Safe and Efficient Exploration of Human Models During Human-Robot Interaction

Ravi Pandya and Changliu Liu

***Abstract*— Many collaborative human-robot tasks require the robot to stay safe and work efficiently around humans. Since the robot can only stay safe with respect to its own model of the human, we want the robot to learn a good model of the human in order to act both safely and efficiently. This paper studies methods that enable a robot to safely explore the space of a human-robot system to improve the robot's model of the human, which will consequently allow the robot to access a larger state space and better work with the human. In particular, we introduce active exploration under the framework of energy-function based safe control, investigate the effect of different active exploration strategies, and finally analyze the effect of safe active exploration on both analytical and neural network human models.**

I. Introduction

In order to improve the effectiveness of robots as collaborators for humans, they need to move both safely and efficiently in shared spaces with humans such as in autonomous driving, collaborative manufacturing or social navigation. Since robots should never harm people, safe control is a fundamental system requirement for human-robot interaction and it appears as a hard constraint in real-time control. Recent work has handled this constraint through a safety monitor or safety controller that keeps the robot safe with respect to the human's dynamics [1], [2]. Human behavior is often noisy and unpredictable, so the robot should also keep a probabilistic model of the human. Additionally, the robot gets observations of the human online and can use this data to adapt the human model [3], [2].

Much work assumes a "human-in-isolation" model [4], [5], meaning that they do not consider the effect that the robot has on the human. In this work, we make use of the fact that the human will respond to the robot's actions by having the robot actively explore in order to learn a better model of the human. This exploration can be thought of as optimizing for long-term efficiency and is treated as soft constraint rather than a hard constraint like safety for the robot.

In the evaluation, we consider the effects of active exploration in safe control under two kinds of uncertainty: *intrinsic* uncertainty which corresponds to uncertainty in how the human moves in certain areas of the environment and *interactive* uncertainty which corresponds to uncertainty in how the human reacts to the robot. We additionally consider a data-driven neural network dynamics model where these two sources of uncertainty may be coupled.

The main contributions of this work are to build on top of safe control by 1) introducing active exploration under an energy-function-based safe control framework 2) investigating different strategies for safe exploration and evaluating their efficiency in information collection and 3) investigating the effects of safe exploration on both analytical and data-driven models. Our results suggest that some safe exploration strategies can improve the robot's model of the human and expand the set of safe reachable states, ultimately meaning the robot can improve both the safety and adaptation efficiency of the overall system.

Authors are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, USA {rapandya, cliu6}@andrew.cmu.edu

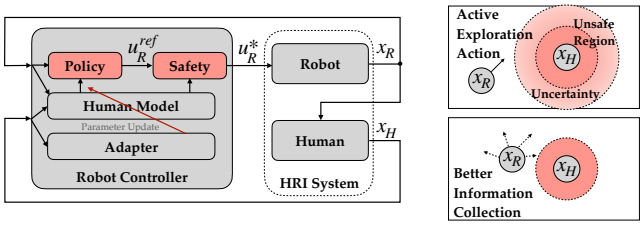

Fig. 1: **Left:** The human-robot system considered in this work, the robot adapts its human model online and is equipped with a safety controller to (probabilistically) guarantee safety. **Right:** Shows that active exploration can result in reduced uncertainty, hence the robot can access same states with higher safety probability (improved safety) and better collect information around human (improved efficiency).

II. Related Work

Safe control in human-robot interaction: Since humans and robots will start to work in physical proximity, much work has gone into measuring safety of human-robot systems from collaborative manufacturing to healthcare [6], [7]. Others have focused on applying energy-function-based methods [1], [8], [9] and reachability-based methods [2], [10] to mathematically guarantee safety of a robot around humans. In this work, we adopt energy-function-based safe control.

Safe control under uncertainty: Since human behavior is often noisy and hard to predict exactly, we care about keeping the human-robot system safe even under uncertainty. The general problem of decision making under uncertainty is often modeled as a Partially Observable Markov Decision Process (POMDP). However, they are computationally intractable to solve explicitly [11]. There is some work which utilizes the exact POMDP solution for decision-making around humans [12], but such approaches do not scale to environments larger than a handful of discrete states and actions. While it is also possible to add hard constraints [13] or chance constraints [14], these problems are harder than standard POMDPs.

Another body of work incorporates uncertainty into the safety monitor, such as by using Bayesian inference over the human's intention to inform the safety monitor which states the human is likely to reach [2], [4]. Other work considers

staying safe with respect to a Gaussian noise model over the human's state [3] and nonparametric models like Gaussian Processes [15], [16]. The main drawback of most methods for dealing with uncertainty in safe control is the conservatism of the resulting policy for the robot. In this work, we aim to reduce this conservatism through active exploration to improve the robot's estimated model of the human.

Information gathering and adaptation: Some existing work considers a robot taking information gathering actions over a human's internal state to improve the robot's model of the human [17], [18]. In particular, [17] considers a model of the human as making its best response to the robot's policy, which is similar to the human model considered in this work, though we encapsulate the human's intentions inside their dynamics.

Another way to reduce the conservatism of the robot is to adapt the robot's model of the human online. Online adaptation has long been used for system identification [19], [20] and has more recently been applied to adapting models of humans [21], [22]. In this work, we use active information gathering to improve the adaptation process by changing the data input to the adaptation algorithm.

Neural network dynamics models: There has recently been lots of work that utilizes neural networks to represent dynamics models [23], [24]. This can be particularly useful in cases where the system can be hard to model explicitly, like human behavior. The authors in [25] also consider uncertainty in the learned dynamics to attain better sample efficiency. Additionally, [26] considers adapting neural network dynamics models online through meta-learning. While there has been work on safe active learning for adaptation of neural network dynamics [27], to the best of our knowledge, we present the first investigation of safe exploration for neural network dynamics models in human-robot interaction.

