Article

The International Journal of
Robotics Research
2020, Vol. 39(2-3) 250–265
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0278364919859436
journals.sagepub.com/home/ijr
SAGE

Confidence-aware motion prediction for real-time collision avoidance[1]

David Fridovich-Keil*, Andrea Bajcsy*, Jaime F Fisac,
Sylvia L Herbert, Steven Wang, Anca D Dragan and Claire J Tomlin

Abstract

One of the most difficult challenges in robot motion planning is to account for the behavior of other moving agents, such as humans. Commonly, practitioners employ predictive models to reason about where other agents are going to move. Though there has been much recent work in building predictive models, no model is ever perfect: an agent can always move unexpectedly, in a way that is not predicted or not assigned sufficient probability. In such cases, the robot may plan trajectories that appear safe but, in fact, lead to collision. Rather than trust a model's predictions blindly, we propose that the robot should use the model's current predictive accuracy to inform the degree of confidence in its future predictions. This model confidence inference allows us to generate probabilistic motion predictions that exploit modeled structure when the structure successfully explains human motion, and degrade gracefully whenever the human moves unexpectedly. We accomplish this by maintaining a Bayesian belief over a single parameter that governs the variance of our human motion model. We couple this prediction algorithm with a recently proposed robust motion planner and controller to guide the construction of robot trajectories that are, to a good approximation, collision-free with a high, user-specified probability. We provide extensive analysis of the combined approach and its overall safety properties by establishing a connection to reachability analysis, and conclude with a hardware demonstration in which a small quadcopter operates safely in the same space as a human pedestrian.

Keywords

Motion planning, human motion prediction, safety, robust control

1. Introduction

Motion planning serves a key role in robotics, enabling robots to automatically compute trajectories that achieve the specified objectives while avoiding unwanted collisions. In many situations of practical interest, such as autonomous driving and unmanned aerial vehicle (UAV) navigation, it is important that motion planning account not just for the current state of the environment, but also for its *predicted* future state. Often, certain objects in the environment may move in active, complex patterns that cannot be readily predicted using straightforward physics models; we shall refer to such complex moving objects as *agents*. Examples of agents considered in this paper include pedestrians and human-driven vehicles. Predicting the future state of these agents is generally a difficult problem. Some of the key challenges include unclear and varying intents of other agents, mismatches between dynamics models and reality, incomplete sensor information, and interaction effects.

One popular approach to addressing the challenge of *a priori* unknown agent intent is to use rule-based or data-driven algorithms to predict *individual trajectories* for each agent, as in Schmerling et al. (2017). Alternatively, Ziebart et al. (2009), Bandyopadhyay et al. (2013), and Kochenderfer et al. (2010) explicitly predict an agent's full state distribution over time; this representation may be better suited to capturing uncertainty in an agent's dynamics and the environment itself. Wang et al. (2019) and Fisac et al. (2018b) pose the prediction problem game-theoretically to model coupled human–robot interaction effects explicitly.

Unfortunately, a significant problem still remains: if an agent suddenly moves in a way that is not predicted, or not assigned sufficient probability, the robot may not react

Department of Electrical Engineering and Computer Sciences,University of California, Berkeley, CA, USA

*Indicates equal contribution

Corresponding author:
David Fridovich-Keil, Department of Electrical Engineering and Computer Sciences,University of California, Cory Hall, Berkeley, CA 94720, USA.
Email: dfk@eecs.berkeley.edu

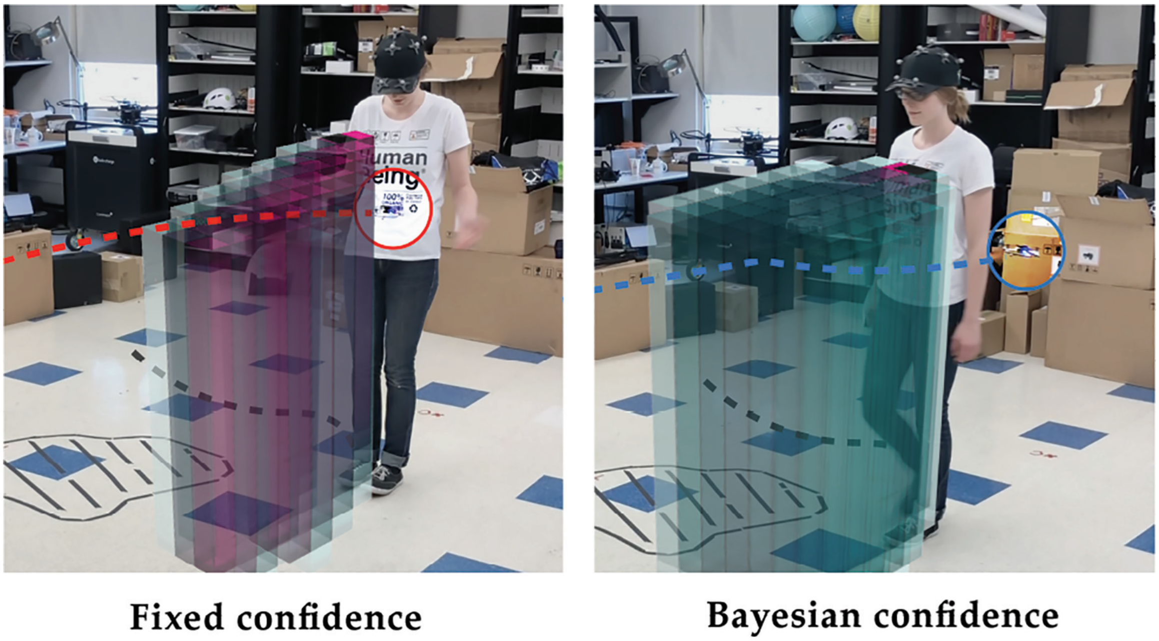

Fig. 1. When planning around humans, accurate predictions of human motion (visualized here in pink and blue, representing high and low probability, respectively) are an essential prerequisite for safety. Unfortunately, these approaches may fail to explain all observed motion at runtime (e.g., human avoids unmodeled spill on the ground), leading to inaccurate predictions, and potentially, collisions (left). Our method addresses this by updating its *predictive model confidence* in real time (right), leading to more conservative motion planning in circumstances when predictions are known to be suspect.

appropriately. For example, in Figure 1 a pedestrian is walking around an obstacle that the robot, a quadcopter, cannot detect. To the robot, such behavior may be assigned very low probability, which could lead the robot to plan a dangerous trajectory. In this particular example, this inaccuracy caused the quadcopter to collide with the pedestrian (Figure 1, left).

To prepare for this eventuality, we introduce the idea of *confidence-aware prediction*. We argue that, in addition to predicting the future state of an agent, it is also crucial for a robot to assess the quality of the mechanism by which it is generating those predictions. That is, a robot should reason about how *confident* it is in its predictions of other agents before attempting to plan future motion. For computational efficiency, the quadcopter uses a simplified model of pedestrian dynamics and decision-making. Thus equipped, it generates a time-varying probability distribution over the future state of the pedestrian, and plans trajectories to a pre-specified goal that maintain a low probability of collision. Figure 1(right) illustrates how this approach works in practice. The quadcopter maintains a Bayesian *belief* over its prediction confidence. As soon as the pedestrian moves in a way that was assigned low probability by the predictive model, the quadcopter adjusts its belief about the accuracy of that model. Consequently, it is less certain about what the pedestrian will do in the future. This leads the quadcopter's onboard motion planner, which attempts to find efficient trajectories with low probability of collision, to generate more cautious, and perhaps less efficient, motion plans.

In order to improve the robustness of generated motion plans, we employ the recent FaSTrack framework from Herbert et al. (2017) for fast and safe motion planning and tracking. FaSTrack quantifies the maximum possible tracking error between a high-order dynamical model of the physical robot and the (potentially lower-order) dynamical model used by its motion planner. Solving an offline Hamilton–Jacobi reachability problem yields a guaranteed *tracking error bound* and the corresponding *safety controller*. These may be used by an out-of-the-box real-time motion planning algorithm to facilitate motion plans with strong runtime collision-avoidance guarantees.

The remainder of this paper is organized as follows. Section 2 places this work in the context of existing literature in human motion modeling and prediction, as well as robust motion planning. Section 3 frames the prediction and planning problems more formally, and introduces a running example used throughout the paper. Section 4 presents our main contribution: confidence-aware predictions. Section 5 showcases confidence-aware predictions in operation in several examples. Section 6 describes the application of the robust motion planning framework from FaSTrack to this setting, in which predictions are probabilistic. Section 7 explores a connection between our approach and reachability theory. Section 8 presents experimental results from a hardware demonstration. Finally, Section 9 concludes with a discussion of some of the limitations of our work and how they might be addressed in specific applications, as well as suggestions for future research.

2. Prior work[2]

2.1. Human modeling and prediction

One common approach for predicting human actions is to collect data from real-world scenarios and train a machine learning model via supervised learning. Such techniques use the human's current state, and potentially her prior state and action history, to predict future actions directly. Amor et al. (2014), Ding et al. (2011), Koppula and Saxena (2013), Lasota and Shah (2015), and Hawkins et al. (2013) demonstrated the effectiveness of this approach for inference and planning around human arm motion. In addition, Hawkins et al. (2013) focused on multi-step tasks such as assembly, and Schmerling et al. (2017) and Driggs-Campbell et al. (2018) addressed the prediction problem for human drivers.

Rather than predicting actions directly, an alternative is for the robot to model the human as a rational agent seeking to maximize an unknown objective function. The human's actions up to a particular time may be viewed as Bayesian evidence from which the robot may infer the parameters of that objective. Assuming that the human seeks to maximize this objective in the future, the robot can predict her future movements (e.g., Bai et al., 2015; Baker et al., 2007; Ng and Russell, 2000; Ziebart et al., 2009). In this paper, we build on this work by introducing a principled online technique for estimating confidence in such a learned model of human motion.

