Progressive Stochastic Motion Planning for Human-Robot Interaction

Ozgur S. Oguz[1], Omer C. Sari[1], Khoi H. Dinh[1] and Dirk Wollherr[1]

Abstract— **This paper introduces a new approach to optimal online motion planning for human-robot interaction scenarios. For a safe, comfortable, and efficient interaction between human and robot working in close proximity, robot motion has to be agile and perceived as natural by the human partner. The robot has to be aware of its environment, including human motions, in order to proactively take actions while ensuring safety, and task fulfillment. Human motion prediction constitutes the fundamental perception input for the motion planner. The prediction system, which is based on probabilistic movement primitives, generates a prediction of human motion as a trajectory distribution learned in an offline phase. The proposed stochastic optimization-based planning algorithm then progressively finds feasible optimization parameters to re-plan the motion online that ensures collision avoidance while minimizing the task-related trajectory cost. Our simulation results show that the proposed approach produces collision-free trajectories while still reaching the goal successfully. We also highlight the performance of our planner in comparison to previous methods in stochastic motion planning.**

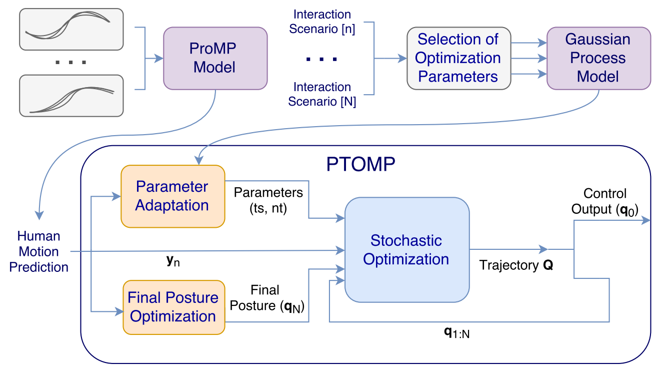


Fig. 1: Overview of the framework: The Probabilistic Movement Primitive (ProMP) model that enables the motion prediction and the GP model that selects the optimization parameters are trained in the offline phase. The online motion planning is an adaptive process as the stochastic optimization is supported by updated prameterization and final posture as well as a human motion prediction.

I. INTRODUCTION

Robots are envisioned to be present in everyday human life as well as in manufacturing. Independent from application scenarios, robots are expected to interact with human partners in a natural way, i.e. as similar as possible to the humans interacting with each other. There are three main requirements to achieve such a seamless interaction: *(R1)* Both, human and robot, have to achieve the goal defined by the task to be solved, *(R2)* robot has to be aware of the changes in the environment, including human motions, to adapt accordingly, and *(R3)* the robot behavior has to be reactive, yet still understandable by humans.

In human-robot close interaction scenarios, the robot trajectory generation is a challenging problem as it has to not only fulfill the aforementioned requirements but also consider the robot's capabilities (e.g. joint limitations) and the variations in human motions. The three requirements (R1-3) are highly interlinked. In order for the robot to react to ongoing human actions (R3), the robot has to be capable of predicting these human actions (R2). For achieving the required goals by both partners (R1), the robot has to plan a goal-directed motion trajectory while allowing the human partner to easily reach their target location (R2-3).

Recently, there has been an increased interest in close HRI. On the one hand, there are studies focusing on the safety aspect of such interactions [1, 2], where the human is treated as a dynamic obstacle which needs to be avoided while being indifferent to how humans perceive the interaction. On the other hand, few studies investigate motion models for the robot that can be perceived as legible [3], predictable [4], and understandable [5] by the human partner. However, all those aspects are interdependent on each other, which demands an integrated approach to motion planning for the robot.

In this paper, we propose a motion planning framework that addresses the challenges in human-robot collaboration in a unified structure. Given that each partner has a task to fulfill, we seek to formulate a stochastic optimization model to generate robot motions that 1) take into account human motion prediction while 2) minimizing the trajectory cost of the robot, and yet 3) avoid collisions without obstructing the human partner's task execution. This establishes a shared workspace in which a human-robot team can work safely and naturally in close proximity.

An overview of the framework is presented in Figure 1. Our approach comprises a human motion prediction module along with a stochastic trajectory optimization method. Human motions are recorded for the set of tasks we are interested in, and then utilized for training probabilistic Movement Primitives (ProMPs) in an offline phase. During interaction, learned ProMPs generate predictive trajectory distributions for the human's motion given observed data. Based on this prediction, a final posture is optimized for the robot to avoid collision with the human as well as the feasible parameters for the motion optimization method is acquired w.r.t. the task conditions. As a last step, the motion trajectory of the robot is optimized efficiently with the

[1]Ozgur S. Oguz, Ömer C. Sarı, Khoi H. Dinh and Dirk Wollherr are with Chair of Automatic Control Engineering (LSR), Technical University of Munich, 80333 Munich, Germany {o.oguz,omercan.sari,k.hoang,dw} at tum.de

identified parameters by taking into account the prediction and the optimal final posture to re-plan the motion online accordingly. The whole process is repeated at each time interval during task execution. This adaptive optimization process enables agile and flexible robot motion generation.

