellow), *baseline 0* (blue) and IOC (green).

	Re-planning								No Re-planning							
	task-space				Joint center distances				task-space				Joint center distances			
Dynamic Time Warping: Leave-One-Out Tests on 7 Trajectories Ending at the Same Goal Region																
Method	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max
<i>baseline 1</i>	20.7	7.4	5.0	35.3	77.1	26.2	29.7	128.2	20.4	7.4	6.1	35.5	76.1	23.5	34.5	120.6
<i>baseline 0</i>	18.3	6.8	5.1	37.3	70.1	22.0	29.7	118.3	18.3	6.2	6.0	31.7	69.7	20.5	32.6	111.6
With IOC	15.5	6.5	7.6	34.7	61.7	25.2	26.3	130.3	17.8	7.3	8.4	32.6	61.3	19.7	27.4	90.4
Dynamic Time Warping: Generalization among Participants																
Method	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max
<i>baseline 1</i>	14.4	14.9	4.0	64.1	55.3	47.7	22.5	205.5	12.8	10.8	4.4	55.1	50.7	34.3	15.9	177.4
<i>baseline 0</i>	10.4	7.5	3.9	54.8	40.3	24.0	19.8	171.8	10.5	8.9	3.4	48.2	43.0	31.0	17.5	173.8
With IOC	9.5	8.9	3.4	44.9	46.4	30.2	20.9	163.4	9.7	7.9	3.8	45.3	46.0	27.3	19.1	158.9
Spectral Analysis: Leave-One-Out Test																
Method	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max
<i>baseline 1</i>	1.0	9.7	-18.6	25.1	2.1	7.0	-13.5	20.0	-0.6	13.5	-20.1	32.5	6.9	10.4	-11.9	36.2
<i>baseline 0</i>	1.2	11.1	-16.6	33.3	2.2	7.7	-11.4	24.9	0.9	15.5	-19.9	49.5	8.4	12.1	-12.2	47.0
With IOC	2.6	10.7	-19.4	27.4	-0.5	6.5	-14.8	13.8	2.4	14.8	-19.0	38.5	5.0	8.3	-10.7	28.4
Spectral Analysis: Generalization among participants																
Method	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max	μ	σ	min	max
<i>baseline 1</i>	25.3	18.2	-6.1	61.4	19.6	16.5	-11.7	56.1	32.7	17.8	-8.9	72.0	29.3	15.8	-6.1	59.9
<i>baseline 0</i>	30.0	19.0	-3.3	73.7	23.7	14.2	-9.0	59.9	34.2	18.1	-0.1	75.8	32.5	15.7	-6.5	72.2
With IOC	25.4	15.4	-14.0	72.1	10.4	10.1	-16.6	33.7	29.6	17.5	-3.7	71.7	23.1	14.9	-5.5	71.7

TABLE I

DTW AND SPECTRAL ANALYSIS PERFORMED BETWEEN THE OBSERVED AND PREDICTED TRAJECTORIES, WITH AND WITHOUT RE-PLANNING FOR THE LEAVE-ONE-OUT TEST AND THE HUMAN SUBJECTS STUDY TEST. RESULTS ARE AVERAGED OVER 10 RUNS.

terms of mean. The segmented motions are in general shorter than for the leave-one-out tests, where the reaching motions go from one extreme to the other, and thus it is easier for the baseline methods to approximate human behavior using direct motions in this case.

In conclusion, for the 27 predicted motions in the three studies, in all except one test, the IOC tuned weights always outperform the baseline methods in terms of Task-space distances. Overall, it outperform the other methods in 6 out of the 8 similarity tests, where *baseline 0* outperforms the others twice. This suggests that: 1) IOC with a rich set of features can be used to predict human motion in collaborative tasks and outperform simple cost functions and, 2) inter-link-distances have low impact in predicting collaborative motions. Since, the scores of using replanning do not significantly ameliorate the predictions, we could conclude that it is not necessary for these types of motion prediction in general.