2.2. Safe robot motion planning

Once armed with a predictive model of the human motion, the robot may leverage motion planning methods that plan around uncertain moving obstacles and generate real-time dynamically feasible and safe trajectories.

To avoid moving obstacles in real time, robots typically employ reactive and/or path-based methods. Reactive methods directly map sensor readings into control, with no memory involved (e.g., Belkhouche, 2009). Path-based methods such as rapidly-exploring random trees from Karaman and Frazzoli (2011) and A* from Hart et al. (1968) find simple kinematic paths through space and, if necessary, time. These path-based methods of planning are advantageous in terms of efficiency, yet, while they have in some cases been combined with probabilistically moving obstacles as in Aoude et al. (2013) and Ziebart et al. (2009), they do not consider the endogenous dynamics of the robot or exogenous disturbances such as wind. As a result, the robot may deviate from the planned path and potentially collide with obstacles. It is common for these plans to try to avoid obstacles by a heuristic margin of error. Herbert et al. (2017) and Fridovich-Keil et al. (2018) proposed FaSTrack, a recent algorithm that provides a guaranteed tracking error margin and corresponding error-feedback controller for dynamic systems tracking a generic planner in the presence of bounded external disturbance. Our work builds upon FaSTrack to create an algorithm that can safely and dynamically navigate around uncertain moving obstacles in real time.

3. Problem setup

We consider a single mobile robot operating in a shared space with a single human agent (e.g., a pedestrian or human-driven car). For simplicity, we presume that the robot has full knowledge of its own state and that of the human, although both would require online estimation in practice. As we present each formal component of this problem, we will provide a concrete illustration using a running example in which a quadcopter is navigating around a pedestrian.

3.1. Dynamical system models and safety

We will model the motion of both the human and the robot as the evolution of two dynamical systems. Let the state of the human be $x_H \in \mathbb{R}^{n_H}$, where n_H is the dimension of the human state space. We similarly define the robot's state, for planning purposes, as $x_R \in \mathbb{R}^{n_R}$. In general, these states could represent the positions and velocities of a mobile robot and a human in a shared environment, the kinematic configurations of a human and a robotic manipulator in a common workspace, or the positions, orientations, and velocities of human-driven and autonomous vehicles in an intersection. We express the evolution of these states over time as a family of ordinary differential equations:

$$\dot{x}_H = f_H(x_H, u_H), \quad \dot{x}_R = f_R(x_R, u_R) \tag{1}$$

where $u_H \in \mathbb{R}^{m_H}$ and $u_R \in \mathbb{R}^{m_R}$ are the control actions of the human and robot, respectively.

Running example: *We introduce a running example for illustration purposes throughout the paper. In this example we consider a small quadcopter that needs to fly to goal location $g_R \in \mathbb{R}^3$ in a room where a pedestrian is walking. For the purposes of planning, the quadcopter's 3D state is given by its position in space $x_R = [p_x, p_y, p_z]$, with velocity controls assumed decoupled in each spatial direction, up to $v_R = 0.25$ m/s. The human can only move by walking and therefore her state is given by planar coordinates $x_H = [h_x, h_y]$ evolving as $\dot{x}_H = [v_H \cos u_H, v_H \sin u_H]$. Intuitively, we model the human as moving with a fixed speed and controlling their heading angle. At any given time, the human is assumed to either move at a leisurely walking speed ($v_H \approx 1$ m/s) or remain still ($v_H \approx 0$).*

Ultimately, the robot needs to plan and execute an efficient trajectory to a pre-specified goal state (g_R), without colliding with the human. We define the keep-out set $\mathcal{K} \subset \mathbb{R}^{n_H} \times \mathbb{R}^{n_R}$ as the set of joint robot–human states to be avoided (for example, because they imply physical collisions). To avoid reaching this set, the robot must reason about the human's future motion when constructing its own motion plan.

Running example: *In our quadcopter-avoiding-pedestrian example, $\mathcal{K}$ consists of joint robot–human states in which the quadcopter is flying within a square of side length $l = 0.3$ m centered around the human's location, while at any altitude, as well as any joint states in which the robot is outside the environment bounds defined as a box with a square base of side $L = 3.66$ m and height $H = 2$ m, regardless of the human's state.*

3.2. Robust robot control

Provided an objective and a dynamics model, the robot must generate a motion plan that avoids the keep-out set $\mathcal{K}$. Unfortunately, this safety requirement is difficult to meet during operation for two main reasons.

1. *Model mismatch.* The dynamical system model f_R will never be a perfect representation of the real robot. This mismatch could lead to unintended collision.
2. *Disturbances.* Even with a perfect dynamics model, there may be unobserved, external "disturbance" inputs such as wind or friction. Without accounting for these disturbances, the system is not guaranteed to avoid $\mathcal{K}$, even if the planned trajectory is pointwise collision-free.

To account for modeling error and external disturbances, we could in principle design a higher-fidelity dynamical model directly in a robust motion planning framework. Unfortunately, however, real-time trajectory optimization in high dimensions can be computationally burdensome, particularly when we also require some notion of robustness to external disturbance. Ideally, we would like to enjoy the computational benefits of planning with a lower-fidelity model while enforcing the safety constraints induced by the higher-fidelity model. To characterize this model mismatch, we consider a higher fidelity and typically higher-order dynamical representation of the robot, with state representation $s_R \in \mathbb{R}^{n_S}$. This dynamical model will also explicitly account for external disturbances as unknown bounded inputs, distinct from control inputs. In order to map between this higher-fidelity "tracking" state s_R and the lower-fidelity "planning" state x_R, we shall assume a known projection operator $\pi : \mathbb{R}^{n_S} \to \mathbb{R}^{n_R}$. Fortunately, we can plan in the lower-dimensional state space at runtime, and guarantee robust collision avoidance via an offline reachability analysis that quantifies the effects of model mismatch and external disturbance. This framework, called FaSTrack and first proposed by Herbert et al. (2017), is described in further detail in Section 6.

Running example: *We model our quadcopter with the following flight dynamics (in the near-hover regime, at zero yaw with respect to a global coordinate frame):*

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \quad \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} a_g \tan u_\theta \\ -a_g \tan u_\phi \\ u_T - a_g \end{bmatrix} \tag{2}$$

where $[p_x, p_y, p_z]$ is the quadcopter's position in space and $[v_x, v_y, v_z]$ is its velocity expressed in the fixed global frame. We model its control inputs as thrust acceleration u_T and attitude angles (roll u_ϕ and pitch u_θ), and denote the acceleration due to gravity as a_g. The quadcopter's motion planner generates nominal kinematic trajectories in the lower-dimensional $[p_x, p_y, p_z]$ position state space. Therefore, we have a linear projection map $\pi(s_R) = [I_3, 0_3]s_R$, that is, x_R retains the position variables in s_R and discards the velocities.

3.3. Predictive human model

In order to predict the human's future motion, the robot uses its internal model of human dynamics, f_H. Under this modeling assumption, the human's future trajectory depends upon the choice of control input over time, $u_H(\cdot)$. Extensive work in econometrics and cognitive science, such as that of Von Neumann and Morgenstern (1945), Luce (1959), and Baker et al. (2007), has shown that human behavior (that is, u_H) can be well modeled by utility-driven optimization. Thus, the robot models the human as optimizing a reward function, $r_H(x_H, u_H; \theta)$, that depends on the human's state and action, as well as a set of parameters θ. This reward function could be a linear combination of features as in many inverse optimal control implementations (where the goal or feature weighting θ must be learned, either online or offline), or more generally learned through function approximators such as deep neural networks, where θ are the trained weights as in Finn et al. (2016).

We assume that the robot has a suitable human reward function r_H, either learned offline from prior human demonstrations or otherwise encoded by the system designers. Thus, endowed with r_H, the robot can model the human's choice of control action as a probability distribution over actions conditioned on state. Under maximum-entropy assumptions (Ziebart et al., 2008) inspired by noisy-rationality decision-making models (Baker et al., 2007), the robot models the human as more likely to choose (discrete) actions u_H with high expected utility, in this case the state–action value (or Q-value):

$$P(u_H | x_H; \beta, \theta) = \frac{e^{\beta Q_H(x_H, u_H; \theta)}}{\sum_{\tilde{u}} e^{\beta Q_H(x_H, \tilde{u}; \theta)}} \tag{3}$$

We use a temporally and spatially discretized version of human dynamics, $\tilde{f}_H$. These discrete-time dynamics may be found by integrating f_H over a fixed time step of Δt with fixed control u_H over the interval. Section 5 provides further details on this discretization.

Running example: *The quadcopter's model of the human assumes the human intends to reach some target location $g_H \in \mathbb{R}^2$ in a straight line. The human's reward function is given by the distance traveled over time step Δt, i.e., $r_H(x_H, u_H; g_H) = -v_H \Delta t$, and human trajectories are constrained to terminate at g_H. The state–action value,* *parameterized by* $\theta = g_H$, *captures the optimal cost of reaching* g_H *from* x_H *when initially applying* u_H *for a duration* Δt: $Q_H(x_H, u_H; g_H) = -v_H\Delta t - \| x_H + v_H\Delta t[\cos u_H, \sin u_H]^\top - g_H\|_2$.

Often, the coefficient β is termed the *rationality coefficient*, because it quantifies the degree to which the robot expects the human's choice of control to align with its model of utility. For example, taking $\beta \downarrow 0$ yields a model of a human who appears "irrational," choosing actions uniformly at random and completely ignoring the modeled utility. At the other extreme, taking $\beta \uparrow \infty$ corresponds to a "perfectly rational" human, whose actions exactly optimize the modeled reward function. As we will see in Section 4, β can also be viewed as a measure of the robot's *confidence* in the predictive accuracy of Q_H.

Note that $Q_H(x_H, u_H; \theta)$ only depends on the human state and action and not on the robot's. Thus far, we have intentionally neglected discussion of human–robot interaction effects. These effects are notoriously difficult to model, and the community has devoted a significant effort to building and validating a variety of models (e.g., Sadigh et al., 2016; Trautman and Krause, 2010). In that spirit, we could have chosen to model human actions u_H as dependent upon robot state x_R in (3), and likewise defined Q_H to depend upon x_R. This extended formulation is sufficiently general as to encompass all possible (Markov) interaction models. However, in this work we explicitly do not model these interactions; indeed, one of the most important virtues of our approach is its robustness to precisely these sorts of modeling errors.