This paper concentrates on the formulation of the stochastic motion planner which progressively optimizes the trajectory of the robot during interaction. We implement, test, and then analyze our approach both in simulation and in a preliminary human trial. Our analysis demonstrate the trade-off between safety and interaction related costs. Results show that the proposed method, by acknowledging this trade-off, enables a cautious interaction while providing locally optimal solutions for task achievement. Comparison to some state-of-the-art motion planners reveals the benefits of our approach both in terms of optimality and safety.

The novel approach we propose is a fundamental step toward safe and natural HRI in close proximity. The main contributions of this work are:

- We propose an integrated approach where human motion prediction is incorporated into a trajectory optimization method efficiently.
- The trajectory optimization is achieved online by a novel parameter adaptation method and progressively recomputed depending on the interaction dynamics.
- The proposed method solves for locally optimal trajectories given time constraints while still providing seamless interaction.

II. RELATED WORK

Existing methods in motion planning for robotics arms are generally classified in two main branches that have received most significant attention over the last two decades, which are sampling-based and optimization-based approaches. Sampling-based approaches randomly generate sampled points in the configuration space at each iteration and check the collision of these points with obstacles by using a collision-checking component. A graph (roadmap) is then constructed by a set of collision-free points and is extended at every iteration until a solution from the start to the final position is found. As a result, these approaches become very effective in high-dimensional spaces since they do not require a full representation of the environment explicitly.

The most influential sampling-based motion planning algorithms to date include probabilistic roadmaps (PRMs) [6] and rapidly-exploring random trees (RRTs) [7]. Both algorithms rely on the idea of connecting points sampled randomly from the state space, although they differ in the way of constructing a graph connecting these points. One major advantage of sampling-based approaches is fast computation and therefore they can be easily applied online. However most of these methods treat human co-workers as constraints or as obstacles that the robot needs fulfill or avoid [8]. To the best of our knowledge, there is no existing sampling-based motion planning framework that considers joint collaboration between human and robot while planning.

Unlike sampling-based approaches, optimization-based methods exploit the whole configuration space and seek for the best possible solution which minimizes (maximizes) a performance metric. In robot motion planning, optimization-based approaches help to save energy and imitate human motion by defining proper cost functions and consider different type of constraints in robot motions. With the broad validity of the cost functions, these planners can be easily adapted in different human-robot collaboration scenarios.

Optimization-based approaches for motion planning problem can be classified into two different branches: gradient-based optimization and stochastic optimization. A well-known gradient-based method is the CHOMP algorithm [9] which utilizes covariant gradient techniques for optimization calculations. The sampled trajectory is optimized within consecutive iterations where the covariant gradient update rule ensures convergence of CHOMP to a locally optimal trajectory. However, the main drawback of this method is that, the cost and constraint representations need to be differentiable, which is impractical for many real-life cases, especially in HRI, where uncertainties and disturbances exist in most of the cases.

An approach to bypass this problem is stochastic optimization, where STOMP [10] is a typical candidate. In STOMP, the numerical optimization is considered as a stochastic direct optimization. Here, instead of determining the gradient to update the trajectory numerically, STOMP uses a stochastic gradient estimation. This is basically done by generating noisy trajectories around a feasible initial trajectory and evaluating their performance in terms of the cost function to determine the gradient update. However, this prohibits the flexibility and adaptability. As the whole trajectory is computed prior to execution, it may become invalid when the environment changes.

To enable flexibility and adaptability in robot motion, planning while executing is considered. The ITOMP algorithm takes STOMP as starting point and introduces the interleaving of motion generation and execution [11]. By doing this, ITOMP gains the ability to be used for real-time motion generation.

Even though optimization-based methods are widely used in robotics, the application in human-robot interaction, or especially joint collaboration is still very limited. One of the reasons is that there is no proper way to define human-related cost functions which are varied depending on different kinds of tasks and human partners. A very recent research [12] uses inverse optimal control to learn the cost functions from sampled trajectories and then applies these cost functions to a STOMP-based optimization process to predict human motions and therefore provides better reactive motions for the robot. However, the method is only applied for a very specific, simple pick-and-place task.

A major drawback of optimization-based approaches is the heavy computation time, which makes it inapplicable in online scenarios if the environment or tasks are complex. There are some approaches that try to overcome this drawback i.e. combing optimization with machine learning techniques [13] or dynamic movement primitives [14]. However, they are still limited in the static cases or they neglect the optimality aspect in the online phase as a trade-off.

III. STOCHASTIC MOTION PLANNING

Here, we explain briefly the general structure of a stochastic trajectory optimization formulation [10] as the foundation of our motion planning framework for human-robot interaction in close proximity.

