

A Data-driven Framework for Proactive Intention-Aware Motion Planning of a Robot in a Human Environment

Rahul Peddi, Carmelo Di Franco, Shijie Gao, and Nicola Bezzo

Abstract—For safe and efficient human-robot interaction, a robot needs to predict and understand the intentions of humans who share the same space. Mobile robots are traditionally built to be *reactive*, moving in unnatural ways without following social protocol, hence forcing people to behave very differently from human-human interaction rules, which can be overcome if robots instead were *proactive*. In this paper, we build an intention-aware proactive motion planning strategy for mobile robots that coexist with multiple humans. We propose a framework that uses Hidden Markov Model (HMM) theory with a history of observations to: i) predict future states and estimate the likelihood that humans will cross the path of a robot, and ii) concurrently learn, update, and improve the predictive model with new observations at run-time. Stochastic reachability analysis is proposed to identify multiple possibilities of future states and a control scheme that leverages temporal virtual physics inspired by spring-mass systems is proposed to enable safe proactive motion planning. The proposed approach is validated with simulations and experiments involving an unmanned ground vehicle (UGV) performing go-to-goal operations in the presence of multiple humans, demonstrating improved performance and effectiveness of online learning when compared to reactive obstacle avoidance approaches.

I. INTRODUCTION

Autonomous mobile robots that share space with humans have become increasingly popular in recent years; we found such robots delivering packages on crowded sidewalks, working in warehouses alongside human workers, or even simply vacuuming a busy household. In many cases, these robots perform their missions by treating surrounding humans as stationary obstacles. People, in turn, are expected to work around and learn to adapt to these robots often operating in unnatural ways. On the other hand, people interact and cooperate with one another seamlessly. This is because humans not only implicitly understand others' intentions, but they also implicitly communicate their intentions by changing their behavior proactively, often well in advance of a possible collision.

In this paper, we address the problem of enabling proactive intention-aware motion on a robot that coexists with humans in environments like airports, train stations, labs, and offices. In our work, *proactive intention-aware motion* refers to robot motion that accommodates the human's future motion while maintaining the robot's desired goal.

In this work, we leverage Hidden Markov Model (HMM) theory to collect observations and emission matrices that are used to predict and understand future human intentions, i.e., where the humans plan to go in the future. Similar to other learning enabled techniques, we use a history of



Fig. 1. Pictorial representation of the motivation behind this work. Our proposed approach computes temporal stochastic reachable sets and plans motion that proactively accommodates human intentions.

observations to make predictions, however in our proposed approach, new observations are considered at run-time to improve and refine predictions, and rapidly incorporate new or unexpected behaviors. Surrounding humans' future state predictions are used to build stochastic reachable sets and to plan proactive actions. Specifically, we compute the control inputs using temporal virtual physics based on spring-mass systems to avoid the generated reachable sets.

The rest of this paper is organized as follows: in Section II, we discuss relevant work in this field, and in Section III, we formally define the problem. In Section IV, we describe each component of our approach. In V, we present our results through simulations and experiments. Lastly, we present conclusions and future work in Section VI.

II. RELATED WORK

With technological advances in mobile robotics, there has been widespread interest in enabling robots to navigate human environments the way a human does, that is, in a proactive and socially acceptable manner. While advances in dynamic obstacle avoidance, like human avoidance, suffer from the *freezing robot* problem, which is when the robot sees no path forward and is “frozen” for the remainder of the operation [1], [2], recent works have shown progress in reducing this problem. In [3], [4], the authors identify safe motion planning methods using graph search and directional rules, given a priori assumptions of human trajectories based on heading and velocity. These assumptions, however, do not account for uncertainty or rapid changes that can be seen in human motion. The authors in [5] use principles of virtual physics to develop “social-force” model, which produces attractive and repulsive forces based on personal space and other qualitative attributes of human behavior. This model does not predict the future actions of people, and suffers if there is an unexpected violation of the “social force” model by an uncontrolled agent. In [6], [7], authors present

Rahul Peddi, Carmelo Di Franco, Shijie Gao, and Nicola Bezzo are with the Department of Systems and Information Engineering and the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA. Email: {rp3cy, cd8gm, sg9dn, nb6be}@virginia.edu

reachability-based and confidence-based approaches that do consist of an online update, but rely on simplified models of human motion, and suffers from applicability and scalability issues as a result. The notion of reachability for predictions, however, remains useful. In this paper, we take into account observations of human behavior in the presence of a moving robot to create a model, and obtain probabilistic predictions and stochastic reachable sets that describe possible future human motion.

Machine learning based approaches that also aim to enable socially acceptable motion for robots include [8], where the authors use recurrent neural networks (RNNs) or long short term memory networks (LSTMs) to predict future motion of humans. While these methods are effective predictive models for human trajectories, they suffer under an unknown set of inputs, as they are subject to vanishing or exploding gradients. Authors in [9] use deep reinforcement learning (DRL) to attain socially acceptable behaviors, and in [10], the authors use DNNs to achieve similar results. While the RNN, DRL, and DNN based approaches demonstrate good robot behaviors, none are able to improve at run-time. The approach presented in this paper on the other hand, collects the training observations and emission probabilities, so that it can easily identify and learn from new observations at run-time, improving the predictive model at every iteration.