3.4. Probabilistically safe motion planning

Ideally, the robot's motion planner should generate trajectories that reach a desired goal state efficiently, while maintaining safety. More specifically, in this context "safety" indicates that the physical system will never enter the keep-out set $\mathcal{K}$ during operation, despite human motion and external disturbances. That is, we would like to guarantee that $(\pi(s_R), x_H) \notin \mathcal{K}$ for all time.

To make this type of strong, deterministic, *a priori* safety guarantee requires the robot to avoid the set of all human states x_H which could possibly be occupied at a particular time, i.e., the human's *forward reachable set*. If the robot can find trajectories that are safe for *any* possible human trajectory then there is no need to predict the human's next action. Unfortunately, the forward reachable set of the human often encompasses such a large volume of the workspace that it is impossible for the robot to find a guaranteed safe trajectory to the goal state. This motivates refining our notion of prediction: rather than reasoning about all the places where the human *could* be, the robot can instead reason about *how likely* the human is to be at each location. This probabilistic reasoning provides a guide for planning robot trajectories with a quantitative degree of safety assurance.

Our probabilistic model of human control input (3) coupled with dynamics model f_H allows us to compute a probability distribution over human states for every future time. By relaxing our conception of safety to consider only collisions that might occur with sufficient probability P_{th}, we dramatically reduce the effective volume of this set of future states to avoid. In practice, P_{th} should be chosen carefully by a system designer in order to trade off overall collision probability with conservativeness in motion planning.

The proposed approach in this paper follows two central steps to provide a quantifiable, high-confidence collision avoidance guarantee for the robot's motion around the human. In Section 4 we present our proposed Bayesian framework for reasoning about the uncertainty inherent in a model's prediction of human behavior. Based on this inference, we demonstrate how to generate a real-time probabilistic prediction of the human's motion over time. Next, in Section 6, we extend a state-of-the-art, provably safe, real-time robotic motion planner to incorporate our time-varying probabilistic human prediction.

4. Confidence-aware human motion prediction

Any approach to human motion prediction short of computing a full forward reachable set must, explicitly or implicitly, reflect a model of human decision-making. In this work, we make that model explicit by assuming that the human chooses control actions in a Markovian fashion according to the probability distribution (3). Other work in the literature, such as that of Schmerling et al. (2017), aims to learn a generative probabilistic model for human trajectories; implicitly, this training procedure distills a model of human decision-making. Whether explicit or implicit, these models are by nature imperfect and liable to make inaccurate predictions eventually. One benefit of using an explicit model of human decision-making, such as (3), is that we may reason directly and succinctly about its performance online.

In particular, the entropy of the human control distribution in (3) is a decreasing function of the parameter β. High values of β place more probability mass on high-utility control actions u_H, whereas low values of β spread the probability mass more evenly between different control inputs, regardless of their modeled utility Q_H. Therefore, β naturally quantifies how well the human's motion is expected to agree with the notion of optimality encoded in Q_H. The commonly used term "rationality coefficient," however, seems to imply that discrepancies between the two indicate a failure on the human's part to make the "correct" decisions, as encoded by the modeled utility. Instead, we argue that these inevitable disagreements are primarily a result of the model's inability to fully capture the human's behavior. Thus, instead of conceiving of β as a rationality measure, we believe that β can be given a more pragmatic interpretation related to the accuracy with which the robot's model of the human is able to explain the human's motion. Consistently, in this paper, we refer to β as *model confidence*.

An important related observation following from this interpretation of β is that the predictive accuracy of a human model is likely to change over time. For example, the human may change their mind unexpectedly, or react suddenly to some aspect of the environment that the robot is unaware of. Therefore, we shall model β as an unobserved, time-varying parameter. Estimating it in real time provides us with a direct, quantitative summary of the degree to which the utility model Q_H explains the human's current motion. To do this, we maintain a Bayesian *belief* about the possible values of β. Initially, we begin with a uniform prior over β and over time this distribution evolves given measurements of the human's state and actions.

4.1. Real-time inference of model confidence

We reason about the model confidence β as a hidden state in a hidden Markov model (HMM) framework. The robot starts with a prior belief b_-^0 over the initial value of β. In this work, we use a uniform prior, although that is not strictly necessary. At each discrete time step $k \in \{0, 1, 2, \ldots\}$, it will have some belief about model confidence $b_-^k(\beta)$.[3] After observing a human action u_H^k, the robot will update its belief to b_+^k by applying Bayes' rule.

The hidden state may evolve between subsequent time steps, accounting for the important fact that the predictive accuracy of the human model may change over time as unmodeled factors in the human's behavior become more or less relevant. As, by definition, we do not have access to a model of these factors, we use a naive "ϵ-static" transition model: at each time k, β may, with some probability ϵ, be re-sampled from the initial distribution b_-^0, and otherwise retains its previous value. We define the belief over the next value of β (denoted by β') as an expectation of the conditional probability $P(\beta'|\beta)$, i.e., $b_-^k(\beta') := \mathbb{E}_{\beta \sim b_+^{k-1}}[P(\beta'|\beta)]$. Concretely, this expectation may be computed as

$$b_-^k(\beta') = (1 - \epsilon) b_+^{k-1}(\beta') + \epsilon b_-^0(\beta') \tag{4}$$

By measuring the evolution of the human's state x_H over time, we assume that, at every time step k, the robot is able to observe the human's control input u_H^k. This observed control may be used as evidence to update the robot's belief b_-^k about β over time via a Bayesian update:

$$b_+^k(\beta) = \frac{P(u_H^k | x_H^k; \beta, \theta) b_-^k(\beta)}{\sum_{\tilde{\beta}} P(u_H^k | x_H^k; \tilde{\beta}, \theta) b_-^k(\tilde{\beta})} \tag{5}$$

with $b_+^k(\beta) := P(\beta | x_H^{0:k}, u_H^{0:k})$ for $k \in \{0, 1, \ldots\}$, and $P(u_H^k | x_H^k; \beta, \theta)$ given by (3).

It is critical to be able to perform this update rapidly to facilitate real-time operation; this would be difficult in the original continuous hypothesis space $\beta \in [0, \infty)$, or even in a large discrete set. Fortunately, our software examples in Section 5 and hardware demonstration in Section 8 suggest that maintaining a Bayesian belief over a relatively small set of $N_\beta = 5$ discrete values of β distributed on a log scale achieves significant improvement relative to using a fixed value.

The "ϵ-static" transition model leads to the desirable consequence that old observations of the human's actions have a smaller influence on the current model confidence distribution than recent observations. In fact, if no new observations are made, successively applying time updates asymptotically contracts the belief towards the initial distribution, that is, $b_-^k(\cdot) \to b_-^0(\cdot)$. The choice of parameter ϵ effectively controls the rate of this contraction, with higher ϵ leading to more rapid contraction.

4.2. Human motion prediction

Equipped with a belief over β at time step k, we are now able to propagate the human's state distribution forward to any future time via the well-known Kolmogorov forward equations, recursively. In particular, suppose that we know the probability that the human is in each state x_H^κ at some future time step κ. We know that (according to our utility model) the probability of the human choosing control u_H^κ in state x_H^κ is given by (3). Accounting for the otherwise deterministic dynamics model $\tilde{f}_H$, we obtain the following expression for the human's state distribution at the following time step $\kappa + 1$:

$$P(x_H^{\kappa+1}; \beta, \theta) = \sum_{x_H^\kappa, u_H^\kappa} P(x_H^{\kappa+1} | x_H^\kappa, u_H^\kappa; \beta, \theta) \cdot P(u_H^\kappa | x_H^\kappa; \beta, \theta) P(x_H^\kappa; \beta, \theta) \tag{6}$$

for a particular choice of β. Marginalizing over β according to our belief at the current step time k, we obtain the overall occupancy probability distribution at each future time step κ:

$$P(x_H^\kappa; \theta) = \mathbb{E}_{\beta \sim b^k} P(x_H^\kappa; \beta, \theta) \tag{7}$$

Note that (6) is expressed more generally than is strictly required. Indeed, because the only randomness in dynamics model $\tilde{f}_H$ originates from the human's choice of control input u_H, we have $P(x_H^{\kappa+1} | x_H^\kappa, u_H^\kappa; \beta, \theta) = \mathbb{1}\{x_H^{\kappa+1} = \tilde{f}_H(x_H^\kappa, u_H^\kappa)\}$.

4.3. Model confidence with auxiliary parameter identification

Thus far, we have tacitly assumed that the only unknown parameter in the human utility model (3) is the model confidence, β. However, often one or more of the auxiliary parameters θ are also unknown. These auxiliary parameters could encode one or more human goal states or intents, or other characteristics of the human's utility, such as her preference for avoiding parts of the environment. Further, much like model confidence, they may change over time.

In principle, it is possible to maintain a Bayesian belief over β and θ jointly. The Bayesian update for the hidden state (β, θ) is then given by

$$b_{+}^{k}(\beta, \theta) = \frac{P(u_H^k | x_H^k; \beta, \theta) b_{-}^{k}(\beta, \theta)}{\sum_{\tilde{\beta}, \tilde{\theta}} P(u_H^k | x_H^k; \tilde{\beta}, \tilde{\theta}) b_{-}^{k}(\tilde{\beta}, \tilde{\theta})} \tag{8}$$

with $b_{+}^{k}(\beta, \theta) := P(\beta, \theta | x_H^{0:k}, u_H^{0:k})$ the running posterior and $b_{-}^{k}(\beta, \theta) := P(\beta, \theta | x_H^{0:k-1}, u_H^{0:k-1})$ the prior at time step k.

