Towards Safe Motion Planning in Human Workspaces: A Robust Multi-agent Approach

Shih-Yun Lo[1], Benito Fernandez[1], Peter Stone[1,2] and Andrea L. Thomaz[1]
[1]the University of Texas at Austin, [2]Sony AI
{yunl,benito}@utexas.edu, pstone@cs.utexas.edu, athomaz@ece.utexas.edu

***Abstract*— It is becoming increasingly feasible for robots to share a workspace with humans. However, for them to do so safely while maintaining agile performance, they need the ability to smoothly handle the dynamics and uncertainty caused by human motions. Markov Decision Processes (MDPs) serve as a common framework to formulate robot planning problems. However, because of its single-agent formulation, such planner cannot account for human reaction when evaluating robot actions. The robot can thus suffer from unsafe motions and move in ways that are hard for nearby humans to understand. To resolve this, we instead model robot planning in human workspaces as a Stochastic Game, and contribute a robust planning algorithm, which enables the robot to account for its prediction errors in human responses to prevent collision, while not losing agility, opposed to traditional maximin optimization techniques, by applying maximin operation only at "critical states". We validate the approach under partial knowledge of pedestrian behaviors, and show that our approach encounters zero collision despite imperfect prediction, while improving path efficiency, compared to baselines.**

I. Introduction

In robotics, cost-minimization solutions using MDPs have shown success to generate efficient trajectories in static workspaces [1][2]. In environments with dynamic objects, however, the MDP formulation is limited by its intrinsic *static environment assumptions*: *1)* state transition function is time-invariant, *2)* reward function is time-invariant, and therefore *3)* state value function is time-invariant. As these assumptions no longer hold in dynamic environments, traditional motion planning literature suffers from poor performance when applied in human workspaces.

Common strategies to deal with the above disadvantages include frequent replanning and short-horizon planning; nevertheless, the lack of awareness of future variations leads the planner to shortsighted decisions (causing socially incompetent behavior for human interaction [3]), or overly conservative decisions for long-horizon planning (referred to as the freezing-robot problem [4]). One scenario that traditional planners fail to realize is the commonly-seen flow-following strategy in crowd navigation [5] – people follow one another to reach shared short-term subgoals. This strategy relies on policy evaluation based on the future paths of nearby agents, which traditional static cost formulations cannot incorporate.

To resolve the issue, we first formulate robot planning in human workspaces as a multi-agent problem using stochastic games, to compute the dynamic state-action values that are influenced by the states and actions of *other agents* (here, the humans). We develop a sampling-based algorithm to plan on this formulation, and address safety, or collision prevention, by applying maximin operations on samples (of pedestrian motions) where collision may incur under worst-case prediction, using off-the-shelf collision detection techniques. While traditional robust planning methods can prevent unwanted events and ensure worst-case performance (by applying worst-case prediction throughout the whole planning process), the resultant behavior can be overly conservative [6], which degrades robot motion agility and smoothness around humans [7]. Our proposed approach only applies worst-case prediction at "critical" states that collision could incur, to prepare preventive motions early while not assuming adversarial pedestrians elsewhere. The approach contributes by *planning carefully only when it matters*, and yields improved path efficiency while maintaining safety.

As prior crowd navigation approaches, which assume homogeneous (and reciprocal) crowd behaviors for robot plan evaluation, were shown to suffer from prediction error when deployed in real world [8] [9], we therefore proposed heterogeneous pedestrian models based on field study for evaluation, to validation safety under partial model knowledge. We constructed a baseline that replicates the reciprocal behavior in learning-based approaches in the literature [4] [10], and show that our planner prevents 55% of unanticipated stop (safety-ensured maneuver), and is safe encountering all types of pedestrians, despite its incomplete knowledge of pedestrian models.

II. Related Work

Despite the efforts to introduce human factors into planning [11] [12], traditional motion planning algorithms have shown to generate motions that appear socially incompetent [13] [3] – inconsistent motion arose when solving for the optimal trajectory under highly dynamic environment.

On dynamic obstacle avoidance approaches [14] [15], used on low-level control to navigate towards a local direction, their constant-speed model assumption leads to myopic decision in response to human motion; they are therefore used as collision detector during plan execution [9] [16] [8], which yields unsmooth motion under unanticipated events.

Recently, a community proposed to solve robot planning in human workspaces as a multi-agent joint-dynamics learning problem, and uses motion models learned from crowd

*This work was supported by the US NSF (IIS-1564080, IIS-1724157) and US ONR (N000141612835, N000141612785).

data as the robot planner, as if it was one of the crowd members [4] [10] [17] [18]. This method has been shown to outperform traditional motion planning approaches by producing smooth human-emulating trajectories. One major drawback, however, is that the predicted interaction, learned from human crowds, do not well represent human behavior when responding to a robot. The planner is then ineffective in scenarios where human exhibits rarely seen behaviors in interaction with another human, when they are around robots [17] [8] [9]. Learning-based methods also have limited capability to apply for task-dependent robot objectives, e.g. being urgent or just wandering around casually, which the planning-based formulation can achieve.

