

An Online Interactive Approach for Crowd Navigation of Quadrupedal Robots

Bowen Yang¹, Jianhao Jiao¹, Lujia Wang^{1,2}, Ming Liu^{1,3}

Abstract—Robot navigation in human crowds remains the challenge of understanding human behaviors in different scenarios. We present an approach for interactive and human-friendly crowd navigation in complex static environments. The planner models the online interactions among the robot, humans, and the static environment based on game theory. It recurrently expands and optimizes the estimated trajectories for the robot and neighboring agents and provides human-friendly navigation commands. We use various indicators to evaluate the social awareness of the planners and show that our method outperforms existing approaches in success rate to reach the goals and compatibility with humans while maintaining low navigation times. The planner is successfully deployed on a real-world quadrupedal robot, demonstrating safe and interactive crowd navigation with real-time performance.

I. INTRODUCTION

Robot navigation with mobile agents in constrained static environments requires the cooperation of individuals [1] and requires the robot to capture crowd interactions for safe and cooperative movements. Although a massive amount of research has been conducted on the socially aware navigation problem, challenges arise from the prediction and planning, behavior modeling, and choice of evaluation method [2].

Reactive-based, optimization-based, and learning-based approaches are widely applied to solve crowd navigation problems [3]. Reactive-based approaches only consider the instant agent states, which results in unsatisfactory performance. Optimization-based approaches first predict human trajectories using physical models or data-driven methods, then plan and optimize the path for the ego agent. However, the decoupled planning and prediction process assumes “non-responsive crowds”, which leads to unnatural robot actions or failure to find a feasible path [2]. The proposed optimization-based approach regards crowd navigation as a successive decision-making procedure. It iteratively estimates the online interaction among robot, humans, and static environment within a specific time horizon in each planning loop, which captures the robot’s impact on the crowd. Learning-based approaches implicitly model the agents’ relationships, which achieve high success rate and low travel time [4], [5].

¹ Authors are with the Hong Kong University of Science and Technology, Hong Kong SAR, China. {byangar, jjiao}@connect.ust.hk

²Lujia Wang is the corresponding author and is also with the Clear Water Bay Institute of Autonomous Driving. eewanglj@ust.hk

³Author is also with the Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511400, Guangdong, China and HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. eelium@ust.hk

This work was supported by Guangdong Basic and Applied Basic Research Foundation, under project 2021B1515120032, awarded to Prof. Lujia Wang.



Fig. 1: The proposed local planner performs online interaction estimation with mobile agents and the static environment, providing efficient and cooperative crowd navigation solutions that fully utilize the agile motion capabilities of quadrupedal robots in highly dynamic scenarios.

However, they either only focus on open area scenarios [4]–[7] or often fail in long-range applications [8], [9].

Game theory is adopted in various robotic applications including competitive racing [10], [11] and autonomous driving [12]–[14]. However, they heavily rely on the quality of initial trajectories in complex scenarios for convergence. In this paper, we leverage the advantage of game theory in modeling multi-agent relationships and apply it in interactive crowd navigation scenarios of quadrupedal robots. Our approach (GTICN) is proposed to provide robust and human-friendly commands without requiring initial trajectories of agents and take full advantage of the robot’s agile motion capabilities. We deploy our planner on a quadrupedal robot to demonstrate its agile collision avoidance and interactive navigation capabilities (Fig. 1). Our main contributions include:

- A local path planner that fully utilizes the agile motion abilities of quadrupedal robots to perform human-friendly crowd navigation.
- An approach that estimates the online crowd interaction in complex environments based on game theory and incentive force model.
- Extensive evaluations of the planning performance with multiple criteria to measure the robot’s impact on the crowd and its compatibility with humans.
- Real-world demonstrations of the proposed planner on a quadrupedal robot for human-friendly crowd navigation with real-time performance.

II. RELATED WORK

A. Traditional Model-based Approaches

Reactive-based approaches understand the crowd as a group of individuals, as is done by Optimal Reciprocal Col-

lision Avoidance (ORCA) [15] and the Social Force Model (SFM) [16]. Their variants are widely adopted in autonomous navigation [17]–[20], pedestrian simulation [21]–[23], and crowd evacuation [24], [25]. Optimization approaches are also applied for crowd navigation, including Timed Elastic Band (TEB) [26], [27] using the g2o-framework [28] and predictive methods using Model Predictive Control (MPC) [3], [29], [30]. However, they seldom consider the impact of the robot on the interactions. Khambaita and Alami [1] introduce an online approach that plans TEBs for both the robot and humans to achieve cooperative navigation. Nishimura *et al.* [31] sample human trajectories using Trajectron++ [32] to obtain robot-conditional predictions for planning. Our approach extends the SFM into the “incentive force model” and further introduces game theory to path optimization to better capture the crowd interactions for human-friendly motions.