This approach can be practical for parameters taking finitely many values from a small, discrete set, e.g., possible distinct modes for a human driver (distracted, cautious, aggressive). However, for certain scenarios or approaches it may not be practical to maintain a full Bayesian belief on the parameters θ. In such cases, it is reasonable to replace the belief over θ with a point estimate $\bar{\theta}$, such as the maximum likelihood estimator or the mean, and substitute that estimate into (6). Depending on the complexity of the resulting maximum likelihood estimation problem, it may or may not be computationally feasible to update the parameter estimate $\bar{\theta}$ at each time step. Fortunately, even when it is computationally expensive to estimate $\bar{\theta}$, we can leverage our model confidence as an indicator of when re-estimating these parameters may be most useful. That is, when model confidence degrades that may indicate poor estimates of $\bar{\theta}$.

5. Prediction examples

We illustrate these inference steps with two sets of examples: our running pedestrian example and a simple model of a car.

5.1. Pedestrian model (running example)

So far, we have presented a running example of a quadcopter avoiding a human. We use a deliberately simple, purely kinematic model of continuous-time human motion:

$$\dot{x}_H = \begin{bmatrix} \dot{h}_x \\ \dot{h}_y \end{bmatrix} = \begin{bmatrix} v_H \cos u_H \\ v_H \sin u_H \end{bmatrix} \tag{9}$$

However, as discussed in Section 3.3, the proposed prediction method operates in discrete time (and space). The discrete dynamics corresponding to (9) are given by

$$x_H^{k+1} - x_H^k \equiv x_H(t + \Delta t) - x_H(t) = \begin{bmatrix} v_H \Delta t \cos u_H(t) \\ v_H \Delta t \sin u_H(t) \end{bmatrix} \tag{10}$$

for a time discretization of Δt.

5.2. Dubins car model

To emphasize the generality of our method, we present similar results for a different application domain: autonomous driving. We will model a human-driven vehicle as a dynamical system whose state x_H evolves as

$$\dot{x}_H = \begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{h}_\phi \end{bmatrix} = \begin{bmatrix} v_H \cos h_\phi \\ v_H \sin h_\phi \\ u_H \end{bmatrix} \tag{11}$$

Observe that, while (11) appears very similar to (9), in this Dubins car example the angle of motion is a *state*, not a *control input*.

We discretize these dynamics by integrating (11) from t to $t + \Delta t$, assuming a constant control input u_H:

$$x_H^{k+1} - x_H^k \equiv x_H(t + \Delta t) - x_H(t) = \begin{bmatrix} \frac{v_H}{u_H(t)} \left(\sin\left(h_\phi(t) + u_H(t)\Delta t\right) - \sin\left(h_\phi(t)\right) \right) \\ -\frac{v_H}{u_H(t)} \left(\cos\left(h_\phi(t) + u_H(t)\Delta t\right) - \cos\left(h_\phi(t)\right) \right) \\ u_H \Delta t \end{bmatrix}$$

For a specific goal position $g = [g_x, g_y]$, the Q-value corresponding to state–action pair (x_H, u_H) and reward function $r_H(x_H, u_H) = -v_H \Delta t$ (until the goal is reached) may be found by solving a shortest path problem offline.

5.3. Accurate model

First, we consider a scenario in which the robot has full knowledge of the human's goal, and the human moves along the shortest path from a start location to this known goal state. Thus, human motion is well-explained by Q_H.

The first row of Figure 2 illustrates the probability distributions our method predicts for the pedestrian's future state at different times. Initially, the predictions generated by our Bayesian confidence-inference approach (right) appear similar to those generated by the low model confidence predictor (left). However, our method rapidly discovers that Q_H is an accurate description of the pedestrian's motion and generates predictions that match the high model confidence predictor (center). The data used in this example was collected by tracking the motion of a real person walking in a motion capture arena. See Section 8 for further details.

Likewise, the first row of Figure 3 shows similar results for a human-driven Dubins car model (in simulation) at an intersection. Here, traffic laws provide a strong prior on the human's potential goal states. As shown, our method of Bayesian model confidence inference quickly infers the correct goal and learns that the human driver is acting in accordance with its model Q_H. The resulting predictions are substantially similar to the high-β predictor. The data used in this example was simulated by controlling a Dubins car model along a pre-specified trajectory.

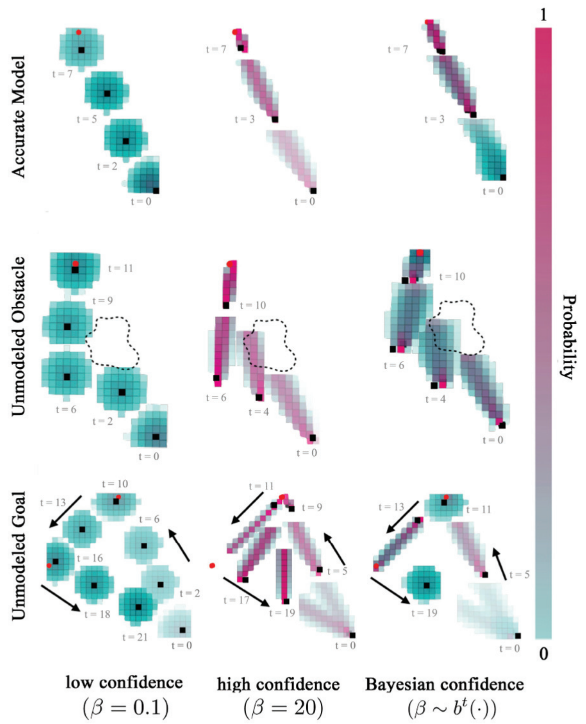

Fig. 2. Snapshots of pedestrian trajectory and probabilistic model predictions. Top row: Pedestrian moves from the bottom right to a goal marked as a red circle. Middle row: Pedestrian changes course to avoid a spill on the floor. Bottom row: Pedestrian moves to one known goal, then to another, then to a third which the robot has not modeled. The first two columns show predictions for low and high model confidence; the third column shows the predictions using our Bayesian model confidence. For all pedestrian videos, see https://youtu.be/lh_E9rW-MJo.

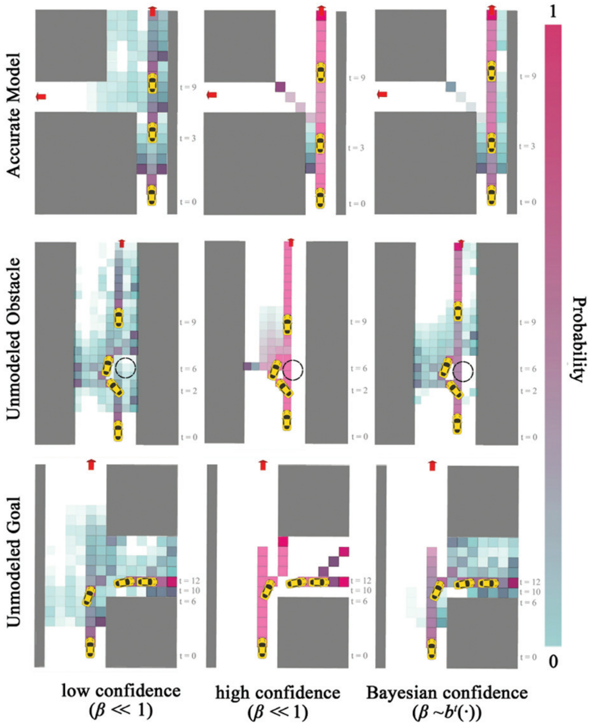

Fig. 3. Snapshots of Dubins car and probabilistic predictions. Top row: Car moves straight ahead toward one of two known goals (red arrows), staying in its lane. Middle row: Car suddenly swerves to the left to avoid a pothole. Bottom row: Car turns to the right, away from the only known goal. The left and center columns show results for low and high confidence predictors, respectively, and the right column shows our approach using Bayesian inferred model confidence. For all Dubins car videos, see https://youtu.be/sAJKNnP42fQ.

5.4. Unmodeled obstacle

Often, robots do not have fully specified models of the environment. Here, we showcase the resilience of our approach to unmodeled obstacles that the human must avoid. In this scenario, the human has the same start and goal as in the accurate model case, except that there is an obstacle along the way. The robot is unaware of this obstacle, however, which means that in its vicinity the human's motion is not well-explained by Q_H, and $b(\beta)$ ought to place more probability mass on higher values of β.

The second rows of Figure 2 and Figure 3 illustrate this type of situation for the pedestrian and Dubins car, respectively. In Figure 2, the pedestrian walks to an *a priori* known goal location and avoids an unmodeled spill on the ground. Analogously, in Figure 3 the car swerves to avoid a large pothole. By inferring model confidence online, our approach generates higher-variance predictions of future state, but only in the vicinity of these unmodeled obstacles. At other times throughout the episode when Q_H is more accurate, our approach produces predictions more in line with the high model confidence predictor.

5.5. Unmodeled goal

In most realistic human–robot encounters, even if the robot does have an accurate environment map and observes all obstacles, it is unlikely for it to be aware of all human goals. We test our approach's resilience to unknown human goals by constructing a scenario in which the human moves between both known and unknown goals.

The third row of Figure 2 illustrates this situation for the pedestrian example. Here, the pedestrian first moves to one known goal position, then to another, and finally back to the start which was not a modeled goal location. The first two legs of this trajectory are consistent with the robot's model of goal-oriented motion, though accurate prediction does require the predictor to infer *which* goal the pedestrian is walking toward. However, when the pedestrian returns to the start, her motion appears inconsistent with Q_H, skewing the robot's belief over β toward zero.

Similarly, in the third row of Figure 3 we consider a situation in which a car makes an unexpected turn onto an unmapped access road. As soon as the driver initiates the turn, our predictor rapidly learns to distrust its internal model Q_H and shift its belief over β upward.

6. Safe probabilistic planning and tracking

Given probabilistic predictions of the human's future motion, the robot must plan efficient trajectories that avoid collision with high probability. In order to reason robustly about this probability of future collision, we must account for potential tracking errors incurred by the real system as it follows planned trajectories. To this end, we build on the recent FaSTrack framework of Herbert et al. (2017), which provides control-theoretic robust safety certificates in the presence of deterministic obstacles, and extend it to achieve approximate probabilistic collision-avoidance.