Stochastic optimization for motion planning relies on generating noisy trajectories around an initial (possibly infeasible) trajectory. Then, those trajectories are evaluated based on their costs. Calculating the cost of a trajectory requires a discretized representation of consecutive states. For a D-DOF robotic arm, a feasible state representation is defined in the joint space for the robot's configuration $\mathbf{q}_n \in \mathbb{R}^D$ at each time-step n. Concatenating the sequential postures of N time-steps, the trajectory representation $\mathbf{Q} \in \mathbb{R}^{N\times D}$ is obtained as:

$$\mathbf{Q} = [\mathbf{q}_1^T, \mathbf{q}_2^T, ..., \mathbf{q}_N^T]^T. \tag{1}$$

The d^{th} column of $\mathbf{Q}$ is indexed with $\mathbf{q}^d \in \mathbb{R}^N$ to represent the trajectory of the d^{th} joint as well. The generalized formulation of the cost function is given as follows:

$$\min_{\mathbf{Q}} \Big[J(\mathbf{Q}) = \sum_{n=1}^{N} C(\mathbf{q}_n, \dot{\mathbf{q}}_n, ...) + \sum_{d=1}^{D} \frac{1}{2} \|\mathbf{A}\mathbf{q}^d\|^2 \Big]$$
$$\text{s.t.} \;\; \mathbf{q}_0 = \mathbf{q}_{\text{init}}, \;\; \mathbf{q}_N = \mathbf{q}_{\text{final}} \quad \mathbf{q}_{\text{init}}, \mathbf{q}_{\text{final}} \text{ are const.}$$

where $C(\mathbf{q}_n, \dot{\mathbf{q}}_n, ...)$ is the sum of arbitrary state-dependent cost functions, $\mathbf{A}$ is a finite differencing matrix and $\frac{1}{2}\|\mathbf{A}\mathbf{q}^T\|^2$ represents the control cost as the norm of the accelerations in quadratic form.

NT number of sample trajectories, each having N number of time-steps, are generated by adding a noise covariance that ensures smoothness. As explained in [10], at every iteration the initial trajectory is updated with weights that are inversely proportional to the costs computed for the noisy trajectories. Running consecutive iterations by following the same procedure over the updated trajectory allows the result to approach to an optimal solution. Note that, the trajectory is planned as a whole, and not modified during the motion [10], thus, preventing any kind of reaction from the robot to the environmental dynamics.

We extend this standard formulation by introducing interleaving of motion planning and execution as in [11]. Interleaving in this context means the same optimization procedures are computed after the robot executes the control signal at each time-step. The motion is divided into $N-1$ intervals and the robot moves from the posture $\mathbf{q}_n$ to $\mathbf{q}_{n+1}$ during which the optimization continues for the trajectory $[\mathbf{q}_{n+1}^T, \mathbf{q}_{n+2}^T, ..., \mathbf{q}_N^T]^T$ similar to *receding horizon control*. The optimal trajectory varies at each interval depending on the updated cost function, and we employ this functionality to provide responsive behavior for the robot.

IV. PTOMP

A. Overview

In this section, we introduce the **P**rogressive **T**rajectory **O**ptimization-based **M**otion **P**lanning algorithm for Human-Robot Interaction Scenarios (PTOMP). This algorithm forms the basis of a novel approach to adapt the previous work on stochastic motion planning to human-robot interaction by integrating human motion prediction, and caring for human partner's perception in the planned motion. The main contribution of our framework is concentrated on (*i*) choosing feasible parameters, i.e. number of noisy trajectories (NT) and number of time-steps (TS), for the optimization method to achieve online capability, (*ii*) incorporating the interaction related cost functions to provide a natural interaction, and (*iii*) enabling safe interaction by taking proactive actions based on an efficient human motion prediction.

Our framework (Figure 1) comprises offline and online stages. In the offline phase, a human motion library is constructed for each task as Probabilistic Movement Primitives (ProMPs). This library enables predicting the human motion online (Section IV-B). The prediction is fed into to trajectory optimization to solve for a collision-free optimal motion. In another offline phase, a comprehensive set of different conditions are simulated with varying optimization parameters, NT, TS. This allows learning the space of trajectory cost values for a multitude of tasks with different set of optimization parameters (Section IV-E).

In the online phase, trajectory optimization is run with a combination of cost functions (Section IV-D) by choosing the feasible parameters from the learned model and integrating human motion prediction, and final posture optimization (Section IV-C) to ensure natural and safe interaction.

B. Prediction

The occupancy estimation of a dynamic obstacle in the workspace is utilized for the collision cost calculation of future time-steps. Even if it is possible to make this estimation from the velocity and acceleration of the obstacle as in [11], its accuracy is likely to remain reliable only for a short period of time. In other words, the trajectory optimization will define a collision cost based on an unreliable estimation, thus hindering the safety of the avoidance behavior.

To avoid collision in the workspace, the human arm can also be considered as a dynamic object to be avoided. In general, human motion estimation requires a specialized prediction method as it depends on arm dynamics and its control results in variation in motion [15]. To imitate such behavior online, we use Probabilistic Movement Primitives (ProMPs) and learn a distribution of a motion behavior by training with multiple trajectories performed for a specific task [16]. ProMPs represent a discrete trajectory $X = \{x_n\}$, $n = 0 \dots N$ defined by states x_n over time N with the formulation

$$\boldsymbol{y}_n = [x_n, \dot{x}_n]^\top = \boldsymbol{\Phi}_n^\top \boldsymbol{\omega} + \boldsymbol{\epsilon}_y, \tag{2}$$

