

MIT Open Access Articles

Provably Safe and Efficient Motion Planning with Uncertain Human Dynamics

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Li, Shen, Figueroa, Nadia, Shah, Ankit and Shah, Julie. 2021. "Provably Safe and Efficient Motion Planning with Uncertain Human Dynamics." Robotics: Science and Systems XVII.

As Published: 10.15607/RSS.2021.XVII.050

Publisher: Robotics: Science and Systems Foundation

Persistent URL: <https://hdl.handle.net/1721.1/145785>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Provably Safe and Efficient Motion Planning with Uncertain Human Dynamics

Shen Li, Nadia Figueroa, Ankit Shah, Julie A. Shah
Computer Science and Artificial Intelligence Laboratory (CSAIL),
Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
{shenli, nadiafig, ajshah, julie_a_shah}@csail.mit.edu

Abstract—Ensuring human safety without unnecessarily impacting task efficiency during human-robot interactive manipulation tasks is a critical challenge. In this work, we formally define human physical safety as collision avoidance or safe impact in the event of a collision. We developed a motion planner that theoretically guarantees safety, with a high probability, under the uncertainty in human dynamic models. Our two-pronged definition of safety is able to unlock the planner’s potential in finding efficient plans even when collision avoidance is nearly impossible. The improved efficiency is empirically demonstrated in both a simulated goal-reaching domain and a real-world robot-assisted dressing domain. We provide a unified view of two approaches to safe human-robot interaction: human-aware motion planners that use predictive human models and reactive controllers that compliantly handle collisions.

I. INTRODUCTION

Ensuring human physical safety is a top priority during human-robot interaction (HRI), but sometimes comes at the cost of task performance. In order to ensure safety, a pessimistic robot might assume that all space could be occupied by its human partner in the future, resulting in the freezing robot problem [41]. Preventing harm without unnecessarily impacting task efficiency is a critical challenge [49, 28]. In this work, with a focus on interactive manipulation tasks, we address this challenge via a motion planner that guarantees human safety, equipped with a two-pronged definition of safety that infuses flexibility into the planner.

Current approaches to safety during interactive manipulation tasks include both “predictive” and “reactive” methods [38]. Predictive approaches allow a robot to anticipate human behavior while simultaneously planning collision-free motion [15]. Through anticipation, such approaches enable a rich set of collaborative behaviors [36, 27, 4, 44]. Reactive approaches enable a robot to detect collisions in real time [11, 15, 32] and react compliantly to reduce contact forces [8, 14].

Many prior works have integrated these two approaches *sequentially* [38, 15], with a robot first employing motion planners to find paths and then using compliant controllers for execution. However, each approach separately optimizes behavior for its own particular goal (collision avoidance for planners and contact force reduction for compliant controllers) rather than a goal *jointly held* by both approaches, ultimately exposing the weaknesses of each. First, most planners don’t incorporate the fact that a compliant controller is employed to reduce contact forces in the event of a collision. As a result,

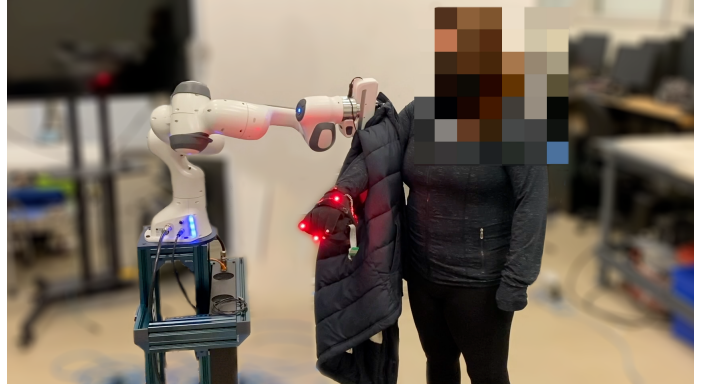


Fig. 1: During robot-assisted dressing, the robot must remain physically close to the human arm to ensure human comfort due to the limited size of the armhole. Planners optimizing for human safety, defined as collision avoidance, might cause the robot to freeze in such a circumstance. In this work, we redefine safety as collision avoidance or safe impact. This definition enables the robot to dress the person while allowing non-harmful impacts to occur.

planners tend to be very conservative, attempting to avoid collisions entirely. Conservative behavior can ensure safety, but could also worsen task performance or even unnecessarily freeze the robot in place [41]. Reasoning about uncertainties and avoiding the uncertain regions – which are necessary and useful in many cases [36, 42, 35] – would exacerbate this issue of over-conservativeness. Consider the case of robot-assisted dressing shown in Fig. 1: avoiding the (uncertain) human arm during task execution is nearly impossible, preventing a safe planner from making progress. Second, compliant controllers usually don’t maintain the robot’s high-level plans, making it challenging to adapt stiffness profiles in order to properly balance safety and precision.

We propose a safe planner for integrating predictive and reactive approaches *jointly* within a framework, in order to reduce system conservativeness while maintaining safety. Human physical safety in predictive and reactive approaches were previously defined as collision avoidance [31, 34, 29] and contact force reduction [8, 14], respectively. By incorporating both definitions, we redefine safety in the context of human-aware motion planning (HAMP). Our new definition is two-pronged: *collision avoidance or safe impact in the event of a collision* [26]. For safe motion planning, we developed a model predictive control (MPC) algorithm, built upon Koller et al. [23]. This algorithm guarantees safety, according to our

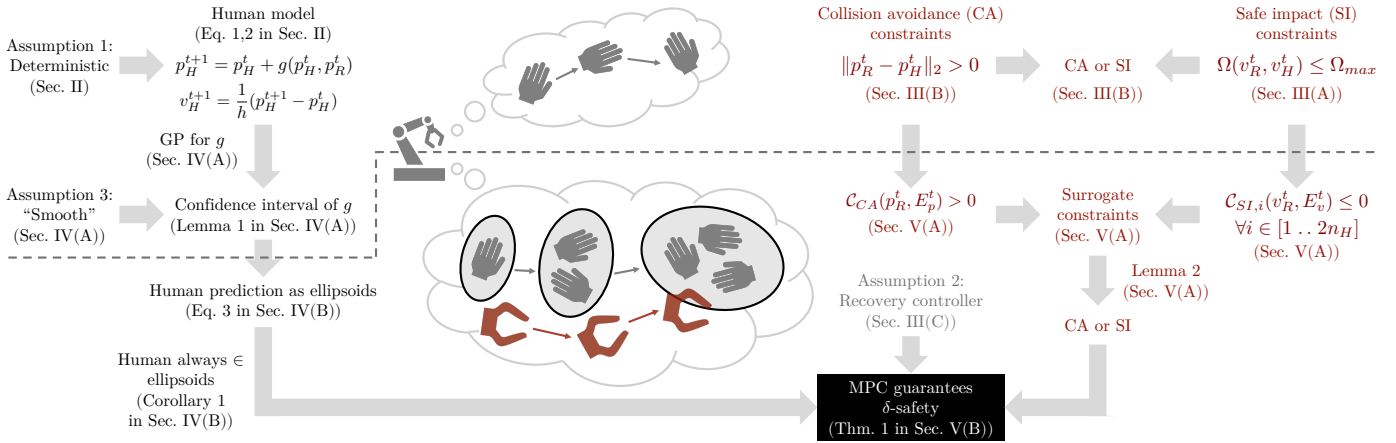


Fig. 2: Flowchart of the assumptions, theorems, and algorithms in this paper. The workflow on the left (black) represents our human model and prediction. The workflow on the right (red) represents our safety constraints, where safety is defined as collision avoidance or safe impact. Both workflows merge at the black block representing our MPC that guarantees δ -safety (defined in Def. 1). Above the dotted line, the human model and safety constraints are formulated for the human position and velocity, (p_H^t, v_H^t) . Below the dotted line, the human prediction outer-approximates the future p_H^t and v_H^t as ellipsoids, E_p^t and E_v^t , respectively. The safety constraints below the dotted line are formulated on (E_p^t, E_v^t) and enable our MPC to guarantee δ -safety. In the middle, the cartoon above the dotted line illustrates the human hand dynamic model. The cartoon below illustrates the ellipsoidal prediction and the robot path (red) that satisfies the safety constraints.

definition, with a high probability, under the uncertainty in human dynamic models. Our two-pronged definition of safety is able to infuse flexibility into the MPC and enable it to find both safe and efficient motions. In a scenario such as that depicted in Fig. 1, our MPC enables the robot to dress the person while allowing non-harmful impacts to occur.

We have evaluated the MPC algorithm in a simulated goal-reaching domain and a real-world robot-assisted dressing domain. Our results show that the two-pronged definition of human physical safety enables the algorithm to achieve a significantly higher task efficiency without sacrificing safety.

In this paper, we make two contributions: (1) We formally define human physical safety in HAMP for interactive manipulation tasks as collision avoidance or safe impact. (2) We propose an MPC that can guarantee human physical safety, according to our definition, with a high probability. To the best of our knowledge, ours is the first work to provide a probabilistic safety guarantee under the uncertainty in human dynamic models for human-robot systems.

We intend our method to be used for robots to personalize their assistance in activities of daily living to meet their users' specific needs [20]. A robot shipped to a user at home, for example, would only have a default model of how a human could interact with it during robot assistance. The misalignment between the default model and the true user behavior could lead to unsafe robot actions [40]. Our algorithm can enable the robot to reason about the uncertainty in its user model, safely assist the user, and efficiently collect on-policy data. With the interactive data, the robot could then adapt its user model and personalize its assistance for the user's needs.

The rest of the paper is organized as follows: In Sec. II, we introduce the dynamical system formulation of the robot and human. In Sec. III, we formally define human physical safety under our system representation. In Sec. IV, we introduce

Gaussian process (GP) and discuss the key steps necessary to conservatively predict human motion based on the GP. Finally, in Sec. V, we combine these elements, present our MPC, and prove its safety guarantee. Fig. 2 is a flowchart highlighting the assumptions, theorems, and algorithms in this paper.

II. SYSTEM REPRESENTATION

Throughout this work, we use subscripts to denote the agent a given variable is related to. Superscripts denote the time index. We use $[a .. b]$ to denote the integer interval, $\{a, a + 1, \dots, b\}$, while $[x]_i$ denotes the i th dimension of a vector, x .

We represent the robot and human as two point masses in Cartesian space. As shown in Sec. V-A, this representation allows us to analytically enforce safety constraints on ellipsoids. A broad range of tasks can be modeled by this representation, including handover and space-sharing tasks [18, 45], where only the robot end-effector and human hand are modeled. As depicted in Sec. VI-B, this representation enables a robot to dress a human arm (under certain task simplifications).

We use $p_R \in \mathcal{P}_R \subset \mathbb{R}^{n_R}$, $v_R \in \mathcal{V}_R \subset \mathbb{R}^{n_R}$, $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$, and $m_R \in \mathbb{R}$ to represent the position, velocity, control, and mass of the robot (respectively), where $n_R = 3$. Here, \mathcal{P}_R , \mathcal{V}_R , and \mathcal{U} encode boundary conditions. Similarly, let $p_H \in \mathcal{P}_H \subset \mathbb{R}^{n_H}$, $v_H \in \mathbb{R}^{n_H}$, and $m_H \in \mathbb{R}$ denote the human position, velocity, and mass (respectively), where $n_H = 3$. Here, \mathcal{P}_R and \mathcal{P}_H refer to the robot's and human's *workspace* (respectively). For the robot-assisted dressing task (Sec. VI-B), we specified both \mathcal{P}_R and \mathcal{P}_H as boxes around the human. We cannot force the human to stay inside \mathcal{P}_H . Instead, we collect the human data inside \mathcal{P}_H , train a human model, and derive robot-dependent probabilistic bounds on the future human positions. Given the bounds, our MPC algorithm can use the robot's control authority to theoretically guarantee

with a high probability that the human will always stay inside \mathcal{P}_H and safe. These bounds require the following assumption:

Assumption 1. *Human behavior inside \mathcal{P}_H is deterministic.*

This assumption applies to low-level human motions – e.g., reaching toward a goal [25, 35] – which are usually embedded in complicated behaviors as primitives. It also emphasizes that our focus is modeling the human behavior inside \mathcal{P}_H .

$$p_H^{t+1} = f_H(p_H^t, p_R^t) = p_H^t + g(p_H^t, p_R^t) \quad (1)$$

$$v_H^{t+1} = \frac{1}{h}(p_H^{t+1} - p_H^t) = \frac{1}{h}g(p_H^t, p_R^t) \quad (2)$$

Given Assumption 1, we model the human behavior inside \mathcal{P}_H as a first order, deterministic, and discrete-time dynamical system, as shown in Eq. 1, where $f_H: \mathcal{P}_H \times \mathcal{P}_R \rightarrow \mathcal{P}_H$. The function g is unknown, but will be learned via Gaussian Process, as presented in Sec. IV-A. We approximate v_H as the rate of change of position as shown in Eq. 2, where h is the length of time-steps. The initial condition, v_H^0 , is specified.

Since we focus on safety under unknown human dynamics, we model robot dynamics as a known, deterministic, discrete-time system: $(p_R^{t+1}, v_R^{t+1}) = f_R(p_R^t, v_R^t, u^{t+1})$. A robot controller π is defined as a function: $u^{t+1} = \pi(p_R^t, v_R^t, p_H^t, v_H^t)$. Let f_π be a function describing the closed-loop human-robot system corresponding with π . Combining f_R, g, π yields $(p_R^{t+1}, v_R^{t+1}, p_H^{t+1}, v_H^{t+1}) = f_\pi(p_R^t, v_R^t, p_H^t, v_H^t)$.

III. SAFETY IN HUMAN-AWARE MOTION PLANNING

In this section, we formally define safety in human-aware motion planning for interactive manipulation tasks. Lasota et al. [26] defined safety as (1) collision avoidance (CA) whenever possible, and (2) safe impact (SI) when collisions are required or unavoidable. We see CA and SI as two approaches to ensuring safety; integrating these approaches allows our planner greater freedom to find less conservative and more efficient solutions without sacrificing safety. Hence, we define safety as **collision avoidance or safe impact in the event of a collision**. Note that we do not impose a preference for either, differing slightly from Lasota et al. [26]’s definition.

First, we formulate safe human-robot impact during collisions in Sec. III-A. Then, in Sec. III-B, we define safety by combining the definitions of CA and SI. In Sec. III-C, we make one assumption about the availability of a recovery controller. Finally, in Sec. III-D, we define probabilistic safety, the main focus of this paper. Later, as shown in Sec. V-B, our MPC uses the recovery controller and guarantees probabilistic safety.