6.1. Background: fast planning, safe tracking

Recall that x_R is the robot's state for the purposes of motion planning, and that s_R encodes a higher-fidelity, potentially higher-dimensional notion of state (with associated dynamics). The recently proposed FaSTrack framework from Herbert et al. (2017) uses Hamilton–Jacobi reachability analysis to quantify the worst-case tracking performance of the s_R-system as it follows trajectories generated by the x_R-system. For further reading on reachability analysis refer to Evans and Souganidis (1984), Mitchell et al. (2005), and Bansal et al. (2017). A byproduct of this FaSTrack analysis is an error feedback controller that the s_R system can use to achieve this worst-case tracking error. The tracking error bound may be given to one of many off-the-shelf real-time motion planning algorithms operating in x_R-space in order to guarantee real-time collision avoidance by the s_R-system.

Formally, FaSTrack precomputes an optimal tracking controller, as well as a corresponding compact set $\mathcal{E}$ in the robot's planning state space, such that $(\pi(s_R(t)) - x_{R,\mathrm{ref}}(t)) \in \mathcal{E}$ for *any* reference trajectory proposed by the lower-fidelity planner. This bound $\mathcal{E}$ is a trajectory tracking certificate that can be passed to an online planning algorithm for real-time safety verification: the dynamical robot is guaranteed to *always* be somewhere within the bound relative to the current planned reference point $x_{R,\mathrm{ref}}(t)$. This tracking error bound may sometimes be expressed analytically; otherwise, it may be computed numerically offline using level set methods (e.g., Mitchell, 2009). Equipped with $\mathcal{E}$, the planner can generate safe plans online by ensuring that the entire tracking error bound around the nominal state remains collision-free throughout the trajectory. Efficiently checking these $\mathcal{E}$-augmented trajectories for collisions with known obstacles is critical for real-time performance. Note that the planner only needs to know $\mathcal{E}$ (which is computed offline) and otherwise requires no explicit understanding of the high-fidelity model.

Running example: *As dynamics (2) are decoupled in the three spatial directions, the bound $\mathcal{E}$ computed by FaSTrack is an axis-aligned box of dimensions $\mathcal{E}_x \times \mathcal{E}_y \times \mathcal{E}_z$. For further details refer to Fridovich-Keil et al. (2018).*

6.2. Robust tracking, probabilistic safety

Unfortunately, planning algorithms for collision checking against deterministic obstacles cannot be readily applied to our problem. Instead, a trajectory's collision check should return the probability that it *might* lead to a collision. Based on this probability, the planning algorithm can discriminate between trajectories that are *sufficiently safe* and those that are not.

As discussed in Section 3.4, a safe online motion planner invoked at time t should continually check the probability that, at any future time τ, $(\pi(s_R(\tau)), x_H(\tau)) \in \mathcal{K}$. The tracking error bound guarantee from FaSTrack allows us to conduct worst-case analysis on collisions given a human state x_H. Concretely, if no point in the Minkowski sum $\{x_R + \mathcal{E}\}$ is in the collision set with x_H, we can guarantee that the robot is not in collision with the human.

The probability of a collision event for any point $x_R(\tau)$ along a candidate trajectory is then

$$P_{\mathrm{coll}}(x_R(\tau)) := P((x_R, x_H) \in \mathcal{K}) \tag{12}$$

Assuming worst-case tracking error bound $\mathcal{E}$, this quantity can be upper-bounded by the total probability that $x_H(\tau)$ will be in collision with *any* of the possible robot states $\tilde{x}_R \in \{x_R(\tau) + \mathcal{E}\}$. For each robot planning state $x_R \in \mathbb{R}^{n_R}$ we define the set of human states in potential collision with the robot:

$$\mathcal{H}_{\mathcal{E}}(x_R) := \{\tilde{x}_H \in \mathbb{R}^{n_H} : \exists \tilde{x}_R \in \{x_R + \mathcal{E}\}, (\tilde{x}_R, \tilde{x}_H) \in \mathcal{K}\} \tag{13}$$

Running example: *Given $\mathcal{K}$ and $\mathcal{E}$, $\mathcal{H}_{\mathcal{E}}(x_R)$ is the set of human positions within the rectangle of dimensions $(l + \mathcal{E}_x) \times (l + \mathcal{E}_y)$ centered on $[p_x, p_y]$. A human anywhere in this rectangle could be in collision with the quadcopter.*

The following result follows directly from the definition of the tracking error bound and a union bound.

Proposition 1. *The probability of a robot with worst-case tracking error $\mathcal{E}$ colliding with the human at any trajectory point $x_R(\tau)$ is bounded above by the probability mass of $x_H(\tau)$ contained within $\mathcal{H}_{\mathcal{E}}(x_R(\tau))$.*

We consider *discrete-time* motion plans. The probability of collision along any such trajectory from current time step k to final step $k + K$ is upper-bounded by

$$\begin{aligned} &P_{\mathrm{coll}}^{k:k+K} \leqslant \overline{P_{\mathrm{coll}}^{k:k+K}} := \\ &1 - \prod_{\kappa=k}^{k+K} P(x_H^{\kappa} \notin \mathcal{H}_{\mathcal{E}}(x_R^{\kappa}) | x_H^{\kappa} \notin \mathcal{H}_{\mathcal{E}}(x_R^{s}), k \leqslant s < \kappa) \end{aligned} \tag{14}$$

Evaluating the right-hand side of (14) exactly requires reasoning about the joint distribution of human states over all time steps and its conditional relationship on whether collision has yet occurred. This is equivalent to maintaining a probability distribution over the exponentially large space of trajectories $x_H^{k:k+K}$ that the human might follow. As motion planning occurs in real time, we shall resort to a heuristic approximation of (14).

One approach to approximating (14) is to assume that the event $x_H^{\kappa_1} \notin \mathcal{H}_\mathcal{E}(x_R^{\kappa_1})$ is independent of $x_H^{\kappa_2} \notin \mathcal{H}_\mathcal{E}(x_R^{\kappa_2})$, for all $\kappa_1 \neq \kappa_2$. This independence assumption is equivalent to removing the conditioning in (14). Unfortunately, this approximation is excessively pessimistic; if there is no collision at time step κ, then collision is also unlikely at time step $\kappa + 1$ because both human and robot trajectories are continuous. In fact, for sufficiently small time discretization Δt and nonzero collision probabilities at each time step, the total collision probability resulting from an independence assumption would approach 1 exponentially fast in the number of time steps K.

We shall refine this approximation by finding a tight lower bound on the right-hand side of (14). Because collision events are correlated in time, we first consider replacing each conditional probability $P(x_H^\kappa \notin \mathcal{H}_\mathcal{E}(x_R^\kappa) | x_H^s \notin \mathcal{H}_\mathcal{E}(x_R^s), k \leqslant s < \kappa)$ by 1 for all $\kappa \in \{k+1, \ldots, k+K\}$. This effectively lower bounds $\overline{P^{k:k+K}_{\text{coll}}}$ by the worst-case probability of collision at the current time step k:

$$\overline{P^{k:k+K}_{\text{coll}}} \geqslant 1 - P(x_H^k \notin \mathcal{H}_\mathcal{E}(x_R^k)) = P(x_H^k \in \mathcal{H}_\mathcal{E}(x_R^k)) \quad (15)$$

This bound is extremely loose in general, because it completely ignores the possibility of future collision. However, note that probabilities in the product in (14) may be conditioned in any particular order (not necessarily chronological). This commutativity allows us to generate $K - k + 1$ lower bounds of the form $\overline{P^{k:k+K}_{\text{coll}}} \geqslant P(x_H^\kappa \in \mathcal{H}_\mathcal{E}(x_R^\kappa))$ for $\kappa \in \{k, \ldots, k+K\}$. Taking the tightest of all of these bounds, we can obtain an informative, yet quickly computable, approximator for the sought probability:

$$\overline{P^{k:k+K}_{\text{coll}}} \geqslant \max_{\kappa \in \{k:k+K\}} P(x_H^\kappa \in \mathcal{H}_\mathcal{E}(x_R^\kappa)) \approx P^{k:k+K}_{\text{coll}} \quad (16)$$

To summarize, the left inequality in (16) lower-bounds $\overline{P^{k:k+K}_{\text{coll}}}$ with the greatest marginal collision probability at *any point* in the trajectory. On the right-hand side of (16), we take this greatest marginal collision probability as an approximator of the actual probability of collision over the entire trajectory. In effect, we shall approximate $P^{k:k+K}_{\text{coll}}$ with a tight lower bound *of an upper bound*. While this type of approximation may err on the side of optimism, we note that both the robot's ability to replan over time and the fact that the left-hand side of (16) is an *upper bound* on total trajectory collision probability mitigate this potentially underestimated risk.

6.3. Safe online planning under uncertain human predictions

This approximation of collision probability allows the robot to discriminate between valid and invalid candidate trajectories during motion planning. Using the prediction methodology proposed in Section 4, we may quickly generate, at every time t, the marginal probabilities in (16) at each future time $\kappa \in \{k, \ldots, k+K\}$, based on past observations at times $0, \ldots, k$. The planner then computes the instantaneous probability of collision $P(x_H^\kappa \in \mathcal{H}_\mathcal{E}(x_R^\kappa))$ by integrating $P(x_H^\tau | x_H^{0:k})$ over $\mathcal{H}_\mathcal{E}(x_R^\kappa)$, and rejects the candidate point x_R^κ if this probability exceeds P_{th}.