B. Learning-based Approaches

Learning-based approaches are also widely applied in robotic crowd navigation. Reinforcement learning (RL) is used to directly obtain crowd navigation policies. Chen *et al.* train CADRL [6] under the reciprocity assumption for interactive behaviors and achieve social awareness by rewarding the policy with social norms [7]. Attention mechanisms and graph structures are then introduced in [4], [5], [33] to encode the inter-agent relations better. However, these policy networks are normally trained in open areas and ignore possible interactions with the static environment. Liu *et al.* [8] extend SARL [4] by additionally observing the occupancy map, while Dugas *et al.* [9] further improve it by introducing the transformer architecture. Deep learning can also predict human trajectories using data-driven regressions [34], [35] or generative approaches [36]. The model proposed in [34] is further integrated with an RL policy in a decoupled way to generate motion commands [37]. However, learning-based approaches are still not guaranteed to perform well outside the training scenarios [38]. Our method provides robust performance in complex static environments while maintaining high navigation efficiency compared with learning-based approaches.

C. Game-theoretic Approaches

Game theory is increasingly adopted in robotic applications. It is competent to model the behaviors of individuals in a multi-agent system and has been used in double [39]–[41] or multi-agent [10], [11] competitive scenarios. Williams *et al.* [42] use the Iterated Best Response (IBR) approach to capture the agile interactions between autonomous vehicles. Fridovich-Keil *et al.* [12], [13] propose iLQGames to approach the local Nash equilibrium for traffic simulation and motion planning. Cleac'h *et al.* [14] present an augmented Lagrangian solver and adopt it in autonomous driving. However, it's essential to obtain good initial paths for agents to find a feasible solution. In addition, they typically simplify robots as circles to achieve a rational computation time. We modify the IBR algorithm to integrate the trajectory

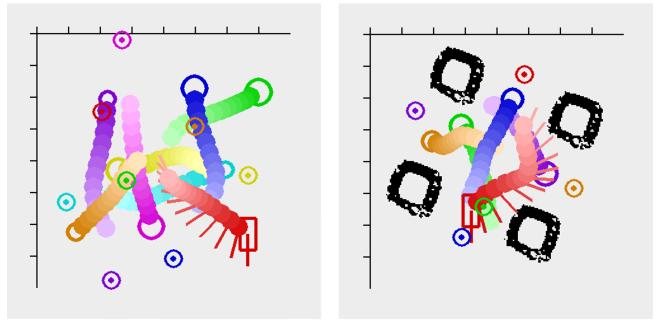


Fig. 2: The planner alternately expands and optimizes the trajectories for the robot (in red) and related agents to their goals (circled dots) using game theory. The trajectories are the interaction estimations considering the impact of the robot and static structures.

expansion and optimization procedures, which releases the requirement on the initial path. Our approach achieves a high success rate in finding feasible solutions even in highly dynamic scenarios with complex static structures. It also better captures the robot's geometric shape to further improve the navigation efficiency without introducing a heavy computational burden.

III. APPROACH

The proposed planner couples the expansion and optimization phases. Each loop performs one-step-ahead expansion and trajectory optimization alternately for robot and related agents using the incentive force model (Section III-B) until the trajectories expand to a specific time horizon and reach stable equilibria (Fig. 2). The robot takes the first step on its trajectory and repeats the process in the next loop.

A. Problem Formulation

The human state $\mathbf{q}_h \in Q_h$ contains the current position $\mathbf{p} = [p_x, p_y]^T$, linear velocity $\mathbf{v} = [v_x, v_y]^T$, and the goal $\mathbf{g} = [g_x, g_y]^T$ in the world frame. Each human has a circular body with a radius r and a preferred walking speed v_p :

$$Q_h = \{\mathbf{q}_h | \mathbf{q}_h = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{g}^T r, v_p]^T\}, \quad (1)$$

where \mathbf{p} , \mathbf{v} , and r are observable. The robot is considered rectangular with length=2 r_l and width=2 r_w . Its state $\mathbf{q}_r \in Q_r$ also contains an orientation ψ and an angular velocity ω :

$$Q_r = \{\mathbf{q}_r | \mathbf{q}_r = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{g}^T, \psi, \omega, r_l, r_w, v_p]^T\}. \quad (2)$$

In each planning loop, the planner gradually expands and optimizes the trajectories of the robot and its neighbors from the current time step $t = 0$ to a horizon ξ . The trajectory $P_i = [\mathbf{q}_i^0, \mathbf{q}_i^1, \dots, \mathbf{q}_i^l]$ of agent i comprises sequential agent states \mathbf{q}_i^t with a constant time interval τ between them. We approximate a linear motion between \mathbf{q}_i^{t-1} and \mathbf{q}_i^t , and define $\mathbf{v}_i^t, \omega_i^t$ to be the end velocities of the previous motion:

$$\mathbf{p}_i^t - \mathbf{p}_i^{t-1} = \mathbf{v}_i^t \tau, \quad \psi_i^t - \psi_i^{t-1} = \omega_i^t \tau. \quad (3)$$

We model the crowd navigation problem as an N -person ξ -stage infinite dynamic game. The Nash equilibrium is

regarded as an optimal form of the multi-agent system, where the navigation costs J_i satisfy the following inequality [43]:

$$J_i(P_0^*; \dots P_i^*; \dots P_N^*) \leq J_i(P_0^*; \dots P_i; \dots P_N^*), \forall i \in N. \quad (4)$$