III. Safe Exploration in HRI

A. The HRI System

We consider the case of a single robot interacting with a single human. The human has dynamics $\pi_\theta(\cdot)$ that are parameterized by the vector θ and depends on the environment state $x(k)$:

$$x(k) = (x_H(k), x_R(k), x_G(k), x_G^R(k)) \tag{1}$$

$$x_H(k+1) = \pi_\theta((x(k)), \tag{2}$$

where the environment state is defined by the human's state $x_H(k)$, the robot's state $x_R(k)$, the human's goal $x_G(k)$ and the robot's goal $x_G^R(k)$. The robot's dynamics are assumed to be control-affine (such that $f_R(\cdot)$ and $g_R(\cdot)$ are potentially nonlinear functions of $x_R(k)$); and it is equipped with a control law $\pi_R(\cdot)$ and an adaptation law to estimate the parameters θ of the human's policy π_θ:

$$x_R(k+1) = f_R(x_R(k)) + g_R(x_R(k))u_R(k) \tag{3}$$

$$u_R(k) = \pi_R(x_R(k), x_H(k), x_G^R(k), \hat{\theta}(k)) \tag{4}$$

$$\hat{\theta}(k+1) = \text{update}(\hat{\theta}(k), x_H(k), x_H(k-1), \ldots, x_R(k), x_R(k-1), \ldots) \tag{5}$$

where $\hat{\theta}(k)$ is the robot's estimate of θ at time k. Note that the robot does not necessarily have access to the form of π_θ, so θ and $\hat{\theta}(k)$ may not have the same semantic meaning.

B. Goal for Robot Control

The goal for robot control is to minimize the prediction error over the distribution of environment states that the human-robot system will reach when both agents are running goal-focused controllers (denoted as $\mathcal{G}$) while additionally minimizing the number of *unsafe states* the system will visit. Based on a predefined safety specification (like keeping the human and robot from colliding), the system has a set of safe states $\mathcal{Z}$, so the robot incurs a penalty if the human-robot system leaves this set. The full cost function for the robot is:

$$\pi_R^* = \underset{\pi}{\text{argmin}} \underset{x\sim\mathcal{G}}{\mathbb{E}} \left[||\pi_\theta(x) - \hat{\pi}_{\hat{\theta}^\pi}(x)|| + \beta_{x\notin\mathcal{Z}}\right] \tag{6}$$

where $\hat{\pi}_{\hat{\theta}^\pi}(\cdot)$ denotes the robot's estimated policy of the human (parameterized by $\hat{\theta}^\pi$) after the robot executes control law π and estimates $\hat{\theta}^\pi$ during the interaction and β is a constant coefficient. Intuitively, minimizing this objective means that the robot should both learn a good predictive model of the human and keep the system outside of unsafe states. The objective function does not require the human's dynamics model and the robot's estimated model to have the same form since it just considers the state prediction error.

C. Safety Assurance

To minimize (6), the robot could either trade off between the two objectives or could treat the safety violation penalty as a constraint while minimizing the prediction error. We consider the latter approach where the robot has a safety monitor that keeps the system safe.

We adopt the safe exploration algorithm [1], [3] for safety assurance. The method belongs to general energy-function-based safe control [9], which ensures forward invariance to a subset of a user-defined safe set. The safety specification considered in this paper is collision avoidance between the human and the robot. Hence the user-defined safe set $\mathcal{Z} = \{x \mid \phi_0 = d_{min} - d(x_R, x_H) \leq 0\}$, where d_{min} is the distance margin and d measures the minimum distance between the human and the robot. The key idea of safe control is to design a safety index (or energy function) $\phi : x \mapsto \mathbb{R}$ such that 1) a control law that satisfies the constraint $\dot{\phi} \leq -\eta_R$ when $\phi \geq 0$ ensures forward invariance to the set $\mathcal{X} := \{x \mid \phi \leq 0, \phi_0 \leq 0\}$; and 2) there always exists a feasible safe control for any state x, i.e., $U_S^R(x) = \{u_R : \dot{\phi} \leq -\eta_R \text{ when } \phi \geq 0\} \neq \emptyset$. The parameter $\eta_R \in \mathbb{R}^+$ is a safety margin. Once such a safety index is constructed, we just need to construct a safety monitor to project all reference control signals to $U_S^R(x)$. Then safety is guaranteed in the sense that if the system trajectory starts from the set $\mathcal{X}$ will always remain in that set.

This work uses a second-order control-affine robot model. Hence according to the design rule in [1], the safety index is designed to be

$$\phi = d_{min}^2 + \rho - d^2 - k_\phi \dot{d}, \tag{7}$$

where $\dot{d}$ is the time derivative of the relative distance, k_ϕ is a positive constant, and ρ is a positve margin. The derivative term $\dot{d}$ is added in order to ensure that the robot control u_R can always affect $\dot{\phi}$. Suppose the next human state follows a Gaussian distribution $x_H(k+1) \sim \mathcal{N}(\hat{x}_H(k+1 \mid k), \Sigma_H(k+1))$ (to be derived in Sec. IV), then

$$\dot{\phi}(x(k)) \sim \mathcal{N}\Big(\frac{\partial \phi}{\partial x_R}\dot{x}_R + \frac{\partial \phi}{\partial x_R}\hat{\dot{x}}_H, \frac{1}{t_s}\left[\left(\frac{\partial \phi}{\partial x_H}\right)\Sigma_H(k+1)\left(\frac{\partial \phi}{\partial x_H}\right)^T\right]^{\frac{1}{2}}\Big), \quad (8)$$