A. Safe Human-Robot Impact

We ensure safe human-robot impact during collisions by enforcing a low impact potential. *Impact potential* is defined as the maximum impact force a robot can create during a collision with a human [16]. Following Heinzmann and Zelinsky [16], and given our point-mass assumption for both human and robot, impact potential can be computed by a function $\Omega(v_R, v_H) = (e + 1)\|v_R - v_H\|_2 / (\frac{1}{m_R} + \frac{1}{m_H})$

(Eq. 7 in App. A), where e is the coefficient of restitution. (The derivation of Ω is available in App. A.)

Intuitively, if v_R and v_H are similar, then $\Omega(v_R, v_H)$ is low; however, if v_R and v_H are vastly different from one another, then $\Omega(v_R, v_H)$ is high, which could be unsafe. Thus, we ensure safe impact by enforcing $\Omega(v_R, v_H) \leq \Omega_{max}$ as a constraint within the robot’s planning process, where Ω_{max} is the maximum allowable impact potential.

B. Safety as Collision Avoidance Or Safe Impact

In this section, we formally combine CA and SI to arrive at our two-pronged definition of safety.

We define *collision avoidance (CA) set* as $\{(p_R, v_R): \|p_R - p_H\|_2 > 0\} = \mathcal{S}_{CA}(p_H)$, with $\|p_R - p_H\|_2$ referring to the Euclidean distance between p_R and p_H . We also define *safe impact (SI) set* as $\{(p_R, v_R): \Omega(v_R, v_H) \leq \Omega_{max}\} = \mathcal{S}_{SI}(v_H)$, where Ω computes the impact potential as introduced in Sec. III-A. Here, \mathcal{S}_{CA} and \mathcal{S}_{SI} denote the set-valued functions needed to compute those sets.

The human is considered physically safe if the system either remains collision-free or ensures safe impact during collisions. Formally, we define “safety” as

$$(p_R, v_R) \in \mathcal{S}_{CA}(p_H) \cup \left[(\mathcal{S}_{CA}(p_H))^c \cap \mathcal{S}_{SI}(v_H) \right] \\ = \mathcal{S}_{CA}(p_H) \cup \mathcal{S}_{SI}(v_H)$$

where the superscript c denotes the set complement operator. The \cap emphasizes that SI is used only in the event of a collision. This above equality implies that “CA or (SI during collision)” is equivalent to “CA or SI.” Our definition of safety indicates two methods of achieving safety: by imposing either a robot position constraint ($\|p_R - p_H\|_2 > 0$) for CA or a robot velocity constraint ($\Omega(v_R, v_H) \leq \Omega_{max}$) for SI.

As described in Sec. II, our human model, g , only captures the human behavior when the system is within the workspace $\mathcal{P}_H \times \mathcal{P}_R$. Applying the concept of safety as CA or SI to the workspace, we define *safe set* as $\mathcal{S} = \{(p_H, v_H, p_R, v_R) \in \mathcal{P}_H \times \mathbb{R}^{n_H} \times \mathcal{P}_R \times \mathcal{V}_R: (p_R, v_R) \in \mathcal{S}_{CA}(p_H) \cup \mathcal{S}_{SI}(v_H)\}$.

C. Safe Recovery Controller

We assume access to a *recovery controller*, π_{rec} , that can keep the human always in the workspace and physically safe. As shown in Sec. V-B, in case the trajectory optimization cannot find feasible solutions, our MPC will deploy π_{rec} to ensure that the human will remain in the workspace and safe.

In this work, we implement π_{rec} as a safety stop, that can keep the robot in place. We define the *recovery set* associated with π_{rec} as $\mathcal{S}_{rec} = \{(p_H, v_H, p_R, v_R) \in \mathcal{S}: v_R = 0\} \subset \mathcal{S}$. Let $f_{\pi_{rec}}$ denote the corresponding closed-loop system. Our assumption regarding π_{rec} is stated formally as follows:

Assumption 2. *The system is given π_{rec} with \mathcal{S}_{rec} , such that:*

$$\forall t_0 \in \mathbb{Z}_{\geq 0}, \forall (p_H^{t_0}, v_H^{t_0}, p_R^{t_0}, v_R^{t_0}) \in \mathcal{S}_{rec}, \forall t \geq t_0: (p_H^t, v_H^t, p_R^t, v_R^t) \in \mathcal{S}$$

where $\forall t \geq t_0: (p_R^{t+1}, v_R^{t+1}, p_H^{t+1}, v_H^{t+1}) = f_{\pi_{rec}}(p_R^t, v_R^t, p_H^t, v_H^t)$.

This assumption states that if the human-robot system is initially within \mathcal{S}_{rec} , then by executing π_{rec} , the system will always stay within the safe set \mathcal{S} . Intuitively, we assume that when $v_R = 0$ and the human is inside the workspace and safe, if the robot activates the safety stop, then the human will always stay inside the workspace and never injure themselves. Our safe MPC supports other implementations of π_{rec} , such as gravity compensation, as long as the above assumption is valid. Trivially executing π_{rec} would not allow the robot to complete any tasks. Thus, in Sec. V-B, we present an MPC algorithm for both task completion and guaranteed safety. The above assumption is necessary for the safety guarantee.

D. Probabilistic Safety

We define *probabilistic safety* (or δ -safety) – i.e., safety with a high probability throughout the operating time – as follows:

Definition 1 (δ -safety). *Let π be a controller with the closed-loop system f_π . Given $\delta \in (0, 1]$ and the initial states $(p_H^0, v_H^0, p_R^0, v_R^0) \in \mathcal{S}_{rec}$, the system is δ -safe under π iff*

$$\Pr[\forall t \in \mathbb{N}: (p_H^t, v_H^t, p_R^t, v_R^t) \in \mathcal{S}] \geq 1 - \delta$$

where $\forall t \in \mathbb{N}: (p_R^t, v_R^t, p_H^t, v_H^t) = f_\pi(p_R^{t-1}, v_R^{t-1}, p_H^{t-1}, v_H^{t-1})$.

Intuitively, δ -safety indicates that a human will remain inside the workspace and safe (as defined by CA or SI) with a high probability, if the human-robot system is initially within the recovery set \mathcal{S}_{rec} . Def. 1 implies that our high probability safety guarantee is *independent* of the duration of system operation [23]. By contrast, many other works ensure high probability safety guarantees *per time step*, in the form of $\forall t \in \mathbb{N}: \Pr[\text{system is safe at time } t] \geq 1 - \delta$. This comparison is noted again after Lemma. 1 in Sec. IV-A.

Problem Statement. *Design an MPC controller, π , that can guarantee a human-robot system is δ -safe during interaction.*

IV. HUMAN MODEL LEARNING AND PREDICTION

As described in Sec. II, the unknown function g in the human dynamic model (defined in Eq. 1) is learned via a Gaussian process (GP). GP allows our system to capture the uncertainty in the model and reliably predict human behavior.

In Sec. IV-A, we describe GP and make certain assumptions about the human behavior which are necessary for our safety guarantee. Then, in Sec. IV-B, we introduce the key steps necessary for reliably predicting human motion.

A. Human Model Learning via Gaussian Process

In this section, we first represent the multi-output function, g , equivalently using a single-output function, g' and then use GP to model g' . The reformulation allows us to use the standard definition of GP with a scalar output. Finally, we make certain regularization assumptions on g' and formulate the reliable confidence intervals of g .

The unknown function $g: \mathcal{P}_H \times \mathcal{P}_R \rightarrow \mathbb{R}^{n_H}$, with $n_H > 1$, is a multi-output function. Let $g_j(p_H, p_R)$ denote the j th dimension of the function output, $g(p_H, p_R)$, given (p_H, p_R) . We assume that all outputs g_1, \dots, g_{n_H} are independent of

each other. We equivalently represent g using a single-output function, $g': \mathcal{P}_H \times \mathcal{P}_R \times \mathcal{J} \rightarrow \mathbb{R}$, where $\mathcal{J} = [1 \dots n_H]$ [3]. Thus, for each $j \in \mathcal{J}$, we have $g_j(\cdot, \cdot) = g'(\cdot, \cdot, j)$.

We use a GP, denoted by $\mathcal{GP}(m, k)$, to represent the system's belief about the single-output function g' . Here, m denotes the prior mean function. Without loss of generality, we assume that $m \equiv 0$. The function k denotes the covariance (or kernel) function. We consider k as a composite kernel that is constructed additively. (We provide the details in App. B-A).

Let $\{\hat{g}^i\}_{i=1}^n$ denote the set of n past measurements of g at the input locations $\{(p_H^i, p_R^i)\}_{i=1}^n$. We assume that our measurements, along all dimensions, are independently corrupted by a zero-mean Gaussian noise with the same variance, denoted by $\lambda^2 \in \mathbb{R}$. Formally, for each $i \in [1 \dots n]$, $j \in \mathcal{J}$, we have $[\hat{g}^i]_j = g_j(p_H^i, p_R^i) + w_j^i$, where $w_j^i \sim \mathcal{N}(0, \lambda^2)$.

We construct a dataset of $n \cdot n_H$ measurements for g' by splitting the n measurements of g along each dimension $j \in \mathcal{J}$ and extending each input location by the index, j . Conditioned on these observations, the GP posterior prediction of g' at the test input (p_H, p_R, j) has a mean $\mu_{n \cdot n_H}(p_H, p_R, j)$ and a variance $\sigma_{n \cdot n_H}^2(p_H, p_R, j)$. The subscript $n \cdot n_H$ emphasizes that given each measurement of g , we obtain n_H measurements of g' – one for each output dimension of g . Given the assumption that g_1, \dots, g_{n_H} are independent of each other, the GP prediction of g' is equivalent to the prediction obtained by modeling each g_j as an independent GP [23]. Formally, for each $j \in \mathcal{J}$, we have that $\mu_{n \cdot n_H}(p_H, p_R, j) = \mu_{n, j}(p_H, p_R)$ and $\sigma_{n \cdot n_H}(p_H, p_R, j) = \sigma_{n, j}(p_H, p_R)$.

It is challenging to ensure safety for arbitrary human behavior, because the speed of human motion can be an order of magnitude faster than that of robots [15], and a moving human can inflict arbitrary collision forces upon robots [16]. In this work, we focus on the human behavior that is “smooth” [5] or has a “low” complexity [39], with smoothness measured under the norm of a reproducing kernel Hilbert space (RKHS). Let \mathcal{H}_k denote the RKHS corresponding to g' , where k is the kernel used by g' . The induced norm, $\|g'\|_k$, is a measure of the smoothness of g' with respect to k [5].

Assumption 3 (Koller et al. [22] Assumption 1). *The function g' has bounded RKHS norm induced by a continuously differentiable kernel k , i.e., $\|g'\|_k \leq B$.*

This assumption implies that the human behavior, equivalently represented by g' , remains smooth not only in collision-free cases, but also in the cases where collisions occur. Despite being strong, this above assumption is valid in practice when the robot is operating at a low speed, and our MPC is equipped with a low-level compliant controller. Compliant controllers [46, 21, 32] can use control authority to render a robot to behave like a virtual mass-spring-damper system that reacts to contact forces in a compliant manner.

Assumption 3 allows our system to use the GP prediction to build reliable confidence intervals of g given unseen inputs. We define information capacity, denoted by $\gamma_{n \cdot n_H}$, as the maximum mutual information between a finite set of $n \cdot n_H$ noisy samples and our function g' [3]. Since our measurements

are corrupted by λ -sub-Gaussian noise, given Assumption 3, we obtain the following confidence intervals:

Lemma 1 (Berkenkamp et al. [3] Lemma 1, Berkenkamp [1] Lemma 2). *Let $\beta_n = B + 4\lambda\sqrt{\gamma_{n,n_H} + 1 + \ln(1/\delta)}$. Then with a probability at least $1 - \delta$, with $\delta \in (0, 1)$, we have for all $j \in [1 \dots n_H]$, $n \in \mathbb{N}$, $(p_H, p_R) \in \mathcal{P}_H \times \mathcal{P}_R$, that*

$$|\mu_{n,j}(p_H, p_R) - g_j(p_H, p_R)| \leq \beta_n \cdot \sigma_{n,j}(p_H, p_R)$$

The lemma states that the confidence intervals (CIs) given by the GP prediction contain the unknown and deterministic function g , with a high probability. The CIs are necessary for one-step human motion prediction. Our system uses the CIs to reliably predict the reachable regions of the human position and velocity at the next time step, as shown in Sec. IV-B1. For multi-step prediction, the system applies one-step prediction *iteratively at every time step*, as presented in Sec. IV-B2. However, the high probability safety guarantee in our work is not ensured *per time step*. In fact, the CIs in Lemma 1 come from a frequentist rather than a Bayesian analysis [7]. This fact allows our guarantee to be *independent* of the duration of system operation, as noted in the end of Sec. III-D. (Please refer to App. B-B for more details).

For more details about GP posterior, GP prediction, RKHS, and information capacity, please refer to Srinivas et al. [39], Chowdhury and Gopalan [5], Koller et al. [23].

B. Robust Human Motion Prediction

Our system reliably predicts the human positions and velocities for multiple time steps into the future, based on the robust multi-step prediction theory introduced by Koller et al. [23]. We denote the finite prediction horizon by $T \in \mathbb{N}$.

Our system constructs ellipsoids to *outer-approximate* the reachable regions of the human positions and velocities in the future. The ellipsoidal outer approximations conservatively bound the true human future positions and velocities, with a high probability. In Sec. V, we design an MPC to guarantee δ -safety by enforcing safety constraints on these ellipsoids.

The rest of this section is organized as follows: In Sec. IV-B1, we formulate *one-step prediction*, that computes two ellipsoids that outer-approximate p_H^{t+1} and v_H^{t+1} , respectively, given p_R^t and an ellipsoid that outer-approximates p_H^t . In Sec. IV-B2, we formulate *multi-step prediction* that computes two sequences of ellipsoids that outer-approximate $\{p_H^t\}_{t=1}^T$ and $\{v_H^t\}_{t=1}^T$, respectively, given $\{p_R^t\}_{t=0}^{T-1}$ and an ellipsoid that outer-approximates p_H^0 . The system iteratively applies one-step prediction to achieve multi-step prediction.