A group of strategies reaches the Nash equilibrium if none of the agents has the incentive to deviate from its strategy unless its neighbors do [43]. Therefore, we define the incentive force models, where the composition of the translational and the rotational forces $\mathbf{f}_T(\mathbf{q}_i^t) = [f_{Tx}(\mathbf{q}_i^t), f_{Ty}(\mathbf{q}_i^t)]^T$ and $h_R(\mathbf{q}_i^t)$ can be interpreted as the motion incentives of agents. The social navigation cost J_i is thus the potential energy of states in the field that described by $\mathbf{f}_T(\mathbf{q}_i^t)$:

$$f_{Tx}(\mathbf{q}_i^t) = \frac{\delta J_i}{\delta p_{xi}^t}, \quad f_{Ty}(\mathbf{q}_i^t) = \frac{\delta J_i}{\delta p_{yi}^t}, \quad h_R(\mathbf{q}_i^t) = \frac{\delta J_i}{\delta \psi_i^t}. \quad (5)$$

The forces on each agent state at any time step are independent and are calculated w.r.t its current pose only. The planner approximates the Nash equilibrium by optimizing the positions \mathbf{p}_i and orientations ψ_i of the trajectory states:

$$P_i^* = \arg \min_{\mathbf{p}_i^t, \psi_i^t} J_i(P_0; \dots \mathbf{q}_i^{t-1}, \mathbf{q}_i^t, \mathbf{q}_i^{t+1}, \dots P_N), \forall t \in \xi. \quad (6)$$

B. Force Models and Constraints

The proposed method references the SFM [16] and TEB [26] to represent the influence of the static environment, the other agents at the same time step, and the self-states from the adjacent steps on a specific agent state (Fig. 3). Each agent state is incentivized by the composition of forces while satisfying several constraints. For each type of incentive, the planner calculates the translational force \mathbf{f}_t and the rotational force h_r , whose signs denote their directions. For the interactive forces, i denotes the agent who receives the force and j denotes the agent who exerts the force on i .

1) Goal Attraction: The translational attractive force \mathbf{f}_{tg} is applied to accelerate the robot to a preferred velocity \mathbf{v}_{pref} to approach the temporary goal \mathbf{g} :

$$\mathbf{v}_{\text{pref}}^t = v_{pi} \frac{\mathbf{g}_i^{t-1} - \mathbf{p}_i^{t-1}}{\|\mathbf{g}_i^{t-1} - \mathbf{p}_i^{t-1}\|_2}, \quad \mathbf{f}_{tg}(\mathbf{q}_i^t) = \frac{\mathbf{v}_{\text{pref}}^t - \mathbf{v}_i^t}{\tau}. \quad (7)$$

To present a clear intention to humans, the rotational attractive force h_{rg} follows the translation direction of the robot:

$$h_{rg}(\mathbf{q}_i^t) = \frac{\arctan(v_{yi}^t/v_{xi}^t) - \psi_i^t}{\tau}. \quad (8)$$

2) Collision Avoidance Constraints: We adopt the cost function from [26] to model the constraints of agents:

$$c_T(a, a_r, \epsilon, S, n) = \left(\frac{a - (a_r - \epsilon)}{S} \right)^n \text{ if } a > a_r - \epsilon \text{ else } 0, \quad (9)$$

where a_r is the bound, ϵ , S , and n affect the approximation, scaling, and polynomial order [26]. \mathbf{f}_{tb} denotes the translational repulsive force from dynamic agents:

$$\mathbf{f}_{tb}(\mathbf{q}_i^t, \mathbf{q}_j^t) = -\nabla c_p(-d_{ij}^t, -(r_i + r_j), \epsilon, S, n), \quad (10)$$

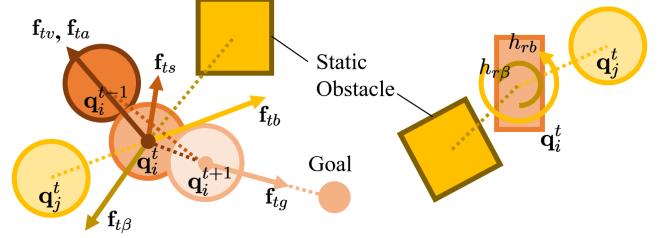


Fig. 3: The planner optimizes the robot poses by minimizing the incentive forces which model the influence from the static environment (yellow squares), other dynamic agents \mathbf{q}_j^t , and the adjacent self-states $\mathbf{q}_i^{t\pm 1}$.

where d_{ij}^t is the distance between agents i and j at step t . When the robot rotates, we consider r_i w.r.t j as a variable:

$$r_i = \min \left(\frac{r_w}{\sin |\theta_{ij}^t|}, \sqrt{r_w^2 + r_l^2} \right), \quad \theta_{ij}^t \in [0, \pi], \quad (11)$$

where θ_{ij}^t is the angle between $\mathbf{p}_j^t - \mathbf{p}_i^t$ and ψ^t . The rotational repulsive forces h_{rb} and $h_{r\beta}$ are also formulated as Eq. 10, where we obtain the gradient w.r.t ψ_i through the chain rule.