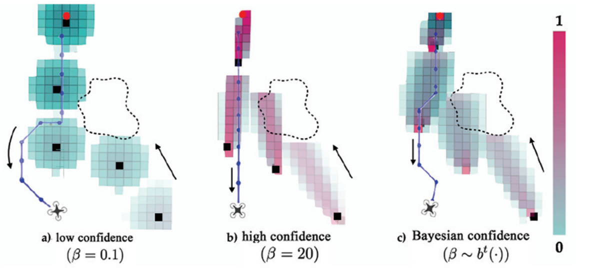

Fig. 4. Scenario from the middle row of Figure 2 visualized with robot's trajectory. When β is low and the robot is not confident, it makes large deviations from its path to accommodate the human. When β is high, the robot refuses to change course and comes dangerously close to the human. With inferred model confidence, the robot balances safety and efficiency with a slight deviation around the human.

Note that for graph-based planners that consider candidate trajectories by generating a graph of time-stamped states, rejecting a candidate edge from this graph is equivalent to rejecting all further trajectories that would contain that edge. This early rejection rule is consistent with the proposed approximation (16) of $P^{k:k+K}_{\text{coll}}$ while preventing unnecessary exploration of candidate trajectories that would ultimately be deemed unsafe.

Throughout operation, the robot follows each planned trajectory using the error feedback controller provided by FaSTrack, which ensures that the robot's high-fidelity state representation s_R and the lower-fidelity state used for planning, x_R, differ by no more than the tracking error bound $\mathcal{E}$. This planning and tracking procedure continues until the robot reaches its desired goal state.

Running example: *Our quadcopter is now required to navigate to a target position shown in Figure 4 without colliding with the human. Our proposed algorithm successfully avoids collisions at all times, replanning to leave greater separation from the human whenever her motion departs from the model. In contrast, robot planning with fixed model confidence is either overly conservative at the expense of time and performance or overly aggressive at the expense of safety.*

7. Connections to reachability analysis

In this section, we present an alternative, complementary analysis of the overall safety properties of the proposed approach to prediction and motion planning. This

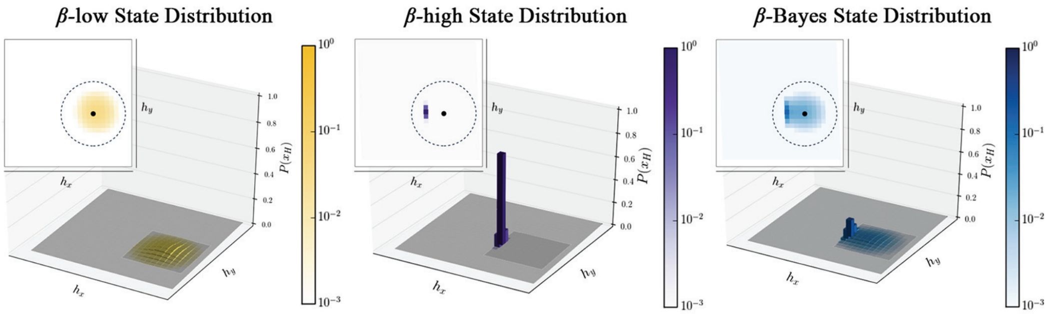

Fig. 5. The human (black dot) is moving west towards a goal. Visualized are the predicted state distributions for 1 second into the future when using low, high, and Bayesian model confidence. Higher-saturation indicates higher likelihood of occupancy. The dashed circle represents the pedestrian's 1 second forward reachable set.

discussion is grounded in the language of reachability theory and worst-case analysis of human motion.

7.1. Forward reachable set

Throughout this section, we frequently refer to the human's time-indexed forward reachable set. We define this set formally in the following.

Definition 1. (Forward reachable set) *For a dynamical system $\dot{x} = f(x, u)$ with state trajectories given by the function $\xi(x(0), t, u(\cdot)) =: x(t)$, the forward reachable set* $\mathrm{FRS}(x, t)$ *of a state x after time t has elapsed is*

$$\mathrm{FRS}(x, t) := \{x' : \exists u(\cdot), x' = \xi(x, t, u(\cdot))\}$$

That is, a state x' is in the forward reachable set of x after time t if it is *reachable* via some applied control signal $u(\cdot)$.

Remark 1. (Recovery of FRS) *For $P_{\text{th}} = 0$ and any finite β, the set of states assigned probability greater than P_{th} is identical to the forward reachable set, up to discretization errors. This is visualized for low, high, and Bayesian model confidence in Figure 5.*

7.2. A sufficient condition for the safety of individual trajectories

In Section 6.2, we construct an approximation to the probability of collision along a trajectory, which we use during motion planning to avoid potentially dangerous states. To make this guarantee of collision avoidance for a motion plan even stronger, it would suffice to ensure that the robot never comes too close to the human's forward reachable set. More precisely, a planned trajectory is safe if $\{x_R(t) + \mathcal{E}\} \cap \mathrm{FRS}(x_H, t) = \emptyset$, for every state $x_R(t)$ along a motion plan generated when the human was at state x_H. The proof of this statement follows directly from the properties of the tracking error bound $\mathcal{E}$ described in Section 6.

While this condition may seem appealing, it is in fact highly restrictive. The requirement of avoiding the full forward reachable set is not always possible in confined spaces; indeed, this was our original motivation for wanting to predict human motion (see Section 3.4). However, despite this shortcoming, the logic behind this sufficient condition for safety provides insight into the effectiveness of our framework.

7.3. Recovering the forward reachable set

Though it will not constitute a formal safety guarantee, we analyze the empirical safety properties of our approach by examining how our predicted state distributions over time relate to forward reachable sets. During operation, our belief over model confidence β evolves to match the degree to which the utility model Q_H explains recent human motion. The "time constant" governing the speed of this evolution may be tuned by the system designer to be arbitrarily fast by choosing the parameter ϵ to be small, as discussed in Section 4.1. Thus, we may safely assume that $b(\beta)$ places high probability mass on small values of β as soon as the robot observes human motion that is not well explained by Q_H.

Figure 6 shows the sets of states with "high enough" ($>P_{\text{th}}$) predicted probability mass overlaid on the human's forward reachable set at time t, which is a circle of radius $v_H t$ centered on x_H for the dynamics in our running example. When β is high (10), we observe that virtually all of the probability mass is concentrated in a small number of states in the direction of motion predicted by our utility model. When β is low (0.05) we observe that the set of states assigned probability above our collision threshold P_{th} occupies a much larger fraction of the reachable set. A typical belief $b(\beta)$ recorded at a moment when the human was roughly moving according to Q_H yields an intermediate set of states.

Figure 7 illustrates the evolution of these sets of states over time, for the unmodeled obstacle example of Section 5.4 in which a pedestrian avoids a spill. Each row corresponds to the predicted state distribution at a particular point in time. Within a row, each column shows the reachable set and the set of states assigned occupancy

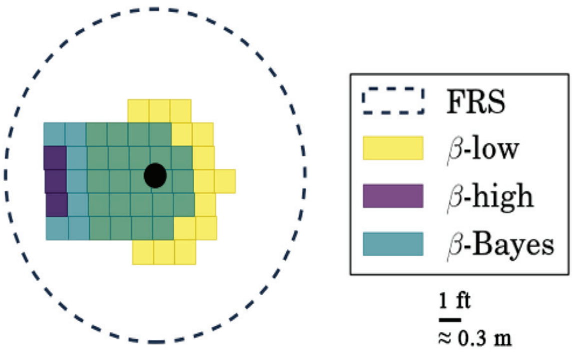

Fig. 6. Visualization of the states with probability greater than or equal to the collision threshold, $P_{\text{th}} = 0.01$. The human's forward reachable set includes the set of states assigned probability greater than P_{th}. We show these "high probability" predicted states for predictors with fixed low and high β, as well as our Bayesian-inferred β.

probability greater than $P_{\text{th}} = 0.01$. The color of each set of states corresponds to the value of β used by the low confidence and high confidence predictors, and the *maximum a posteriori* value of β for the Bayesian confidence predictor. The human's known goal state is marked by a red dot.

Interestingly, as the Bayesian model confidence decreases, which occurs when the pedestrian turns to avoid the spill at $t \approx 6$ s, the predicted state distribution assigns high probability to a relatively large set of states, but unlike the low-β predictor that set of states is oriented toward the known goal. Of course, had $b(\beta)$ placed even more probability mass on lower values of β then the Bayesian confidence predictor would converge to the low confidence one.

In addition, we observe that, within each row as the prediction horizon increases, the area contained within the forward reachable set increases and the fraction of that area contained within the predicted sets decreases. This phenomenon is a direct consequence of our choice of threshold P_{th}. Had we chosen a smaller threshold value, a larger fraction of the forward reachable set would have been occupied by the lower-β predictors.

This observation may be viewed prescriptively. Recalling the sufficient condition for safety of planned trajectories from Section 7.2, if the robot replans every T_{replan} seconds, we may interpret the fraction of $\text{FRS}(\cdot, t + T_{\text{replan}})$ assigned occupancy probability greater than P_{th} by the low-confidence predictor as a rough indicator of the safety of an individual motion plan, robust to worst-case human movement. As this fraction tends toward unity, the robot is more and more likely to be safe. However, for any $P_{\text{th}} > 0$, this fraction approaches zero for $T_{\text{replan}} \uparrow \infty$. This immediately suggests that, if we wish to replan every T_{replan} seconds, we can achieve a particular level of safety as measured by this fraction by choosing an appropriate threshold P_{th}.

In summary, confidence-aware predictions rapidly place high-probability mass on low values of β whenever human motion is not well-explained by utility model Q_H. Whenever this happens, the resulting predictions encompass a larger fraction of the forward reachable set, and in the limit that $P_{\text{th}} \downarrow 0$ we recover the forward reachable set exactly. The larger this fraction, the more closely our approach satisfies the sufficient condition for safety presented in Section 7.2.