Let $\mathcal{E}(c_p^t, Q_p^t) \subset \mathbb{R}^{n_H}$ denote the ellipsoid that outer-approximates the human position, p_H^t . Here, $c_p^t \in \mathbb{R}^{n_H}$ is the center and $Q_p^t \in \mathbb{R}^{n_H \times n_H}$ is the shape matrix (positive semi-definite). This ellipsoid is a set of possible human positions, i.e., $\mathcal{E}(c_p^t, Q_p^t) := \{p_H^t \in \mathbb{R}^{n_H} : (p_H^t - c_p^t)^T (Q_p^t)^{-1} (p_H^t - c_p^t) \leq 1\}$. Similarly, let $\mathcal{E}(c_v^t, Q_v^t) \subset \mathbb{R}^{n_H}$ denote the ellipsoid that outer-approximates the human velocity v_H^t . This ellipsoid is a set of possible human velocities. For notational shorthand, we

use $E_p^t = (c_p^t, Q_p^t)$ and $E_v^t = (c_v^t, Q_v^t)$.

$$E_p^{t+1} = \tilde{m}_p(E_p^t, p_R^t); \quad E_v^{t+1} = \tilde{m}_v(E_p^t, p_R^t) \quad (3)$$

1) *One-step prediction*: At every time step t , based on $p_H^{t+1} = f_H(p_H^t, p_R^t)$ (Eq. 1), we formulate one-step prediction for p_H^{t+1} as a function \tilde{m}_p as shown in Eq. 3. Here, $\mathcal{E}(E_p^{t+1})$ and $\mathcal{E}(E_p^t)$ are the ellipsoids that outer-approximate p_H^{t+1} and p_H^t , respectively. Similarly, based on $v_H^{t+1} = g(p_H^t, p_R^t)/h$ (Eq. 2), we formulate one-step prediction for v_H^{t+1} as a function \tilde{m}_v as shown in Eq. 3. Here, $\mathcal{E}(E_v^{t+1})$ is the ellipsoid that outer-approximates v_H^{t+1} . Intuitively, based on E_p^t , which describes an ellipsoid for p_H^t , and the precise knowledge of p_R^t , the system can build ellipsoids to outer-approximate p_H^{t+1} and v_H^{t+1} . (We provide the derivation for \tilde{m}_p and \tilde{m}_v in App. C-A.)

2) *Multi-step prediction*: Given a prediction horizon T , we formulate multi-step prediction as a function. The input is denoted by $(E_p^0, \{p_R^t\}_{t=0}^{T-1})$. Here, E_p^0 describes an ellipsoid for p_H^0 and $\{p_R^t\}_{t=0}^{T-1}$ denotes a fixed robot trajectory. The outputs, denoted by $(\{E_p^t\}_{t=1}^T, \{E_v^t\}_{t=1}^T)$, describe two sequences of ellipsoids that outer-approximate $\{p_H^t\}_{t=1}^T$ and $\{v_H^t\}_{t=1}^T$, respectively, with a high probability. We implement multi-step prediction by having the system iteratively applying one-step predictions. Formally, for all $t \in [0 \dots T-1]$, the system applies \tilde{m}_p and \tilde{m}_v , as shown in Eq. 3, to roll out the sequences of ellipsoids. Note that the item being “propagated” forward in time is E_p rather than E_v , since E_v^{t+1} is computed based on E_p^t instead of E_v^t . The ellipsoids $\{\mathcal{E}(E_p^t)\}_{t=1}^{T=4}$ in our 2D goal-reaching task are depicted in Fig. 4.

By applying Lemma 1, we obtain the following corollary:

Corollary 1. [Similar to Koller et al. [23] Corollary 7] *Let $\delta \in (0, 1]$. For all $n \in \mathbb{N}$, choose β_n based on Lemma 1. Then, the following holds jointly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}^0) \subset \mathcal{P}_H$, $p_{H,n}^0 \in \mathcal{E}(E_{p,n}^0)$, $\{p_{R,n}^\tau\}_{\tau=0}^{\tau=T-1} \subset \mathcal{P}_R$, $t \in [1 \dots T]$, with a probability at least $1 - \delta$:*

$$p_{H,n}^t \in \mathcal{E}(E_{p,n}^t), v_{H,n}^t \in \mathcal{E}(E_{v,n}^t)$$

where for all $n \in \mathbb{N}$, $\{E_{p,n}^\tau\}_{\tau=1}^T$ and $\{E_{v,n}^\tau\}_{\tau=1}^T$ are computed via multi-step prediction, such that for all $\tau \in [1 \dots T]$, $\mathcal{E}(E_{p,n}^\tau) \subset \mathcal{P}_H$.

We include the proof of this corollary in App. C-B based on Lemma 3, which itself is proved in App. C-A. The above corollary states that, with a high probability, the future human positions and velocities, across all time steps, are jointly contained within the ellipsoids generated by multi-step prediction. However, this holds true only if the ellipsoid $\mathcal{E}(E_{p,n}^t)$ is always inside the workspace, \mathcal{P}_H . Thus, as presented in Sec. V-B, to guarantee probabilistic safety, our MPC uses the robot’s control authority to keep $\mathcal{E}(E_{p,n}^t)$ always inside \mathcal{P}_H , while enforcing safety constraints on all $\mathcal{E}(E_{p,n}^t)$ and $\mathcal{E}(E_{v,n}^t)$.

V. SAFE MODEL PREDICTIVE CONTROLLER

In this section, we present an MPC algorithm that can guarantee δ -safety. In Sec. V-A, we integrate our definition of safety (CA or SI, as defined in Sec. III-B) with ellipsoidal

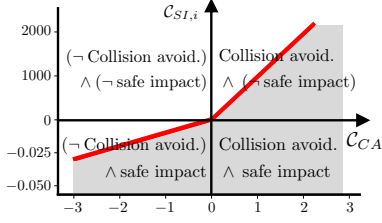


Fig. 3: The CA constraint is $\mathcal{C}_{CA}(p_R, E_p) > 0$. The SI constraints are $\forall i \in [1 \dots 2n_H]: \mathcal{C}_{SI,i}(v_R, E_v) \leq 0$. This plot shows the feasible regions of these constraints for a particular i by treating the function values, \mathcal{C}_{CA} and $\mathcal{C}_{SI,i}$, as independent variables. All pairs of $(\mathcal{C}_{CA}, \mathcal{C}_{SI,i})$ in the first quadrant indicate “CA and not SI”. Note that “not SI” refers to the situation of “unsafe impacts during (hypothetical) collisions”, which is not in conflict with “CA”. All pairs in the union of the first, third, and fourth quadrants indicate “CA or SI”, which implies safety according to our definition. All and only unsafe pairs are located within the second quadrant. The feasible region of the surrogate constraint, $\mathcal{C}_{SI,i} \leq \max(0.01 \mathcal{C}_{CA}, 1000 \mathcal{C}_{CA})$, is plotted in gray. The corresponding equality is represented by the red line segments. Our MPC ensures safety by enforcing the surrogate constraints for all $i \in [1 \dots 2n_H]$.

multi-step prediction. We then introduce the MPC algorithm in Sec. V-B, with proofs for human physical safety.

A. Safety Constraints with Ellipsoidal Uncertainty

In Sec. III-B, we define safety as collision avoidance (CA) or safe impact (SI). In Sec. IV-B, we introduce the steps to predict the human positions and velocities as the ellipsoids, E_p and E_v , respectively. In this section, we formulate safety (CA or SI) as a set of constraints representing the relations between the robot state, p_R, v_R , and the ellipsoids, E_p, E_v .

To ensure safety as CA, since our robot is modeled as a point mass, our system can enforce that the robot always stays outside the ellipsoids for human positions. Formally, we denote the CA constraint as follows: $\mathcal{C}_{CA}(p_R, E_p) = (p_R - c_p)^T (Q_p)^{-1} (p_R - c_p) - 1 > 0$, where $E_p = (c_p, Q_p)$.

To ensure safety as SI, our system can enforce the constraint $\Omega(v_R, v_H) \leq \Omega_{max}$, as described in Sec. III-A, to all $v_H \in E_v$. Through a conservative relaxation, we arrive at a set of SI constraints as follows: $\forall i \in [1 \dots 2n_H]: \mathcal{C}_{SI,i}(v_R, E_v) \leq 0$. (We provide the derivation of this relaxation in App. D-A.)

In order to implement safety as “CA or SI,” we must disjunctively combine the constraints for CA with those for SI. The disjunctive normal form is $[\mathcal{C}_{CA} > 0] \vee [\bigwedge_{i \in [1 \dots 2n_H]} \mathcal{C}_{SI,i} \leq 0]$, with function arguments omitted for notational convenience. Here, \vee and \wedge denote the operators of logical disjunction and conjunction, respectively. Equivalently, the conjunctive normal form is $\bigwedge_{i \in [1 \dots 2n_H]} [\mathcal{C}_{CA} > 0 \vee \mathcal{C}_{SI,i} \leq 0]$. In Fig. 3, the feasible region of the one-dimensional constraint

$$\mathcal{C}_{CA}(p_R, E_p) > 0 \vee \mathcal{C}_{SI,i}(v_R, E_v) \leq 0 \quad (4)$$

is plotted as the union of the first, third, and fourth quadrants.

We conservatively approximate the constraint in Eq. 4 using a *surrogate constraint*: $\mathcal{C}_{SI,i} \leq \max(\theta_1 \mathcal{C}_{CA}, \theta_2 \mathcal{C}_{CA})$, with hyper-parameters $\theta_1 \geq 0$ and $\theta_2 \geq 0$. The feasible region of the surrogate constraint, in the case where $\theta_1 = 0.01$ and

$\theta_2 = 1000$, is the gray region in Fig. 3. Since the gray region is a strict subset of the union of the first, third, and fourth quadrants, satisfying the surrogate constraint is a sufficient but not necessary condition for satisfying the constraint in Eq. 4.

All relaxations in this section conservatively preserve safety, yielding the following lemma, that is proved in App. D-B:

Lemma 2. *Given hyper-parameters $\theta_1 \geq 0$, $\theta_2 \geq 0$, and the ellipsoids described by E_p and E_v , then the following holds*

$$\begin{aligned} & \bigcap_{i \in [1 \dots 2n_H]} \left\{ (p_R, v_R) : \mathcal{C}_{SI,i}(v_R, E_v) \leq \max(\theta_1 \mathcal{C}_{CA}(p_R, E_p), \theta_2 \mathcal{C}_{CA}(p_R, E_p)) \right\} \\ & \subset \bigcap_{p_H \in \mathcal{E}(E_p)} \bigcap_{v_H \in \mathcal{E}(E_v)} [\mathcal{S}_{CA}(p_H) \cup \mathcal{S}_{SI}(v_H)] \end{aligned}$$

This lemma states that satisfying the surrogate constraint, for all $i \in [1 \dots 2n_H]$, is a sufficient condition for safety (CA or SI), for all the human positions and velocities inside the ellipsoids. Admittedly, enforcing the surrogate constraints would prevent some safe solutions from being identified, but would never make any unsafe solutions feasible. Next, our MPC will use these surrogate constraints to guarantee safety.

B. Safe Model Predictive Controller

In this section, we introduce our MPC that can guarantee δ -safety. First, we formulate a trajectory optimization (TO) problem that our system can solve for safe robot trajectories, given a finite planning horizon, denoted by $T \in \mathbb{N}$. Then, we introduce our MPC that solves the TO iteratively in a receding horizon fashion and present the δ -safety guarantee.

Our nonlinear TO problem is formulated as follows:

$$\begin{aligned} & \text{Minimize } J \\ & \text{Subject to} \\ & (a) \forall t \in [1 \dots T]: E_p^t = \tilde{m}_p(E_p^{t-1}, p_R^{t-1}), E_v^t = \tilde{m}_v(E_v^{t-1}, p_R^{t-1}) \\ & (b) \forall t \in [1 \dots T]: \mathcal{E}(E_p^t) \subset \mathcal{P}_H, p_R^t \in \mathcal{P}_R \\ & (c) \forall t \in [1 \dots T], \forall i \in [1 \dots 2n_H]: \\ & \quad \mathcal{C}_{SI,i}(v_R^t, E_v^t) \leq \max(\theta_1 \mathcal{C}_{CA}(p_R^t, E_p^t), \theta_2 \mathcal{C}_{CA}(p_R^t, E_p^t)) \\ & (d) v_R^T = 0 \\ & (e) \forall t \in [1 \dots T]: v_R^t \in \mathcal{V}_R, u^t \in \mathcal{U}, (p_R^t, v_R^t) = f_R(p_R^{t-1}, v_R^{t-1}, u^t) \end{aligned}$$

The TO problem contains an objective function, denoted by J , and a set of constraints. Constraint (a) implements the ellipsoidal human motion prediction as formulated in Sec. IV-B. (b) ensures that all the ellipsoids that outer-approximate the human positions are inside the workspace, \mathcal{P}_H , while all the robot positions are inside \mathcal{P}_R . This constraint is required by Corollary 1, which guarantees with a high probability that the human positions and velocities are always inside the ellipsoids. (c) implements the surrogate constraints, as formulated in Sec. V-A, that enable the system to ensure safety as CA or SI, regarding all the ellipsoids. (d) ensures that the robot could possibly arrive inside the recovery set at the terminal time, T . This constraint allows our MPC to potentially guarantee safety even beyond the finite horizon, T , by deploying the safety stop, as described in Sec. III-C, after T . (e) include the boundary conditions and robot dynamics, as defined in Sec. II.

The solutions to the TO guarantees jointly with a high probability that within the finite horizon T , the human always

stays inside the workspace \mathcal{P}_H and safe, that is proved in App. E-A. To further ensure δ -safety, which requires that the system stays safe for all $t \in \mathbb{N}$, we propose an MPC algorithm.

Our MPC solves the TO iteratively in a receding horizon fashion, and uses the recovery controller, π_{rec} (defined in Sec. III-C) in case the TO cannot find feasible solutions. We design the MPC as follows: At any given time $t \in \mathbb{N}$, if the TO finds a feasible solution (denoted by $\Pi_t = \{u_t^1, \dots, u_t^T\}$), then the robot will execute the first control, u_t^1 . In the next time step, $t + 1$, if the TO cannot find feasible solutions, the system will reuse the solution from the previous iteration at t . In particular, the system will shift Π_t in a receding horizon manner and append π_{rec} to the end. It obtains a new trajectory, $\Pi_{t+1} = \{u_t^2, \dots, u_t^T, \pi_{rec}\}$ and will execute the first control, $u_{t+1}^1 = u_t^2$. The system will then repeat this process until a new feasible solution is found. The pseudo-code is available in Alg. 1 in App. E-B. Our MPC algorithm guarantees δ -safety as stated by the following theorem, which we prove in App. E-B.

Theorem 1 Let $\delta \in (0, 1]$. For all $n \in \mathbb{N}$, choose β_n based on Lemma 1. The system has access to the initial state $(p_H^0, v_H^0, p_R^0, v_R^0) \in \mathcal{S}_{rec}$. Then, the human-robot system is δ -safe under the MPC algorithm.