where t_s is the sampling time, $\dot{x}_R = {}^{(x_R(k+1)-x_R(k))}/_{t_s}$, and $\hat{\dot{x}}_H := {}^{(\hat{x}_H(k+1|k)-x_H(k))}/_{t_s}$. Since the distribution of $\dot{\phi}$ is unbounded, we can only enforce a probabilistic constraint on $\dot{\phi}$ instead of a hard constraint. This paper aims to ensure safety for human behaviors within the 3σ bound. In this way, the 3σ-robust set of safe control $U_S^R(x) = \{u_R : \mathcal{P}(\dot{\phi} \leq -\eta_R) \geq 99.7\% \text{ when } \phi \geq 0\}$ at time k can be computed as described in [3]:

$$U_S^R = \{u_R(k) : L(k)u_R(k) \leq S(k)\} \quad (9)$$

$$L(k) = \frac{\partial \phi}{\partial x_R} g_R \quad (10)$$

$$S(k) = \begin{cases} -\eta_R - \lambda_R^{SEA}(k) - \frac{\partial \phi}{\partial x_H}\hat{\dot{x}}_H - \frac{\partial \phi}{\partial x_R} f_R & \text{if } \phi \geq 0 \\ \infty & \text{if } \phi < 0 \end{cases} \quad (11)$$

where

$$\lambda_R^{SEA}(k) = \frac{3}{t_s}\left[\left(\frac{\partial \phi}{\partial x_H}\right)\Sigma_H(k+1)\left(\frac{\partial \phi}{\partial x_H}\right)^T\right]^{\frac{1}{2}} + \lambda_0. \quad (12)$$

Here, λ_0 is a tunable constant that bounds other uncertainties. In practice, in order to compensate discrete-time implementation error in safe control, we choose the margin ρ as $\eta_R t_s + \lambda_R^{SEA}(k) t_s$, which grows with the uncertainty of $\dot{\phi}$ (that is affected by the uncertainty of human behaviors).

Finally, our safety controller can be written as the solution to a quadratic program that maps some reference control $u_R^{ref}(k)$ (e.g., actions for exploration) to the set of safe controls U_R^S as the control in the safe set that is closest to the reference:

$$u_R^* = \underset{u_R \in U_R^S}{\text{argmin}} \frac{1}{2}(u_R - u_R^{ref})^T(u_R - u_R^{ref}). \quad (13)$$

When there is no uncertainty on human behavior, i.e., $\Sigma_H \equiv 0$, the safe control enforces forward invariance to $\mathcal{X}$. If there is uncertainty and the uncertainty model is correct, i.e., Σ_H matches the statistical error covariance in the human behavior prediction, then the safe control enforces forward invariance with at least probability 99.7%.

D. Active Exploration

The robot's policy will affect (6) by changing the data input to the adaptation, which will in turn change its estimate of the human's policy $\hat{\pi}_{\hat{\theta}^\pi}(\cdot)$. The human's policy could be estimated passively by just receiving information during the interaction, or actively by choosing actions (u_R^{ref}) to reduce the future error. Active exploration may minimize the immediate or long-term error, which fall on the spectrum of different risk preferences, where "risk" refers to the chance of incorrectly predicting the human's state:

Risk-Neutral: the robot does not explicitly reason about uncertainty and adapts its model passively.

Risk-Seeking: the robot tries to actively explore unknown states with high uncertainty, with the hope to reduce uncertainty in the subsequent rounds.

Risk-Averse: the robot actively keeps the system in known states with low uncertainty.

E. Types of Uncertainties

We're interested in understanding the effects of these risk preferences on the adaptation process while the robot acts with its safety monitor to understand *when safe active exploration can improve the robot's model of the human*. We investigate the effect of active exploration when the robot's human model has different kinds of uncertainty:

Intrinsic Uncertainty: The robot is estimating the portion of the human's dynamics that *does not* depend on the robot's state, so is uncertain of how the human moves in the environment.

Interactive Uncertainty: The robot is estimating the portion of the human's dynamics that *depends on* the robot's state, so is uncertain of how the human reacts to the robot.

Full Uncertainty: The robot estimates both the human's intrinsic and interactive dynamics, so is uncertain about how the human moves in the environment and how it reacts to the robot.

IV. Online Adaptation of Human Models

In the standard formulation [28], the robot will estimate a discrete-time parameter-affine system as the human's dynamics model:

$$x_H(k+1) = \Phi(k)\theta(k) + w_H(k), \quad (14)$$

where the matrix $\Phi(k)$ is some observation of the environment (e.g. nonlinear features of the human's state and control), $\theta(k)$ is the parameter to be estimated and $w_H(k)$ is assumed to be zero-mean Gaussian noise with covariance W. In the following discussion, we introduce a linear parameterization of the human model and a recursive least square parameter adaptation (RLS-PA) algorithm to identify unknown parameters. Notation for the RLS-PA is shown in Table I.