III. PROBLEM FORMULATION

Consider a mobile robot tasked to go from an initial position q_0 to a goal q_g while negotiating and accommodating its motion with surrounding actors, in particular humans, $h_i \in S_h(t)$, $i = 1, \dots, n_h$, where $S_h(t)$ is a time varying set of n_h humans in sensing range with the robot. Each human in the environment is following an unknown trajectory, and we assume the robot is capable of localization and static obstacle and human detection. The dynamics of the robot can be represented in the typical state space form, $\dot{x} = f(x, u)$ where $x \in \mathbb{R}^n$ is the state and $u \in \mathbb{R}^m$ is the control input. With such premises, the robot has the objective to predict the intended motion of humans in its sensing range and proactively plan its motion to minimize the human deviations caused by the robot. In doing that, the robot is also implicitly communicating with the humans to acknowledge that it understands their intention. Formally, the problem is:

Problem 1: Intention Aware Proactive Motion Planning and Control. Consider a robot navigating in an environment with humans. Given a set of observed humans S_h , the objective is to predict their reachable states $\mathcal{R}_h(t)$ at run-time over an horizon H , find a policy to anticipate future robot reactions, and plan a trajectory to minimize human deviating maneuvers created by the robot, such that:

$$\|q(t) - q_i(t)\| > \delta, \forall i \in S_h(t), t \geq 0 \quad (1)$$

where $q(t)$ and $q_i(t)$ are the position of the robot and the i^{th} human at time t respectively, and δ is a minimum safe distance.

A secondary problem we investigate is how to improve predictions over time. To this end we use the theory of Hidden Markov Model (HMM) to collect a history of observations and emission probabilities that are constantly updated online to consider run-time behavior that either was never

observed during training or to reinforce/change the expected future predictions.

IV. PREDICTION AND PLANNING FRAMEWORK

In this section we describe the framework adopted for prediction and control of a robot in a human environment. Specifically, we follow the architecture depicted in the block diagram in Fig. 2.

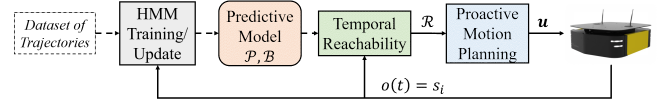


Fig. 2. Block diagram of the framework for human intention prediction and proactive motion planning.

At the core of our framework we consider a HMM that is trained using offline data and constantly updated at run-time for more accurate predictions and to include new behaviors. The predictive model generated by HMM training consists of a set of state observations $\mathcal{O}$ and emission matrix $\mathcal{B}$. Prediction is then executed by building a temporal reachable set $\mathcal{R}$ over a finite time horizon to include future human states and the associated probabilities.

Then, the motion planner that we propose here will consider this prediction to find reactions (deviations from the desired trajectory) of the robot at different time steps and proactively adjust the robot actions toward the point in the trajectory where a deviation was predicted. At run-time new observations are added to $\mathcal{O}$ and are then used to update the HMM, improving and refining $\mathcal{B}$, and thus improving future predictions and inferences. In the next sections, we describe in detail each component of our framework.

A. HMM-based Training

To train the predictive model for human intentions we propose a Hidden Markov Model (HMM) [11] which can be described by the tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{C}, \mathcal{G}, \mathcal{P}, \mathcal{B} \rangle$ where:

- $\mathcal{S} \in \mathbb{R}^N$ is the state space of the system which includes a finite set of unique states $s_i \in \mathcal{S}$, $i = 1, \dots, N$ that can be visited by the system.
- $\mathcal{O} \in \mathbb{R}^T$ is a finite set of observations $o(t) \in \mathcal{O}$ collected over a finite past time horizon T , i.e., $\mathcal{O} = \{o(t-T), o(t-T+1), \dots, o(t)\}$ and such that $o(t) = s_i \in \mathcal{S}$. Note that the states observed during T , $\mathcal{S}' \in \mathbb{R}^n$ are a subset of $\mathcal{S}$, $\mathcal{S}' \subseteq \mathcal{S}$ and $n \leq N$.
- $\mathcal{C} \in \mathbb{R}^T$ is the finite set of emissions, or inferences $c(t)$ that relate to the action taken each state, and $\mathcal{C} = \{c(t-T), c(t-T+1), \dots, c(t)\}$.
- $\mathcal{G} \in \mathbb{R}^M$ is a finite set of M unique inferences that $\mathcal{C}$ can obtain, and $g_k \in \mathcal{G}$, where $k = 1, \dots, M$, with $M \in \mathbb{N}$.
- $\mathcal{P} \in \mathbb{R}^{N \times N}$ is a transition probability matrix, that describes the probability of entering a certain state, $s_j \in \mathcal{S}'$, while currently in observed state $s_i \in \mathcal{S}'$, denoted as $P(s_i \rightarrow s_j)$.
- $\mathcal{B} \in \mathbb{R}^{N \times M}$ is the emission matrix, which lists the probability b_{ik} of obtaining emission g_k given state s_i :

$$b_{ik} = P(g_k(t+1)|s_i(t)) \quad (2)$$

where $i = 1, \dots, N$. Emission probabilities are initialized as $1/M$, and are calculated as follows:

$$b_{ik} = n_{g_{ik}} / n_{g_i^*} \quad (3)$$

where $n_{g_{ik}} \leq n_{g_i^*} \leq T$. Then, the emission matrix is

$$\mathcal{B} = \begin{bmatrix} b_{11} & \dots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{N1} & & b_{NM} \end{bmatrix} \quad (4)$$