Incorporating human prediction for robot planning has mostly been formulated in single-agent settings, which fail to capture the co-dependency in interactions [19] [3]. To deal with the uncertainty in prediction, stochastic dynamic program has been applied to deal with noisy human behaviors [16]. For interactive agent designs in video games, human actions are considered in the multi-agent MDP model to simulate multi-agent planning performance [20] [21], where the AI agent's current actions are assumed known by the humans to simulate their policies. This "omniscient" setting follows the turn-taking game formulation, instead of a simultaneous-action model as in real-world interactions.

In Game Theory, Stochastic Games were proposed to model dependent outcomes among multiple players, and have been used to generate human-like interactive motions [22]. Following the formulation, Markov Games were proposed for multi-agent reinforcement learning [23], to study the interactions among learning agents, for example, how one agent's learning affects the final outcome of the others and how they should learn accordingly [24].

III. Problem Statement

We first define the problem to apply traditional motion planning for robot navigation in human workspaces.

A. Dynamic Environment Dilemma

We define the robot's state x_t in the state space X, and its action a_t in the action space A. The collision-free workspace, a subset of the overall workspace, $W_{free} \subset W$, is defined as the feasibly reachable space given robot kinematics. The motion planning formulation is to minimize the accumulated travel cost C_t, while ensuring robot's final state x_T ends in the specified goal set $X^G \subset W_{free}$. C_t is a function of the state-action pair: $Cost(x_t, a_t)$. A negative terminal state cost is assigned: $Cost_{to-go}(x_T) < 0, \forall x_T \in X^G$, to encourage arrival in X^G. To ensure safety, transitions out of free space W_{free} are assigned with high cost: $Cost(x_t, a_t) = \infty, \forall x_t \notin W_{free}$. The sequential optimization formulation is as follows:

$$a^*_{t:T} = \underset{a_{t:T}}{\operatorname{argmin}} \Sigma_t^T Cost(x_t, a_t) + Cost_{to-go}(x_T), \quad s.t. \quad x_{t+1} = \mathscr{T}(x_t, a_t), \forall t \tag{1}$$

where $\mathscr{T}$ is the state transition function. The sequence of a variable v from t to T is denoted by $v_{t:T}$. This common

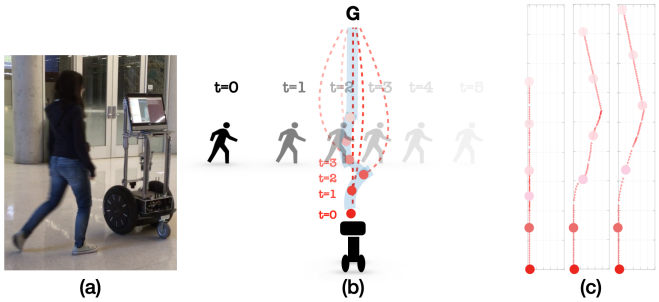

Fig. 1. (a) The robot platform passing pedestrians, used in this research to collect human responses. (b) The problem of robot inconsistent paths, replanned over time (marked by dash red lines). The resultant motion (solid blue curve) was considered *incompetent* in the social navigation literature. (c) The paths generated by our approach, capable of incorporating future pedestrian motion for planning under specified objectives.

motion planning formulation follows the MDP setting, and its solution is later refer to as the *single-agent optimal policy*.

The above formulation relies on the assumption that objects in the robot's environment are static. However, when encountering dynamic objects, W_{free} changes over time. With W_{free} being time-variant, *Cost* also becomes time-variant. The optimal sequence $a^*_{t:T}$ solved at time t then may not be valid at future time instances, as illustrated in Fig. 1-(b).

With online replanning, inconsistent motions are produced, leading to inefficient and even awkward behaviors for humans to interact with [13] [3]. To ensure collision safety, overly-conservative behavior arises, due to the inability to incorporate future variations into the cost formulation. We refer to the situation as the *dynamic environment dilemma*. To resolve this, we re-formulate the planning problem using a *game* setting, which restores the validity of the static environment assumptions by formulating the *simultaneous actions* of other agents and their state into the cost formulation.

B. Planning with Stochastic Games

In Stochastic Games, N agents act at a time t. The joint-action $a_t = (a_t^1, a_t^2, ..., a_t^N) \in A$ is defined in the joint action spaces of all agents $A = A^1 \times A^2 ... \times A^N$. The joint-state $x_t = (x_t^1, x_t^2, ..., x_t^N) \in X$ is defined in the joint state space of all agents $X = X^1 \times X^2 ... \times X^N$. The state transition function $\mathscr{T} : X \times A \to X$ affects agent reward (inverse of *Cost*) over time. Time is discretized by the interval of dt, and game *periods* are defined as follows. At the start of each period t, each agent selects an action $a_t^i, i = 1:N$ and executes it continuously for dt; the transition function $\mathscr{T}$ takes in the current state x_t and determines (probablistically) the state at the beginning of the next period $t+1$. The game starts at the initial period $t = 0$ and terminates at the final period $t = T$, which is a dynamic parameter we will define later based on interaction history. The lookahead H, how long the plan is concerned with, is selected based on the affordable computational resources.