A. Time-Varying Linear Model

We follow prior work by regarding the model of the human as a time-varying linear system with Gaussian uncertainty [3], meaning the robot estimates time-varying parameters A_H and B_H:

$$x_H(k+1) = A_H(k)x_H(k) + B_H(k)u_H(k). \quad (15)$$

We design a feature vector u_H in place of the human's true control vector, since the robot does not have access to it. u_H

	State Estimate	Estimation Error	State Covariance
a priori	$\hat{x}_H(k\|k)$	$\tilde{x}_H(k\|k)$	
a posteriori	$\hat{x}_H(k+1\|k)$	$\tilde{x}_H(k+1\|k)$	$\Sigma_H(k+1)$

TABLE I: Notation for State Estimation

could generally include arbitrary features, but we assume it's a function of the human's state $x_H(k)$, the robot's state $x_R(k)$ and the human's goal $x_G(k)$:

$$u_H(k) = g_H(x_H(k), x_R(k), x_G(k)). \tag{16}$$

At time $k+1$, the robot uses its previous estimate of the state and dynamics to get the *a priori* state estimate:

$$\hat{x}_H(k+1|k) = \hat{A}_H(k)\hat{x}_H(k|k) + \hat{B}_H(k)u_H(k). \tag{17}$$

The state estimation error is defined as

$$\tilde{x}_H(k+1|k) = x_H(k+1) - \hat{x}_H(k+1|k). \tag{18}$$

For simplicity, we assume that the robot can access the ground truth human state, hence the *a posteriori* state estimate has no error, i.e., $\hat{x}_H(k|k) = x_H(k)$ and $\tilde{x}_H(k|k) = 0$.

B. State and Parameter Estimation in the Belief Space

The robot keeps a Gaussian uncertainty model on its estimate of the human

$$\hat{x}_H(k+1) \sim \mathcal{N}(\hat{x}_H(k+1|k), \Sigma_H(k+1)), \tag{19}$$

where $\hat{x}_H(k+1|k)$ is the *a priori* estimate of the human's next state and $\Sigma_H(k+1)$ is the state covariance.

To put the human model into a parameter-affine form (14), we define the matrix $C = \begin{bmatrix} A_H(k) & B_H(k) \end{bmatrix}$, and then split it into its rows to consider the flattened $\theta = \begin{bmatrix} C_1 & \dots & C_{n_h} \end{bmatrix}^T$ where n_h is the dimension of the human state and C_i denotes the ith row of C. The robot's estimate of θ is $\hat{\theta}(k)$. Next, the observation matrix $\Phi(k)$ is

$$\Phi(k) = \begin{bmatrix} \varphi^T(k) & 0 & \dots & 0 \\ 0 & \varphi^T(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varphi^T(k) \end{bmatrix} \tag{20}$$

where $\varphi(k)$ is the observation vector

$$\varphi(k) = \begin{bmatrix} \hat{x}_H(k|k) \\ u_H(k) \end{bmatrix}. \tag{21}$$

This allows us to write our *a priori* state estimate simply as

$$\hat{x}_H(k+1|k) = \Phi(k)\hat{\theta}(k) \tag{22}$$

The parameter estimation error is $\tilde{\theta}(k) = \theta(k) - \hat{\theta}(k)$. This lets us express the state estimation error as

$$\tilde{x}_H(k+1|k) = \Phi(k)\tilde{\theta}(k) + w_H(k) \tag{23}$$

Now we consider the state covariance, or mean squared estimation error $\Sigma_H(k+1) = \mathbb{E}[\tilde{x}_H(k+1|k)\tilde{x}_H(k+1|k)^T]$. Using (23), we can rewrite this as

$$\Sigma_H(k+1) = \Phi(k)\Sigma_{\tilde{\theta}\tilde{\theta}}(k)\Phi^T(k) + W \tag{24}$$

where W is the (known) measurement noise covariance and $\Sigma_{\tilde{\theta}\tilde{\theta}}(k) = \mathbb{E}[\tilde{\theta}(k)\tilde{\theta}(k)^T]$ is the covariance of the model error.

Now, we'll consider how the robot adapts its model of the human in the belief space:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\Phi^T(k)\tilde{x}_H(k+1|k) \tag{25}$$

where $F(k+1)$ is the learning gain which is updated as:

$$\begin{aligned} F(k+1) = \frac{1}{\lambda}[F(k) - \\ F(k)\Phi^T(k)(\lambda I + \Phi(k)F(k)\Phi^T(k))^{-1}\Phi(k)F(k)]. \end{aligned} \tag{26}$$

The parameter estimation error is

$$\tilde{\theta}(k+1) = \tilde{\theta}(k) - F(k+1)\Phi^T(k)\tilde{x}_H(k+1|k) + \Delta\theta \tag{27}$$

where $\Delta\theta = \theta(k+1) - \theta(k)$. The true value of $\tilde{\theta}(k)$ is unknown, but we can calculate its expectation as

$$\mathbb{E}[\tilde{\theta}(k+1)] = [I - F(k+1)\Phi(k)\Phi^T(k)]\mathbb{E}[\tilde{\theta}(k)] + d\theta \tag{28}$$

where $d\theta$ is set to an average time varying rate, since the true $\Delta\theta$ is also unknown.

Finally, this lets us write an explicit update for our model parameter covariance as:

$$\begin{aligned} \Sigma_{\tilde{\theta}\tilde{\theta}}(k+1) = {} & F(k+1)\Phi(k)\Sigma_H(k+1)\Phi(k)F(k+1) \\ & - \Sigma_{\tilde{\theta}\tilde{\theta}}(k)\Phi^T(k)\Phi(k)F(k+1) \\ & - F(k+1)\Phi^T(k)\Phi(k)\Sigma_{\tilde{\theta}\tilde{\theta}}(k) \\ & + \mathbb{E}[\tilde{\theta}(k+1)]d\theta^T + d\theta\mathbb{E}[\tilde{\theta}(k+1)]^T \\ & - d\theta d\theta^T + \Sigma_{\tilde{\theta}\tilde{\theta}}(k) \end{aligned} \tag{29}$$

V. Exploration Strategies

We now substantiate the exploration strategies that we introduced earlier using the estimates from the online adaptation algorithm. The robot's future uncertainty can be measured by looking at the norm of the parameter uncertainty matrix $||\Sigma_{\tilde{\theta}\tilde{\theta}}(k+1)||$. However, since $\Sigma_{\tilde{\theta}\tilde{\theta}}(k+1)$ does not depend on the robot's action at time k, we need to optimize for the uncertainty two steps in the future: $\Sigma_{\tilde{\theta}\tilde{\theta}}(k+2)$. Note that the baseline risk-neutral controller is the same controller used in [3].