Differently from a traditional HMM, the “hidden” part applies to the emissions, rather than the states, which are directly observable (i.e., measurable) and therefore, have observable transitions. Another key difference is that we utilize the set of observations $\mathcal{O}$, to make informed predictions for stochastic reachability in addition to the emission matrix, which provides important behavioral information used for motion planning. Because of the human-robot interaction problem we consider, the specific states recorded in $\mathcal{O}$ are:

$$\mathbf{s} = [d_x \quad d_y \quad \theta] \quad (5)$$

where d_x , d_y , and θ are the relative x - y positions and heading of a person in the robot local frame, respectively.

On the other hand, the emissions will capture qualitative inferences on how a human will behave when approaching the robot: specifically in this work we are interested in predicting whether a human starting from a state $\mathbf{s}_i$ will **cross** or **not cross** the robot’s path at some point in the future during the operation, which impacts the way we plan robot motion; the goal is to accommodate more to a person who is crossing the robot’s path.

The initial offline HMM training shown in Fig. 3 is executed on a set of trajectories specifically designed to capture the accommodating and avoiding behaviors of a humans moving from random initial positions to random goals in the environment around a robot that ignores the human and just follows a straight path starting from $(0, -2.5)$ m to $(0, 3)$ m at $v = 0.5$ m/s. Collisions are not included in our training set because we assume that people in general do not behave adversarially, and therefore will not intentionally try to collide with the robot. The state vector (5) is discretized to prevent an infinite or exploding state-space. The dimension and discretization of the state space are selected based on the capabilities of the robot. A larger state space and finer discretization results in better approximation, but at an increased computational complexity.

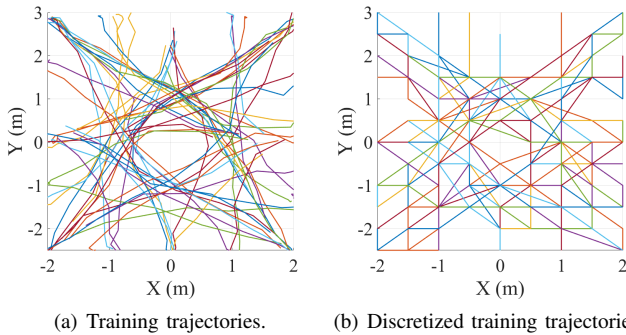


Fig. 3. (a) Set of real trajectories used for the training process in the experiments. (b) Discretized trajectories according to a 0.5m-spaced grid.

At run-time, observations are rounded to the nearest state in the discrete state space. Then, future state predictions and reachable sets are computed with the approximated state, and are geometrically transformed back to the observed state.

Executing the training procedure creates a set of observations $\mathcal{O}$ and emission matrix $\mathcal{B}$, which serve as the predictive model for the likelihood of future states and implicit behaviors given a new state observation at run-time.

B. Online Prediction and Stochastic Reachability

At run-time, given an observation of a person’s state, $o(t) = \mathbf{s}_i \in \mathcal{S}'$, we predict all possible future states and associated probabilities over a finite horizon H to enable proactive motion planning. A low H (e.g., $H = 1$ s) can be ineffective for proactive motion planning resulting in mostly reactive behaviors, while a very large H can be wasteful, as uncertainties can grow too large with time in continually evolving and unstructured environments.

To obtain the future states and probabilities over the selected H , we propose to use Reachability Analysis (RA), which is the process of computing the set of all reachable states for a system by taking into account its dynamic model and state transitions over a future horizon H [12]. The collection of all reachable states at a certain future time forms a *reachable set*, while a *reachable tube* is a temporal sequence of reachable sets. In addition to reachable states, a *stochastic reachable set* includes probabilities associated to each state [6], but generating such stochastic reachable sets can become computationally complex [13].

In this work, stochastic reachability analysis is performed using the finite set of observations, $\mathcal{O}$. At run-time, we consider the future states and associated conditional probabilities of all the paths observed during training originating from the observed state at time t , $\mathbf{s}_i(t)$ over an interval $[t+1, t+H]$. The probability of any reachable future state $\mathbf{s}_j \in \mathcal{S}$ is given by $p(\mathbf{s}_j(t+1)|\mathbf{s}_i(t))$ and this is computed for any state along the path conditioned on the previous states. To limit the computational complexity of such approach, we maintain a list of the N_h most recent paths initiating from each state, obtaining a maximum of $N \times N_h$ trajectories that are used to perform such prediction. In this way, we also remove old and obsolete data and consider only the most recent data in our prediction.