The planner uses the Euclidean Distance Field (EDF) to understand the complex static environments, with r_j replaced by an inflation distance r_o . We obtain $\mathbf{f}_{t\beta}(\mathbf{q}_i^t)$ by calculating the 2D map gradients. For the robot, we adopt a "two-circle" footprint with a front center $\mathbf{q}_{\text{front}}$ and a rear center \mathbf{q}_{rear} both having radius r_w . We define the robot's $\mathbf{f}_{t\beta}$ as the mean of the respective repulsive forces and $h_{r\beta}$ as the tangential component of the half-difference between them.

3) Velocity and Acceleration Constraints: The cost function in Eq. 9 also limits the positional difference between \mathbf{q}_i^t and \mathbf{q}_i^{t-1} to fulfill the motion constraints. Therefore, we formulate the translational constraint forces \mathbf{f}_{tv} and \mathbf{f}_{ta} :

$$\mathbf{f}_{tv}(\mathbf{q}_i^t) = -\nabla c_v(\|\mathbf{q}_i^t - \mathbf{q}_i^{t-1}\|_2, v_{\text{max}}\tau, \epsilon, S, n), \quad (12)$$

$$\mathbf{f}_{ta}(\mathbf{q}_i^t) = -\nabla c_a(\|\mathbf{v}_i^t - \mathbf{v}_i^{t-1}\|_2, a_{\text{max}}\tau, \epsilon, S, n). \quad (13)$$

The angular forces h_{rw} and h_{ra} are also obtained similarly.

4) Trajectory Smoothing: The position and velocity transitions between adjacent states are smoothed using the cost $c_t^t = \|\mathbf{p}_i^t - \mathbf{p}_i^{t-1}\|_2^2$. A variation of a state position will affect the costs of its previous and next motions, so we obtain the translational smoothing force \mathbf{f}_{ts} :

$$\mathbf{f}_{ts}(\mathbf{q}_i^t) = -\nabla(c_t^t + c_{t+1}^t). \quad (14)$$

Similarly, we define h_{rs} to smooth the orientation changes, where the rotational cost c_r and h_{rs} are:

$$c_r^t = |\psi^t - \psi^{t-1}|^2, \quad h_{rs}(\mathbf{q}_i^t) = -\nabla(c_r^t + c_r^{t+1}). \quad (15)$$

5) Composition of Forces: The overall motion incentive of a state \mathbf{q}_i^t is the composition of forces it receives, where N is the number of dynamic agents in the perception range:

$$\begin{aligned} \mathbf{f}_T(\mathbf{q}_i^t) = & \mathbf{f}_{tg}(\mathbf{q}_i^t) + \sum_{j=1}^N \mathbf{f}_{tb}(\mathbf{q}_i^t, \mathbf{q}_j^t) + \mathbf{f}_{t\beta}(\mathbf{q}_i^t) \\ & + \mathbf{f}_{tv}(\mathbf{q}_i^t) + \mathbf{f}_{ta}(\mathbf{q}_i^t) + \mathbf{f}_{ts}(\mathbf{q}_i^t), \end{aligned} \quad (16)$$

$$h_R(\mathbf{q}_i^t) = h_{rg}(\mathbf{q}_i^t) + \sum_{j=1}^N h_{rb}(\mathbf{q}_i^t, \mathbf{q}_j^t) + h_{r\beta}(\mathbf{q}_i^t) + h_{r\omega}(\mathbf{q}_i^t) + h_{r\alpha}(\mathbf{q}_i^t) + h_{rs}(\mathbf{q}_i^t). \quad (17)$$

C. Coupled Expansion and Optimization

The planner adopts a coupled scheme for trajectory expansion and optimization, where the two phases operate alternately and expand the trajectories gradually to a horizon.

1) Expansion Phase: In expansion phases, the planner finds the k -th nearest global path point ahead of the current end state as the temporary goal, while k affects how strictly the planner follows the global plan. For humans, we take a normal walking speed as their v_p and estimate their goals through linear propagation. From the current end state \mathbf{q}_i^l , the expansion phase takes the initial guess of the next step based on the attractive and repulsive forces received by \mathbf{q}_i^l . To provide a good initial guess for the expansion, the planner adopts the collision prediction model [21] to calculate \mathbf{f}_{tb} :

$$\mathbf{f}_{tb}(\mathbf{q}_i^t, \mathbf{q}_j^t) = \frac{\|\mathbf{v}_i^t\|_2}{t_i} e^{-d_{ij}^t/b} \frac{\mathbf{p}_j^t(t_i) - \mathbf{p}_i^t(t_i)}{\|\mathbf{p}_j^t(t_i) - \mathbf{p}_i^t(t_i)\|_2}, \quad (18)$$