Risk-Neutral: Risk-neutral behavior will not account for uncertainty and just move the robot towards its own goal with state feedback control gain K (hand-tuned):

$$u_R^{ref}(k) = -K(x_R(k) - x_G^R(k)). \tag{30}$$

Risk-Seeking: Risk-seeking behavior will try to maximize the objective function $J(\cdot)$, which is the norm of the future covariance matrix:

$$u_R^{ref}(k) = \underset{u_R}{\text{argmax}}\ J(u_R) = ||\Sigma_{\tilde{\theta}\tilde{\theta}}(k+2)||. \tag{31}$$

Risk-Averse: Risk-averse behavior will try to minimize $J(\cdot)$, the norm of the future covariance matrix:

$$u_R^{ref}(k) = \underset{u_R}{\text{argmin}}\ J(u_R) = ||\Sigma_{\tilde{\theta}\tilde{\theta}}(k+2)||. \tag{32}$$

The types of uncertainties in θ that need to be explored are summarized below:

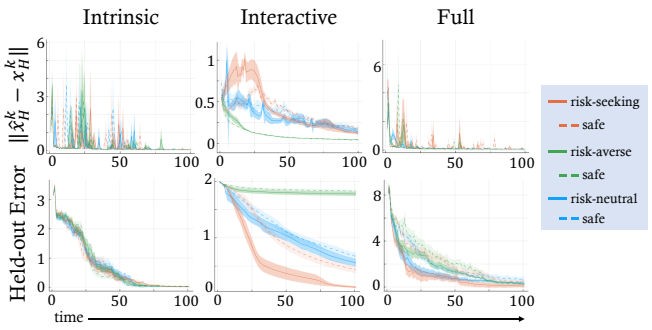

Fig. 2: Each column corresponds to simulations with different kinds of uncertainty (Sec. V). Each row corresponds to a different metric (Sec. VI-C). Each curve is averaged over 10 different initial conditions and the shaded areas show the standard error.

	risk-seeking	risk-averse	risk-neutral
intrinsic uncertainty	4.6 ± 0.2	4.6 ± 0.3	5.2 ± 1.7
interactive uncertainty	19.7 ± 2.3	0.9 ± 0.09	6.3 ± 2.0
full uncertainty	13.2 ± 3.5	4.5 ± 0.7	7.0 ± 1.4
neural network dynamics	7.2 ± 1.3	4.6 ± 2.5	4.2 ± 1.8

TABLE II: Number of Safety Interventions (mean $\pm$ SD)

Intrinsic Uncertainty: The robot estimates only A_H.
Interactive Uncertainty: The robot estimates only B_H.
Full Uncertainty: The robot estimates both A_H and B_H.

VI. Analytical Human Model Experiments

A. Analytical Human Model

The robot's state $x_R(k)$ and the human's state $x_H(k)$ consist of each agent's own position and velocity. The goal state for the robot $x_G^R(k)$ and the goal state for the human $x_G(k)$ are defined as points in the state space with 0 velocity. We hand-design features for the simulated human's control $u_H(k)$ (16). Since the human's objective in the task is to move towards their goal while avoiding collisions with the robot, we consider a potential field model for the human's control—the human is attracted to the goal and repelled from the robot:

$$u_H(k) = -K_1\left(x_H(k) - x_G(k)\right) + \frac{\gamma}{d^2} K_2\left(x_H(k) - x_R(k)\right), \tag{33}$$

where K_1 and K_2 are constant gains and d is the distance between the two agents. The second component of the control vector captures the idea that the robot's influence on the human will decrease as the distance between the two increases and this influence is controlled by the parameter γ. In the evaluation (unless otherwise noted), $\gamma = 30$.

B. Hypotheses

H1: The risk-seeking safe exploration strategy will improve the quality of the robot's estimated model of the human.
H2: The presence of the safety controller will decrease the efficiency of adaptation.
H3: Active exploration will be most beneficial when the robot has interactive uncertainty in the human's dynamics.

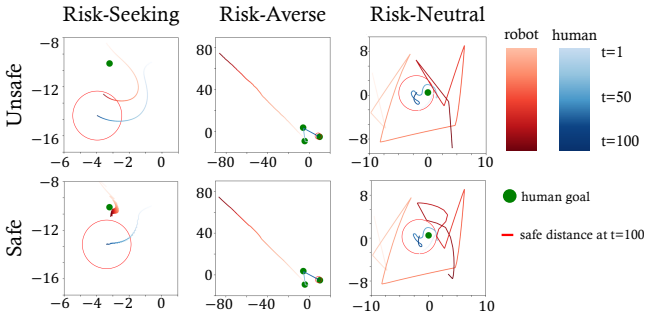

Fig. 3: One example trajectory for each risk preference when the robot has interactive uncertainty (Sec. VI-E). **Top row:** without safety controller **Bottom row:** with safety controller.