The collection of all reachable paths for a human in state $\mathbf{s}_i$ over the entire horizon forms the reachable tube, $\mathcal{R}_{\mathbf{s}_i}^h \in \mathbb{R}^{n_{\mathbf{s}_i} \times H}$,

$$\mathcal{R}_{\mathbf{s}_i}^h = \begin{bmatrix} \mathbf{s}_1(t+1) & \dots & \mathbf{s}_1(t+H) \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{n_{\mathbf{s}_i}}(t+1) & \dots & \mathbf{s}_{n_{\mathbf{s}_i}}(t+H) \end{bmatrix} \quad (6)$$

where each column of $\mathcal{R}_{\mathbf{s}_i}^h$ consists of reachable states at the respective time-step. Then, the probabilities of each temporal future state $\mathbf{s}_j(t+\tau), \forall \mathbf{s}_j(t+\tau) \in \mathcal{R}_{\mathbf{s}_i}^h$ are computed by counting its occurrences as follows:

$$p(\mathbf{s}_j(t+\tau)) = n_{\mathbf{s}_j(t+\tau)} / n_{\mathbf{s}_i} \quad (7)$$

where $n_{\mathbf{s}_j(t+\tau)}$ is the number of paths that contain state $\mathbf{s}_j$ at time $t+\tau$, and $\tau \in [1, H]$. We finally consider the emission matrix, $\mathcal{B}$, to infer the likelihood that each state

$s_j \in \mathcal{R}_{s_i}^h$ in the reachable tube will lead to a crossing behavior: $p_e(s_j) = \mathcal{B}_{s_j,1}$. Reachable tubes for each human and associated probabilities are concatenated and stored to create one reachable tube, $\mathcal{R} \in \mathbb{R}^{n \times H}$, that encompasses all n possible paths for all sensed humans. Then, $\mathcal{R}$ is deconstructed into temporal reachable sets to be used for motion planning:

$$\mathcal{R}(t+\tau) = \begin{bmatrix} s_1(t+\tau-1) & s_1(t+\tau) \\ \vdots & \vdots \\ s_n(t+\tau-1) & s_n(t+\tau) \end{bmatrix} \quad (8)$$

where $\tau \in [1, H]$, and $\mathcal{R}(t+\tau) \in \mathbb{R}^{n \times 2}$. Note that a temporal reachable set $\mathcal{R}(t+\tau)$ includes reachable states at $\tau-1$ and at τ to capture the motion between two consecutive time steps, to prevent cross collisions in paths between discrete states. An example of the proposed reachability analysis over horizon $H = 5s$ for a human is shown in Fig 4.

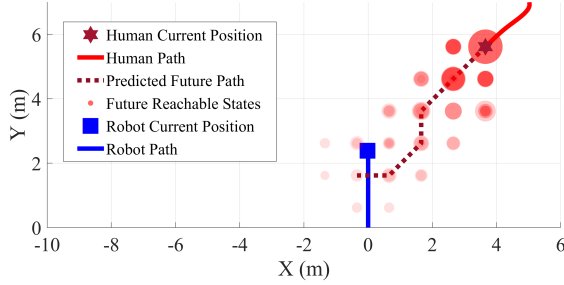


Fig. 4. Stochastic reachable set of a human in the robot's sensing range. The dotted line shows the most likely future path for the human. Red markers represent reachable states, and the color fades temporally along the horizon, with the lightest at $t+H$. The probability of reachable states is shown by the size of the markers, which decrease as probability decreases.

The most likely future path, connected by the dotted line, is nearly consistent with a linear path. Considering this path alone provides little advantage over well-known approaches [2], [4], that explicitly assume that dynamic obstacles will maintain their current speed and/or direction with minor uncertainty. On the other hand, stochastic reachability in our approach captures uncertainties and irregularities that exist in human motion, addressing the possibility that a human can enter states outside of the most likely path.

C. Robot Motion Planning

In this section, we present our motion planning technique that takes into account the temporal stochastic reachable sets $\mathcal{R}(t+\tau)$ developed in Section IV-B to proactively avoid and accommodate future motion of surrounding humans. We utilize virtual spring-mass-damper interactions [14] to generate robot motion that avoids and accommodates humans future states while reaching the goal.

The repulsive spring force directs the robot from its position $\mathbf{q}_r(t)$ away from the temporal reachable states $s_j \in \mathcal{R}(t+\tau)$. The springs for each reachable state have constants $k(s_j) = p(s_j)p_e(s_j)$ that depend on the state probabilities output from stochastic reachability analysis, $p(s_j)$, and the crossing probability of the state from the emission matrix $p_e(s_j) \in \mathcal{B}$. The inclusion of crossing probability creates a stronger reaction to a crossing person and vice versa. The extension of a repulsive spring for each reachable state is

defined as $d_{ho} = l_h(s_j) - l_o$, where $l_h(s_j) = \|\mathbf{q}_r(t) - \mathbf{q}(s_j)\|$ is the distance from the robot to the position of each temporal reachable state, and l_o is a safe distance to maintain between robot and human. In crowded situations, even with good predictions, a human may get very close to the robot, compromising the safety of the operation. In such cases, we introduce a fail-safe distance, l_d , subject to $l_d < l_o$, which will produce stronger repulsive forces.