where ω is the weighting vector over the $k \times 2$ dimensional time-dependent basis matrix $\boldsymbol{\Phi}_n = [\phi_n, \dot{\phi}_n]$ with k being the number of basis functions and $\boldsymbol{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_y)$ is zero-mean independent Gaussian noise, while $\boldsymbol{\Phi}_n^\top \boldsymbol{\omega}$ gives the mean of the trajectory. Introducing a Gaussian distribution to also represent variance $p(\boldsymbol{\omega};\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\omega}|\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}})$ over the weighting vector ω results in the following distribution for the trajectory:

$$\begin{aligned} p(\boldsymbol{y}_n;\boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{\Phi}_n^\top \boldsymbol{\omega}, \boldsymbol{\Sigma}_y)\mathcal{N}(\boldsymbol{\omega}|\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}})d\boldsymbol{\omega} \\ &= \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{\Phi}_n^\top \boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\Phi}_n^\top \boldsymbol{\Sigma}_{\boldsymbol{\omega}} \boldsymbol{\Phi}_n + \boldsymbol{\Sigma}_y). \end{aligned} \tag{3}$$

Using a set of motion observations, the parameters $\boldsymbol{\mu}_{\boldsymbol{\omega}}$, $\boldsymbol{\Sigma}_{\boldsymbol{\omega}}$ can be estimated with maximum likelihood estimation [17].

By this formulation, the ProMPs enable an online human motion prediction, where a trajectory along with the variance for each time point is generated. This variance information is useful for human-robot interaction scenarios where the robot should also consider the uncertainties of human behaviors.

C. Final Posture Optimization

While interacting with a human, the robot should also be expected to reach a final posture that conforms to a legible behavior. As the standard formulation of stochastic trajectory optimization uses a fixed final state, we also incorporate a final posture optimization step which precedes the motion planning phase. Moreover, in close interaction scenarios where both the robot and the human reach nearby targets, the final posture of the robot should be optimized so that it avoids collision. The algorithm searches for a final posture of $\mathbf{q}_N^*$ that satisfies the condition

$$\mathbf{q}_N^* = \arg\min_{\mathbf{q}_N} \left[J(\mathbf{Q}_{1\ldots N}^*) \;\forall\; \mathbf{q}_N \in \mathbb{R}^D \right] \tag{4}$$

and thereby enables the most optimal trajectory while still assuring the task of reaching the target position for the end-effector. As the optimization result could not be known beforehand we target an estimation that satisfies the equation 4. Here we assume that the optimal final posture that minimizes the cost of the current trajectory would also minimize the cost of the optimized trajectory. Therefore, during the run of motion planning we make the final posture optimization over the cost of the instant trajectory.

D. Cost Functions

1) Interaction Cost: An efficient human-robot collaboration highly depends on the fluency of interaction between both sides. Just as the necessity of human motion prediction by the robot, the legibility of robot motion for the human subject is also crucial. Following the results of recent studies claiming that the legibility of a robot's motion comes from its closeness to human motion behavior [18, 19], we look into models to generate human-like motion. In motor control field, human motion generation has been studied as an optimization problem. Considering the total energy [20], torque change [21], standard deviation [22] of the motion has been suggested to be used as cost functions that describe human motion. Recent studies extended this claim by showing that human motion planning tends to reduce the effort while still ensuring a smooth motion that reaches the target [23]. To represent the effort we introduce the state dependent energy to the cost function.

$$\begin{aligned} J_{\text{effort}} &= \sum_{i \in C} (m_i h_i + \frac{1}{2} m_i V_i^2), \\ C &= \{\text{Upper arm}, \text{Lower arm}, \text{Hand}\} \end{aligned} \tag{5}$$

which is the sum of kinetic and potential energy during the motion. Alongside the total energy, the minimization of the hand jerk in cartesian space is also used [24]:

$$J_{\text{jerk}} = \int_0^T \dddot{x}^2 + \dddot{y}^2 + \dddot{z}^2 dt \tag{6}$$

with $[x, y, z]$ giving the cartesian space coordinates derived from joint configurations of the robot. The combination of those costs is referred as the *interaction cost*

$$J_{\text{interaction}} = J_{\text{effort}} + J_{\text{jerk}} \tag{7}$$

2) Control Cost: As humans can perform smooth and consistent motions in interaction scenarios, they also favor robot partners that can smoothly move in a shared workspace [18]. As the optimization is run on the kinematic level of the robot, we take the acceleration of the motion as the input signal to be minimized. The control cost, i.e. the smoothness cost, is defined as the sum of norms of acceleration in quadratic form at each time-step, and is represented by the term:

$$J_{\text{control}} = \sum_{d=1}^{D} \frac{1}{2} \|\mathbf{A}\mathbf{q}^d\|^2 \tag{8}$$

3) Collision Cost: One of the main task in human-robot interaction is to ensure collision avoidance in order to satisfy the safety restrictions and provide efficient collaboration. As collision avoidance is a part of the natural interaction, it also helps the legibility of the robot's motion.