C. Evaluation

In our simulated navigation environment, the robot needs to adapt its model of the human because its initial guesses for A_H and B_H are incorrect. We care about three metrics to measure the quality of the adapted model and the effect of the safety controller:
Runtime Error: Ultimately, we want the robot to end up with a better model of the human based on the actions it selects, so we measure this with the 1-step prediction accuracy of the model on the current trajectory.
Held-out Error: To understand if the robot's estimated model generalizes, we measure the first term of the objective function (6) by computing the average 1-step state prediction error of the estimated model on rollouts of a set of different initial conditions (without further adaptation) where the robot is only taking goal-oriented actions.
Safety Interventions: When the robot is getting close to violating the minimum safe distance constraint, the safety controller will activate to keep the robot safe, so we can measure how often this happens to understand how exploration is affected by the presence of the the safety controller.

D. Intrinsic Uncertainty

Each column in Fig. 2 shows results for a different kind of uncertainty and each row shows a different metric. Each curve on a single plot shows the value of the metric over a 100-timestep interaction between the human and robot, averaged over 10 initial conditions (randomly selected starting locations and goals in the xy-plane for both agents). Safety interventions are shown in Table II.

When the robot has only intrinsic uncertainty, the performance of all risk preferences with and without safety are basically the same, shown in the left column of Fig. 2 (the spikes in prediction error occur when the human's goal changes). This result gives evidence against **H1** and **H2**, since neither active exploration nor the safety controller had any effect on the robot's adaptation process. However, this makes sense because the robot's model of the human is not a function of the robot's state, so the robot is able to estimate the human's intrinsic dynamics well regardless of the robot's trajectory. This result also supports **H3**, since neither risk-seeking nor risk-averse behavior improved the robot's adapted model of the human when the robot has intrinsic uncertainty.

The top row of Table II shows that the number of interventions from the safety controller is similar for all three risk preferences, telling us that the safety controller affected the different risk preferences similarly.

E. Interactive Uncertainty

When the robot has only interactive uncertainty, we see significant differences between the different risk preferences, shown in the middle column of Fig. 2. The risk-averse controller converges to low runtime error quickly while the other risk preferences eventually also reach similarly low values. This happens because the risk-averse controller moves away from the human (Fig. 3), so it will barely affect the human's trajectory, but since it already knows A_H, it can easily predict the future states of the human.

We did not see this pattern when the robot had only intrinsic uncertainty, so this result also supports **H3**. The risk-seeking controller results in much lower held-out error than the baseline risk-neutral controller, which supports **H1** and **H3**. The risk-averse controller, however, results in worse held-out error even though its prediction error is low. We can again understand why by looking at the middle column of Fig. 3—the risk-averse controller runs away from the human to keep its uncertainty low, so it does not get to learn the effect it has on the human and thus its estimated model does not generalize well.

Comparing safe exploration (dashed curves) to unsafe exploration (solid curves), we see that the safety controller does slightly increase the held-out model error for all risk preferences, supporting **H2**, though this difference is most pronounced for risk-seeking controller. We can understand why by looking at Fig. 3, which shows one example interaction for each risk preference with and without the safety controller. The safety controller changes the risk-seeking controller's trajectory drastically, but not the other two risk preferences' trajectories. This is a direct result of the risk-seeking controller trying to stay close to the human, so the safety controller will need to be active more often.

We quantitatively see this in the second row of Table II—the safety controller was activated an average of 19.7 times during a 100-timestep interaction for the risk-seeking controller, while it was activated less than once on average for the risk-averse controller and 6.3 times for risk-neutral. This confirms that the risk-seeking controller was more affected by the presence of the safety controller than the other risk preferences were, showing that **H2** being true depends on both the kind of uncertainty and risk preference considered.

F. Full Uncertainty

In this case, the robot's dynamics estimate has both intrinsic and interactive uncertainty. The results in these simulations (right column of Fig. 2) show the same trends for held-out error as in the interactive uncertainty case, which makes sense because this model also includes interactive uncertainty, so this again supports **H3**. We also see that the risk-seeking controller results in the lowest held-out model error, supporting **H1**.

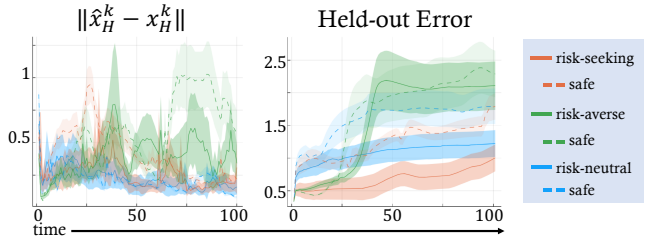

Fig. 4: Each column corresponds to a different metric (Sec. VI-C) in the case when the robot is estimating a neural network to represent the human's dynamics. Each curve is averaged over 10 different initial conditions and the shaded areas represent the standard error of the mean.

Introducing the safety controller makes the held-out error higher for all three risk preferences, which lines up with **H2** just as in the interactive uncertainty case. We again see the same pattern that the risk-seeking controller's model gets affected the most, which can be explained by considering the number of safety interventions (third row of Table II)—the risk-seeking controller has the most number of interventions, so its trajectory is affected most by the safety controller.

VII. Neural Network Human Model Experiments

Human behavior is diverse and may not always be captured well by an analytical model. To capture the diversity of human behavior, data-driven neural network models are often used. In the following discussion, we test the same risk preferences and safe controller while the robot keeps a neural network estimate of the human's dynamics.