8. Hardware demonstration

We implemented confidence-aware human motion prediction (Section 4) and integrated it into a real-time, safe probabilistic motion planner (Section 6), all within the Robot Operating System (ROS) software framework of Quigley et al. (2009). To demonstrate the efficacy of our methods, we tested our work for the quadcopter-avoiding-pedestrian example used for illustration throughout this paper. Human trajectories were recorded as (x, y) positions on the ground plane at roughly 235 Hz by an OptiTrack infrared motion capture system, and we used a Crazyflie 2.0 micro-quadcopter, also tracked by the OptiTrack system.[4]

Figure 4 illustrates the unmodeled obstacle case from Section 5.4, in which the pedestrian turns to avoid a spill on the ground. Using a low model confidence results in motion plans that suddenly and excessively deviate from the ideal straight-line path when the pedestrian turns to avoid the spill. By contrast, the high-confidence predictor consistently predicts that the pedestrian will walk in a straight line to the goal even when they turn; this almost leads to collision, as shown in detail in Figure 8. Our proposed approach for Bayesian model confidence initially assigns high confidence and predicts that the pedestrian will walk straight to the goal, but when they turn to avoid the spill, the predictions become less confident. This causes the quadcopter to make a minor course correction, shown in further detail in Figure 9.

9. Conclusion

When robots operate in complex environments in concert with other agents, safety often depends upon the robot's ability to predict the agents' future actions. While this prediction problem may be tractable in some cases, it can be extremely difficult for agents such as people who act with *intent*. In this paper, we introduce the idea of *confidence-aware* prediction as a natural coping mechanism for predicting the future actions of intent-driven agents. Our approach uses each measurement of the human's state to reason about the accuracy of its internal model of human decision-making. This reasoning about model confidence is expressed compactly as a Bayesian filter over the possible values of a single parameter, β, which controls the entropy of the robot's model of the human's choice of action. In effect, whenever the human's motion is not well-explained by this model, the robot predicts that the human could occupy a larger volume of the state space.

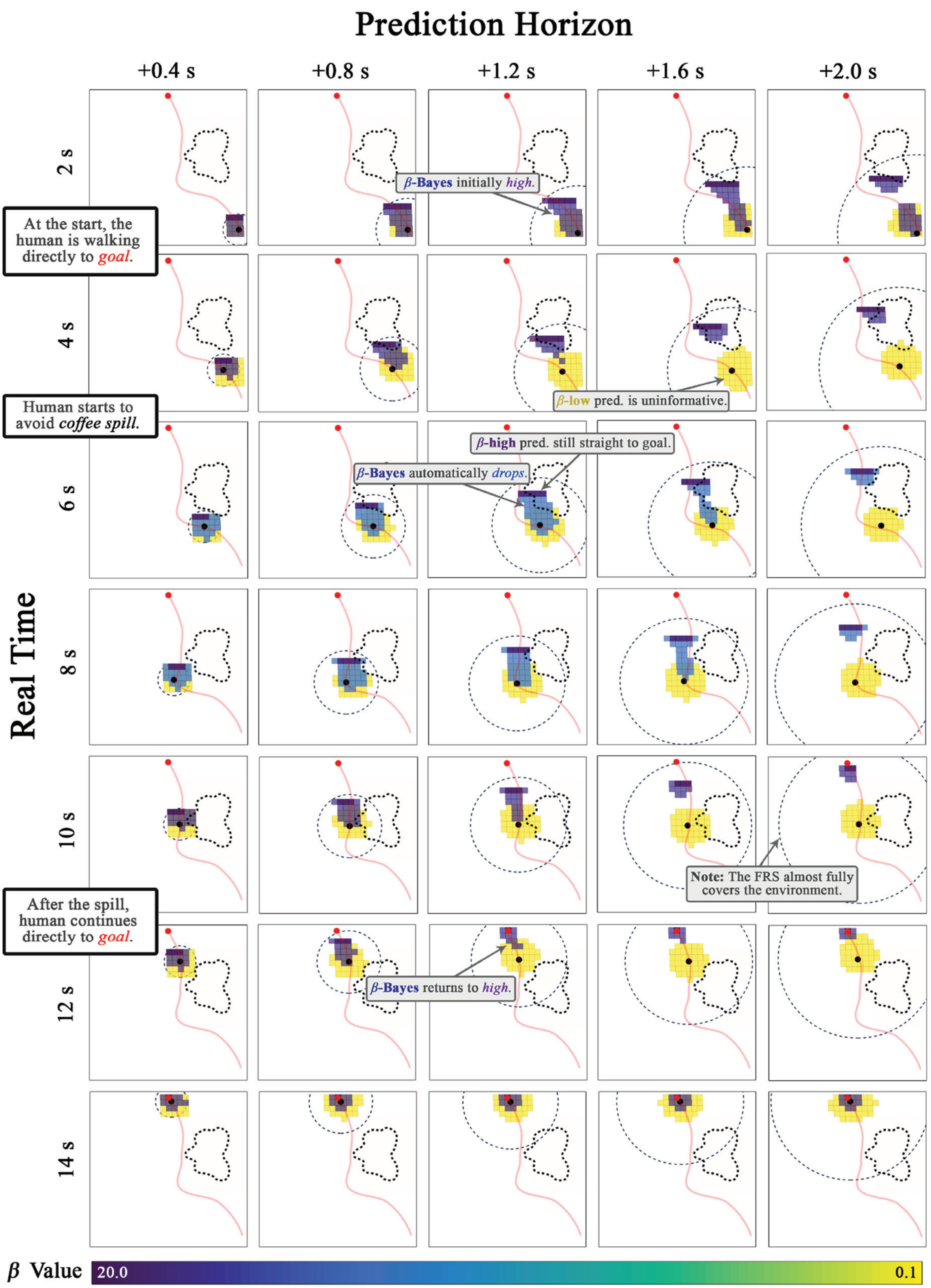

Fig. 7. The human (black dot) is walking towards the known goal (red dot) but has to avoid an unmodeled coffee spill on the ground. Here we show the snapshots of the predictions at various future times (columns) as the human walks around in real time (rows). The visualized states have probability greater than or equal to $P_{\text{th}} = 0.01$. Each panel displays the human prediction under low confidence (in yellow), high confidence (in dark purple), and Bayesian confidence (colored as per the most likely β value), as well as the forward reachable set. The human's actual trajectory is shown in red.

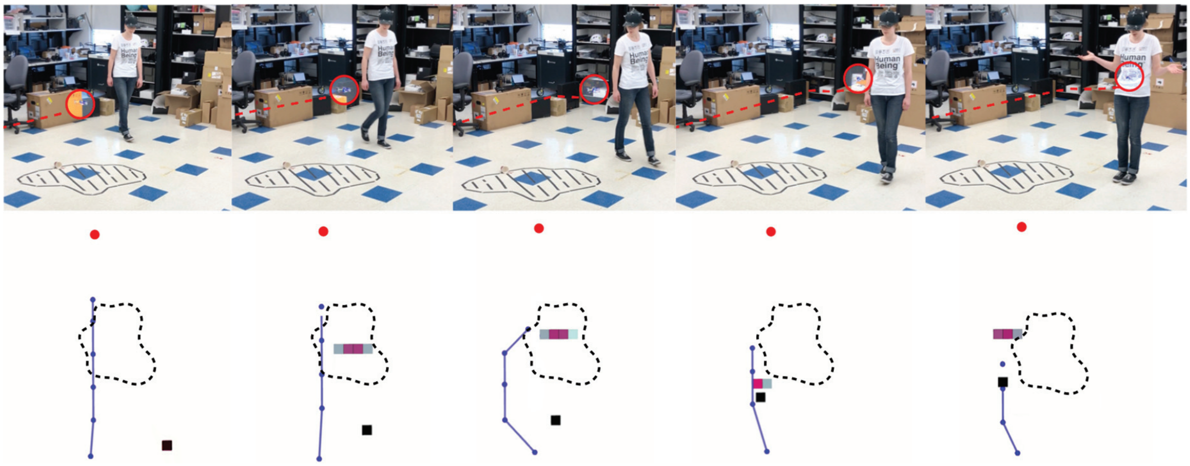

Fig. 8. Predicting with fixed-β (in this case, $\beta = 20$) can yield highly inaccurate predictions (and worse, confidently inaccurate ones). The subsequent motion plans may not be safe; here, poor prediction quality leads to a collision.

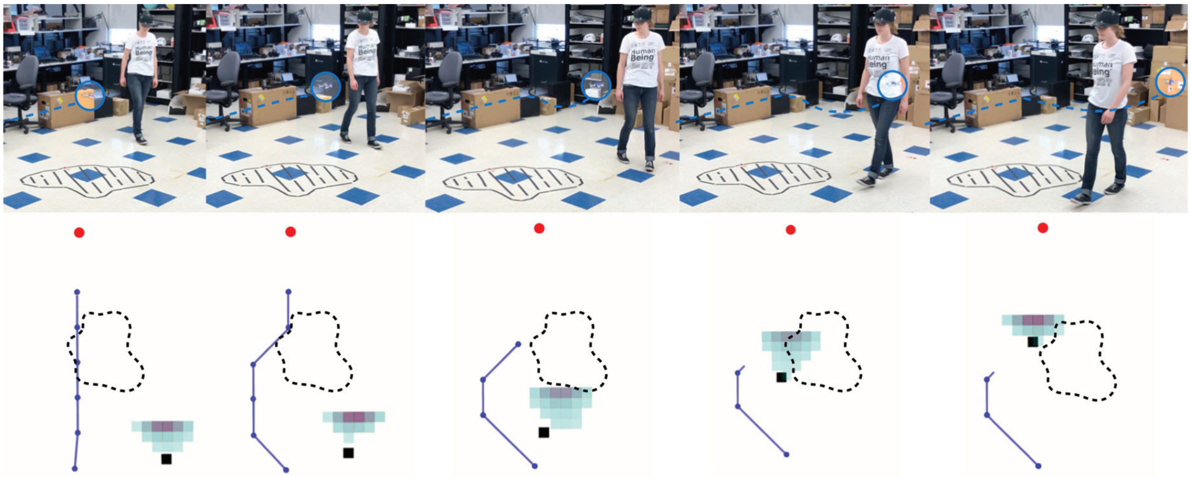

Fig. 9. Inferring β leads to predicted state distributions whose entropy increases whenever the utility model Q_H fails to explain observed human motion. The resulting predictions are more robust to modeling errors, resulting in safer motion plans. Here, the quadcopter successfully avoids the pedestrian even when they turn unexpectedly.