VI. EXPERIMENT

Our focus is the empirical evaluation of improvements to task efficiency resulting from our two-pronged definition of safety. Hence, we benchmarked our MPC (henceforth referred to as *CASI*), wherein safety is defined as CA or SI, along with its variation (henceforth referred to as *CA*), wherein safety is defined only as CA. We evaluated these algorithms using two tasks: a simulated 2D goal-reaching task and a real-world robot-assisted dressing task. Our hypothesis was that CASI would result in a smaller number of iterations necessary for the completion of both tasks. For empirical evaluation regarding the proved safety guarantee, please refer to Koller et al. [23].

We considered a realistic scenario in which a system designer could first collect non-interactive data of the human performing a task without the robot, then allow the robot to operate around the human in order to collect more interactive data, as described at the end of Sec. I. Accordingly, we collected a small, non-interactive dataset by fixing the robot at a predefined location, p_R^0 , and letting the human (simulated or real) roll out their policy. Then, we ran the MPC with the trained GP until the robot reached the goal, where only the GP's posterior (and not its hyper-parameters) was updated between iterations. During our experiments, we expected our robot to act safely and efficiently under the uncertainty caused by the distribution shift from the initial training scenarios where $p_R = p_R^0$ to the testing scenarios where p_R moved.

For both tasks, we implemented π_{rec} as a safety stop, as described in Sec. III-C. In keeping with previous work [2, 23], since the theoretical confidence intervals for the GP model are conservative, we chose to fix $\beta = 2$ to enable task completion under uncertainty.

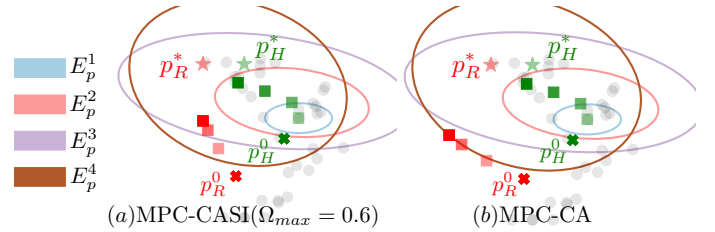


Fig. 4: A comparison, in the form of robot trajectories and human ellipsoidal predictions, of the feasible solutions found by CASI and CA in the 2D goal-reaching domain. The red \blacksquare = the robot positions for $t \in [1 \dots 4]$ along the trajectory. Here, the robot positions for $t = 3$ and 4 overlap due to the constraint (d) in our TO as formulated in Sec. V-B. For $t \in [1 \dots 4]$, the human ellipsoids, E_p^t , are plotted in different colors, whose centers, c_p^t , are the green \blacksquare . The grey dots indicate the input human positions within the initial dataset for GP training. CASI produced a more efficient path by allowing the path to enter the ellipsoids, which CA does not allow. Thus, defining safety as CA or SI provided more flexibility than CA alone, allowing the planner to be less conservative while still guaranteeing safety.

A. 2D Goal-Reaching

We simulated human behavior within a 2D environment by online-optimizing a continuous trajectory to track a pre-computed discrete MDP policy. This policy aimed to reach a predefined goal, denoted as p_H^* , while avoiding obstacles in the environment. We generated five environments with randomly located obstacles denoted as $Env1, \dots, Env5$; these obstacles were used only for the human, not the robot.

We designed three objective functions for the human's trajectory optimization. The first, *H-Indep-R*, was designed to track the MDP policy. The other two, *H-To-R* and *H-Away-R*, were minimizing and maximizing (respectively) the distance to the current robot position, besides policy tracking. We anticipated that our human model g in Eq.1 could capture the dependency of the human's behavior upon the robot states.

The robot's task was to efficiently reach the goal while ensuring the simulated human's safety, with a predefined starting location of p_R^0 and goal location of p_R^* . Our initial dataset contained 45 input-output pairs collected from 3 rollouts of the simulated human. Fig. 4 shows a snapshot of the robot and human in the environment along with the human prediction.

We benchmarked two variations of CASI, CASI ($\Omega_{max} = 0.6$) and CASI ($\Omega_{max} = 0.3$), against CA. Here, Ω_{max} denotes the maximum allowable impact potential as defined in Sec. III-A and we set its values according to Heinzmann and Zelinsky [16]. We ran 30 trials for each condition and evaluated performance based on the following measurements: (1) *#Itr*: the number of iterations taken for the robot to reach p_R^* (per trial); (2) *#SafeColl*: the number of collisions involving safe impact (per trial); (3) *#UnsafeColl*: the number of collisions involving unsafe impact (per trial); and (4) *PlanTime* (s): the amount of time taken to solve the TO (per iteration).

B. Robot-Assisted Dressing

We deployed our algorithm to perform a real-world robot-assisted dressing task, wherein the robot must dress a sleeveless jacket onto a human arm (the human's fist is already inside the armhole upon task initiation). In this task, the robot

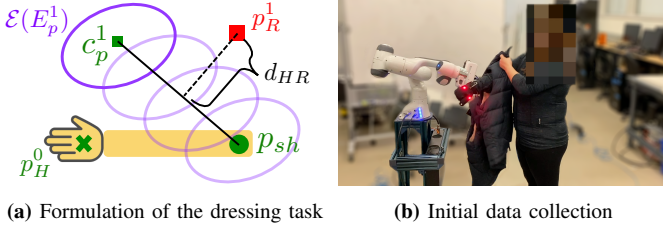


Fig. 5: (a) We modeled the human hand as a point mass, and approximated the arm by linearly interpolating between the hand position, p_H , and the shoulder position, p_{sh} . At $t = 0$, the hand is at p_H^0 (green \times) and the approximated arm is drawn in orange. The system computed an ellipsoid $\mathcal{E}(E_p^1)$ (in dark purple) to bound the hand position, p_H^1 . Then, the system duplicated $\mathcal{E}(E_p^1)$ multiple times (in light purple) to approximately bound the arm configuration along the line between the center of $\mathcal{E}(E_p^1)$ (green \blacksquare) and p_{sh} . The armhole constraint at time $t = 1$ enforced that the distance d_{HR} (black curly brackets) between p_R^1 and the interpolated line (black solid line) must be $\leq d_{HR}^{max}$. (b) Initial non-interactive data was collected by allowing the subject to dress herself.

runs the MPC, integrated with a framework that interleaves planning and execution [43], to find paths for its end-effector to reach the goal, p_R^* , near the human shoulder position, denoted by p_{sh} . Both p_R^* and p_{sh} are assumed to be known and fixed.

We assumed that the human elbow never bends during the task, and accordingly modeled the human hand as a point mass and approximated the arm configuration by linear interpolation between the hand and p_{sh} . Our system also duplicated the ellipsoidal predictions for the hand positions to bound the arm configuration, as described in Fig. 5a. Accordingly, the TO used all ellipsoids for hand position and arm configuration, as well as ellipsoids for hand velocity, when ensuring safety constraints. We have considered lifting this assumption in future work by drawing insights from prior art [9, 50, 20, 6, 19].

As shown in Fig. 1, the robot must remain physically close to the human arm due to the limited size of the armhole. We encoded the armhole constraint approximately by enforcing $d_{HR}^t \leq d_{HR}^{max}$ for all $t \in [1 \dots T]$. Here, d_{HR}^t denotes the distance between p_R^t and the line between p_{sh} and c_p^t , the center of the ellipsoid $\mathcal{E}(E_p^t)$, as illustrated in Fig. 5a. The threshold d_{HR}^{max} encodes the size of the armhole.

Our initial dataset contained 16 input-output pairs collected from two human rollouts during which the human used her left arm to dress her right arm while the robot remained static, as shown in Fig. 5b. We conducted a case study to compare CASI ($\Omega_{max} = 1$) against CA under three conditions wherein d_{HR}^{max} was set to 0.08, 0.085, 0.09m. We measured $\#Itr$ and $TotalTime$ (s), which is the total amount of time taken by the robot to reach p_R^* (different from $\#PlanTime$).

VII. RESULTS

A. 2D Goal-Reaching

We benchmarked the 3 algorithms – CASI ($\Omega_{max} = 0.6$), CASI ($\Omega_{max} = 0.3$), and CA – using 15 different simulated human behaviors (5 environments \times 3 objective functions). The results of $\#Itr$, $\#SafeColl$, and $PlanTime$ under all conditions are presented in Table II in App. F. For each mea-

surement, we conducted Wilcoxon signed-rank test to perform pairwise comparisons among the 3 algorithms. In this section, we discuss the results of $\#UnsafeColl$, $\#Itr$, and $PlanTime$. (The discussion of the results of $\#SafeColl$ is in App. F.)

In all conditions, $\#UnsafeColl$ was always 0, verifying the safety guarantee provided by all three algorithms.

The results of $\#Itr$ are plotted in Fig. 6. Both CASI ($\Omega_{max} = 0.6$) and CASI ($\Omega_{max} = 0.3$) had a significantly lower $\#Itr$ than CA (both $p < 0.001$ resulting from the Wilcoxon signed-rank test); hence, CASI produced significantly more efficient plans than CA. The definition of safety as CA or SI, rather than just CA, allowed the planner to be less conservative while still guaranteeing safety. This is illustrated in Fig. 4, where even though the ellipsoids in CASI and in CA had roughly similar areas, CASI produced a more efficient path by allowing the path to enter the ellipsoids, which CA does not allow. In addition, CASI ($\Omega_{max} = 0.6$) had a significantly lower $\#Itr$ than CASI ($\Omega_{max} = 0.3$) ($p < 0.001$), which implies that the benefit to efficiency increases as Ω_{max} increases.

Both CASI ($\Omega_{max} = 0.6$) and CASI ($\Omega_{max} = 0.3$) had a significantly longer $PlanTime$ than CA (both $p < 0.001$). One explanation is that CASI needs to construct ellipsoids for both human positions and velocities, while CA only needs to do so for human positions. In addition, CASI ($\Omega_{max} = 0.3$) had a significantly longer $PlanTime$ than CASI ($\Omega_{max} = 0.6$) ($p < 0.001$), which implies that solving the TO becomes harder as Ω_{max} decreases. In real-time control scenarios where robots need to replan at 50Hz [17], the extra computation in our MPC could counteract the benefits in achieving a lower $\#Itr$. However, in many HRI scenarios, where the robot is equipped with compliant controllers, replanning at a much lower frequency could suffice. In such cases, our MPC’s strength in finding safe and efficient solutions could potentially outweigh its weakness in longer planning time, resulting in a shorter total time in task completion, which is demonstrated in the robot-assisted dressing domain in Sec. VII-B.

B. Robot-Assisted Dressing

We ran our system with one subject as a case study, leaving a full user study for future work. Fig. 7 depicts the trajectory of robot positions recorded during a single task execution along three dimensions (x, y, z) in Cartesian space. Subfigure (a) represents the execution in the case of $d_{HR}^{max} = 0.08m$, meaning that the robot end-effector had to stay within 0.08m of the human arm. Our planner, CASI ($\Omega_{max} = 1$), was able to reach the goal in 11 iterations and 29.30s, while CA reached the goal in 49 iterations and 223.52s. One explanation for the inefficiency observed with CA is that it is challenging to satisfy both the armhole constraints ($d_{HR} \leq d_{HR}^{max}$) and collision avoidance constraints while under large uncertainty in the human model. As depicted in Fig. 5a, if the uncertainty represented by the ellipsoids is huge, then collision avoidance motivates p_R^1 to stay very far from the human arm (solid black line). However, the armhole constraint requires p_R^1 to stay close to the human arm; hence, CA needed to collect much more data in order to significantly reduce the uncertainty (size

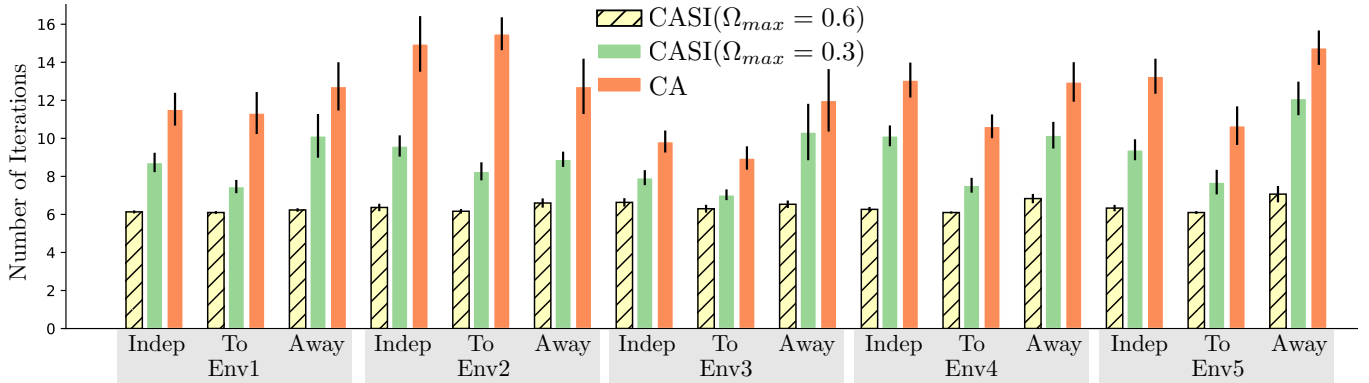


Fig. 6: The mean and standard error of #Itr in the 2D goal-reaching domain. The benchmark included running CASI ($\Omega_{max} = 0.6$), CASI ($\Omega_{max} = 0.3$), and CA, with 15 different simulated human behaviors (5 environments \times 3 objective functions). CASI ($\Omega_{max} = 0.6$) achieved the highest efficiency, while CA yielded the lowest.

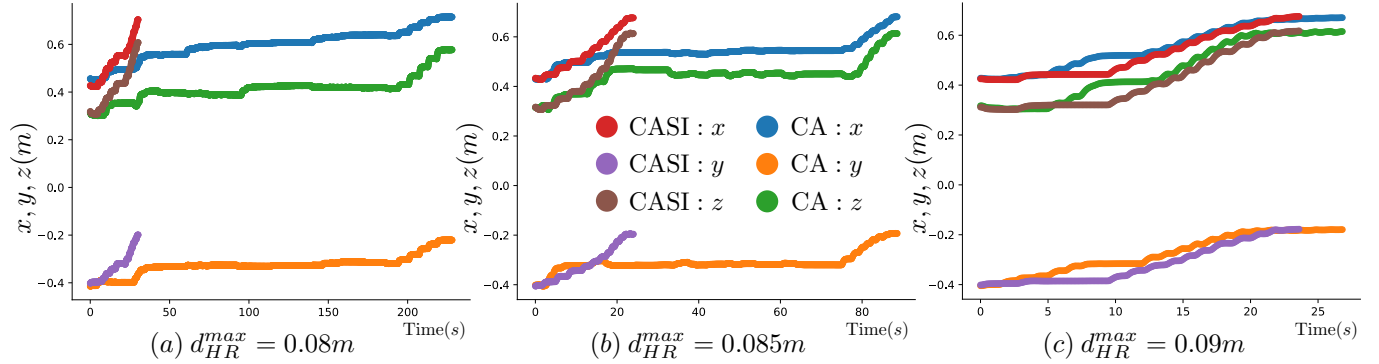


Fig. 7: The trajectories of the robot end-effector positions, recorded during one execution of the assisted dressing task. Each subplot depicts trajectories along three dimensions (x, y, z) produced by two algorithms, CASI ($\Omega_{max} = 1$) and CA; hence, each subplot includes six trajectories. (a) presents the case of $d_{HR}^{max} = 0.08m$, which means the robot end-effector had to remain within $0.08m$ of the human arm. (b) and (c) present the case of $d_{HR}^{max} = 0.085m$ and $0.09m$, respectively. In both (a) and (b), CASI achieved much higher efficiency than CA. In (c), CASI and CA achieved a similar efficiency, though CASI was slightly more efficient.

of the ellipsoids) before finding feasible plans. In contrast, CASI allowed greater flexibility by ensuring safety as CA or SI. Thus, even when uncertainty about future human motion is relatively high, the robot can still find feasible plans by ensuring safe impact, resulting in more efficient task completion.