A. Dataset and Architecture

We created a dataset collected from the human model in (33). The dataset $\mathcal{D}$ consists of trajectories of length $T = 100$ of the human's state, the human's goal and robot's state $\tau = (x_H^0, x_R^0, x_G^0, ..., x_H^T, x_R^T, x_G^T)$. We then split these trajectories into segments with a short history of length N to use as training inputs $(x_H^{k-N}, x_R^{k-N}, x_G^{k-N}, ..., x_H^k, x_R^k, x_G^k)$ and their corresponding labels x_H^{k+1}.

We train a 4-layer feedforward ReLU neural network to learn the human's dynamics function. The network is trained by minimizing an MSE loss with the Adam optimizer.

B. Last Layer Adaptation

Since the robot gets to train a good model offline in this case, we make adaptation necessary by generating the training data with $\gamma = 50$ but the human's true value is $\gamma = 30$ online.

To adapt the model online, we can adapt the last layer's weights (a common paradigm for fine-tuning neural network models [29]). Using the flattened post-activation output of the second to last layer of the network as $\varphi(k)$ and the weights of the last layer to be $\hat{\theta}(k)$, we can keep the same form of the dynamics as in (14). We adapt $\hat{\theta}(k)$ online and keep track of our parameter uncertainty $\Sigma_{\tilde{\theta}\tilde{\theta}}(k)$ as before.

C. Results

Unlike in the linear system case, we can't easily separate the robot's uncertainty into intrinsic and interactive uncertainty, so the robot has full uncertainty by being uncertain about the parameters in the last layer of the network.

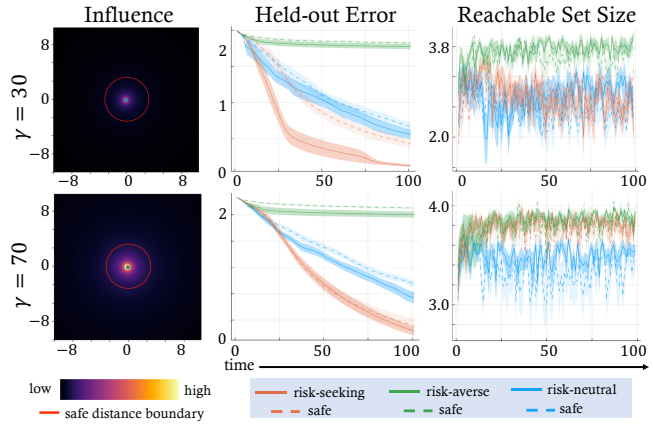

Fig. 5: Shows the held-out error and reachable set size (Sec. VIII-B) for different levels of influence (the color represents how far the human would move at the next timestep if the robot was at that point in space) when the robot has interactive uncertainty.

Looking at Fig. 4, the prediction error does not show any strong patterns in with either the risk preferences or the safety controller, besides that the the risk-averse controller has slightly higher error (this is likely due to the robot's state going far outside the range of values in the training data for the neural network when it runs away from the human).

In the held-out error, we see a similar pattern as in the linear system case emerge with the risk preferences and the safety controller. The risk-seeking controller results in the best learned model while the risk-averse controller has the worst, which supports **H1**. The held-out error does slightly increase over time, which is not unexpected since the goal of adaptation is to overfit to temporally local data, so it may "forget" its previous training. This effect doesn't appear in the linear system case because improving local prediction accuracy will also improve global accuracy.

For the risk-seeking and risk-averse controllers, the presence of the safety controller reduces the quality of the learned model, which again supports **H2**. The decrease in model quality is again higher for the risk-seeking controller than the others, which is likely a result of the same phenomenon observed in the linear system. We can again see evidence for this in the bottom row of Table II where the safety controller was activated most often for the risk-seeking controller.

VIII. Effect of Influence

A. Effect on Held-out Error

We saw previously that under interactive uncertainty, the safety controller significantly changes the reference control from the risk-seeking controller, meaning the robot has to stay farther away from the human. This results in the robot's estimate of the human's dynamics being worse than it would be without the safety controller active since observing these close-proximity states is likely ideal for adaptation—this is exactly the underlying tension between optimizing for safety and optimizing for efficiency.

To test this idea, we vary γ in (33) when the robot has interactive uncertainty and a linear model, since this is where we find the most stark differences between controllers. Each row of Fig. 5 visualizes a different value of γ. The left column visualizes the influence that the robot has on the human where the human is in the center, the red circle shows the minimum safe distance, and the color represents how far the human would move at the next timestep if the robot was at that point in space. When $\gamma = 30$, the influence is close to 0 when the robot is outside the safe distance boundary, whereas the robot still has a notable influence on the human outside this bubble when $\gamma = 70$.

When $\gamma = 70$, the safety controller *does not* significantly affect the held-out error under any risk preference. This is in contrast to the $\gamma = 30$ case where introducing the safety controller significantly reduced the quality of the estimated model for the risk-seeking controller. This means **H1** is still true with safe control and that **H2** is not necessarily true, it depends on how much influence the robot has on the human. This result tells us that if the human is sufficiently affected by the robot's state outside of the safe bubble, the robot can actually make use of active exploration to improve the robot's model of the human while staying safe.

B. Effect on Reachable Set of States

Finally, we want to understand how safe exploration can reduce the conservatism of the safety monitor, so we show a preliminary analysis here. The right column of Fig. 5 shows the average size of the 1-step reachable states under safe control from the current state (x_0), i.e, $|\{x \mid \exists u \in U_S^R, \text{ s.t., } x = f_R(x_0) + g_R(x_0)u\}|$. This set measures the conservatism of the constraint U_S^R in (9) under the current human model and uncertainty. Computation-wise, it is calculated by sampling controls uniformly between the control bounds, computing the resulting safe states using $\Sigma_H(k)$, then again sampling the state space to estimate the volume of the resulting safe reachable set of states. The curves show how the size of this set changes during the interaction.