For the collision cost calculation, the algorithm needs to determine the collision point at first. To reduce the calculation time, we define the lower and upper arms of both the robot and the human subject as cylindrical shapes. Through a trigonometric calculation, we determine the distance and the location of collision as:

$$[dist_n, loc_n] = \text{collision}(\mathbf{q}_n, \mathbf{y}_n) \tag{9}$$

where the location is the distance of collision to the robot's end-effector. As the collision cost minimization should be driving the robot away from the human arm, we define a continuous cost function that decreases towards the end-effector of the robot:

$$J_{\text{collision}} = \begin{cases} K * loc_n, & \text{if } dist_n \leq dist\_limit \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

with $dist\_limit$ indicating the collision limit.

TABLE I: Summary of how the increase on the parameters influences the optimization.

Parameters	Number of Iterations	Efficiency of a single Iteration
Duration of the Motion	Linearly Proportional	No Relation
Number of Time-steps	Quadratically Inverse Proportional	Proportional
Number of Noisy Trajectories	Linearly Inverse Proportional	Proportional

4) Joint Limitations: As the servo motors of the robot can perform limited rotations, the motion planning requires a limitation for the joint rotation to comply with the physical limitations of the robot. Moreover, to satisfy a legible motion, the robot needs to be restricted to a similar rotation limitation as human joints. Such limitations can be provided by setting a threshold limit. Instead of using such threshold limits as constraints in the optimization, we incorporate the joint limitation as a another cost function which remains continuous and dramatically increases at the limitations. For example, to keep the d^{th} joint rotation between $-L$ and $+L$ degrees, we set the joint limitations function as:

$$J_{\text{limit}} = K_1 * \mathrm{e}^{K_2\left(|\mathbf{q}^d|-L\right)} \tag{11}$$

where the higher values of the coefficient K_2 provides a sharper increase of cost at the limitation.

E. Parameter Adaptation

As real-time motion planning requires the optimization to run simultaneously with the execution, there is a certain time allowed for the calculations at each interval. In stochastic optimization, the calculation consists of consecutive iterations, where the trajectory converges to optimality. If the time limitation does not allow to run enough number of iterations to converge to the optimality, then the resulting trajectory is considered as sub-optimal. This sub-optimal result might still satisfy the requirements defined by the cost functions. However, the collision must not be tolerated as it concerns the safety criteria and hinders the task execution.

In our analysis we have observed that even under the same scenario, different trajectories are generated when the optimization is run with different parameterizations. These results are obtained as the optimization parameters influence the performance of optimization. Reducing the calculation time for each optimization iteration proportionally increases the number of performed iterations which is expected to support optimality, however, the quality (efficiency) of iterations also play a role. The three main parameters that influence both the calculation time and the quality of each iteration are (1) the number of time-steps discretizing the trajectory in time, (2) the number of noisy trajectories generated at each iteration and (3) the duration of the robot motion.

The discretization (resolution) of the trajectory changes the calculation time of a single optimization interval as more time-steps for the trajectory means higher load of cost calculation. However, choosing a very low discretization reduces the smoothness of the trajectory. It also hinders the robot's reaction to the environmental dynamics due to the reduced cost function update frequency. Similarly, having higher number of generated noisy trajectories increases the calculation time at each iteration, but at the same time provides smoother trajectories. Lastly, an increase on the motion duration clearly allows more time for the optimization, but results in a slower motion for the robot which can prevent efficient interaction. A summary of how these three parameters influence the number of iteration run in a single interval and the efficiency of a single iteration is given in Table I. This analysis indicates that, the most feasible optimization parameters can not be easily selected as there is a trade-off in terms of their effect on optimization performance.

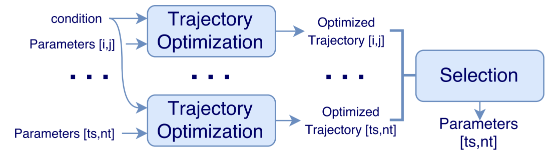
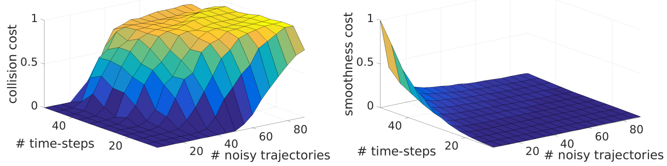


Fig. 2: Parameter selection from simulations under a constant condition.

1) Criteria of Parameter Selection: To determine the feasible parameters we evaluate their performance in terms of the costs over the optimized trajectories. We run simulations with varying parameterization over a grid under a constant condition as shown in Figure 2. For this evaluation, we consider the collision and the control cost of the optimized trajectories. The visualizations of how these costs are effected by the parameterization under an example condition is given in Fig 3. An acceptable performance requires to be capable of ensuring obstacle avoidance. Therefore, any parameter set that has a non-zero collision cost is considered as poorly performing and filtered out for the parameter selection. A higher discretization of trajectory increases the response frequency to environmental dynamics since the optimization is recomputed for each time-step. Hence, the parameter selection algorithm opts for choosing as many time-steps as possible after the initial filtering. However, this causes an increase in the control cost (Figure 3b). Therefore, a strategy for choosing a feasible parameter considering the smoothness cost is necessary as well. From our preliminary analysis, we found setting a limitation of 10% of maximum smoothness cost as a criteria for the secondary filtering provides good results consistently. Thus, the maximum values are chosen from the filtered subset of parameters. The procedure of this parameter selection is also detailed out in Algorithm 1 as a pseudocode.