where b is the interaction range, t_i is the "collision time" defined in [21], and $\mathbf{p}^t(t_i)$ is the linearly propagated position after t_i . We assign a higher priority for reaching the goal than maintaining the path smoothness, so the smoothing force is excluded from the expansion phase and the trajectory can automatically adapt to the growth during optimization. Therefore, we obtain the translational expansion force \mathbf{f}_E and plan the position of \mathbf{q}_i^{l+1} based on the agent dynamics:

$$\mathbf{f}_E(\mathbf{q}_i^l) = \mathbf{f}_{tg}(\mathbf{q}_i^l) + \sum_{j=1}^N \mathbf{f}_{tb}(\mathbf{q}_i^t, \mathbf{q}_j^l) + \mathbf{f}_{t\beta}(\mathbf{q}_i^l), \quad (19)$$

$$\mathbf{v}_i^{l+1} = \mathbf{v}_i^l + \tau \mathbf{f}_E(\mathbf{q}_i^l), \quad \mathbf{p}_i^{l+1} = \mathbf{p}_i^l + \tau \mathbf{v}_i^{l+1}, \quad (20)$$

where we clamp the growth length to fulfill the velocity and acceleration constraints of the new state. The orientation ψ_i^{l+1} of the new state is obtained using the same approach.

2) Optimization Phase: Although we assume that all the agents behave rationally and interactively, as they move in a non-communicative scenario, the expansions may be sub-optimal or even lead to collisions. Therefore, the trajectories are further optimized after each expansion through iterative gradient descent, where \mathbf{f}_{tg} is only applied on the current end state of the trajectory to provide more freedom to the collision avoidance. In each iteration, the poses are updated by taking a small step along with the composition of forces they receive with a stepping rate $\gamma \in (0, 1]$:

$$\mathbf{p}_i^t \leftarrow \mathbf{p}_i^t + \gamma \mathbf{f}_T(\mathbf{q}_i^t), \quad \psi_i^t \leftarrow \psi_i^t + \gamma h_R(\mathbf{q}_i^t). \quad (21)$$

In this way, a single state exerts influence on its neighboring agents at the same step through the repulsive force, while also affecting its own previous states through the smoothing process. This is how we achieve online interaction estimation with the neighboring agents. We note that the planner aims to provide a human-friendly navigation solution rather than focusing on accurately predicting human actions. The real

Algorithm 1: Game-Theoretic Interactive Crowd Navigation

Input: self state \mathbf{q}_i^0 and neighbor-agent states $\{\mathbf{q}_j^0\}$

Output: motion command \mathbf{v}_i^1 and ω_i^1

- 1: Expand one step \mathbf{q}^1 for i and $\{j\}$ (Eq. 20)
 - 2: **for** $iteration = 1, 2, \dots$ **do**
 - 3: **while** not converged **do**
 - 4: Calculate $\mathbf{f}_T(\mathbf{q}^t)$ and $h_R(\mathbf{q}^t)$ **for** $t = 1 : l$
 - 5: Optimize and update trajectories (Eq. 21)
 - 6: **if** $l + 1 < \text{horizon } \xi$ **then**
 - 7: Expand one step \mathbf{q}^{l+1} for i and $\{j\}$ (Eq. 20)
 - 8: **return** the first action \mathbf{v}_i^1 and ω_i^1
-

actions of neighboring agents may be different from the estimated states. However, the "feedback nature" of the receding horizon assists the planner in reacting to possible changes in human intentions and avoiding collisions [11].

After the trajectories converge, the expansion phase takes another group of initial guesses and expands the trajectories. Then the optimization phase again iteratively adjusts them until they reach a new equilibrium. The above process is continuously operated in a single loop until the trajectories reach a certain horizon at which the planner takes the first step $\mathbf{v}_i^1, \omega_i^1$ of its trajectory P_i as the motion command. The pseudo-code of a planning loop is presented in Algorithm 1.

IV. EXPERIMENTS

A. Setup

We test the proposed planner in various simulated and real-world environments. As suggested by [22], we set an appropriate time horizon $\tau\xi = 3s$ and interval $\tau = 0.25s$ to capture the crowd interactions while maintaining a rational computational burden. The planner takes a $10 \times 10m$ occupancy map to construct EDF with a resolution of $0.08m$ and receives the observable states of the neighboring agents within $5.0m$. We compare the success rate R_S , average time $\overline{T_R}$ for the robot to reach the goal, and the average velocity change $\overline{\Delta V_R}$, which indicates the motion smoothness. As we focus on providing human-friendly navigation solutions, we also analyze the average crowd time $\overline{T_C}$ for all agents to reach their goals, $\overline{\Delta V_C}$ that reflects the robot's impact on human actions, and the average minimum separation rate $\overline{S_M}$. We also introduce the average directional cost $\overline{C_D}$ in [1] as a measure of human compatibility with the robot, while $\overline{C_D}$ is high if the robot and the human are moving towards each other at a small distance and high speed:

$$S_{Mij} = \frac{\|\mathbf{p}_j^0 - \mathbf{p}_i^0\|_2}{r_i + r_j}, \quad (22)$$

$$C_{Dij} = \frac{S_{Mij}}{S_{Mij} - 1} \cdot \frac{\mathbf{v}_i^0 \cdot (\mathbf{p}_j^0 - \mathbf{p}_i^0) + \mathbf{v}_j^0 \cdot (\mathbf{p}_i^0 - \mathbf{p}_j^0)}{\|\mathbf{p}_j^0 - \mathbf{p}_i^0\|_2^2}, \quad (23)$$

where we use the boundary distance and the variable robot radius in Eq. 11 to calculate $\overline{S_M}$ and $\overline{C_D}$ for agents with different radii. For planners without orientation optimizations, the robot's orientations simply follow the moving directions.