Then, the repulsive spring force at each time $t+\tau$, with $\tau \in [1, H]$, for each state $s_j \in \mathcal{R}(t+\tau)$ is computed as:

$$\mathbf{u}_{rep,(s_j)}(t+\tau) = \begin{cases} k(s_j)d_{ho}\vec{\mathbf{d}}_{ho}, & l_d < l_h(s_j) \leq l_o \\ d_{hd}\vec{\mathbf{d}}_{hd}, & l_h(s_j) \leq l_d \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $\vec{\mathbf{d}}_h$ indicates a unit vector in the direction away from the human's reachable state. Note that probabilities are left out of the case $l_h(s_j) \leq l_d$, and the spring stiffness is simply 1, which is the maximum value $k(s_j)$ can take, creating the strongest repulsive forces.

The attractive force directs the robot from its position $\mathbf{q}_r(t+\tau)$ towards the goal, $\mathbf{q}_g = [x_g, y_g]^T$, and is computed as follows:

$$\mathbf{u}_{att}(t+\tau) = k_{att}(\|\mathbf{q}_r(t+\tau) - \mathbf{q}_g\|)\vec{\mathbf{d}}_g \quad (10)$$

where $\vec{\mathbf{d}}_g$ is the unit vector directed towards the goal and k_{att} is the spring constant. Here, the distance to goal is used as the extension of the spring, as the ultimate target of the robot is to reach the goal. The summation of attractive and repulsive forces yields an input for the robot at each time:

$$\mathbf{u}(t+\tau) = \mathbf{u}_{att}(t+\tau) + \sum_{j=1}^n \mathbf{u}_{rep,q(s_j)}(t+\tau) - c_d v(t) \quad (11)$$

where n is the number of states reachable at $t+\tau$ and $c_d v(t)$ is the spring damping effect. At each time-step, the input is used to compute the robot's next position, and the entire procedure is repeated for the remainder of the horizon, resulting in a time series of inputs for the robot.

Instead of applying the first input in the predicted series (reactive approach), we proactively replan the trajectory to accommodate the future states by finding the first time τ^* in which the robot deviates from a direct path to the goal,

$$\tau^* = \arg \min_{\tau} (\mathbf{u}(t+\tau)), \quad (12)$$

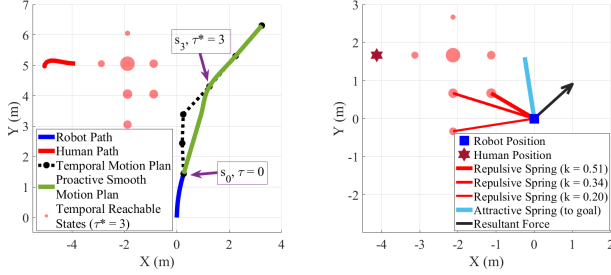
$$\text{s.t. } \mathbf{u}(t+\tau) \neq \mathbf{u}_{att}(t+\tau)$$

Then, we compute a new set of inputs that directly sends the vehicle towards the planned position at τ^* , accommodating the deviation caused by future states of surrounding humans:

$$\mathbf{u}'(t+\tau) = \frac{\sum_{\tau=1}^{\tau^*} \mathbf{u}(t+\tau)}{\tau^*}, \quad \tau \in [1, \tau^*] \quad (13)$$

where the numerator is the vector sum of the inputs between 1 and τ^* . Dividing this resultant by τ^* provides the value of the inputs to use to move directly toward the deviated position at τ^* . When $\tau^* < H$, the previously calculated inputs, $\mathbf{u}(t+\tau)$, $\tau \in (\tau^*, H]$, are included to populate the complete series of inputs for the horizon, which is then smoothed with cubic spline interpolation. The final smoothed

trajectory is sent to the robot, and is replanned at every time-step, as the presence and motion of surrounding humans is constantly changing and evolving. Fig. 5(a) shows an example of such motion plan. The robot is in state s_0 at $\tau = 0$ and wants to reach a goal g . After running the prediction, a deviation from the planned straight line trajectory occurs at $\tau^* = 3$ resulting in s_3 . To be proactive, the robot replans its trajectory to go directly to s_3 from s_0 . Fig. 5(b) shows the predicted action (black arrow computed with (11)) of the robot as a resultant of the repulsive spring forces (red lines) and attractive force (light blue line) pushing the robot away toward the right side of its desired trajectory. Note that, per (9), only the reachable states within the range l_o impact the robot. Taking into account the probabilities



(a) Overall plan of the robot at $\tau = 0$ (b) The springs formed with the pre-dicted human positions at $\tau^* = 3$ and the resultant motion, seen from the local frame of the robot.

Fig. 5. Motion plan prediction example.

of reachable states and emission probabilities ensures that more likely or crossing states create a stronger reaction from the robot. Considering only temporal reachable sets to plan at each time provides an advantage over a potentially circuitous or “frozen” paths generated by approaches that consider the entire range of future predictions at one time [2]. In addition, replanning inputs based on future deviation generates robot behavior that accommodates humans’ future intentions, creating an advantage over reactive dynamic obstacle avoidance approaches that only consider the current positions of surrounding humans.

D. Online Model Updates

Our motion planner is predictive and proactive with respect to future states of humans, but due to the dynamic nature of the environment, many new behaviors can be observed online. If the predictive model can be updated online, the expectation that a new behavior could occur at a later time can be exploited to improve predictions, and by extension, motion planning.