The reward $\mathbf{r}_t^i \in \mathbb{R}$ of an agent i at time t is based on its reward function r^i: $\mathbf{r}_t^i = r^i(x_t, a_t^i, a_t^{-i})$, where a_t^{-i} denotes actions of all other agents except agent i. This formulation incorporates x_t, the time-variant states of all agents; it brings out the notion of *planning while considering the effects of the simultaneous actions of other agents*, which resolves the dynamic workspace issue.

We then solve for the optimal action sequence, given other agents executing their policies π^{-i}. The optimal state value function of agent i's optimal policy, $V^{i|\pi^{-i}}$, is defined as:

$$V^{i|\pi^{-i}}(x_t) = \max_{a_t^i} \mathbb{E}_{a_t^{-i}\sim\pi^{-i}(x_t)}[Q^{i|\pi^{-i}}(x_t, a_t^i, a_t^{-i})], \quad (2)$$

where $Q^{i|\pi^{-i}}$, the optimal state-action value of agent i given π^{-i}, is defined recursively, similar to the Bellman's Equation, solvable through dynamic programming:

$$Q^{i|\pi^{-i}}(x_t) = \max_{a_t^i} \mathbb{E}_{a_t^{-i}\sim\pi^{-i}(x_t)}[r^i(x_t, a_t^i, a_t^{-i}) + V^{i|\pi^{-i}}(x_{t+1})]. \quad (3)$$

The optimal action of agent i at time t is therefore,

$$a_t^{i*} = \operatorname*{argmax}_{a_t^i} \mathbb{E}_{a_t^{-i}\sim\pi^{-i}(x_t)}[Q^{i|\pi^{-i}}(x_t, a_t^i, a_t^{-i})], \quad (4)$$

which is defined in the joint state space X, and it depends on agent i's estimate of other agents' policies π^{-i}.

C. Planning Problem in Real World

Despite the benefits of using stochastic games to solve for the "optimal" plan, accurate modeling of human behavior for π^{-i} remains ongoing research. Prediction error not only yields inefficient performance but also collision risks during real-world deployment, for the robot had to constantly slow down to resolve unanticipated collision threats [9][8].

1) Collision Prevention under Prediction Error: To address the safety concern induced by prediction error, in the literature, worst-case predictions have been commonly applied to prevent events of high costs [25]. Within the multi-agent formulation, it is then to be concerned with adversarial others for prediction:

$$a_{0:T}^{i*} = \operatorname*{argmax}_{a_{0:T}^i} \min_{a_{0:T}^{-i}} \mathbb{E}_{x_{0:T}}[\sum_{t=0}^{T-1} r^i(x_t, a_t^i, a_t^{-i}) + Q_T^i(x_T, a_T^i, a_T^{-i})]. \quad (5)$$

This method addresses worst-case performance, such as to safely maneuver from people who intentionally block the robot. The behavior assumption however leads to *overly conservative* decisions preventing the robot from engaging in any risk, e.g., to slow down early or navigate far away when encountering crowds, which leads to inefficiency. We therefore contribute a robust planning algorithm to improve the efficiency while maintaining safety, detailed in Section. IV.

2) Validation under Modeling Error: In addition to data insufficiency, people exhibit individual differences in their responses to the robot, which contribute to prediction inaccuracy that is hard to prevent. We therefore conduct field study, detailed in Sec.V, and propose various pedestrian models for performance validation under partial model knowledge.

IV. Methodology

We propose to improve the performance of maximin planners as in Eq. 5, first by considering collision avoidance as a *finite-period game* which ends at the time when collision threat is resolved, and second by planning with worst-case prediction *only in the final period*, which is the critical timing to prevent collision. We then introduce our proposed search algorithm to plan in stochastic games with final-period worst-case prediction, and the real-time computation.

A. Robust Planning with Final-period Worst-case Prediction

When agents coordinate for collision avoidance, they adjust motions early, as such (history of) actions affect the value of the final passing. After collision threat is resolved at time T, which we define as when two agents' individual single-agent optimal policies no longer lower the plan values of each other, the game terminates. The game also early terminates if collision occurs, in which case, large penalty is assigned.

To guarantee safety, the robot needs to consider worst-case scenarios up to time T when calculating the cumulative rewards and final-period coordination value Q_T^i:

$$a_{0:T}^{i*} = \operatorname*{argmax}_{a_{0:T}^i} \min_{a_{0:T}^{-i}} \mathbb{E}_{x_{0:T}}[\sum_{t=0}^{T-1} r^i(x_t, a_t^i, a_t^{-i}) + Q_T^i(x_T, a_T^i, a_T^{-i})]. \quad (6)$$

T is dynamically updated given online observation. With each agent $-i$, individual T value is calculated; depending on action history of agent i and $-i$, T can vary. We here denote Q_T^i, instead of $Q_T^{i|\pi^{-i}}$, since we expect no interaction after the final period [1].