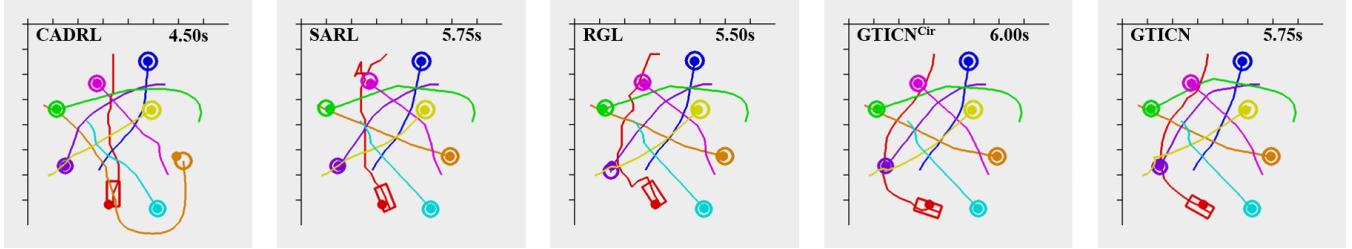


Fig. 4: An example of the resulting paths in experiment one. The robot (red) moves through the crowd from top to the goal at the bottom. Both versions of our GTICN approach provide smooth paths in rational travel times without significantly disturbing human behaviors.

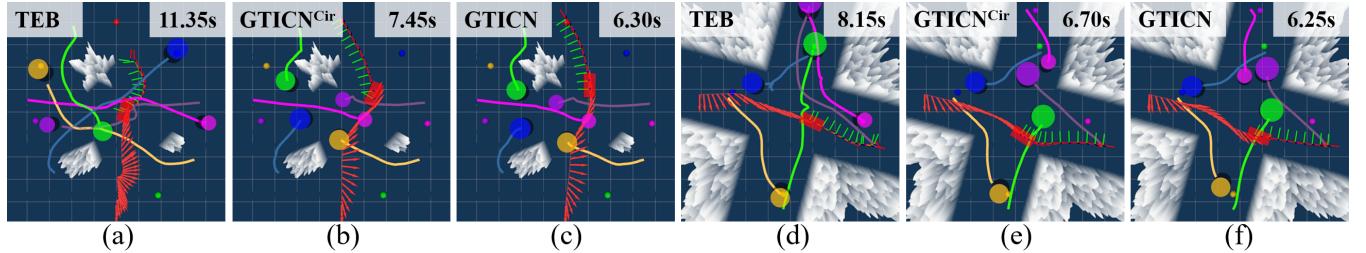


Fig. 5: Examples in *obstacle-avoidance* (a–c) and *crossroads* (d–f). Our approaches capture the crowd interaction with the static environment for efficient and smooth navigation. The travel time is further reduced by optimizing the robot’s orientations in our full version.

First, we evaluate the crowd navigation performance in the simulated open area scenario. The robot is $1.0 \times 0.5\text{m}$ in shape with $v_p = 1.5\text{m/s}$, while each neighboring agent is randomly assigned a radius $r \in [0.3, 0.5]\text{m}$ and a $v_p \in [1.0, 1.5]\text{m/s}$. We prepare 2000 scenarios, with each containing eight agents with randomly assigned initial positions and goals around a circular region and with travel distances that are four times their v_p . The interactive humans use the ORCA algorithm that tries to keep a 0.1m distance from the robot and other agents, while one of the humans adopts the “invisible robot” setting to increase the crowd uncertainty. We evaluate our approach (GTICN) while choosing CADRL [6], which is a cooperative RL policy for crowd navigation, and the SARN [4] and RGL [5] policies, which respectively use a self-attention mechanism and relational graph to capture the crowd-robot interaction, for comparison. CADRL, SARN, and RGL are trained using the simulator in [5] with holonomic kinematics, and the r and v_p of agents are randomized using the experiment settings. We further implement GTICN^{Cir}, which assumes a circular robot, to show our advantage of capturing the robot’s geometric shape.

The next experiment contains both humans and static structures, while r and v_p are randomized in the same way. Two scenarios are used to set up the static environment: In the first scenario, the agents perform *obstacle-avoidance* in a map with static objects, while the robot and another five agents manage to move from one side to the other side around a 4.0m circle (Fig. 5 (a–c)). The second scenario is set at a *crossroads* with the road width= 3.0m , and where we randomly set the robot and another five agents to move from one of the entrances to another (Fig. 5 (d–f)). Both scenarios are tested on 150 sets of agents with different start and goal positions. We use GTICN^{Cir} to simulate the predictive and interactive behavior of humans in static environments, while

all the agents still operate in a decentralized way without communication. We compare our approach with the TEB [26] local planner that assumes a constant velocity model for dynamic obstacles. The TEB uses holonomic kinematics and adopts a ‘two-circle’ footprint for the robot. All the agents take way-points from a simple A* global planner.