3) *Significant interference*: However, to show the capability 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 motions used in the leave-one-out phase, but does not include the trajectory from which we extract the start and end configurations for prediction. The motions obtained with and without re-planning are shown in Figure 10, and the DTW results are shown in Table II. 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.³

³In this example the goal set algorithms were not used, and the tuning of the task-space metric was different that used for the other experiments presented in this paper.

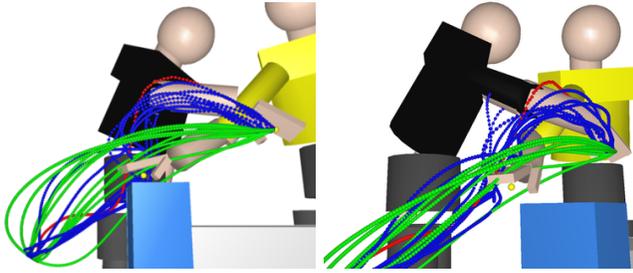


Fig. 10. Two view angles of a demonstration of the benefits of re-planning on a difficult example. Original motion (red) and predicted motions with (blue) and without (green) re-planning.

Type	μ	σ	min	max	μ	σ	min	max
	task-space				Joint center distances			
No re-plan	52.8	9.6	39.9	67.1	49.2	8.2	37.7	63.8
With re-plan	44.9	6.6	36.1	55.2	36.2	8.1	24.81	50.8

TABLE II

DTW PERFORMED BETWEEN THE OBSERVED AND PREDICTED TRAJECTORIES OF FIGURE 10. RESULTS ARE AVERAGED OVER 10 RUNS.

B. Smoothness Analysis

To assess the capability of our approach to produce motions which exhibit the smoothness property of human motions, we use the technique introduced in [40]. This measure evaluates the spectral arc-length metric on the movement speed profile of the kinematic quantity of interest. Thus we first compute the speed profiles in task-space and joint center distances, we then compute the profiles' Fourier magnitude spectrum, which allows us to compute the spectral arc length (we use a frequency cutoff of 20 Hz and K value of 1000). In a similar fashion to what is performed for the DTW similarity measures, we report the score difference in percentage between the predicted motions and the observed motions scores for the tasks in Table I.

Positive values indicate that the predicted motion is smoother than the human motion. For the leave-one-out phase, the score differences are very low indicated by the low mean and standard deviation values. This shows that the motions exhibit human-level smoothness. However for the larger dataset generated among different participants, the standard deviations are higher. In this case predicted motions tend to be much smoother than the observed human behavior (by 30% of the smoothness measure). This can be explained by the larger diversity in the recorded motions, where the recording process was challenged by potential marker occlusions due to the interference between the participants.

C. Comparison to GMM-GMR prediction

To compare with standard motion recognition techniques using probabilistic graphical models, we implemented an algorithm based on GMM and Gaussian Mixture Regression (GMR). The 73 training examples, used in the previous study presented in Section V-A.2, were manually classified into four sets, corresponding to their goal regions. A motion for each class was computed using GMR, similarly to our prior work [15].

Method	GMM-GMR prediction							
	task-space				Joint center distances			
	μ	σ	min	max	μ	σ	min	max
Active human used for prediction								
5	26.2	14.2	6.3	54.6	72.2	38.3	20.0	158.9
10	25.0	14.2	6.3	54.6	68.5	38.4	20.0	158.9
30	21.8	12.6	6.3	47.9	60.6	33.6	20.0	128.2
Both humans used for prediction								
5	31.9	18.4	6.3	92.2	87.1	53.0	20.2	268.6
10	20.4	9.9	6.3	36.8	54.2	24.9	20.0	97.7
30	19.1	9.4	6.3	34.6	51.3	25.1	20.0	97.7