Moreover, we propose that the worst-case prediction is not needed until reaching *critical states*, in which actions can have unrecoverably great impacts on the optimal value $V_t^{i|\pi^{-i}}$. For example, entering a narrow hallway with possible dead end, with a vehicle that cannot do reverse driving, can prevent the robot from completing its task. While such critical state can be intractable to compute in general problem settings, in collision coordination (despite that actions in every period affect the optimal value), it is only until the *final period* can actions evoke the large collision penalty that the planner seeks to prevent.

Therefore, to ensure safety, the robot only needs to account for worst-case prediction and the associated action value, until immediate collision (under the worst-case prediction) is detected, which is at the final period of the searched branch. The planner then can still use a nominal prediction model π^{-i} for reward estimate for $t = 0 : T-1$:

$$a_{0:T}^{i*} = \operatorname*{argmax}_{a_{0:T}^i} \mathbb{E}_{a_{0:T}^{-i}, x_{0:T}|\pi^{-i}}[\sum_{t=0}^{T-1} r^i(x_t, a_t^i, a_t^{-i})] + \min_{a_T^{-i}} \quad Q^i(x_T, a_T^i, a_T^{-i}). \quad (7)$$

We refer to this behavior as *safe yet not overly conservative*, and the strategy as **planning carefully only when it matters**.

B. Planning on Stochastic Games with Safety Guarantees

To find the optimal solution, we consider tree search to apply forward simulation/state transitions of robot dynamics with non-holonomic constraints. A tree starts with a root node x_t, and it expands by forward simulating the state-action pair (possibly through a stochastic function) $x_{t+1} \sim \mathcal{T}(x_t, a_t)$, and a reward $r_t = r(x_t, a_t)$ is received. An illustration of a

[1]This assumption holds among goal-oriented agents, but does not hold among adversarial agents, for whom longer periods have to be concerned. We do not consider adversarial agents here.

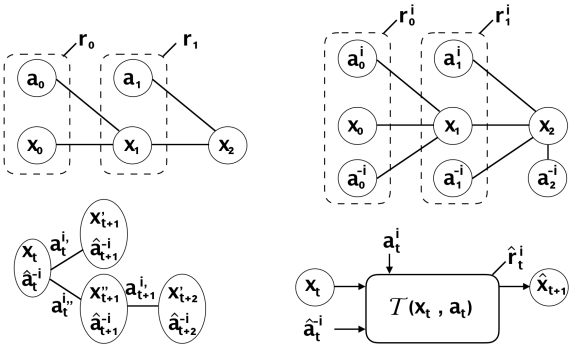

Fig. 2. Comparison between single-agent MDPs (Top-Left) and stochastic games (Top-Right). A tree with all-agent simulation expands its node x_t (Bottom-Left) with choice of actions (here, a_t^i, or $a_t^{i,,}$) to the children nodes (here, x_{t+1}' or x_{t+1}''). Since actions of other agents a_t^{-i} are unknown, to expand from a node x_t, the state transition function $\mathscr{T}$ needs to sample (K) potential actions of other agents $\hat{a}_t^{-i}$ ($a_{t,k}^{-i}, k = 1..K$), to estimate the reward $\hat{r}_t^i$ after taking action a_t^i and arriving at the next state $\hat{x}_{t+1}$ (Bottom-Right).

graphical model of single-agent MDP is shown in Fig. 2: Top-Left. When planning in Stochastic Games, both the reward function $r^i(x_t, a_t^i, a_t^{-i})$ and the transition function $\mathscr{T}(x_t, a_t^i, a_t^{-i})$ involve other agents' actions a_t^{-i}, which may be highly probabilistic and are not available until observed at t+1 (Fig. 2: Top-Right). Therefore, we sample their potential actions for reward estimate $\hat{r}_t^i$ and state transition (Fig. 2: Bottom-Right, to expand the tree with all-agent rollout (Fig. 2: Bottom-Left).

1) Nominal prediction: Each node contains the state of agent i, x_t^i, and its action, a_t^i. For action selection, $a_t^i \in A^i$ are sampled (shown in Fig. 2:Bottom-Left). Each node also contains K sets of sampled states and actions for each agent $-i$, as the *belief states* and *belief actions*, to account for prediction uncertainty of x_t^{-i} and a_t^{-i}. The belief actions at time t, $a_{t,k}^{-i}, k = 1..K$, can be sampled from models that are Markovian [5], or history-dependent [9]. The reward r_t^i is then estimated through the unweighted sample averages,

$$\hat{r}_t^i = \frac{1}{K} \sum_{k=1}^{K} r^i(x_{t,k}, a_t^i, a_{t,k}^{-i}). \tag{8}$$

The state transition x_{t+1} is estimated by K sets of samples, $x_{t+1,k} = \mathscr{T}([x_t^i, x_{t,k}^{-i}], [a_t^i, a_{t,k}^{-i}])$. In practice, the choice of K should be domain dependent, e.g., maximum likelihood prediction showed success in some navigation domains [26].