In real-world experiments, we deploy GTICN on a Jueying Mini [44] quadrupedal robot. The planner runs onboard at 5Hz on an Intel i7-7600U. The robot receives its own odometry and that of its neighboring agents through an external OptiTrack motion capture system, while humans are set to have $r = 0.3\text{m}$. We conduct experiments in three scenarios: the “*random-walk*” where four people randomly walk around the robot to reach their targets, the “*box-relay*” where four people randomly pass on two boxes, and the “*corridor*” where the robot moves with humans along a 2.2m -width corridor around the planning region. In the first two scenarios, the robot tries to move between the top-left and the bottom-right corner and interacts with humans. In the “*corridor*” environment, the robot moves along a U-shaped path between the top-left and the top-right corner.

B. Results

Table I and Fig. 6 show the average performance of the planners in the first experiment. CADRL achieves short travel times, but the low social separation $\overline{S_M}$ and high directional cost $\overline{C_D}$ indicate low compatibility with humans. SARN and RGL passively avoid humans, which results in low crowd travel times, but the high $\overline{\Delta V_R}$ may cause disturbance to humans in real-world applications. Our GTICN^{Cir} approach stably performs human-friendly motions by maintaining high social distances, smooth velocity transitions, and low direction costs, while our full version (GTICN) further improves travel efficiency and achieves a higher success rate R_S .

TABLE I: Average Performance in the Open Area.

| Local planners | R_S (%) | \bar{T}_R (s) | \bar{T}_C (s) | $\bar{\Delta V}_R$ (m/s ²) | $\bar{\Delta V}_C$ (m/s ²) | \bar{S}_M (1) | \bar{C}_D (1) |
|----------------------|--------------|--------------------|--------------------|---|---|--------------------|--------------------|
| CADRL | 92.05 | 5.59 | 9.78 | 1.78 | 0.54 | 1.14 | 4.67 |
| SARL | 95.85 | 5.92 | 9.12 | 2.79 | 0.54 | 1.21 | 3.64 |
| RGL | 98.50 | 5.62 | 8.88 | 2.18 | 0.52 | 1.26 | 3.42 |
| GTICN ^{Cir} | 97.45 | 6.26 | 9.34 | 1.06 | 0.49 | 1.28 | 2.52 |
| GTICN | 98.50 | 5.60 | 9.25 | 0.89 | 0.49 | 1.32 | 2.71 |

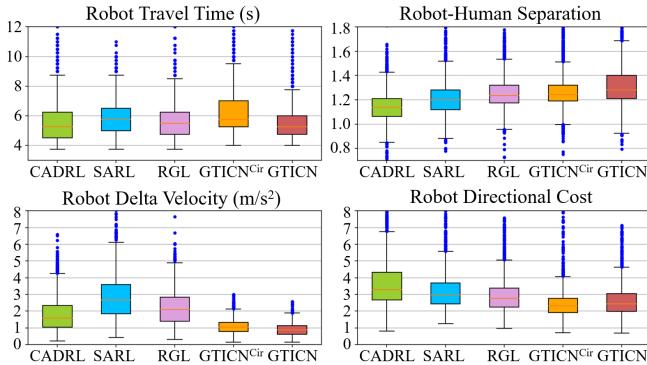


Fig. 6: The average performance of the tested planners in the first experiment. Our approach stably provides interactive navigation solutions with longer social distances, smoother velocity transitions, and higher human compatibility, while also achieving a high navigation efficiency as it better captures the robot’s geometric shape.

We further visualize one of the planning results as an example in Fig. 4. Although the CADRL policy takes the minimum time to reach the goal, it apparently affects the behavior of the agent in orange. Both versions of our approach keep a low level of influence on the crowd while the resulting robot paths are much smoother compared with other planners. This indicates that our approach assists the robot to present clear motion intentions to humans without bringing disturbance or inconvenience to them.

Table II presents the results of the second experiment, where the top three rows are from the *obstacle-avoidance* scenario and the bottom rows are from the *crossroads*. Both versions of our approach have a high success rate and achieve a high navigation efficiency with shorter travel times. In addition, they provide smooth transitions with a lower $\bar{\Delta V}_R$ and maintain better human compatibility. The TEB planner often suffers from the “freezing robot” problem because it assumes a constant velocity model for agents and fails to find feasible solutions in these highly dynamic scenarios, which results in a low success rate and long travel times. Fig. 5 presents visualizations of the planning results.

Finally, we deploy the GTICN planner on a quadrupedal robot in real-world experiments. The robot presents interactive and human-friendly motions during multiple repetitions. The planner shows cooperative behaviors when a human overtakes or meets the robot in the *corridor* environment (Fig. 1). It can also handle different crowd motion patterns in the *“random-walk”* and the *“box-relay”* scenarios (Fig. 7)

so that humans can safely concentrate on their own actions. Even when the humans have uncooperative actions, the robot can still promptly adjust itself and avoid potential collisions, which presents the feedback capability of our approach.