As we slightly relaxed the armhole constraint, the robot gained more freedom for collision avoidance. When $d_{HR}^{max} = 0.085m$, CA accomplished the task in 40 iterations and 87.02s — still much longer than CASI ($\Omega_{max} = 1$), which completed the task in 13 iterations and 23.22s. When $d_{HR}^{max} = 0.09m$, both algorithms yielded similar performance, with CASI ($\Omega_{max} = 1$) still a bit more efficient than CA (12 iterations and 22.57s compared with 32 iterations and 25.78s, respectively). In conclusion, CASI was able to both accomplish the task and ensure safety, even when close contact between the human and robot was unavoidable.

VIII. RELATED WORK

In this work, we formally define human safety by incorporating insights from the fields of HAMP and compliant control. Our safe planner is a HAMP algorithm and could serve as a complement to compliant control approaches.

In the field of HAMP, Leung et al. [28] developed a reachability-based controller that assures human safety by maintaining the availability of a collision-free maneuver, given a known human model. Our work focuses on ensuring safety under the uncertainty in learned human dynamic models. Park et al. [35] developed a planner that enables a robot to avoid collisions by predicting both the human short-term motion and subtasks. Sadigh and Kapoor [37] proposed a controller synthesis algorithm that can guarantee safety that is specified as probabilistic predicates. Lütjens et al. [30] developed a safe navigation method by leveraging the uncertainty in a black-box human prediction model. Our method focuses on both estimating the uncertainty in human models and conservatively propagating it forward in time for safe planning.

Fisac et al. [10], Fridovich-Keil et al. [12] developed a confidence-aware safe HAMP algorithm. There are a couple of differences between their work and ours. Their human model is a noisy-rationality Boltzmann model, which is less restricted than ours that assumes that the human behavior is deterministic. However, our human model captures the dependence of human motion on robot states, which could be useful in HRI tasks where the human and robot are tightly coupled, such

as robot-assisted dressing. Our planner also provides a high probability safety guarantee that is independent of the duration of system operation, as noted after Def. 1.

IX. CONCLUSION

In this paper, we proposed a two-pronged definition of safety for human-aware motion planning: “collision avoidance or safe impact.” We developed a motion planner able to guarantee human physical safety and leverage our safety definition to find efficient plans. We deployed our algorithm for a simulated task and a real-world robot-assisted dressing task, and demonstrated its benefit for completing these tasks safely and efficiently even when collision avoidance was difficult.

X. FUTURE WORK

We offer a novel way to combine insights from the fields of motion planning and compliant control. One future direction is to develop planners to schedule the stiffness of compliant controllers for safe and efficient human-robot force interaction.

Our work introduces a principled method for safe, on-policy interactive data collection. One next step is to verify the performance of personalization, as discussed in Sec.I, via user studies. Our framework with safety guarantees also opens up avenues for investigating human psychological safety [26] that could be measured by eye trackers [33] and questionnaires.

From the theoretical perspective, extending the point-mass representation and lifting model assumptions provide new opportunities for ensuring safety in real-world HRI problems. Future directions include developing safe planners to handle complex geometry, multibody robot models, and second order, discontinuous, multi-modal, and stochastic human models.

XI. ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research (Sponsor Award ID N00014-18-1-2815). We greatly appreciate anonymous reviewers for constructive, thorough, and rigorous reviews, Felix Berkenkamp and Theodoros Stouraitis for invaluable advice on improving this paper.

REFERENCES

- [1] Felix Berkenkamp. *Safe exploration in reinforcement learning: Theory and applications in robotics*. PhD thesis, ETH Zurich, 2019. 5
- [2] Felix Berkenkamp, Matteo Turchetta, Angela P Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, 2017. 7
- [3] Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *The International Journal of Robotics Research*, 2018. 4, 5, 13
- [4] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. Trust-aware decision making for human-robot collaboration: Model learning and planning. *ACM Transactions on Human-Robot Interaction (THRI)*, 2020. 1
- [5] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, 2017. 4, 5
- [6] Alexander Clegg, Zackory Erickson, Patrick Grady, Greg Turk, Charles C Kemp, and C Karen Liu. Learning to collaborate from simulation for robot-assisted dressing. *Robotics and Automation Letters*, 2020. 8
- [7] Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. In *Advances in Neural Information Processing Systems*, 2020. 5, 13
- [8] Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006. 1
- [9] Zackory Erickson, Maggie Collier, Ariel Kapusta, and Charles C Kemp. Tracking human pose during robot-assisted dressing using single-axis capacitive proximity sensing. *Robotics and Automation Letters*, 2018. 8
- [10] Jaime F Fisac, Andrea Bajcsy, Sylvia L Herbert, David Fridovich-Keil, Steven Wang, Claire J Tomlin, and Anca D Dragan. Probabilistically safe robot planning with confidence-based human predictions. 2018. 9
- [11] Fabrizio Flacco, Torsten Kröger, Alessandro De Luca, and Oussama Khatib. A depth space approach to human-robot collision avoidance. In *IEEE International Conference on Robotics and Automation*, 2012. 1
- [12] David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. Confidence-aware motion prediction for real-time collision avoidance. *The International Journal of Robotics Research*, 2020. 9
- [13] Shromona Ghosh, Felix Berkenkamp, Gireeja Ranade, Shaz Qadeer, and Ashish Kapoor. Verifying controllers against adversarial examples with bayesian optimization. In *International Conference on Robotics and Automation*, 2018. 13
- [14] Sami Haddadin, Alin Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008. 1
- [15] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 2017. 1, 4
- [16] Jochen Heinzmann and Alexander Zelinsky. Quantitative safety guarantees for physical human-robot interaction. *The International Journal of Robotics Research*, 2003. 3, 4, 7, 13
- [17] Lukas Hewing, Juraj Kabzan, and Melanie N Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, 2019. 8
- [18] Guy Hoffman. Evaluating fluency in human-robot collaboration. *IEEE Transactions on Human-Machine Systems*, 2019. 2
- [19] Ravi Prakash Joshi, Nishanth Koganti, and Tomohiro Shibata. A framework for robotic clothing assistance by imitation learning. *Advanced Robotics*, 2019. 8
- [20] Ariel Kapusta, Zackory Erickson, Henry M Clever, Wenhao Yu, C Karen Liu, Greg Turk, and Charles C Kemp. Personalized collaborative plans for robot-assisted dressing via optimization and simulation. *Autonomous Robots*, 2019. 2, 8
- [21] Arvid QL Keemink, Herman van der Kooij, and Arno HA Stienen. Admittance control for physical human-robot interaction. *The International Journal of Robotics Research*, 2018. 4
- [22] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *IEEE Conference on Decision and Control*, 2018. 4, 15
- [23] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, Joschka Boedecker, and Andreas Krause. Learning-based model predictive control for safe exploration and reinforcement learning. 2019. URL <https://arxiv.org/abs/1906.12189>. 1, 4, 5, 7, 14, 15, 16, 17
- [24] Andreas Krause and Cheng Soon Ong. Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, 2011. 13
- [25] Przemyslaw A Lasota and Julie A Shah. Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration. *Human factors*, 2015. 3
- [26] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*, 2017. 1, 3, 10
- [27] Séverin Lemaignan, Mathieu Warnier, E Akin Sisbot, Aurélie Clodic, and Rachid Alami. Artificial cognition for social human-robot interaction: An implementation. *Artificial Intelligence*, 2017. 1
- [28] Karen Leung, Edward Schmerling, Mengxuan Zhang, Mo Chen, John Talbot, J Christian Gerdes, and Marco Pavone. On infusing reachability-based safety assurance within planning frameworks for human-robot vehicle interactions. *The International Journal of Robotics Research*, 2020. 1, 9
- [29] Shen Li and Julie A Shah. Safe and efficient high dimensional motion planning in space-time with time parameterized prediction. In *International Conference on Robotics and Automation*, 2019. 1
- [30] Björn Lütjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *International Conference on Robotics and Automation*, 2019. 9
- [31] Jim Mainprice, E Akin Sisbot, Léonard Jaillet, Juan Cortés, Rachid Alami, and Thierry Siméon. Planning human-aware motions using a sampling-based costmap planner. In *2011 IEEE International Conference on Robotics and Automation*, 2011. 1
- [32] Eleonora Mariotti, Emanuele Magrini, and Alessandro De Luca. Admittance control for human-robot interaction using an industrial robot equipped with a f/t sensor. In *International Conference on Robotics and Automation*, 2019. 1, 4
- [33] Benjamin A. Newman, Reuben M. Aronson, Siddhartha S. Srinivasa, Kris Kitani, and Henny Admoni. HARMONIC: A Multimodal Dataset of Assistive Human-Robot Collaboration. 2018. URL <https://arxiv.org/abs/1807.11154>. 10
- [34] Jae Sung Park, Chonhyon Park, and Dinesh Manocha. Intention-aware motion planning using learning based human motion prediction. In *Robotics: Science and Systems*, 2017. 1
- [35] Jae Sung Park, Chonhyon Park, and Dinesh Manocha. I-planner: Intention-aware motion planning using learning-based human motion prediction. *The International Journal of Robotics Research*, 2019. 1, 3, 9
- [36] Stefania Pellegrinelli, Henny Admoni, Shervin Javdani, and Siddhartha Srinivasa. Human-robot shared workspace collaboration via hindsight optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016. 1
- [37] Dorsa Sadigh and Ashish Kapoor. Safe control under uncertainty with probabilistic signal temporal logic. In *Proceedings of Robotics: Science and Systems*, 2016. 9
- [38] Riccardo Schiavi, Antonio Bicchi, and Fabrizio Flacco. Integration of active and passive compliance control for safe human-robot coexistence. In *IEEE International Conference on Robotics and Automation*, 2009. 1
- [39] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process bandits without regret: An experimental design approach. In *International Conference on Machine Learning*, 2010. URL <http://arxiv.org/abs/0912.3995>. 4, 5
- [40] Gokul Swamy, Jens Schulz, Rohan Choudhury, Dylan Hadfield-Menell, and Anca Dragan. On the utility of model learning in hri. *arXiv preprint arXiv:1901.01291*, 2019. URL <https://arxiv.org/abs/1901.01291>. 2
- [41] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010. 1
- [42] Vaibhav V Unhelkar and Julie A Shah. Learning models of sequential decision-making with partial specification of agent behavior. In *AAAI Conference on Artificial Intelligence*, 2019. 1
- [43] Vaibhav V Unhelkar, Przemyslaw A Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and*

Automation Letters, 2018. 8

- [44] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. Decision-making for bidirectional communication in sequential human-robot collaborative tasks. In *ACM/IEEE International Conference on Human-Robot Interaction*, 2020. 1
- [45] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. Semi-supervised learning of decision-making models for human-robot collaboration. In *Conference on Robot Learning*, 2020. 2
- [46] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, MG Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and autonomous systems*, 2013. 4
- [47] Ian D Walker. Impact configurations and measures for kinematically redundant and multiple armed robot systems. *IEEE transactions on robotics and automation*, 1994. 13
- [48] Jens Wittenburg. *Dynamics of multibody systems*. Springer-Verlag Berlin Heidelberg, 2008. 13
- [49] Keenan A Wyrobek, Eric H Berger, HF Machiel Van der Loos, and J Kenneth Salisbury. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *International Conference on Robotics and Automation*, 2008. 1
- [50] Fan Zhang, Antoine Cully, and Yiannis Demiris. Probabilistic real-time user posture tracking for personalized robot-assisted dressing. *Transactions on Robotics*, 2019. 8

APPENDIX A
IMPACT POTENTIAL

Heinzmann and Zelinsky [16] formulated a constraint on a robot’s position and velocity to ensure safe human-robot impact during collisions, while assuming that the human remains static. In this section, we adapt that formulation to the case in which the human is moving. Given our point-mass assumption for both human and robot, our safe impact constraint depends on the human’s and robot’s velocities.

We consider the case in which two general bodies collide. By assuming that the collision occurs within an infinitesimally small period of time $\Delta t \rightarrow 0$, we can treat both bodies during impact as rigid bodies (according to CH.6.1 in Wittenburg [48]). Let $p_H, v_H, p_R,$ and v_R denote the human’s position, velocity, robot’s position, and velocity immediately before a collision, respectively. Let Δv_H and Δv_R denote the changes to v_H and v_R immediately following the collision, respectively. As $\Delta t \rightarrow 0$, during Δt , both p_H and p_R don’t change, while v_H and v_R remain finite. The impulse, denoted by $\hat{F} = \lim_{\Delta t \rightarrow 0} \int_t^{t+\Delta t} F(s) ds$, remains finite, where F is the impulsive force that tends toward infinity as $\Delta t \rightarrow 0$, according to CH6.1 in Wittenburg [48].

According to Eq.(8) in Walker [47] or Eq.(6.9) in Wittenburg [48], we have the following kinematic relationship:

$$[(v_H + \Delta v_H) - (v_R + \Delta v_R)]^T \mathbf{n} = -e(v_H - v_R)^T \mathbf{n} \quad (5)$$

where \mathbf{n} is the unit normal vector to the common tangent plane at the point of collision [48]. The parameter $e \in [0, 1]$ denotes the *coefficient of restitution*. The value of e is 0 for purely plastic collisions and 1 for purely elastic collisions [47].