TABLE III

DTW SCORES FOR 5, 10, AND 30 % OF THE TRAJECTORY

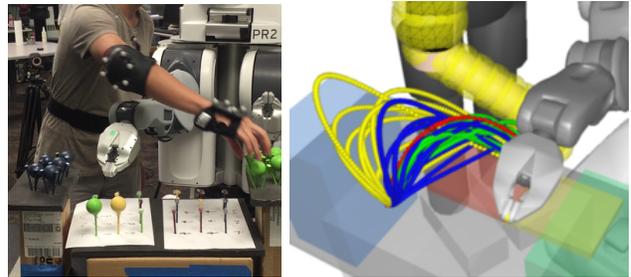


Fig. 11. Human-robot experiment (left). Trajectories predicted with Goalset-STOMP (right).

We first train the GMM using only the active human configuration using 150 kernels for classification and 25 for regression to increase the classification rate while keeping the regressed profile smooth. We then trained a second GMM using the passive human configuration in addition to the active human for which we used 450 kernels.

Early classifications of the 20 motions in our testing dataset were computed with 5%, 10%, 30% of the trajectory execution for both cases. We then performed DTW between the regressed motions of the class identified by the GMM classification, and the recorded trajectories. The values reported in Table III, when compared to the last rows of Table I, are significantly higher. This shows that prediction using motion planning, and IOC in particular, outperforms this approach on our data. We did not observe significant improvement when using the joint distribution (using the two humans' configurations to train and predict the motion).

VI. HUMAN-ROBOT WORKSPACE SHARING EXPERIMENT

In addition to recording human-human interactions, we created a human-robot workspace sharing experiment to evaluate the ability of this method to predict human motion when working with a robot instead of with another human. In this experiment, the human subject performs exactly the same task as described in Section IV-C while a PR2 robot executes a predetermined sequence of straight line trajectories between goal regions (see Figure 14). The set of robot trajectories were created with the intention of occluding the experiment workspace, while still allowing the human collaborator to complete their task. A total of 16 subjects participated in the human-robot study of which 15 recordings produced reliable data. In addition to being read experiment

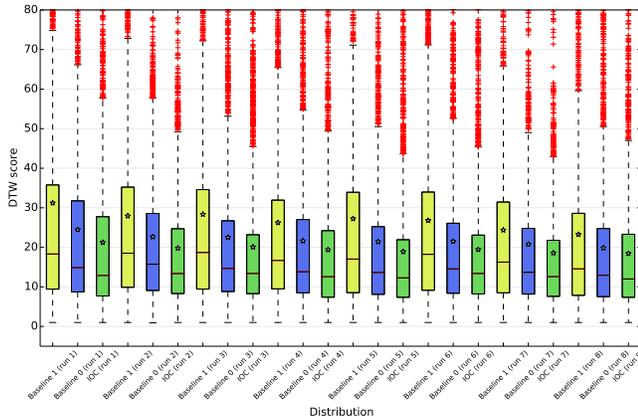


Fig. 12. Distribution of DTW task-space scores for the three methods of tuning, baseline 0, baseline 1, and IOC for each run.

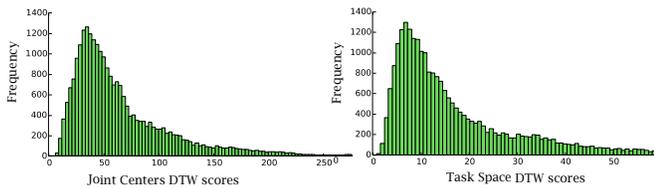


Fig. 13. DTW score distributions in joint center distances and task-space observed when predicting human motions in the human-robot experiment with our framework.

instructions, each subject performed three demo runs of their task in which the PR2 was held static to ensure proper familiarity with task execution. Immediately following the demo runs subjects performed 8 runs of the experiment in which the PR2 executed the previously described sequence. Finally, the 120 trials of the experiment involving the PR2 were segmented into 18 trajectories forming a library of 2120 human reaching motions with a robot collaborator.