The approach is related to sequential importance sampling and Partially Observable Monte Carlo Planning [27], a POMDP planner which samples observations (here, the belief actions) to maintain belief states, but without the pruning step. We omit this belief update step, for the clarity of demonstrating the the ability to address the modeling error issue. While this step could potentially improve prediction accuracy on-the-fly and therefore plan quality, we consider it as future extension but do not detail here.

Replanning is initiated each time a new observation o_{t+1} is received, and a new search tree is started. The planner computation depends on the agent number N, number of sampled actions K per agent, sampled robot actions $|A^i|$, and search depth H , with the complexity $(|A^i| + NK)|A^i|^{H-1}$ for node expansion. Sampling $|A^i|$ actions per agent $-i$ along with sampling $a_t^i \in A^i$ can have complexity $(N|A^{-i}||A^i|)^H$; here we maintain fixed number of samples for state transition $x_{t+1,1:K}^{-i}$, to prevent it from growing exponentially in H. We here do not consider further reduction to make sure all node expansions have joint state transitions, for collision check in future periods. If collision threat is detected as resolved, single-agent policy can be restored. Such implementation should be delayed in case that T estimate is noisy.

Here we use heuristics to guide the search; other action selection mechanism can be used, e.g., UCT in MCTS [28].

2) Worst-case prediction: Collisions are checked under the condition that the distance between two agents is within 0.8 m and the robot's velocity is greater than 0.3 m/s. This condition applies for both planning and execution time for validation. We chose to apply a safety speed (0.3 m/s) instead of full stop, since people are very capable of avoiding low-speed objects. Full stop was also suggested as unnatural in human crowds [4].

We detect collision by solving for the minimum pedestrian velocity change to induce collisions with sampled robot motion, checking if the velocity change is within a predefined value: v_h, human maximum speed change from nominal speed. We here apply $v_h = 0.7m/s^2$, which can be online adjusted. v_h can apply a larger value for conservative detection. The approach is similar to velocity obstacles [14]; other collision detection programs can be used [7] [8] [9] [17].

As here we want to ensure safety, nodes with potential collision detected can be directly removed from the tree. For computational efficiency, before node expansion, one should first apply worst-case prediction and directly abandon the node if potential collision is detected.

3) Algorithm: our robust planning technique can be seen in Algorithm 1. It begins with initializing state x_t^i, belief state samples $x_{t,1:K}^{-i}$, root node b_t, the search queue $qSet$, and the heuristic function for cost-to-go estimate $V^{i|-i*}$ (Line: 1). Nodes are composed of a state x_t^i, belief state samples $x_{t,1:K}^{-i}$, action a_t^i, and belief actions $a_{t,1:K}^{-i}$, as shown in Fig. 2:Bottom-Left. The algorithm then enters a loop, which repeats the search and node expansion until $qSet$ is empty or until time is out (Line: 2-19). The optimal action sequence is returned as the history actions to reach the final searched node b_t (Line: 20-21). The ActionHistory function takes in b_t and the index of action(s) to retrieve: a_t^i is indexed 0, and agent $-i$'s sampled actions $a_{t,1:K}^{-i}$ are indexed 1 to K.

Within the repeat loop, a node is first selected (Line: 3-4). Entering the first for-loop (Line: 5-8), belief actions are sampled using π^{-i}, represented as a history-dependent function (Line: 6). Then agent $-i$'s state transition is applied (Line: 7). Entering the second for-loop with robot action sampled $a_t^i \in A^i$ (Line: 9-18), collision is first checked by a Worst-casePredict function (Line: 10), which returns *True* if collision is detected (Line: 11-13), it continues with the next sampled action $a_t^i \in A^i$ for collision check. If no collision is detected given a_t^i, state transition is applied (Line: 14) for node expansion (Line: 15-16). The accumulated reward till the expanded node b_{t+1} is updated with unweighted sample averages as in Eq. 8 (Line: 17).