Under our representation where both bodies are point masses, as introduced in Sec. II, by following Sec. II(B) in Walker [47], we obtain that $\Delta v_H = -\hat{F}/m_H$ and $\Delta v_R = \hat{F}/m_R$. Here, we follow the convention that F is the force exerted by the human and applied to the robot. By plugging both expressions into Eq. 5, we get the following equation:

$$\hat{F}^{mag} = \frac{-(e+1)(v_R^T - v_H^T) \cdot \mathbf{n}}{\frac{1}{m_R} + \frac{1}{m_H}} \quad (6)$$

where \hat{F}^{mag} denotes the magnitude of \hat{F} , i.e., $\hat{F} = \hat{F}^{mag} \mathbf{n}$. Our Eq. 6 is equivalent to Eq.(6.16) in Wittenburg [48] applied to point masses.

We further adopt the assumption made by Heinzmann and Zelinsky [16] that at the moment of contact, there is sufficient friction to align \hat{F} and $(v_R - v_H)$. Given this assumption, we have $\mathbf{n} = -(v_R - v_H) / \|v_R - v_H\|_2$. By plugging this expression into Eq. 6, we arrive at the following equation:

$$\hat{F}_e^{mag} = \frac{(e+1)\|v_R - v_H\|_2}{\frac{1}{m_R} + \frac{1}{m_H}} \quad (7)$$

where \hat{F}_e^{mag} is the *effective impact force* magnitude [16].

Now we define *impact potential* as the maximum impact force that a robot can create in a collision with a human [16]. It is a scalar value that provides an upper limit for any impact between the robot and human. According to Eq. (11) in

Heinzmann and Zelinsky [16], impact potential is computed as the maximum effective impact force among all points of collision on the surfaces of the robot and human. Given our point-mass assumption, there is only one possible point of collision. Hence, our impact potential is equivalent to \hat{F}_e^{mag} .

APPENDIX B

HUMAN MODEL LEARNING VIA GAUSSIAN PROCESS

A. *Multi-output GP*

As discussed in Sec. IV-A, we equivalently represent g using a single-output function $g': \mathcal{P}_H \times \mathcal{P}_R \times \mathcal{J} \rightarrow \mathbb{R}$, where $\mathcal{J} = [1 \dots n_H]$ [3]. Then we use $\mathcal{GP}(m, k)$ to represent the system’s belief about g' . The function $m: \mathbb{R}^{n_H} \times \mathbb{R}^{n_R} \times \mathbb{R} \rightarrow \mathbb{R}$ denotes the prior mean function. Without loss of generality, we assume that $m \equiv 0$. The function $k: \mathbb{R}^{n_H+n_R+1} \times \mathbb{R}^{n_H+n_R+1} \rightarrow \mathbb{R}$ denotes the covariance (or kernel) function.

We consider k as a composite kernel function. As mentioned in Sec. IV-A, we assume that the function outputs, g_1, \dots, g_{n_H} are independent of each other. Hence, for every g_j along the dimension $j \in \mathcal{J}$, we design one independent kernel function, denoted by $k_j: \mathbb{R}^{n_H+n_R} \times \mathbb{R}^{n_H+n_R} \rightarrow \mathbb{R}$. Then, we construct the composite kernel k as an additive combination of these independent kernels. Formally, we have:

$$\begin{aligned} &k((p_H, p_R, j), (p'_H, p'_R, j')) \\ &= \mathbb{1}_{j=j'=1} k_1((p_H, p_R), (p'_H, p'_R)) + \mathbb{1}_{j=j'=2} k_2((p_H, p_R), (p'_H, p'_R)) \\ &\quad + \dots + \mathbb{1}_{j=j'=n_H} k_{n_H}((p_H, p_R), (p'_H, p'_R)) \end{aligned}$$

Here, $\mathbb{1}_{j=j'=\cdot}$ denotes the indicator function. For bounding the information capacity for composite kernels, please refer to Krause and Ong [24]. More details on the surrogate function g' are available in Berkenkamp et al. [3], Ghosh et al. [13].

B. *Confidence Intervals*

As introduced in Sec. IV-A, Lemma 1 states that the confidence intervals given by the GP prediction contain the unknown and deterministic function g , with a high probability.

The confidence intervals come from a frequentist analysis that assumes that there is an *a priori* fixed underlying function, g (or equivalently, g'), of which we observe noisy measurements [7]. The confidence intervals are not from a Bayesian analysis, which inherently models a belief over a random function [7]. In fact, g' could be chosen adversarially [7], as long as it has bounded norm in the RKHS according to Assumption 3.

Our system uses the confidence intervals for one-step human motion prediction, as formulated in Sec. IV-B1. For multi-step prediction, the system applies one-step prediction *iteratively at every time step*, as presented in Sec. IV-B2. However, the high probability safety guarantee in our work is not ensured *per time step*. Instead, Lemma 1 allows our safety guarantee to be *independent* of the duration of system operation, as noted after Def. 1 in the end of Sec. III-D. As presented in Sec. V-B, the high probability safety guarantee is provided by our MPC algorithm.

APPENDIX C
ROBUST HUMAN MOTION PREDICTION

A. Robust One-step Prediction

First, we denote a hyperrectangle by $\mathcal{R}(a \pm b) \subset \mathbb{R}^{n_H}$, where $a \in \mathbb{R}^{n_H}$ denotes the center and $b \in \mathbb{R}^{n_H}$ denotes the half side length. Formally, $\mathcal{R}(a \pm b) := [a_1 - b_1, a_1 + b_1] \times [a_2 - b_2, a_2 + b_2] \times \dots \times [a_{n_H} - b_{n_H}, a_{n_H} + b_{n_H}]$, where a_j and b_j denote the j th dimension of a and b , respectively. For notational shorthand, we use $R = a \pm b$. In this section, hyperrectangles are built to represent a set of possible human positions and velocities.

	Confidence region for p_H^{t+1}	Confidence region for v_H^{t+1}
PCR	$\mathcal{R}(R_p^{t+1}) \subset \mathbb{R}^{n_H}$ $R_p^{t+1} = \tilde{m}_p(p_H^t, p_R^t)$	$\mathcal{R}(R_v^{t+1}) \subset \mathbb{R}^{n_H}$ $R_v^{t+1} = \tilde{m}_v(p_H^t, p_R^t)$
ECR	$\mathcal{E}(E_p^{t+1}) \subset \mathbb{R}^{n_H}$ $E_p^{t+1} = \tilde{m}_p(E_p^t, p_R^t)$	$\mathcal{E}(E_v^{t+1}) \subset \mathbb{R}^{n_H}$ $E_v^{t+1} = \tilde{m}_v(E_p^t, p_R^t)$
Relation	$\mathcal{E}(\tilde{m}_p(E_p^t, p_R^t)) \supset \bigcup_{p_H^t \in \mathcal{E}(E_p^t)} \mathcal{R}(\tilde{m}_p(p_H^t, p_R^t))$	$\mathcal{E}(\tilde{m}_v(E_p^t, p_R^t)) \supset \bigcup_{p_H^t \in \mathcal{E}(E_p^t)} \mathcal{R}(\tilde{m}_v(p_H^t, p_R^t))$

TABLE I: Notations for one-step prediction for the human position p_H^{t+1} and velocity v_H^{t+1} , where $p_H^{t+1} = p_H^t + g(p_H^t, p_R^t)$ (Eq. 1) and $v_H^{t+1} = g(p_H^t, p_R^t)/h$ (Eq. 2). The point-input confidence regions (PCRs), $\mathcal{R}(R_p^{t+1})$ and $\mathcal{R}(R_v^{t+1})$, represent the hyperrectangles bounding p_H^{t+1} and v_H^{t+1} , respectively. The PCRs are computed via the functions, \tilde{m}_p and \tilde{m}_v , respectively, given the points, (p_H^t, p_R^t) . The ellipsoid-input confidence regions (ECRs), $\mathcal{E}(E_p^{t+1})$ and $\mathcal{E}(E_v^{t+1})$, represent the ellipsoids bounding p_H^{t+1} and v_H^{t+1} , respectively. The ECRs are computed via the functions, \tilde{m}_p and \tilde{m}_v , respectively, given (E_p^t, p_R^t) . Each ECR outer-approximates the union of all the corresponding PCRs.

As described in Sec. IV-B, at every time step t , based on $p_H^{t+1} = f_H(p_H^t, p_R^t)$ (Eq. 1), we formulate one-step prediction for p_H^{t+1} as a function $E_p^{t+1} = \tilde{m}_p(E_p^t, p_R^t)$. Similarly, based on $v_H^{t+1} = g(p_H^t, p_R^t)/h$ (Eq. 2), we formulate one-step prediction for v_H^{t+1} as a function $E_v^{t+1} = \tilde{m}_v(E_p^t, p_R^t)$. Here, $\mathcal{E}(E_p^{t+1}) \subset \mathbb{R}^{n_H}$ and $\mathcal{E}(E_v^{t+1}) \subset \mathbb{R}^{n_H}$ are the ellipsoidal outer approximations for p_H^{t+1} and v_H^{t+1} , respectively. Since both the inputs to \tilde{m}_p and to \tilde{m}_v contain E_p^t which describes an ellipsoid, we refer to both $\mathcal{E}(E_p^{t+1})$ and $\mathcal{E}(E_v^{t+1})$ as the *ellipsoid-input confidence regions* (ECRs). The notations of ECRs are listed in Table I.

By contrast, we define a *point-input confidence region* (PCR) as a hyperrectangular outer approximation that is computed based on points, rather than ellipsoids. As listed in Table I, let $\mathcal{R}(R_p^{t+1})$ and $\mathcal{R}(R_v^{t+1})$ denote the PCRs for p_H^{t+1} and v_H^{t+1} , respectively. The functions needed to compute these PCRs are denoted by \tilde{m}_p and \tilde{m}_v , respectively. Both the inputs to \tilde{m}_p and to \tilde{m}_v involve only the points (p_H^t, p_R^t) .

In App. C-A1, we derive the PCRs for p_H^{t+1} and v_H^{t+1} , given the points (p_H^t, p_R^t) . Then in App. C-A2, we derive the ECRs for p_H^{t+1} and v_H^{t+1} , given (E_p^t, p_R^t) .

1) *Point-input confidence region:* We denote g 's inputs by $z^t = (p_H^t, p_R^t)$. As introduced in Sec. IV-A, given n measurements, let $\mu_n = [\mu_{n,1}, \dots, \mu_{n,n_H}]^T$ denote the stacked

GP predictive mean function. Let $\sigma_n = [\sigma_{n,1}, \dots, \sigma_{n,n_H}]^T$ denote the stacked predictive variance function.

By linearizing μ_n around a point \bar{z} , we get $P_{\mu_n}^{\bar{z}}(z^t) = \mu_n(\bar{z}) + J_{\mu_n}^{\bar{z}}(z^t - \bar{z})$, where $J_{\mu_n}^{\bar{z}} = [A_{\mu_n}^{\bar{z}}, B_{\mu_n}^{\bar{z}}]$ is the Jacobian of μ_n at \bar{z} . Then, we bound the error between the linearized function $P_{\mu_n}^{\bar{z}}$ and the true function g . Following Sec. V.A in Koller et al. [23], we apply Lagrangian remainder theorem and Lemma 1. Then, we obtain the following bound:

$$|P_{\mu_n}^{\bar{z}}(z^t) - g(z^t)| \leq \epsilon(z^t)$$

$$\epsilon(z^t) = \frac{1}{2} \|z^t - \bar{z}\|_2^2 L_{\nabla \mu_n} + \beta_n(\sigma_n(\bar{z}) + \|z^t - \bar{z}\|_2 L_{\sigma_n})$$

This equation is similar to the summation of Eq. 31 and 32 in Koller et al. [23]. Here, $L_{\nabla \mu_n} \in \mathbb{R}^{n_H}$ and $L_{\sigma_n} \in \mathbb{R}^{n_H}$ denote the Lipschitz constants of $\nabla \mu_n$ and σ_n , respectively. Since $\nabla \mu_n$ and σ_n are known functions, we can use tools from the global optimization literature to identify $L_{\nabla \mu_n}$ and L_{σ_n} [23]. Then we obtain a hyperrectangular confidence region around the true function g at a given point z^t . Formally:

$$g(z^t) \in \mathcal{R}(P_{\mu_n}^{\bar{z}}(z^t) \pm \epsilon(z^t)) \quad (8)$$

Since Lemma 1 has been used in the derivation, Eq. 8 holds uniformly with a high probability for all $n \in \mathbb{N}$, $z^t \in \mathcal{P}_H \times \mathcal{P}_R$.

Thus, the PCR for p_H^{t+1} is the hyperrectangle, $\mathcal{R}(\tilde{m}_p(z^t))$, where $\tilde{m}_p(z^t) = p_H^t + P_{\mu_n}^{\bar{z}}(z^t) \pm \epsilon(z^t)$. Similarly, the PCR for v_H^{t+1} is the hyperrectangle, $\mathcal{R}(\tilde{m}_v(z^t))$, where $\tilde{m}_v(z^t) = [P_{\mu_n}^{\bar{z}}(z^t) \pm \epsilon(z^t)]/h$.

2) *Ellipsoid-input confidence region:* We derive the ECRs for p_H^{t+1} and v_H^{t+1} given (E_p^t, p_R^t) . Intuitively, we derive the ECRs by outer-approximating the union of the PCRs for all $(p_H^t, p_R^t) \in \mathcal{E}(E_p^t) \times \{p_R^t\}$.

First, we derive $\mathcal{E}(E_p^{t+1})$, the ECR for p_H^{t+1} , given the input (E_p^t, p_R^t) , where $E_p^t = (c_p^t, Q_p^t)$. We have already derived that the PCR for p_H^{t+1} is $\mathcal{R}(p_H^t + P_{\mu_n}^{\bar{z}}(z^t) \pm \epsilon(z^t))$, which can be decomposed to $p_H^t + P_{\mu_n}^{\bar{z}}(z^t) + \mathcal{R}(0 \pm \epsilon(z^t))$. Accordingly, we choose $\bar{z} = (c_p^t, p_R^t)$ and derive the ECR in three steps: (1) We construct an ellipsoid to outer-approximate this set:

$$\bigcup_{z^t \in \mathcal{E}(E_p^t) \times \{p_R^t\}} \{p_H^t + P_{\mu_n}^{\bar{z}}(z^t)\}$$

By leveraging the linearity of $p_H^t + P_{\mu_n}^{\bar{z}}(z^t)$, we obtain that the ellipsoid is $\mathcal{E}(c_p^t + \mu_n(\bar{z}), (A_{\mu_n}^{\bar{z}} + I_{n_H})Q_p^t(A_{\mu_n}^{\bar{z}} + I_{n_H})^T)$. Here I_{n_H} denotes the identity matrix of size n_H . (2) We construct another ellipsoid to outer-approximate this following set:

$$\bigcup_{z^t \in \mathcal{E}(E_p^t) \times \{p_R^t\}} \mathcal{R}(0 \pm \epsilon(z^t))$$

(3) We sum both ellipsoids via Minkowski sum and obtain the ECR. For details in these steps, please refer to Sec. V.A in Koller et al. [23]. The ECR $\mathcal{E}(E_p^{t+1})$ is an ellipsoidal outer approximation for p_H^{t+1} given (E_p^t, p_R^t) . As shown in Table I, we denote these three steps by the function $E_p^{t+1} = \tilde{m}_p(E_p^t, p_R^t)$.