A. Results

To predict human motion with the PR2, the weight vector learned for the study presented in Section V-A.2 is used and the feature function for link distances is mapped to the robot kinematics (wrist, elbow and shoulder and pelvis).

We ran the three tuning methods (i.e., *baseline 1*, *baseline 0*, and IOC) for 2120 elementary motions without replanning and we report the DTW scores distributions for each run individually in Figure 12. Note that the IOC prediction outperforms the baseline methods in each run. However many outlier exist in the distribution. These are instances where either the human hesitates or the motion planner is unable to find a collision free motion within the budgeted iterations. Figure 13 shows the combined distribution of the IOC DTW scores for all motions.

To examine the behavior of our prediction system as the human gets more acclimated to the robot we restrict the distribution to the first 95% of the IOC-based method and report the mean and standard deviation for each run in Figure 14. In the first run of the experiment, the participants are not acclimated to the robot and their behavior is more hesitant and thus less predictable by Goalset-STOMP, which aims to find optimal motions. Hesitations cause the human to stop,

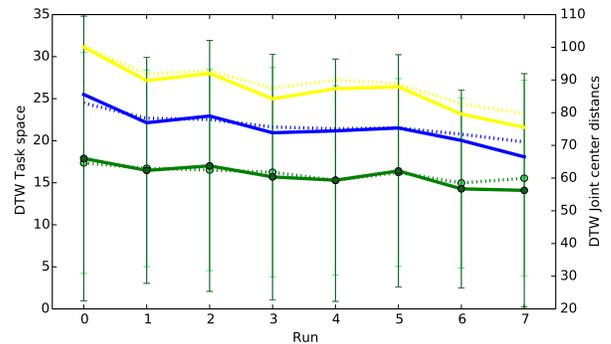


Fig. 14. Convergence of DTW scores for the IOC method (green) with the number of runs where the 5% highest scores have been removed from the data. Baseline 0 (yellow) and baseline 1 (blue) are shown for reference. Solid lines denote joint center distance, dotted lines denote task-space distance.

while our method assumes the human will always move to the goal. However, as the human becomes more acclimated, the predictions made by the motion planning algorithms improve, as denoted by the reduction in mean and standard deviation of the IOC-based method from the 1st (highest) to the 7th run (lowest).

VII. 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) and Goal Set Iterative replanning 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. We have also shown that our learned cost function outperforms baseline tunings of the cost function when the human works with a robot. Our prediction also improves as the human acclimates to the robot's motion.

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

We would like to thank the Max Planck Society and Pr. Stefan Schaal for partially supporting this work, and thank Alexander Herzog and Nathan Ratliff for fruitful discussions concerning the goal set trajectory sampling algorithm.

REFERENCES

- [1] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *ICRA*, 2011.
- [2] 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.