2) Formulation of Parameter Selection: For the online selection of the optimization parameters, we need to provide a formulation that returns the parameter set when the conditions defining the dynamics of the scene are given. As this formulation cannot be achieved simply by a heuristic method online, we rely on building a generalized model that can map interaction conditions as the input to the optimization parameters as the output. This necessitates defining the



(a) Plot of collision cost (b) Plot of control cost

Fig. 3: Example figures for a collision avoidance scenario simulated under the same environmental conditions, but with different parameterizations.

Algorithm 1 Parameter Selection Algorithm

For a given condition *cond*, max number of time-steps TS, and noisy trajectories NT:

for $i = 0,\ i \leq \text{TS},\ i{+}{+}$ **do**
  for $j = 0,\ j \leq \text{NT},\ j{+}{+}$ **do**
    if $J_{\text{collision}}(traj([i, j] \mid \text{cond})) = 0$ **and** $J_{\text{control}}(traj([i, j] \mid \text{cond})) \leq 0.1 * max(J_{\text{control}})$ **then**
      Store $\mathbb{A} \leftarrow [i, j]$
    end if
  end for
end for
$\text{ts} = max(k \mid [k, l] \in \mathbb{A})$
$\text{nt} = max(l \mid [ts, l] \in \mathbb{A})$

conditions of the environment that influence the optimization the most. The *prediction uncertainty* and the *prediction error* are two metrics that require the optimization to be more reactive against a possible collision. The *distance between the robot and human arm* and the *remaining time until the collision* for the sub-optimized trajectory also determine how quickly the optimization requires to reach a solution such that it ensures a collision-free motion. Finally, the *immediate collision cost* indicates how close the current trajectory is to optimality. Combination of these five metrics define the state of the condition and serve as the inputs of the parameter selection function.

The parameter selection function is trained with Gaussian Process Regression Models [25]. Here, the training requires a set of real input and output data to fit a function estimation. To train for this formulation we run a set of simulations, by varying environmental conditions and recording a list of the feasible parameters selected. The metrics that define the condition of the scenario are taken as the input data while the selected optimization parameters are the outputs. As the function estimation of the Gaussian Process returns a single output, we need to train different models for different output parameters separately.

V. RESULTS

Here, we present the results of our approach, PTOMP, on collision avoidance performance in online HRI scenarios and compare to a state-of-the-art motion planner, ITOMP. The efficiency of the GP-model based parameter selection, and its effect on the performance of PTOMP are also analyzed. Lastly, the results for the final posture optimization is provided. In these analysis we use the KUKA LWR4™ robot with 6 degrees of freedom.

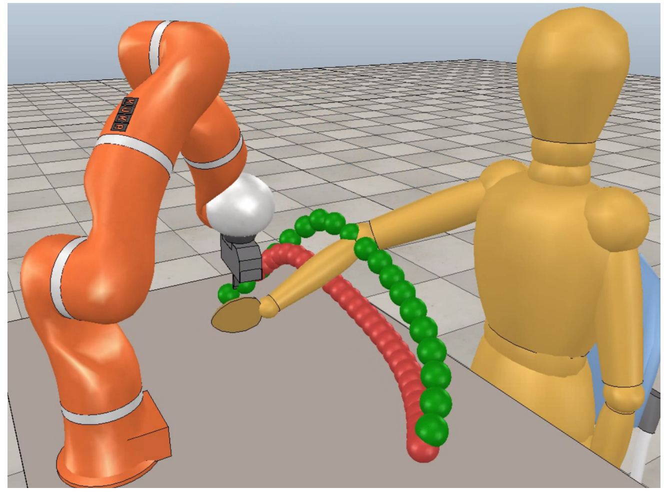
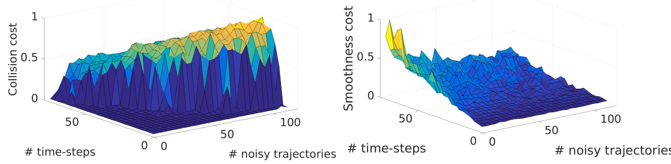


Fig. 4: Interaction scene in an example HRI scenario. The green trajectory represents the robot's motion optimized by PTOMP. A second motion optimized with infeasible parameters is also shown with the red trajectory.

In order to emphasize the influence of optimization parameters (NT, TS) on the resulting robot motion trajectory, an example HRI scenario is simulated where both agents execute their reaching motions, and the robot utilizes two different strategies (Figure 4). One subject's recorded data is used for simulation synchronously with robot's motion execution. The human motion is predicted based on the ProMP model learned offline for similar reaching tasks. PTOMP uses this prediction to generate a collision free trajectory early on by using GP-model based parameter selection (green in Figure 4). In our example, we choose the duration of robot's reaching motion as 1.5 seconds. The results of the parameter selection function under the given conditions were 42.87 time-step discretization and 30.23 noisy trajectories per iteration, which are rounded down to (42, 30) (note that the most feasible parameters are (45, 32) if the simulation analysis is directly used rather than GP-model output). To visualize a comparison we run a second simulation with 55 time-steps and 50 noisy trajectories, which are close to the first case but arbitrarily chosen. However, these parameters turn out to be infeasible for this scenario (Figure 4 red trajectory). This shows that, without a proper parameter identification, such a motion planner is prone to fail in avoiding the collision.