Algorithm 1 Multi-agent Tree Search with Safety Guarantees

1: **Initialize:** state x_t^i, belief state samples $x_{t,1:K}^{-i}$
state transition function $\mathscr{T}^i$, $\mathscr{T}^{-i}$
root node $b_t \leftarrow \langle [x_t^i, x_{t,1:K}^{-i}][\,]\rangle$, search queue $qSet \leftarrow b_t$
reward function r
accumulated reward $g(b_t) \leftarrow 0$, heuristic function $V^{i|-i*}$
2: **repeat**
3: $b_t \leftarrow$NodeSelect($qSet$,g,$V^{i|-i*}$)
4: update $x_t^i, x_{1:K}^{-i}$ based on those in b_t
5: **for** k from 1 to K **do**
6: $a_{t,k}^{-i} \sim \pi^{-i}([x_t^i, x_{t,k}^{-i}]|$ ActionHistory$(b_t, [0:K]))$
7: $x_{t+1,k}^{-i} \leftarrow \mathscr{T}^{-i}(x_{t,k}^{-i}, a_{t,k}^{-i})$
8: **end for**
9: **for** sampled $a_t^i \in A^i$ **do**
10: $Collide \leftarrow$Worst-casePredict$(x_t^i, x_{t,1:K}^{-i}, a_t^i, A^{-i})$
11: **if** $Collide$ **then**
12: **continue**
13: **end if**
14: $x_{t+1}^i \leftarrow \mathscr{T}^i(x_t^i, a_t^i)$
15: $b_{t+1} \leftarrow \langle [x_{t+1}^i, x_{t+1,1:K}^{-i}], [a_t^i, a_{t,1:K}^{-i}]\rangle$
16: $qSet$.append(b_{t+1})
17: $g(b_{t+1}) \leftarrow g(b_t) + \frac{1}{K}\Sigma_{k=1}^K r(x_t^i, x_{t,k}^{-i}, a_t^i, a_{t,k}^{-i})$
18: **end for**
19: **until** $qSet == empty$ **or** TimeOut()
20: $a_{0:t-1}^i \leftarrow$ActionHistory$(b_t, 0)$
21: **return** $a_{0:t-1}^i$

C. Implementation

1) Search: we apply A^* in the tree structure, and Euclidean distance as the admissible heuristic estimate for all agents (without collision detection). With A^*, the *NodeSelect* function in Line 3 outputs node b_t based on $g + \hat{V}^{i|-i*}$, and removes b_t from *qSet* afterwards. The planner runs till time budget is out or certain lookahead H is reached, and replans online at each period, in a receding-horizon fashion.

2) Action Sampling: we sample actions at nominal speed with constant angular velocity of [-30, 30] deg/s to encourage path exploration, and safety action a_s at the same angular range at safety speed ($0.3m/s$). We also sample human-like yielding motions [29]. When within 3.5 s before arriving at path intersection, we also sample slightly-accelerating motions using 5th-order polynomials, which were shown to yield intent-expressive avoidant motion [30]. Other motion primitives can apply accounting for dynamical constraints. Here we consider a_s from a motion planning perspective, and omit the discussion on other implementations of a_s for dynamically challenging platforms.

3) Real-time Computation: We consider $|A^i| = 5$ during nominal operation, $|A^i| = 8$ during collision coordination, $H = 4$, and keep the worst-case complexity under 1000 nodes of expansion. The time duration of each period is 1 sec, which ensures the robot can cover 4-sec prediction, reacting to collisions at least 3 s ahead of time. Humans usually react to collision threats $2.5s$ ahead on average [31].

D. Performance Implication

When collision is detected at b_T under worst-case prediction, the coordination with sampled a_t^i corresponds in both agents "dare" in a Chicken game. Infinite penalty is then received, for which we directly abandon b_T for plan output. The robot therefore will only produce actions from branches that do not detect collision threat (at b_T); they may contribute lengthy trajectories due to early adjustment, and may contain a_s in its action sequence. Since nominal behavior prediction is applied elsewhere, worst-case evaluation for plan selection, e.g., for both agents to take a_s (to yield) which induces long accumulated delay, is not concerned in our planner, improving the conservativeness in Eq. 5.

Example robot trajectories are illustrated (based on real outputs) in Fig. 1-(c). Robot altruistic/collaborative/aggressive behaviors are shown, given reward function: mostly on others/ evenly on all agents/ mostly on the robot. The altruistic (Left) always waits until the pedestrian passes. The aggressive (Right) has high travel efficiency by reaching the farthest. The cooperative (Middle) less hinders the pedestrian. Despite different values were reached, coordination was successful (without collision) among the three cases.

V. Human Behavior Modeling

To simulate crowd dynamics, homogeneous models were proposed in agent-based modeling, where individuals share the same multi-agent policy to interact with one another [5] [32]. Here we use Social Force Model with Collision Prediction (SF-CP) [29], which generates pedestrian coordinating motions based their crossing-point arrival timing estimate. The one with later timing generates yielding motion, whereas the other generates passing-in-front motion.

For those models, all agents share the same objective and it is of common knowledge (all agents know they share an objective and they know others know that and so on). It is as if one agent has full control of the others, to optimize that agent's objective (maximizing all agents' rewards):

$$a_{0:T}^{i*} = \underset{a_{0:T}^i}{\operatorname{argmax}} \max_{a_{0:T}^{-i}} \mathbb{E}_{x_{0:T}}[\sum_{t=0}^{T-1} r^i(x_t, a_t^i, a_t^{-i}) + Q_T^i(x_T, a_T^i, a_T^{-i})]. \tag{9}$$

Here agent i is collaborative and assumes others to also be collaborative; we refer to it as the *reciprocal* behavior.