We couple this notion of confidence-aware prediction with a reachability-based robust motion planning algorithm, FaSTrack, which quantifies the robot's ability to track a planned reference trajectory. Using this maximum tracking error allows us to bound an approximation of the probability of collision along planned trajectories. In addition, we present a deeper connection between confidence-aware prediction and forward reachable sets, which provides an alternative explanation of the safety of our approach. We demonstrate the proposed methodology on a ROS-based quadcopter testbed in a motion capture arena.

9.1. Limitations

There are several important limitations of this work, which we summarize and discuss in the following.

9.1.1. State discretization. As presented, our approach to prediction requires a discrete representation of the human's state space. This can be tractable for the relatively simple dynamical models of human motion we consider in this work. Fortunately, one of the strongest attributes of confidence-aware prediction is that it affords a certain degree of robustness to modeling errors by design. Still, our approach is effectively limited to low-order dynamical models.

9.1.2. FaSTrack complexity. FaSTrack provides a strong safety guarantee vis-à-vis the maximum tracking error that could ever exist between a higher-fidelity dynamical model of the robot and a lower-order model used for motion planning. Unfortunately, the computational complexity of finding this maximum tracking error and the corresponding safety controller scales exponentially with the dimension of the high-fidelity model. In some cases, these dynamics are decomposable and analytic solutions exist (e.g., Chen et al., 2018; Fridovich-Keil et al., 2018), and in other cases conservative approximations may be effective (e.g., Chen et al., 2016; Royo et al., 2018).

9.1.3. Boltzmann distributional assumption. We model the human's choice of control input at each time step as an independent, random draw from a Boltzmann distribution (3). This distributional assumption is motivated from the literature in cognitive science and econometrics and is increasingly common in robotics, yet it may not be accurate in all cases. Maintaining an up-to-date model confidence belief $b(\beta)$ can certainly mitigate this inaccuracy, but only at the cost of making excessively conservative predictions.

9.1.4. Safety certification. Our analysis in Section 7 makes connections to forward reachability in an effort to understand the safety properties of our system. As shown, whenever our confidence-aware prediction method detects poor model performance it quickly yields predictions that approximate the human's forward reachable set. Although this approximation is not perfect, and hence we cannot provide a strong safety certificate, the connection to reachability is in some sense prescriptive. That is, it can be used to guide the choice of collision probability threshold P_{th} and replanning frequency. However, even if we could provide a strong guarantee of collision avoidance for a *particular* motion plan, that would not, in general, guarantee that future motion plans would be *recursively safe*. This recursive property is much more general and, unsurprisingly, more difficult to satisfy.

9.2. Future directions

Future work will aim to address each of these shortcomings. We are also interested in extending our methodology for the multi-robot, multi-human setting; our preliminary results are reported by Bajcsy et al. (2018). In addition, we believe that our model confidence inference approach could be integrated with other commonly used probabilistic prediction methods besides the Boltzmann utility model. Finally, we are excited to test our work in hardware in other application spaces, such as manipulation and driving.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is supported by and NSF CAREER award, the Air Force Office of Scientific Research (AFOSR), NSF's CPS FORCES and VeHICal projects, the UC-Philippine-California Advanced Research Institute, the ONR MURI Embedded Humans and the SRC CONIX Center.

Notes

1. This work is an extension of Fisac et al. (2018a), which was originally presented at *Robotics: Science and Systems, 2018*. In this paper, we provide further details of the proposed method, introduce another example to emphasize its generality, and present a connection to reachability analysis.
2. This section is taken from our initial work Fisac et al. (2018a) with minimal modification.
3. To avoid confusion between discrete and continuous time, we shall use superscripts to denote discrete time steps (e.g., x_H^k) and parentheses to denote continuous time (e.g., $x_H(t)$).
4. We note that in a more realistic setting, we would require alternative methods for state estimation using other sensors, such as lidar and/or camera(s). A video recording may be found at https://youtu.be/2ZRGxWknENg.

ORCID iD

David Fridovich-Keil https://orcid.org/0000-0002-5866-6441

References

Amor HB, Neumann G, Kamthe S, Kroemer O and Peters J (2014) Interaction primitives for human–robot cooperation tasks. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2831–2837.

Aoude GS, Luders BD, Joseph JM, Roy N and How JP (2013) Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots* 35(1): 51–76.

Bai H, Cai S, Ye N, Hsu D and Lee WS (2015) Intention-aware online pomdp planning for autonomous driving in a crowd. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 454–460.

Bajcsy A, Herbert SL, Fridovich-Keil D, et al. (2018) A scalable framework for real-time multi-robot, multi-human collision avoidance. *arXiv preprint arXiv:1811.05929*.

Baker CL, Tenenbaum JB and Saxe RR (2007) Goal inference as inverse planning. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 29.

Bandyopadhyay T, Won KS, Frazzoli E, Hsu D, Lee WS and Rus D (2013) Intention-aware motion planning. In: *Algorithmic Foundations of Robotics X*. New York: Springer, pp. 475–491.

Bansal S, Chen M, Herbert S and Tomlin CJ (2017) Hamilton–Jacobi reachability: A brief overview and recent advances. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, pp. 2242–2253.

Belkhouche F (2009) Reactive path planning in a dynamic environment. *IEEE Transactions on Robotics* 25(4): 902–911.

Chen M, Herbert S and Tomlin CJ (2016) Fast reachable set approximations via state decoupling disturbances. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 191–196.

Chen M, Herbert SL, Vashishtha M, Bansal S and Tomlin CJ (2018) Decomposition of reachable sets and tubes for a class of nonlinear systems. *IEEE Transactions on Automatic Control* 63(11): 3675–3688.

Ding H, Reißig G, Wijaya K, Bortot D, Bengler K and Stursberg O (2011) Human arm motion modeling and long-term prediction for safe and efficient human–robot-interaction. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5875–5880.

Driggs-Campbell K, Dong R and Bajcsy R (2018) Robust, informative human-in-the-loop predictions via empirical reachable sets. *IEEE Transactions on Intelligent Vehicles* 3(3): 300–309.

Evans LC and Souganidis PE (1984) Differential games and representation formulas for solutions of Hamilton–Jacobi–Isaacs equations. *Indiana University Mathematics Journal* 33(5): 773–797.

Finn C, Levine S and Abbeel P (2016) Guided cost learning: Deep inverse optimal control via policy optimization. In: *International Conference on Machine Learning*, pp. 49–58.

Fisac JF, Bajcsy A, Herbert SL, et al. (2018a) Probabilistically safe robot planning with confidence-based human predictions. In: *Robotics: Science and Systems*.

Fisac JF, Bronstein E, Stefansson E, Sadigh D, Sastry SS and Dragan AD (2018b) Hierarchical game-theoretic planning for autonomous vehicles. *arXiv preprint arXiv:1810.05766*.

Fridovich-Keil D, Herbert SL, Fisac JF, Deglurkar S and Tomlin CJ (2018) Planning, fast and slow: A framework for adaptive real-time safe trajectory planning. In: *IEEE Conference on Robotics and Automation*.

Hart PE, Nilsson NJ and Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4(2): 100–107.

Hawkins KP, Vo N, Bansal S and Bobick AF (2013) Probabilistic human action prediction and wait-sensitive planning for responsive human–robot collaboration. In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 499–506.

Herbert SL, Chen M, Han S, Bansal S, Fisac JF and Tomlin CJ (2017) Fastrack: a modular framework for fast and guaranteed safe motion planning. In: *IEEE Conference on Decision and Control*.

Karaman S and Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30(7): 846–894.

Kochenderfer MJ, Edwards MW, Espindle LP, Kuchar JK and Griffith JD (2010) Airspace encounter models for estimating collision risk. *Journal of Guidance, Control, and Dynamics* 33(2): 487–499.

Koppula HS and Saxena A (2013) Anticipating human activities for reactive robotic response. In: *IROS*, p. 2071.

Lasota PA and Shah JA (2015) Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human Factors* 57(1): 21–33.

Luce RD (1959) *Individual Choice Behavior: A Theoretical Analysis*. New York: John Wiley & Sons.

Mitchell I (2009) *A Toolbox of Level Set Methods*. http://people.cs.ubc.ca/∼*mitchell*/ToolboxLS/index.html.

Mitchell IM, Bayen AM and Tomlin CJ (2005) A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control* 50(7): 947–957.

Ng AY and Russell SJ (2000) Algorithms for inverse reinforcement learning. In: *ICML*, pp. 663–670.

Quigley M, Conley K, Gerkey BP, et al. (2009) ROS: An open-source robot operating system. In: *ICRA Workshop on Open Source Software*.

Royo VR, Fridovich-Keil D, Herbert S and Tomlin CJ (2018) Classification-based approximate reachability with guarantees applied to safe trajectory tracking. *arXiv preprint arXiv:1803.03237*.

Sadigh D, Sastry S, Seshia SA and Dragan AD (2016) Planning for autonomous cars that leverage effects on human actions. In: *Robotics: Science and Systems (RSS)* .

Schmerling E, Leung K, Vollprecht W and Pavone M (2017) Multimodal probabilistic model-based planning for human–robot interaction. *arXiv preprint arXiv:1710.09483*.

Trautman P and Krause A (2010) Unfreezing the robot: Navigation in dense, interacting crowds. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 797–803.

Von Neumann J and Morgenstern O (1945) *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press.

Wang Z, Spica R and Schwager M (2019) Game theoretic motion planning for multi-robot racing. In: *Proceedings of the International Symposium on Distributed Autonomous Robotics Systems (DARS)*, pp. 225–238.

Ziebart BD, Maas AL, Bagnell JA and Dey AK (2008) Maximum entropy inverse reinforcement learning. In: *AAAI*.

Ziebart BD, Ratliff N, Gallagher G, et al. (2009) Planning-based prediction for pedestrians. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*. IEEE, pp. 3931–3936.