When the influence is low, the risk-averse controller keeps the safe set the largest while the risk-seeking and risk-neutral controllers are indistinguishable. When the influence is high, however, both active exploration controllers (risk-seeking and risk-averse) keep the safe reachable set larger than risk-neutral behavior, but for different reasons. The risk-seeking controller learns a good predictive model of the human, so its reachable set will enlarge while it stays close to the human, while the risk-averse controller enlarges this set by staying far away from the human (Fig. 3).

IX. Conclusion

A. Summary

We have looked into the effect of introducing active exploration for adaptation in an energy-function-based safe control framework. We investigated the effects of different risk preferences (risk-seeking, risk-neutral and risk-averse) on different kinds of uncertainty (intrinsic and interactive) both on an analytical and neural network model of a human partner. The risk-seeking controller generally learns the best model of the human when there is interactive uncertainty present, though active exploration does not change the adapted model when there is only intrinsic uncertainty. Safe

exploration can improve the robot's model of the human and as a result reduce the conservatism of the safety monitor.

We have also seen that the benefit of using a risk-seeking controller can be smaller when a safety controller is active since the risk-seeking controller tries to stay too close to the human. However, if the human is sufficiently influenced by robot's position, this difference can disappear. Broadly, this is a good sign for future work involving physical humans and robots, since it means we can enable robots to explore safely without negatively impacting the adaptation process, assuming the human is sufficiently influenced by the robot.

B. Limitations and Future Work

While our work focuses on the effects of different safe exploration strategies in human-robot interaction, this work is in simulation without real humans in the loop. Future work will focus on speeding up the computation of the risk-seeking and risk-averse controllers to solve them in real-time around humans. Extending this work to physical situations like collaborative manufacturing is an exciting future direction since it would require the human and robot to stay in close proximity to each other while staying safe, so having a good model of the human is paramount to the team's success.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1745016 and DGE2140739 and additionally under Grant No. 2144489. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human-robot interactions," in *ASME Dynamic Systems and Control Conference*, vol. 3, 2014.

[2] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," in *Robotics: Science and Systems Conference (RSS)*, 2018.

[3] C. Liu and M. Tomizuka, "Safe exploration: Addressing various uncertainty levels in human robot interactions," in *IEEE American Control Conference (ACC)*, 2015, pp. 465–470.

[4] A. Bajcsy, S. Bansal, E. Ratner, C. J. Tomlin, and A. D. Dragan, "A robust control framework for human motion prediction," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 24–31, 2020.

[5] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3931–3936.

[6] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, "Design considerations for safe human-robot collaborative workplaces," *Procedia CIrP*, vol. 37, pp. 248–253, 2015.

[7] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 281–297, 2003.

[8] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[9] T. Wei and C. Liu, "Safe control algorithms using energy functions: A unified framework, benchmark, and new directions," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 238–243.

[10] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin, "A scalable framework for real-time multi-robot, multi-human collision avoidance," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 936–943.

[11] C. Papadimitriou and J. Tsitsiklis, "The complexity of markov decision processes," in *Mathematics of Operations Research*, vol. 12, 1987, pp. 441–450.

[12] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *International Conference on Human-Robot Interaction*, 2018.

[13] J. D. Isom, S. P. Meyn, and R. D. Braatz, "Piecewise linear dynamic programming for constrained pomdps." in *AAAI*, vol. 1, 2008, pp. 291–296.

[14] S. Thiébaux, B. Williams *et al.*, "Rao*: An algorithm for chance-constrained pomdp's," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[15] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for gaussian process regression with application to safe control," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] F. Berkenkamp and A. P. Schoellig, "Safe and robust learning control with gaussian processes," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 2496–2501.

[17] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[18] A. Bestick, R. Pandya, R. Bajcsy, and A. D. Dragan, "Learning human ergonomic preferences for handovers," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3257–3264.

[19] R. Isermann, "Parameter adaptive control algorithms—a tutorial," *Automatica*, vol. 18, no. 5, pp. 513–528, 1982.

[20] C. Rohrs, L. Valavani, M. Athans, and G. Stein, "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics," *IEEE Transactions on Automatic Control*, vol. 30, no. 9, pp. 881–889, 1985.

[21] R. Liu and C. Liu, "Human motion prediction using adaptable recurrent neural networks and inverse kinematics," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1651–1656, 2020.

[22] L. Wang, Y. Hu, L. Sun, W. Zhan, M. Tomizuka, and C. Liu, "Hierarchical adaptable and transferable networks (hatn) for driving behavior prediction," *Workshop on Machine Learning for Autonomous Driving at Neural Information Processing Systems*, 2021.

[23] A. Nagabandi, G. Yang, T. Asmar, R. Pandya, G. Kahn, S. Levine, and R. S. Fearing, "Learning image-conditioned dynamics models for control of underactuated legged millirobots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4606–4613.

[24] T. Wei and C. Liu, "Safe control with neural network dynamic models," in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, vol. 168. PMLR, 23–24 Jun 2022, pp. 739–750.

[25] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Conference on Neural Information Processing Systems*, 2018.

[26] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," *International Conference on Learning Representations*, 2019.

[27] T. Lew, A. Sharma, J. Harrison, A. Bylard, and M. Pavone, "Safe active dynamics learning and control: A sequential exploration-exploitation framework," *IEEE Transactions on Robotics*, 2022.

[28] L. Ljung *et al.*, "Theory for the user," *System Identification*, 1987.

[29] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Conference on Neural Information Processing Systems*, 2014.