To this end, we propose online updates of observations and the emission matrix used for prediction and planning. The set of observations $\mathcal{O}$, main input to our stochastic reachability, is updated with a new observed state transition at run-time $o(t) = s_j$. The emission matrix, $\mathcal{B}$, a key part of our motion planning, is updated using the procedure described in (2)-(4), by incrementing the instances of emissions and occurrences of the state, resulting in an updated $\mathcal{B}'$ that includes behavioral information from the new observations. Because the updated matrix is bounded by the size of the state space N and emission space M , and the update procedure is an element-wise operation, the worst-case computational

complexity cannot exceed $O(NM)$ [15]. This update can occur within one iteration of robot operation, rendering the updated predictive model immediately usable.

Updating and learning online rapidly is unique to our approach, as most learning-based techniques, such as DNNs [10], consist of training complex connections between inputs and outputs that cannot necessarily be accessed [16], making them difficult to update without fully retraining the network, which can be computationally intensive and time consuming.

V. SIMULATION AND EXPERIMENTAL RESULTS

The case study investigated in this work and presented in this section consists of a robotic vehicle performing a go-to-goal operation in the presence of moving humans. The robot is expected to predict the intentions of humans, accommodate, and avoid them as it completes its mission.

A. Simulations

In the following simulation the robot is tasked to move in a 11m by 12m environment from an initial point (0,0)m to a goal at (0,12)m while navigating around two humans moving in unknown trajectories. The simulated robot trajectories and the distance maintained between the robot and surrounding humans are shown in Fig. 6. Specifically, in Figs. 6(a) and 6(b), we compare our predictive approach before and after model updates respectively: In Fig. 6(b) the same simulation was run 20 times updating the model after every run following the approach described above demonstrating that over time the behavior improves, becoming smoother if similar behaviors are recorded several times. In Fig. 6(c) we show the results for a reactive-based planner, in which the robot is pushed away from the humans once they are in close proximity, resulting in motion away from the goal.

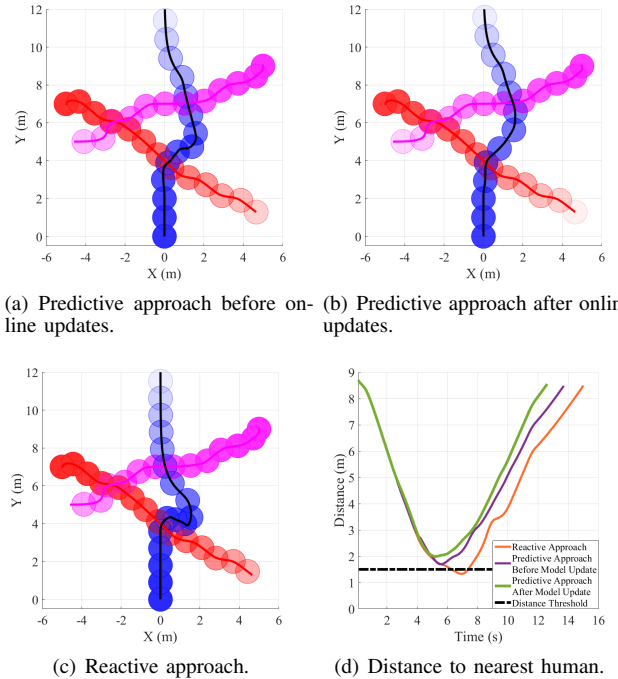


Fig. 6. Scenarios with a robot (blue markers, black line) navigating to its goal in the presence of two people (magenta and red markers and lines). The markers fade as time increases and the actors reach their goals. The distance threshold is 1.5m.

In these simulations, the maximum velocity of the robot, $v_{max} = 1\text{m/s}$, was chosen to approximate average human walking speed, which is usually between 0.7m/s and 1.4m/s . The resting length is set $l_o = 2\text{m}$, and the fail-safe distance threshold is set to $l_d = 1.5\text{m}$. The time horizon for prediction and control was set to $H = 5\text{s}$, and the robot has a 5m radius.

The online update procedure increases probabilities of previously observed future states, so that surrounding reachable states exert weaker repulsive forces (see (9)). Thus, the model updates reduce robot interactions with extraneous states, which can also be seen in the added time to goal. The robot time to goal in the presented approach was recorded to $t = 13.7\text{s}$ before and $t = 12.6\text{s}$ after model updates, while in the reactive approach, the completion time was $t = 15\text{s}$.

We also extensively test our approach on longer trajectories, where the robot traverses a 60m long corridor in the presence of approximately 50 people who walk and stop intermittently. Comparative results with reactive spring-mass-damper planners and with ORCA [17] over 100 trials are shown in Table 1. The target time for the trajectory is

TABLE I
COMPARATIVE SIMULATION RESULTS

Approach	Added Time (%)	Minimum Distance (m)	Mission Success (%)	Collision (%)
Presented Approach	15%	1.614	96%	0%
Reactive Virtual Springs	34%	1.436	68%	0%
ORCA	18%	1.453	93%	0%

60s, consistent with a straight path to goal at the maximum velocity, $v_{max} = 1\text{m/s}$. Our presented approach adds on average 15% extra time, while the reactive approach and ORCA add 34% and 18% extra time, respectively. The reactive approach often takes circuitous paths, and ORCA is prone to stopping when surrounding humans behave in irregular ways [17], which contributes to the added time. Our approach, on the other hand, predicts the evolution of the scenario and is able to find a path forward, even in the presence of unexpected and irregular human behaviors.