Second, we derive $\mathcal{E}(E_v^{t+1})$, the ECR for v_H^{t+1} , given the input (E_p^t, p_R^t) . We have already derived that the PCR for v_H^{t+1} is $\mathcal{R}([P_{\mu_n}^{\bar{z}}(z^t) \pm \epsilon(z^t)]/h)$, which can be decomposed

to $P_{\mu_n}^{\bar{z}}(z^t)/h + \mathcal{R}(0 \pm \epsilon(z^t)/h)$. Similar to the derivation presented in the last paragraph, we choose $\bar{z} = (c_p^t, p_R^t)$ and construct an ellipsoid to outer-approximate the set of all the values $P_{\mu_n}^{\bar{z}}(z^t)/h$. By leveraging the linearity, we obtain this ellipsoid as $\mathcal{E}(\mu_n(\bar{z})/h, (A_{\mu_n}^{\bar{z}}/h)Q_p^t(A_{\mu_n}^{\bar{z}}/h)^T)$. Then we construct another ellipsoid to outer-approximate the union of all the hyperrectangles $\mathcal{R}(0 \pm \epsilon(z^t)/h)$. Finally, we sum both ellipsoids via Minkowski sum and obtain the ECR. The ECR $\mathcal{E}(E_v^{t+1})$ is an ellipsoidal outer approximation for v_H^{t+1} given (E_p^t, p_R^t) . As shown in Table I, we denote these three steps by the function $E_v^{t+1} = \tilde{m}_v(E_p^t, p_R^t)$.

In summary, we have derived the PCRs, $\mathcal{R}(\tilde{m}_p(p_H^t, p_R^t))$ and $\mathcal{R}(\tilde{m}_v(p_H^t, p_R^t))$, to bound p_H^{t+1} and v_H^{t+1} , respectively, given (p_H^t, p_R^t) . We have derived the ECRs, $\mathcal{E}(\tilde{m}_p(E_p^t, p_R^t))$ and $\mathcal{E}(\tilde{m}_v(E_p^t, p_R^t))$, to bound p_H^{t+1} and v_H^{t+1} , respectively, given (E_p^t, p_R^t) . Our derivation implies that each ECR bounds the union of all the corresponding PCRs, as shown in Table I.

By applying Lemma 1, we obtain the following lemma:

Lemma 3 (Similar to Koller et al. [22] Lemma 3). *Let $\delta \in (0, 1]$. For all $n \in \mathbb{N}$, choose β_n according to Lemma 1. Then, with a probability greater than $1 - \delta$, we have that:*

$$\begin{aligned} \forall p_{H,n} \in \mathcal{E}(E_{p,n}): p_{H,n} + g(p_{H,n}, p_{R,n}) &\in \mathcal{E}(\tilde{m}_p(E_{p,n}, p_{R,n})) \\ \forall p_{H,n} \in \mathcal{E}(E_{p,n}): g(p_{H,n}, p_{R,n})/h &\in \mathcal{E}(\tilde{m}_v(E_{p,n}, p_{R,n})) \end{aligned}$$

uniformly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}) \subset \mathcal{P}_H$, $p_{R,n} \in \mathcal{P}_R$.

Proof: Define the following functions:

$$\begin{aligned} m_p(p_H, p_R) &= p_H + \mu_n(p_H, p_R) \pm \beta_n \sigma_n(p_H, p_R) \\ m_v(p_H, p_R) &= \mu_n(p_H, p_R)/h \pm \beta_n \sigma_n(p_H, p_R)/h \end{aligned}$$

By applying Lemma 1, we obtain that the following holds jointly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}) \subset \mathcal{P}_H$, $p_{R,n} \in \mathcal{P}_R$, with a high probability:

$$\begin{aligned} \forall p_{H,n} \in \mathcal{E}(E_{p,n}): p_{H,n} + g(p_{H,n}, p_{R,n}) &\in \mathcal{R}(m_p(p_{H,n}, p_{R,n})) \\ \forall p_{H,n} \in \mathcal{E}(E_{p,n}): g(p_{H,n}, p_{R,n})/h &\in \mathcal{R}(m_v(p_{H,n}, p_{R,n})) \end{aligned}$$

Eq. 8 holds uniformly with a high probability for all $n \in \mathbb{N}$, $z^t \in \mathcal{P}_H \times \mathcal{P}_R$. Thus, for all $n \in \mathbb{N}$, $p_{H,n} \in \mathcal{E}(E_{p,n}) \subset \mathcal{P}_H$, $p_{R,n} \in \mathcal{P}_R$, we can use Eq. 8 to construct PCRs, as described in this section. It can be shown that the PCRs, $\mathcal{R}(\tilde{m}_p(p_{H,n}, p_{R,n}))$ and $\mathcal{R}(\tilde{m}_v(p_{H,n}, p_{R,n}))$, outer-approximate $\mathcal{R}(m_p(p_{H,n}, p_{R,n}))$ and $\mathcal{R}(m_v(p_{H,n}, p_{R,n}))$, respectively. Then, we construct ECRs to outer-approximate the union of PCRs, as described in this section. We conclude that the following holds jointly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}) \subset \mathcal{P}_H$, $p_{R,n} \in \mathcal{P}_R$, with a high probability:

$$\begin{aligned} \bigcup_{p_{H,n} \in \mathcal{E}(E_{p,n})} \{p_{H,n} + g(p_{H,n}, p_{R,n})\} &\subset \bigcup_{p_{H,n} \in \mathcal{E}(E_{p,n})} \mathcal{R}(m_p(p_{H,n}, p_{R,n})) \\ &\subset \bigcup_{p_{H,n} \in \mathcal{E}(E_{p,n})} \mathcal{R}(\tilde{m}_p(p_{H,n}, p_{R,n})) \subset \mathcal{E}(\tilde{m}_p(E_{p,n}, p_{R,n})) \\ \bigcup_{p_{H,n} \in \mathcal{E}(E_{p,n})} \{g(p_{H,n}, p_{R,n})/h\} &\subset \bigcup_{p_{H,n} \in \mathcal{E}(E_{p,n})} \mathcal{R}(m_v(p_{H,n}, p_{R,n})) \\ &\subset \bigcup_{p_{H,n} \in \mathcal{E}(E_{p,n})} \mathcal{R}(\tilde{m}_v(p_{H,n}, p_{R,n})) \subset \mathcal{E}(\tilde{m}_v(E_{p,n}, p_{R,n})) \end{aligned}$$

Intuitively, this lemma states that, with a high probability, the human position and velocity at the next time step are contained within the ellipsoids, $\mathcal{E}(\tilde{m}_p(E_{p,n}, p_{R,n}))$ and $\mathcal{E}(\tilde{m}_v(E_{p,n}, p_{R,n}))$, respectively, given that the current human position is known to be inside an ellipsoid, $\mathcal{E}(E_{p,n})$. This holds true only if the ellipsoid, $\mathcal{E}(E_{p,n})$, is inside the workspace \mathcal{P}_H and the current robot position, $p_{R,n}$, is inside \mathcal{P}_R .

B. Robust Multi-step prediction

As described in Sec. IV-B, multi-step prediction is a function with the input, $(E_p^0, \{p_R^t\}_{t=0}^{T-1})$, and the output, $(\{E_p^t\}_{t=1}^T, \{E_v^t\}_{t=1}^T)$. The system iteratively applies $E_p^{t+1} = \tilde{m}_p(E_p^t, p_R^t)$ and $E_v^{t+1} = \tilde{m}_v(E_p^t, p_R^t)$ (Eq. 3) from $t = 0$ to $T - 1$ to roll out the sequences of ellipsoids.

By applying Lemma 3, we obtain the following corollary:

Corollary 1. [Similar to Koller et al. [23] Corollary 7] *Let $\delta \in (0, 1]$. For all $n \in \mathbb{N}$, choose β_n based on Lemma 1. Then, the following holds jointly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}^0) \subset \mathcal{P}_H$, $p_{R,n}^0 \in \mathcal{E}(E_{p,n}^0)$, $\{p_{R,n}^\tau\}_{\tau=0}^{T-1} \subset \mathcal{P}_R$, $t \in [1 \dots T]$, with a probability at least $1 - \delta$:*

$$p_{H,n}^t \in \mathcal{E}(E_{p,n}^t), v_{H,n}^t \in \mathcal{E}(E_{v,n}^t)$$

where for all $n \in \mathbb{N}$, $\{E_{p,n}^\tau\}_{\tau=1}^T$ and $\{E_{v,n}^\tau\}_{\tau=1}^T$ are computed via multi-step prediction, such that for all $\tau \in [1 \dots T]$, $\mathcal{E}(E_{p,n}^\tau) \subset \mathcal{P}_H$.

Proof: Since we have $p_{H,n}^0 \in \mathcal{E}(E_{p,n}^0) \subset \mathcal{P}_H$ and $p_{R,n}^0 \in \mathcal{P}_R$, we can apply Lemma 3 and obtain that the following holds jointly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}^0) \subset \mathcal{P}_H$, $p_{H,n}^0 \in \mathcal{E}(E_{p,n}^0)$, $p_{R,n}^0 \in \mathcal{P}_R$, with a high probability:

$$\begin{aligned} p_{H,n}^1 &= p_{H,n}^0 + g(p_{H,n}^0, p_{R,n}^0) \in \mathcal{E}(\tilde{m}_p(E_{p,n}^0, p_{R,n}^0)) = \mathcal{E}(E_{p,n}^1) \\ v_{H,n}^1 &= g(p_{H,n}^0, p_{R,n}^0)/h \in \mathcal{E}(\tilde{m}_v(E_{p,n}^0, p_{R,n}^0)) = \mathcal{E}(E_{v,n}^1) \end{aligned}$$

Given that $\mathcal{E}(E_{p,n}^1) \subset \mathcal{P}_H$ and $p_{R,n}^1 \in \mathcal{P}_R$, we can again apply Lemma 3. We continue applying Lemma 3 till we obtain that the following holds jointly for all $n \in \mathbb{N}$, $\mathcal{E}(E_{p,n}^{T-1}) \subset \mathcal{P}_H$, $p_{H,n}^{T-1} \in \mathcal{E}(E_{p,n}^{T-1})$, $p_{R,n}^{T-1} \in \mathcal{P}_R$, with a high probability: $p_{H,n}^T \in \mathcal{E}(E_{p,n}^T)$ and $v_{H,n}^T \in \mathcal{E}(E_{v,n}^T)$. ■

As described in App. B-B, the confidence intervals in Lemma 1 come from a frequentist analysis rather than a Bayesian analysis. Hence, in the proof above, even though we iteratively apply Lemma 3 for multiple time steps, the result of joint inclusions holds with a probability that is still at least $1 - \delta$. This fact reaffirms that our high probability safety guarantee is not ensured *per time step*, as noted after Def. 1 in the end of Sec. III-D.

APPENDIX D

SAFETY WITH ELLIPSOIDAL UNCERTAINTY

A. Safe Impact with Ellipsoidal Uncertainty

As formulated in Sec. III-A, our system can ensure safe impact (SI) by constraining the robot velocity, v_R , based on the constraint, $\Omega(v_R, v_H) \leq \Omega_{max}$, given the human velocity, v_H . In this section, we formulate a set of constraints that can ensure SI, given an ellipsoidal outer approximation of the human velocity, denoted by $\mathcal{E}(E_v)$. This constraint enables our system to ensure SI for all $v_H \in \mathcal{E}(E_v)$.

$$\forall v_H \in \mathcal{E}(E_v): \|v_R - v_H\|_2 \leq \rho \quad (9)$$

$$\forall v_H \in \mathcal{E}(E_v), \forall j \in [1 \dots n_H]: |v_{R,j} - v_{H,j}| \leq \frac{\rho}{\sqrt{n_H}} \quad (10)$$

$$\forall v_H \in \mathcal{E}(E_v): L v_H \leq L v_R + \left[\frac{\rho}{\sqrt{n_H}} \right]_{2n_H \times 1} = l \quad (11)$$

$$\forall i \in [1 \dots 2n_H]: [L]_{i, \cdot} c_v + \sqrt{[L]_{i, \cdot} Q_v [L]_{i, \cdot}^T} \leq [l]_i \quad (12)$$

As formulated in Sec. III-A, the SI constraint, given v_H , is $\Omega(v_R, v_H) = (e+1)\|v_R - v_H\|_2 / (\frac{1}{m_R} + \frac{1}{m_H}) \leq \Omega_{max}$. Extending this constraint to all $v_H \in \mathcal{E}(E_v)$ yields the constraint in Eq. 9 with the constant $\rho = \Omega_{max}(\frac{1}{m_R} + \frac{1}{m_H}) / (e+1)$.

We conservatively relax Eq. 9 to Eq. 10, where $v_{R,j}$ and $v_{H,j}$ denote the j th dimension of v_R and v_H , respectively. The feasible region of $\|v_R - v_H\|_2 \leq \rho$, given v_H , is a hypersphere. The feasible region of $\forall j \in [1 \dots n_H]: |v_{R,j} - v_{H,j}| \leq \rho/\sqrt{n_H}$, given v_H , is the maximum-volume hyper-rectangle inscribed in this hypersphere. Thus, satisfying the constraint in Eq. 10 is a sufficient but not necessary condition for satisfying the constraint in Eq. 9.

Then, we equivalently rewrite Eq. 10 as Eq. 11, which is a polytopic constraint. Here, $L = [-I_{n_H}, I_{n_H}]^T \in \mathbb{R}^{2n_H \times n_H}$ and I_{n_H} denotes the identity matrix of size n_H . Let $[\rho/\sqrt{n_H}]_{2n_H \times 1}$ denote the vector $[\rho/\sqrt{n_H}, \rho/\sqrt{n_H}, \dots, \rho/\sqrt{n_H}]^T \in \mathbb{R}^{2n_H}$. We also denote the right hand side of Eq. 11 by $l \in \mathbb{R}^{2n_H}$.

Following Eq. 41 in Koller et al. [23], we can enforce the Polytopic constraint in Eq. 11 analytically to the ellipsoid $\mathcal{E}(E_v)$ where $E_v = (c_v, Q_v)$. Formally, Eq. 11 is equivalently reformulated as $2n_H$ individual 1-dimensional restraints, as shown in Eq. 12. Here, $[l]_i$ denotes the i th dimension of the vector l and $[L]_{i, \cdot}$ denotes the i th row of the matrix L .