For evaluating the effectiveness and reliability of parameter selection method, we run similar tests under 450 different conditions. Here, we train the GP model with 400 conditions and spare the other 50 for the evaluation which is based on comparing the results of the parameter selection model to the results of the simulation analysis. The error between both outputs shows a standard deviation of 4.17 for the number of time-steps, and a standard deviation of 5.48 for the number of noisy trajectories. Hence, the GP-model



(a) Plot of collision cost

(b) Plot of control cost

Fig. 5: The plots show that the obstacle avoidance is provided under lower number of time-steps and noisy trajectories. Here the obstacle cost appears to be higher with less time-steps when collision is not avoided. This is due to the uncertainty of the human motion which was not included in Figure 3.

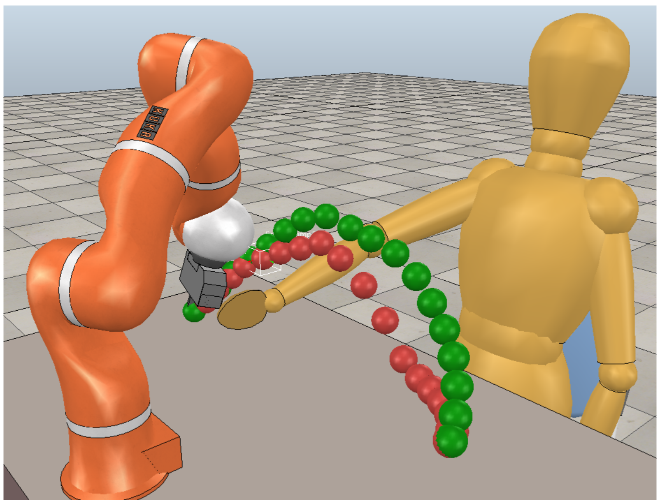
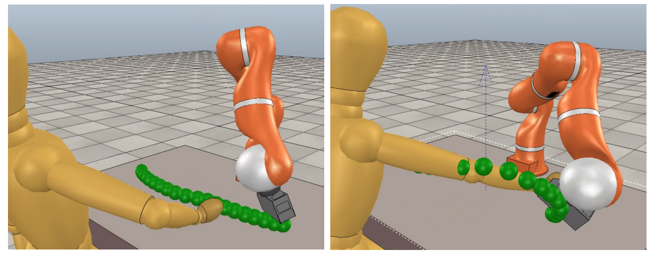


Fig. 6: Comparison of PTOMP (green) and ITOMP (red) generated trajectories. PTOMP relies on ProMP-based human motion prediction whereas ITOMP uses its default short-sighted prediction. The motion planning of ITOMP fails to react until the human hand almost interrupts its trajectory, while PTOMP enables taking a proactive action and thereby plan a legible optimal motion.

based parameter selection remains in a close range to the actual values used for training. In addition, with parameter adaptation, the collision avoidance is achieved with a success rate of 89 percent while it reaches 98 percent when we apply a heuristic by reducing both the number of time-steps and noisy trajectories by the standard deviation found in the analysis phase.

We evaluate the contribution of human motion prediction to the optimization problem. Here, we make the comparison with ITOMP's velocity-based occupancy estimation approach. We run these tests under the same scenario used for Figure 4. The trajectories followed by both algorithms are given in Figure 6. We also run these comparison tests for other 450 scenarios that we used for parameter selection evaluation. Running these tests with the same parameter selection output both for ITOMP and PTOMP, ITOMP succeeded to provide a collision-free trajectory only at 67 cases which correspond to a success rate of 14.88 percent in comparison to PTOMP's success rate of 89.34 percent. With reducing the number of time-steps and noisy trajectories again with their corresponding standard deviations in parameter selection



(a) Final posture without the human interruption

(b) Final posture in close human collaboration

Fig. 7: (a) shows that the final posture optimization drives the robot to lean right in order to comply with the interaction costs. However, when the human hand occupies the right side of the target as seen on (b) the optimization orients the posture to avoid collision.

function, ITOMP's success rate increases to 27.11 percent.

Finally, we evaluate the performance of the final posture optimization in two scenarios. In the first case (Figure 7a) the robot executes a pick and place task as the human motion does not interrupt its trajectory. Here, the final posture satisfies the legible motion as the robot leans to the side that it is approaching from. This behavior is due to the total energy minimization cost over the trajectory. In the second scenario we introduce a human motion that occupies a space where it causes a collision at the initial final posture as shown in Figure 7b. To avoid the collision, final posture optimization drives the robot to approach its target from the other side while trajectory optimization plans a collision-free trajectory that complies with the new final posture.

VI. CONCLUSIONS

We have presented PTOMP, an optimization-based online motion planning algorithm for human-robot interaction scenarios. Our approach concentrates on providing the interaction efficiency to robot motion planning in close human-collaboration by introducing legibility criteria while increasing the optimization performance of the online computation. Our algorithm is powered by a high-precision human motion prediction method, ProMP, that both allows the algorithm to run the optimization for future states early on as well as enabling the parameter adaptation of the optimization algorithm for a performance boost. We have highlighted the optimization performance of the motion planning under the human motion prediction and also shown the task execution success increased by the online parameter adaptation. The test results support that our algorithm exceeds the performance of the previous work in terms of robot motion legibility and trajectory optimality.