The presented approach outperforms others in keeping the minimum distance, because the reactive approach only considers current position and ORCA expects other actors (humans) to follow reciprocal velocities, both of which perform poorly when humans take irregular paths with uncertain velocities. For mission success, which is defined as reaching the goal within 90s, our approach succeeds in 96% of trials. Unsuccessful trials coincide with those of the other approaches, largely due to a very high density of humans in the corridor, allowing for no safe path forward, due to the safe distance constraints in each motion planning approach.

B. Experiments

The proposed approach was also validated experimentally using a Clearpath Robotics Ridgeback Omnidirectional Platform in an indoor environment. Two kinds of experiments were performed : 1) using our motion capture system to track the positions of humans and 2) using an on-board ASUS Xtion RGB-D camera with the SPENCER people tracking package [18] to locate humans in the environment. HMM predictions, reachable sets, and the motion plan are computed in MATLAB, and the robot is controlled using

the Robotics System Toolbox to interface MATLAB with ROS. The training for both experiments consisted of 100 trajectories depicted in Fig. 3(a).

The robot is tasked to avoid and accommodate humans while moving from $(-3.0, 0)\text{m}$ to $(3.0, 0)\text{m}$ at a maximum velocity of $v_{max} = 0.5\text{m/s}$, which is reduced from our simulations due to Lab space constraints.

In Fig. 7, we show a comparison of the presented proactive approach and a reactive approach. The proactive approach

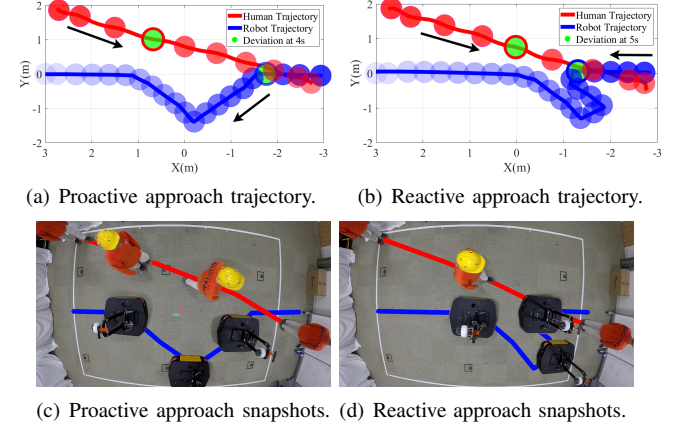


Fig. 7. Comparative results of the presented proactive approach and a reactive virtual physics based approach. The markers fade as time increases. Green markers represent deviation points.

depicted in Figs.7(a,c) predicts the person's future states creating a trajectory to accommodate the person motion while maintaining the direction to the goal. In the reactive case in Figs.7(b,d), the robot is pushed backwards by the person moving diagonally.

In the second experiment, we recreate a similar situation as our simulations in Fig. 6. The robot behaves similarly and successfully navigates around both humans, accommodating their intentions (Fig. 8(a)). Notably, the robot reacts to the purple trajectory by moving in the $+y$ direction, despite the distance never nearing the threshold (Fig. 8(c)). These types

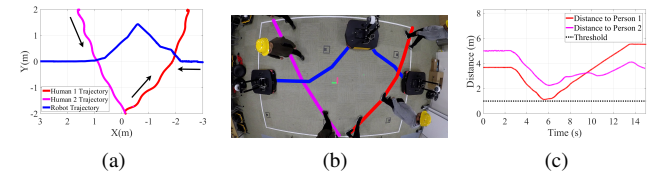
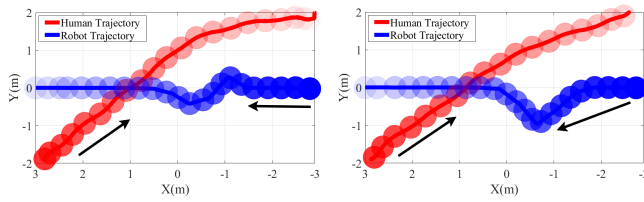


Fig. 8. Results from two-person experiments. (a,b) show the trajectories of the humans and the robot, while (c) shows the distance maintained between the robot and each person.

of experiments were carried out with over 30 trajectories in the lab environment, resulting in no collisions and no violations of the safe distance.

To test the effectiveness of online updates, we performed an experiment with a model that is trained on an incomplete subset of trajectories, comparing before and after the model updated during run-time. Initially, we observe the robot going toward the human's path (Fig. 9(a)), as it makes incorrect intention predictions. The person, in this case, takes a slightly wider path, reacting to the robot's incorrect behavior. The model is updated and reinforced with the observed trajectory at run-time, as discussed in Section IV-D. When a new similar scenario happens again, the robot correctly predicts

the human intention and deviates to accommodate the future path as shown in Fig. 9(b).



(a) Trajectory prior to model update. (b) Trajectory after model update.

Fig. 9. (a) A robot with incorrectly predicts human intention, moving into the path of the human. After updating the model, the robot proactively accommodates the correct human intention by moving to its left (b).