- [3] A. Biess, D. G. Liebermann, and T. Flash, "A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics," *The Journal of Neuroscience*, vol. 27, no. 48, pp. 13045–13064, 2007.
- [4] 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.
- [5] Wu *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.
- [6] 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.
- [7] L. Rozo Castañeda, S. Calinon, D. Caldwell, P. Jimenez Schlegl, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *AAAI*, pp. 1422–1428, 2013.
- [8] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning objective functions for manipulation," in *ICRA*, 2013.
- [9] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *ICRA*, 2015.
- [10] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 9, pp. 884–900, 1999.
- [11] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *IEEE-AFGC*, 2002.
- [12] 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.
- [13] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *RSS*, 2013.
- [14] Y. Jiang and A. Saxena, "Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs," in *Robotics: Science and Systems, RSS*, 2014.
- [15] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IROS*, 2013.
- [16] 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.
- [17] D. P. Bertsekas, *Dynamic programming and optimal control*. No. 2, Athena Scientific, 1995.
- [18] 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.
- [19] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *ICML*, 2000.
- [20] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML*, 2004.
- [21] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI*, pp. 1433–1438, 2008.
- [22] A. Boularias, J. Kober, and J. R. Peters, "Relative entropy inverse reinforcement learning," in *International Conference on Artificial Intelligence and Statistics*, 2011.
- [23] N. Aghasadeghi and T. Bretl, "Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals," in *IROS*, 2011.
- [24] 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.
- [25] F. Stulp and O. Sigaud, "Robot Skill Learning: From Reinforcement Learning to Evolution Strategies," *Paladyn, Journal of Behavioral Robotics*, vol. 4, pp. 1–13, Sept. 2013.
- [26] S. Albrecht *et al.*, "Imitating human reaching motions using physically inspired optimization principles," in *Humanoid Robots (Humanoids), 11th IEEE-RAS International Conference on*, 2011.
- [27] B. Berret, E. Chiovetto, F. Nori, and T. Pozzo, "Evidence for composite cost functions in arm movement planning: an inverse optimal control approach," *PLoS Comput Biol*, vol. 7, no. 10, 2011.
- [28] N. Sylla, V. Bonnet, G. Venture, N. Armande, and P. Friaese, "Assessing neuromuscular mechanisms in human-exoskeleton interaction," in *IEEE-EMBC*, 2014.
- [29] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9–10, pp. 1164–1193, 2013.
- [30] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using gpus," in *ICRA*, 2013.
- [31] G. Andrew and J. Gao, "Scalable training of l_1 -regularized log-linear models," in *ICML*, 2007.
- [32] 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.
- [33] A. D. Dragan, N. D. Ratliff, and S. S. Srinivasa, "Manipulation planning with goal sets using constrained trajectory optimization," in *IEEE Int. Conf. Robot. And Autom. (ICRA)*, 2011.
- [34] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [35] D. Berenson, S. Srinivasa, and J. Kuffner, "Task Space Regions: A framework for pose-constrained manipulation planning," *International Journal Of Robotic Research*, vol. 30, pp. 1435–1460, Oct. 2011.
- [36] 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.
- [37] J. Mainprice, M. Gharbi, T. Siméon, and R. Alami, "Sharing effort in planning human-robot handover tasks," in *ROMAN*, 2012.
- [38] E. T. Hall, "A system for the notation of proxemic behavior," *American anthropologist*, 1963.
- [39] T. Siméon, J.-P. Laumond, and F. Lamiraux, "Move3d: A generic platform for path planning," in *Assembly and Task Planning, Proceedings of the International Symposium on*, 2001.
- [40] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A robust and sensitive metric for quantifying movement smoothness," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 8, pp. 2126–2136, 2012.



Jim Mainprice is a Postdoctoral fellow in the Autonomous Motion Department at the Max Planck Institute for Intelligent Systems (Tübingen, Germany) since January 2015. He received his M.S. from Polytech' Montpellier, France, and his Ph.D. in robotics and computer science from the University of Toulouse, France, in 2009 and 2012 respectively. His research interests include motion planning, task planning, machine learning, human-robot collaboration and human-robot interaction.



Rafi Hayne recently received a BS in Computer Science from Worcester Polytechnic Institute where he is currently pursuing a MS degree in Computer Science. Upon completion of this degree he hopes to continue research in robotics. Although still developing, his research interests focus on understanding human motions as they relate to robot motion planning.



Dmitry Berenson received a BS in Electrical Engineering from Cornell University in 2005 and received his Ph.D. degree from the Robotics Institute at Carnegie Mellon University in 2011. He completed a post-doc at UC Berkeley in 2011 and started as an Assistant Professor in Robotics Engineering and Computer Science at WPI in 2012. He founded and directs the Autonomous Robotic Collaboration (ARC) Lab at WPI, which focuses on motion planning, manipulation, and human-robot collaboration. He received the IEEE

RAS Early Career award in 2016.