In the real world, as observed that humans interact with robots much differently from that with humans [9][8], to build realistic behavior models, under exempt IRB approval, we deployed a robot in a public atrium, to observe pedestrian responses in an uncontrolled setting. We observed some pedestrians to exhibit the reciprocal behavior.

Some people avoided carefully, far from the robot. We refer to these people as being *cautious*, and simulate their motions by planning for the worst case, as in Eq. 6. Some people appeared the opposite, exhibited non-yielding behavior and passed in front of the robot closely. We refer to such behavior as being *aggressive*, and simulate such behavior by applying a self-centered objective on Eq. 9, imposing the assumption that the other party $-i$ compliantly maximizes i's self reward. To simulate human-like motions, we apply the coordinating strategies (to dare or to yield) solved by the above to SF-CP, and simulate to-yield motions by inputting reduced agent speed, such that its arrival timing is later than the other. The opposite is applied to simulate to-dare motions.

VI. VALIDATION

We first validate the performance of our approach when encountering the three types of pedestrian models, introduced in Section. V: the aggressive, cautious, and reciprocal models. We report the impact of modeling errors on the coordination interaction and quality, in a two-agent path crossing setting. We then report the performance compared to baselines under a 4-agent crossing setting with randomly sampled pedestrian types. Initial locations are randomly sampled with all (2 or 4) agents' arriving timings within the range of [-1,1] s difference, to simulate convoluted coordination processes. We conducted 100 trials in the 2-agent test and 20 trials in the 4-agent test. Pedestrians are simulated at 1.0 m/s, with the robot at 0.8 m/s as their nominal speeds. We evaluate the performance under the *reciprocal* assumption, running the reciprocal pedestrian model for all planner prediction.

Metrics: we consider counts of collisions as the safety metric, and counts of trials with the robot executing a_s as an indication of unsmooth avoidance. We also show counts of trials with a_s in the plan at $t = 0$, as an indication of, first, how difficult the crossing is (to navigate around smoothly) among test scenarios, and second, how conservative the prediction is among different planners. $-V_T$ is calculated as time delay at $t = T$ compared to a straight-to-goal planner, as the inverse metric for path efficiency.

I) The 2-agent test result in Table I: with 32% predicted a_s at $t = 0$, the planner sometimes predicts the path conflicts cannot be resolved without a_s (with good $-V_T$), which is a bit higher than the actual number of trials with a_s execution, due to the last-horizon worst-case prediction. Due to prediction error, with aggressive and cautious agents, more a_s are executed at $t = T$ (74 and 30) than that with reciprocal agents (21): the aggressive pedestrian may insist on passing in front when it is predicted to yield; the cautious pedestrian may slow down to yield while predicted to pass in front of the robot. Similar trends are reported in $-V_T$. This result supports the observation in prior work that performance evaluated through crowd model [4] [10] may not reproduce but degrade in the wild [9] [8]. Our planner had *zero* collision with any of the three types of pedestrians.

II) The 4-agent test result in Table II: we consider two baselines here, both with safety guarantees:

1. Maximin-baseline: the conservative planner using Eq. 6, which always predicts the worst case.

2. Safety-check reciprocal baseline: equipped with 1-period worst-case collision-detection at execution time (and executes a_s if detected), it is the optimal planner based on Eq. 4, using the reciprocal model for prediction. This planner's behavior resembles that of the human-mimicking (learning [4], [10] and planning [22]) approaches, by using the nominal model for both prediction and motion generation and using a collision detector for safety [9], [8], [16].

	Collision	Predicted a_s	Executed a_s	$-V_T$
Aggressive	0	32%	74%	1.60 ($\pm$ 0.91) s
Cautious	0	32%	30%	1.09 ($\pm$ 1.11) s
Reciprocal	0	32%	21%	0.66 ($\pm$ 0.55) s

TABLE I. Our planner evaluated under three types of pedestrian models.

	Predicted a_s	Executed a_s	$-V_T$
Maximin-baseline	100%	100%	4.95 ($\pm$ 0.74) s
Safety-check-baseline	30%	100%	3.35 ($\pm$ 0.68) s
Our planner	85%	100%	**3.22** ($\pm$ 0.60) s

TABLE II. Performance comparison in the 4-agent crossing scenario.

This coordination scenario is more difficult than the 2-agent setting, as more predicted a_s are reported by our planner at t=0 (85%). The Maximin-baseline predicted and executed a_s in all trials: it started to execute a_s at early timings, leading to least efficiency with the overly conservative prediction (-4.95s). The Safety-check-baseline predicted the fewest a_s: it went straight towards the goal, expecting others to slow down when supposed to (according to the reciprocal model prediction), yet had to constantly execute a_s due to prediction error. Our planner avoided dangerous states through the worst-case collision-detection, therefore experienced less delay in situations where aggressive agents force their ways to pass in front or cautious agents slow down when they are supposed to pass first. Due to the worst-case prediction, when interaction with the reciprocal agents, our planner can be less efficient than the Safety-check baseline.