In our algorithm the legibility of the robot motion depends on the human motion planning criteria in obstacle-free space. The interaction is then provided by considering the human arm as a dynamic obstacle to be avoided. This combination does not take into account the human response to robot's motion. The observations on real human-human interaction scenarios can be utilized in order to replicate a similar interaction. In that regard, a possibility for future work is to adapt the PTOMP algorithm to a humanoid robot.

References

[1] J. Mainprice and D. Berenson, “Human-robot collaborative manipulation planning using early prediction of human motion,” in *Int. Conf. on Intelligent Robots and Systems, Tokyo, Japan*, IEEE, 2013, pp. 299–306.

[2] F. Flacco, T. Kroeger, A. D. Luca, and O. Khatib, “A depth space approach for evaluating distance to objects-with application to human-robot collision avoidance,” *J. Intell. Rob. Syst.*, vol. 80, pp. 7–22, 2015.

[3] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa, “Legibility and predictability of robot motion,” in *Int. Conf. on HRI, Tokyo, Japan*, IEEE/ACM, 2013, pp. 301–308.

[4] F. Stulp, J. Grizou, B. Busch, and M. Lopes, “Facilitating intention prediction for humans by optimizing robot motions,” in *Int. Conf. on Intelligent Robots and Systems, IROS, Hamburg, Germany*, IEEE, 2015, pp. 1249–1255.

[5] D. Carton, W. Olszowy, and D. Wollherr, “Measuring the effectiveness of readability for mobile robot locomotion,” *Int. J. of Social Robotics*, vol. 8, no. 5, pp. 721–741, 2016.

[6] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[7] J. J. Kuffner and S. M. LaValle, “Rrt-connect: An efficient approach to single-query path planning,” in *Robotics and Automation (ICRA), IEEE Int. Conf. on*, vol. 2, 2000, 995–1001 vol.2.

[8] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, “A human aware mobile robot motion planner,” *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 874–883, 2007.

[9] M. Zucker, N. Ratliff, A. Dragan, M. Pivtoraiko, M. Klingensmith, C. Dellin, J. A. D. Bagnell, and S. Srinivasa, “Chomp: Covariant hamiltonian optimization for motion planning,” *Int. J. Rob. Res.*, 2013.

[10] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, “Stomp: Stochastic trajectory optimization for motion planning,” in *Robotics and Automation (ICRA), IEEE Int. Conf. on*, 2011, pp. 4569–4574.

[11] C. Park, J. Pan, and D. Manocha, “ITOMP: incremental trajectory optimization for real-time replanning in dynamic environments,” in *ICAPS*, AAAI, 2012.

[12] J. Mainprice, R. Hayne, and D. Berenson, “Goal set inverse optimal control and iterative re-planning for predicting human reaching motions in shared workspaces,” *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 897–908, 2016.

[13] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, “Trajectory planning for optimal robot catching in real-time,” in *Robotics and Automation (ICRA), IEEE Int. Conf. on*, 2011, pp. 3719–3726.

[14] R. Weitschat, S. Haddadin, F. Huber, and A. Albu-Schaffer, “Dynamic optimality in real-time: A learning framework for near-optimal robot motions,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 5636–5643.

[15] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in *ICINCO (1)*, INSTICC Press, 2004, pp. 222–229.

[16] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, “Probabilistic movement primitives,” in *NIPS*, 2013, pp. 2616–2624.

[17] A. Lazaric and M. Ghavamzadeh, “Bayesian multi-task reinforcement learning,” in *ICML-27th Int. Conf. on Machine Learning*, 2010, pp. 599–606.

[18] Á. C. González, H. Admoni, and B. Scassellati, “Effects of form and motion on judgments of social robots’ animacy, likability, trustworthiness and unpleasantness,” *Int. J. Hum.-Comput. Stud.*, vol. 90, pp. 27–38, 2016.

[19] D. Matsui, T. Minato, K. F. MacDorman, and H. Ishiguro, “Generating natural motion in an android by mapping human motion,” in *IROS*, IEEE, 2005, pp. 3301–3308.

[20] R. M. Alexander, “A minimum energy cost hypothesis for human arm trajectories,” *Biol. Cybern.*, vol. 76, no. 2, pp. 97–105, 1997.

[21] M. Kawato, Y Maeda, Y Uno, and R Suzuki, “Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion,” *Biol. Cybern.*, vol. 62, no. 4, pp. 275–288, 1990.

[22] C. M. Harris and D. M. Wolpert, “Signal-dependent noise determines motor planning,” *Nature*, vol. 394, no. 6695, pp. 780–784, 1998.

[23] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, “Imitating human reaching motions using physically inspired optimization principles,” in *Humanoids*, IEEE, 2011, pp. 602–607.

[24] T. Flash and N. Hogan, “The coordination of arm movements: An experimentally confirmed mathematical model,” *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, 1985.

[25] C. E. Rasmussen, “Gaussian processes for machine learning,” 2006.