In the second experiment without MOCAP, using only the on-board RGB-D camera, the robot successfully accommodates four people and reaches the goal, showing that our approach scales to more people and performs well despite noisy camera measurements and uncertainty in person identification and tracking. An overlaid sequence of snapshots and a first-person view of the robot are displayed in Fig. 10.

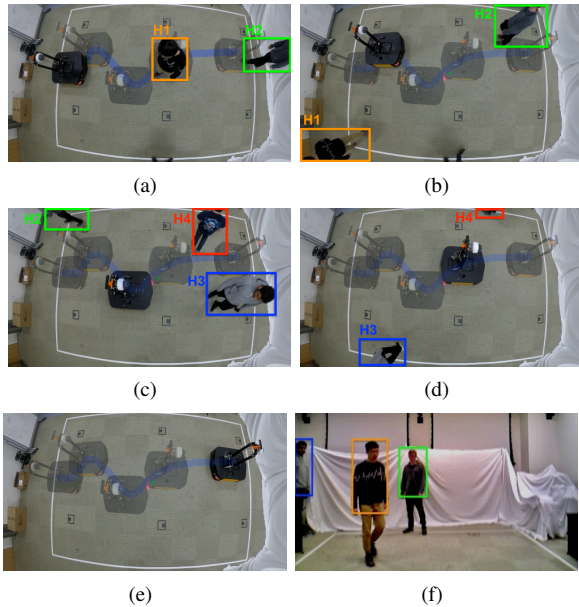


Fig. 10. Results from 4 person camera experiment. Fig 10(f) shows a first-person view of the robot when 3 of the 4 people are in the frame.

VI. CONCLUSIONS & FUTURE WORK

In this work, we have presented an approach for prediction and planning of a robot traversing a shared environment with multiple humans. We leverage Hidden Markov Model theory to predict the human intention and propose temporal stochastic reachability that is coupled with a virtual spring-mass system-based method to generate proactive intention-aware motion for the robot. Results show that our approach performs better than reactive dynamic obstacle avoidance approaches and ORCA. Unique from other works in this field, a key feature here is that the predictive model is constantly updated at run-time improving the behavior as more observations are made. The improvement happens because human motion is not random and over time, the system is able to learn and converge to common social behaviors. Although temporal reachable sets reduce the possibility of the robot to completely stop, the “freezing robot” problem

may arise when the robot is surrounded by multiple people. In future work, we aim to address this problem by encoding and analyzing the effects of complex, high-density crowd dynamics on our framework.

VII. ACKNOWLEDGEMENT

This material is based upon work supported by NSF under grant number #1816591 and #1823325 and by AFRL and DARPA under Contract No. FA8750-18-C-0090.

REFERENCES

- [1] P. Trautman and A. Krause, “Unfreezing the robot: Navigation in dense, interacting crowds,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 797–803.
- [2] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [3] M. Phillips and M. Likhachev, “Sipp: Safe interval path planning for dynamic environments,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 5628–5635.
- [4] M. S. M. Hashim, T. Lu, and H. H. Basri, “Dynamic obstacle avoidance approach for car-like robots in dynamic environments,” in *2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, Dec 2012, pp. 130–135.
- [5] Y. Gao, P. B. Luh, H. Zhang, and T. Chen, “A modified social force model considering relative velocity of pedestrians,” in *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2013, pp. 747–751.
- [6] S. Bansal, A. Bajcsy, E. Ratner, A. D. Dragan, and C. J. Tomlin, “A hamilton-jacobi reachability-based framework for predicting and analyzing human motion for safe planning,” 2019.
- [7] 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,” *CoRR*, vol. abs/1806.00109, 2018. [Online]. Available: <http://arxiv.org/abs/1806.00109>
- [8] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4674–4683.
- [9] Y. F. Chen, M. Everett, M. Liu, and J. P. How, “Socially aware motion planning with deep reinforcement learning,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, 2017.
- [10] Y. Xu, Z. Piao, and S. Gao, “Encoding crowd interaction with deep neural network for pedestrian trajectory prediction,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 5275–5284.
- [11] Z. Wang, P. Jensfelt, and J. Folkesson, “Multi-scale conditional transition map: Modeling spatial-temporal dynamics of human movements with local and long-term correlations,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 6244–6251.
- [12] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 2242–2253.
- [13] L. Liebenwein, C. Baykal, I. Gilitschenski, S. Karaman, and D. Rus, “Sampling-based approximation algorithms for reachability analysis with provable guarantees,” in *Robotics: Science and Systems*, 2018.
- [14] T. X. Lin, E. Yel, and N. Bezzo, “Energy-aware persistent control of heterogeneous robotic systems,” in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 2782–2787.
- [15] L. Nasraoui, L. N. Atallah, and M. Siala, “Performance study of a reduced complexity time synchronization approach for ofdm systems,” in *Third International Conference on Communications and Networking*, March 2012, pp. 1–5.
- [16] Y. Dong, H. Su, J. Zhu, and B. Zhang, “Improving interpretability of deep neural networks with semantic information,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 975–983.
- [17] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, “Reciprocal collision avoidance for multiple car-like robots,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 360–366.
- [18] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, “On multi-modal people tracking from mobile platforms in very crowded and dynamic environments,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5512–5519.