For notational shorthand, we rewrite the set of constraints, presented in Eq. 12 as $\forall i \in [1 \dots 2n_H]: \mathcal{C}_{SI,i}(v_R, E_v) \leq 0$. Here, $\mathcal{C}_{SI,i}(v_R, E_v) = [L]_{i, \cdot} c_v + \sqrt{[L]_{i, \cdot} Q_v [L]_{i, \cdot}^T} - [l]_i \in \mathbb{R}$.

B. Safety as CA or SI with Ellipsoidal Uncertainty

Lemma 2. Given hyper-parameters $\theta_1 \geq 0$, $\theta_2 \geq 0$, and the ellipsoids described by E_p and E_v , then the following holds

$$\begin{aligned} & \bigcap_{i \in [1 \dots 2n_H]} \left\{ (p_R, v_R): \mathcal{C}_{SI,i}(v_R, E_v) \leq \max(\theta_1 \mathcal{C}_{CA}(p_R, E_p), \theta_2 \mathcal{C}_{CA}(p_R, E_p)) \right\} \\ & \subset \bigcap_{p_H \in \mathcal{E}(E_p)} \bigcap_{v_H \in \mathcal{E}(E_v)} [\mathcal{S}_{CA}(p_H) \cup \mathcal{S}_{SI}(v_H)] \end{aligned}$$

Proof: Given that $\theta_1 \geq 0$ and $\theta_2 \geq 0$, we obtain that:

$$\begin{aligned} & \bigcap_{i \in [1 \dots 2n_H]} \left\{ (p_R, v_R): \mathcal{C}_{SI,i}(v_R, E_v) \leq \max(\theta_1 \mathcal{C}_{CA}(p_R, E_p), \theta_2 \mathcal{C}_{CA}(p_R, E_p)) \right\} \\ & \subset \{ (p_R, v_R): \mathcal{C}_{CA}(p_R, E_p) > 0 \} \\ & \cup \left[\bigcap_{i \in [1 \dots 2n_H]} \{ (p_R, v_R): \mathcal{C}_{SI,i}(v_R, E_v) \leq 0 \} \right] \quad (13) \end{aligned}$$

According to the CA constraint formulated in Sec. V-A, we have $\{ (p_R, v_R): \mathcal{C}_{CA}(p_R, E_p) > 0 \} = \bigcap_{p_H \in \mathcal{E}(E_p)} \mathcal{S}_{CA}(p_H)$. Here, $\mathcal{S}_{CA}(p_H)$ denotes the CA set defined in Sec. III-B.

In App. D-A, we have conservatively relaxed the original SI constraint, as shown in Eq. 9, to the set of constraints $\forall i \in [1 \dots 2n_H]: \mathcal{C}_{SI,i}(v_R, E_v) \leq 0$. Hence, we have $\bigcap_{i \in [1 \dots 2n_H]} \{ (p_R, v_R): \mathcal{C}_{SI,i}(v_R, E_v) \leq 0 \} \subset \bigcap_{v_H \in \mathcal{E}(E_v)} \mathcal{S}_{SI}(v_H)$. Here, $\mathcal{S}_{SI}(v_H)$ denotes the SI set that is defined in Sec. III-B.

Now we plug the set relations in the previous two paragraphs into Eq. 13 and obtain the followings:

$$\begin{aligned} & \{ (p_R, v_R): \mathcal{C}_{CA}(p_R, E_p) > 0 \} \\ & \cup \left[\bigcap_{i \in [1 \dots 2n_H]} \{ (p_R, v_R): \mathcal{C}_{SI,i}(v_R, E_v) \leq 0 \} \right] \\ & \subset \left[\bigcap_{p_H \in \mathcal{E}(E_p)} \mathcal{S}_{CA}(p_H) \right] \cup \left[\bigcap_{v_H \in \mathcal{E}(E_v)} \mathcal{S}_{SI}(v_H) \right] \end{aligned}$$

■

A. Safe Trajectory Optimization

The solutions to the trajectory optimization (TO) problem, formulated in Sec. V-B, guarantee safety within the horizon T , with a high probability. This is formally stated as follows:

Lemma 4. *Let $\delta \in (0, 1]$. For all $n \in \mathbb{N}$, choose β_n based on Lemma 1. Then the following holds jointly for all $n \in \mathbb{N}$, $(p_{H,n}^0, v_{H,n}^0, p_{R,n}^0, v_{R,n}^0) \in \mathcal{S}$, $t \in [1 \dots T]$, with a probability at least $1 - \delta$:*

$$(p_{H,n}^t, v_{H,n}^t, p_{R,n}^t, v_{R,n}^t) \in \mathcal{S}$$

under the closed-loop system corresponding with a feasible solution found by the TO, denoted by $\{u_n^\tau\}_{\tau=1}^T$, for all $n \in \mathbb{N}$.

Proof: As assumed in Sec. II, the robot dynamics f_R is known and deterministic. We have encoded f_R in our TO's constraint (e), as presented in Sec. V-B. Thus, the actual robot motion is the same as the desired trajectory found by the TO. Thus, for all $n \in \mathbb{N}$, let $\{(p_{R,n}^\tau, v_{R,n}^\tau, E_{p,n}^\tau, E_{v,n}^\tau)\}_{\tau=1}^T$ denote the desired (and actual) robot trajectory and the human ellipsoids as part of the feasible solution found by the TO.

Our TO's constraint (b) (Sec. V-B) ensures that $\mathcal{E}(E_{p,n}^t)$ and $p_{R,n}^t$ are inside the workspace \mathcal{P}_H and \mathcal{P}_R , respectively, for all $n \in \mathbb{N}$, $t \in [1 \dots T]$. Hence, we can apply Corollary 1 and obtain that: with a high probability, we have $p_{H,n}^t \in \mathcal{E}(E_{p,n}^t)$ and $v_{H,n}^t \in \mathcal{E}(E_{v,n}^t)$, uniformly for all $n \in \mathbb{N}$, $(p_{H,n}^0, v_{H,n}^0, p_{R,n}^0, v_{R,n}^0) \in \mathcal{S}$, $t \in [1 \dots T]$.

Given our TO's constraint (c) (Sec. V-B), by applying Lemma 2, we obtain that for all $t \in [1 \dots T]$: $(p_{R,n}^t, v_{R,n}^t) \in \bigcap_{p_H \in \mathcal{E}(E_{p,n}^t)} \bigcap_{v_H \in \mathcal{E}(E_{v,n}^t)} [\mathcal{S}_{CA}(p_H) \cup \mathcal{S}_{SI}(v_H)]$. In the previous paragraph, we have shown that $p_{H,n}^t$ and $v_{H,n}^t$ are indeed always inside the ellipsoids, jointly with a high probability. Thus, with a high probability, we have $(p_{R,n}^t, v_{R,n}^t) \in \mathcal{S}_{CA}(p_{H,n}^t) \cup \mathcal{S}_{SI}(v_{H,n}^t)$ and $p_{H,n}^t \in \mathcal{E}(E_{p,n}^t) \subset \mathcal{P}_H$, uniformly for all $n \in \mathbb{N}$, $(p_{H,n}^0, v_{H,n}^0, p_{R,n}^0, v_{R,n}^0) \in \mathcal{S}$, $t \in [1 \dots T]$. ■

B. Safe Model Predictive Controller

In Sec. IV-A, we assume that our measurements are corrupted by Gaussian noise. Let \hat{f}_H denote the noisy measurement of $f_H(p_H, p_R)$ (defined in Eq. 1) given (p_H, p_R) . Thus, for each $j \in \mathcal{J}$, we have $[\hat{f}_H]_j = [f_H(p_H, p_R)]_j + w_j = [p_H]_j + [\hat{g}]_j$ where w_j and \hat{g} are defined in Sec. IV-A. Alg. 1 presents our MPC algorithm that is introduced in Sec. V-B.

In Alg. 1, we implicitly assume that we have access to the true human positions, p_H , during planning, while only have \hat{f}_H , the noisy observations of the human positions, for updating the GP. This is a standard assumption when the robot is acting in a Markov decision process (MDPs). We can extend our theory to noisy observations during planning. For more details, please refer to Remark 9 in Koller et al. [23].

Algorithm 1: MPC (similar to Alg. 1 in Koller et al. [23])

Input: π_{rec} , Gaussian process (GP) model (μ_0, Σ_0)

- 1 $\Pi_{-1} \leftarrow \{u_{-1}^1 = \pi_{rec}, \dots, u_{-1}^T = \pi_{rec}\};$
- 2 **for** $t = 0, 1, \dots$ **do**
- 3 feasible, $\Pi \leftarrow$ solve TO (Sec. V-B);
- 4 **if** feasible **then** $\Pi_t \leftarrow \Pi$;
- 5 **else** $\Pi_t \leftarrow \{u_{t-1}^2, \dots, u_{t-1}^T, \pi_{rec}\};$
 // u_t^1 = the first control in Π_t .
- 6 $p_R^{t+1}, v_R^{t+1}, p_H^{t+1}, \hat{f}_H^{t+1} \leftarrow$ execute u_t^1 and observe;
 // Compute v_H^{t+1} based on Eq. 2.
- 7 $v_H^{t+1} \leftarrow (p_H^{t+1} - p_H^t)/h;$
- 8 $(\mu_{t+1}, \Sigma_{t+1}) \leftarrow$ update GP with $(p_H^t, p_R^t, \hat{f}_H^{t+1});$
- 9 **end**

Theorem 1 *Let $\delta \in (0, 1]$. For all $n \in \mathbb{N}$, choose β_n based on Lemma 1. The system has access to the initial state $(p_H^0, v_H^0, p_R^0, v_R^0) \in \mathcal{S}_{rec}$. Then, the human-robot system is δ -safe under the MPC algorithm.*

Proof: The proof is similar to the proof for Theorem 8 in [23]. Our goal is to prove that for all $t \in \mathbb{N}$, the system has a controller that can guarantee δ -safe by induction on t .

a) *Base case:* At time $t = 0$, if our TO finds a feasible solution, Π_0 , then executing Π_0 can guarantee with a high probability that the system state remains inside \mathcal{S} for all $t \in [1 \dots T]$, based on Lemma 4. Our TO's constraint (d), as formulated in Sec. V-B, combined with Lemma 4, implies that the system state at time T is inside \mathcal{S}_{rec} , with a high probability. According to Assumption 2, by switching to π_{rec} at time T , the system can ensure that the system state will remain inside \mathcal{S} from the time T , with a high probability. Thus, the system has a δ -safe controller.

At time $t = 0$, if no feasible solutions are found, the system will keep executing π_{rec} . According to Assumption 2, the system state will always remain inside \mathcal{S} . Thus, the system has a δ -safe controller.

b) *Inductive step:* Assume that for a particular time $t \in \mathbb{N}$, the system has a δ -safe controller, Π_t .

At time $t + 1$, if our TO finds a feasible solution, Π_{t+1} , the system will first deploy Π_{t+1} and switch to π_{rec} at time $t + T$. Based on Lemma 4, our TO's constraint (d) in Sec. V-B, and Assumption 2, the system has a δ -safe controller.

At time $t + 1$, if no feasible solutions are found, then Π_t will lead the system to safely travel to a state inside \mathcal{S}_{rec} , from which the system can switch to π_{rec} . Thus, the system has a δ -safe controller. ■

	Env1			Env2			Env3			Env4			Env5		
	H-Indep-R	H-To-R	H-Away-R	H-Indep-R	H-To-R	H-Away-R	H-Indep-R	H-To-R	H-Away-R	H-Indep-R	H-To-R	H-Away-R	H-Indep-R	H-To-R	H-Away-R
CASI ($\Omega_{max} = 0.6$) #Itr	6.133	6.100	6.233	6.367	6.167	6.600	6.633	6.300	6.533	6.267	6.100	6.833	6.333	6.100	7.067
CASI ($\Omega_{max} = 0.3$) #Itr	8.733	7.467	10.133	9.600	8.267	8.900	7.933	7.033	10.333	10.133	7.533	10.167	9.400	7.700	12.100
CA #Itr	11.533	11.333	12.733	14.967	15.500	12.733	9.833	8.967	12.000	13.067	10.633	12.967	13.267	10.667	14.767
CASI ($\Omega_{max} = 0.6$) #SafeColl	0.767	1.267	0.500	0.467	0.600	0.600	0.333	0.600	0.333	1.300	1.133	0.833	0.367	0.867	0.400
CASI ($\Omega_{max} = 0.3$) #SafeColl	0.633	0.833	0.567	0.533	0.767	0.433	0.167	0.633	0.333	1.133	1.133	0.967	0.533	0.933	0.367
CA #SafeColl	1.033	1.000	0.633	0.333	0.667	0.267	0.600	0.500	0.400	1.300	1.200	0.733	0.933	1.000	0.433
CASI ($\Omega_{max} = 0.6$) PlanTime (s)	0.299	0.469	0.307	0.335	0.464	0.332	0.355	0.731	0.369	0.298	0.494	0.337	0.312	0.590	0.397
CASI ($\Omega_{max} = 0.3$) PlanTime (s)	0.432	0.636	0.510	0.571	0.847	0.614	0.412	0.718	0.494	0.454	0.661	0.496	0.567	0.662	0.522
CA PlanTime (s)	0.145	0.283	0.132	0.146	0.258	0.141	0.135	0.312	0.127	0.147	0.315	0.153	0.154	0.232	0.167

TABLE II: The means of #Itr, #SafeColl, and PlanTime in the 2D goal-reaching domain. The benchmark included running CASI ($\Omega_{max} = 0.6$), CASI ($\Omega_{max} = 0.3$), and CA, with 15 different simulated human behaviors (5 environments \times 3 objective functions), and 30 trials for each condition. The algorithm that achieved the lowest value among the three algorithms for each measurement is highlighted in bold.

APPENDIX F ADDITIONAL RESULTS

The results of #SafeColl is presented in Table II. We conducted a Wilcoxon signed-rank test for each pair among the three algorithms. No significant pairwise differences were found regarding #SafeColl. The p -value from the Wilcoxon signed-rank test for the pair, CASI ($\Omega_{max} = 0.6$) and CA, was 0.583. The p -value for CASI ($\Omega_{max} = 0.3$) and CA was 0.202. The p -value for CASI ($\Omega_{max} = 0.6$) and CASI ($\Omega_{max} = 0.3$) was 0.552.

CASI allows collisions to occur, as long as the impacts are safe, while CA does not allow collisions at all. However, the insignificant result here implies that CASI does not always result in higher numbers of safe collisions than CA.