In terms of "unanticipated" safety maneuvers, which can cause extra travel delay and degrade motion smoothness during real-time execution, our planner, compared to the Safety-check-baseline, improves 55 % of such maneuvers. All planners had **0** collision in all trials, whereas the optimal planner (using Eq. 4 without safety check) would experience collision in 70 % of the trials.

III) Discussion: as observed in the two tests, modeling error contributes greatly to $-V_T$; to improve travel efficiency, online prediction refinement, either as an add-on to existing planners, or incorporated (into our planner) as belief update, can be expected, e.g., to identify pedestrian types [33]. Here our suggested evaluation procedure accounts for unanticipated events, by simulating unmodeled behaviors, to address the concerns from real-world deployments [17] [9] [8].

VII. CONCLUSION

We contributed an algorithm for safe robot planning in human workspaces, and demonstrated its ability to prevent collision when coordinating with multiple other pedestrians under partial model knowledge. The approach was formulated using stochastic games, which resolved the dynamic environment dilemma in the motion planning literature. The proposed technique applies maximin operation only at end nodes of the search for collision prevention, which was shown to save 35% of travel delay, compared to the maximin baseline. The planner also saved 55% of unanticipated slow-down counts, compared to a naive reciprocal baseline. As modeling error remains unresolved for robot planning in human workspaces, the proposed approach and validation help ensure more smooth and safe robot deployment. Further inference, learning, and adaption online can be considered.

REFERENCES

[1] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[2] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993, pp. 802–807.

[3] T. Kruse, P. Basili, S. Glasauer, and A. Kirsch, "Legible robot navigation in the proximity of moving humans," in *Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on*. IEEE, 2012, pp. 83–88.

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

[5] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[6] S. Mannor, D. Simester, P. Sun, and J. N. Tsitsiklis, "Bias and variance approximation in value function estimates," *Management Science*, vol. 53, no. 2, pp. 308–322, 2007.

[7] N. E. Du Toit, "Robot motion planning in dynamic, cluttered, and uncertain environments: the partially closed-loop receding horizon control approach," Ph.D. dissertation, California Institute of Technology, 2010.

[8] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2096–2101.

[9] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[10] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation." in *Robotics: science and systems*. Citeseer, 2012.

[11] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 981–986.

[12] P. Papadakis, P. Rives, and A. Spalanzani, "Adaptive spacing in human-robot interactions," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2627–2632.

[13] C. Lichtenthäler, T. Lorenzy, and A. Kirsch, "Influence of legibility on perceived safety in a virtual human-robot path crossing task," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 676–681.

[14] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[16] N. E. Du Toit and J. W. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2011.

[17] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.

[18] C. I. Mavrogiannis, V. Blukis, and R. A. Knepper, "Socially competent navigation planning by deep learning of multi-agent path topologies," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6817–6824.

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

[20] T.-H. D. Nguyen, D. Hsu, W. S. Lee, T.-Y. Leong, L. P. Kaelbling, T. Lozano-Perez, and A. H. Grant, "Capir: Collaborative action planning with intention recognition." in *AIIDE*, 2011.

[21] O. Macindoe, L. P. Kaelbling, and T. Lozano-Pérez, "Pomcop: Belief space planning for sidekicks in cooperative games." in *AIIDE*, 2012.

[22] A. Turnwald and D. Wollherr, "Human-like motion planning based on game theoretic decision making," *International Journal of Social Robotics*, vol. 11, no. 1, pp. 151–170, 2019.

[23] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 157–163.

[24] J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, "Learning with opponent-learning awareness," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 122–130.

[25] A. Nilim and L. El Ghaoui, "Robust control of markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[26] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," 2010.

[27] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in neural information processing systems*, 2010, pp. 2164–2172.

[28] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Springer, 2006, pp. 282–293.

[29] I. Karamouzas, P. Heil, P. van Beek, and M. H. Overmars, "A predictive collision avoidance model for pedestrian simulation," in *International Workshop on Motion in Games*. Springer, 2009, pp. 41–52.

[30] S.-Y. Lo, K. Yamane, and K.-i. Sugiyama, "Perception of pedestrian avoidance strategies of a self-balancing mobile robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'19*. IEEE, 2019.

[31] S. Paris, J. Pettré, and S. Donikian, "Pedestrian reactive navigation for crowd simulation: a predictive approach," in *Computer Graphics Forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 665–674.

[32] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 3, pp. 7280–7287, 2002.

[33] J. Godoy, I. Karamouzas, S. J. Guy, and M. Gini, "Moving in a crowd: Safe and efficient navigation among heterogeneous agents," in *Proc. Int. Joint Conf. on Artificial Intelligence*, 2016.