TABLE II: Average Performance with Static Structures.

| Local planners | R_S (%) | \bar{T}_R (s) | \bar{T}_C (s) | $\bar{\Delta V}_R$ (m/s ²) | $\bar{\Delta V}_C$ (m/s ²) | \bar{S}_M (1) | \bar{C}_D (1) |
|----------------------|--------------|--------------------|--------------------|---|---|--------------------|--------------------|
| TEB | 76.67 | 12.08 | 15.67 | 2.46 | 0.72 | 1.33 | 3.22 |
| GTICN ^{Cir} | 94.67 | 9.05 | 15.66 | 0.67 | 0.70 | 1.26 | 2.21 |
| GTICN | 99.33 | 8.57 | 15.64 | 0.68 | 0.65 | 1.27 | 2.12 |
| TEB | 86.67 | 8.09 | 12.52 | 1.98 | 0.76 | 1.33 | 3.18 |
| GTICN ^{Cir} | 98.00 | 7.65 | 11.85 | 0.78 | 0.81 | 1.32 | 2.01 |
| GTICN | 99.33 | 6.78 | 11.36 | 0.84 | 0.73 | 1.37 | 2.19 |



Fig. 7: Real-world experiment in the “box-relay” scenario. The robot safely moves through the crowd which has complex motion patterns without introducing apparent disturbances to the humans.

V. CONCLUSION

We presented an approach for interactive crowd navigation in complex static environments. It models the agents’ behaviors based on the Nash equilibrium and adopts integrated trajectory expansion and optimization phases to estimate the online interaction among the mobile agents and the static environment. The planner was evaluated in various dynamic scenarios using multiple social-awareness indicators, presenting efficient, safe, and interactive navigation that outperformed the existing approaches. We successfully deployed the planner on a quadrupedal robot in the real world. The planner performed human-friendly navigation with real-time performance, demonstrating its potential in developing vast human-interactive applications.

REFERENCES

- [1] H. Khambaita and R. Alami, “Viewing robot navigation in human environment as a cooperative activity,” in *Robotics Research*, (Cham), pp. 285–300, Springer International Publishing, 2020.
- [2] C. I. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, “Core challenges of social robot navigation: a survey,” *arXiv preprint arXiv:2103.05668*, 2021.
- [3] Y. Chen, F. Zhao, and Y. Lou, “Interactive model predictive control for robot navigation in dense crowds,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2021.

- [4] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019.
- [5] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10007–10013, 2020.
- [6] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 285–292, 2017.
- [7] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, 2017.
- [8] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5671–5677, 2020.
- [9] D. Dugas, J. I. Nieto, R. Siegwart, and J. J. Chung, "NavRep: unsupervised representations for reinforcement learning of robot navigation in dynamic human environments," *arXiv preprint arXiv:2012.04406*, 2020.
- [10] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Distributed Autonomous Robotic Systems*, pp. 225–238, 2019.
- [11] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1313–1325, 2021.
- [12] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1475–1481, 2020.
- [13] D. Fridovich-Keil, V. Rubies-Royo, and C. J. Tomlin, "An iterative quadratic method for general-sum differential games with feedback linearizable dynamics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2216–2222, 2020.
- [14] L. Cleac'h, M. Schwager, Z. Manchester, et al., "Algames: a fast augmented lagrangian solver for constrained dynamic games," *Autonomous Robots*, pp. 1–15, 2021.
- [15] S. J. van den Berg, Jurand Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, pp. 3–19, 2011.
- [16] D. Helbing, P. Molnar, I. Farkas, and K. Bolay, "Self-organizing pedestrian movement," *Environment and Planning B: Urban Analytics and City Science*, vol. 28, pp. 361–383, 05 2001.
- [17] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "PORCA: modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.
- [18] 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*, pp. 360–366, 2012.
- [19] G. Ferrer and A. Sanfeliu, "Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1730–1735, 2014.
- [20] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1688–1694, 2013.
- [21] F. Zanlungo, T. Ikeda, and T. Kanda, "Social force model with explicit collision prediction," *EPL (Europhysics Letters)*, vol. 93, p. 68005, 03 2011.
- [22] I. Karamouzas, B. Skinner, and S. Guy, "Universal power law governing pedestrian interactions," *Physical Review Letters*, vol. 113, 12 2014.
- [23] N. Rinke, C. Schiermeyer, F. Pascucci, V. Berkahn, and B. Friedrich, "A multi-layer social force approach to model interactions in shared spaces using collision prediction," *Transportation Research Procedia*, vol. 25, pp. 1249–1267, 2017. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.
- [24] Y. Jiang, B. Chen, X. Li, and Z. Ding, "Dynamic navigation field in the social force model for pedestrian evacuation," *Applied Mathematical Modelling*, vol. 80, pp. 815–826, 2020.
- [25] J. Wei, W. Fan, Z. Li, Y. Guo, Y. Fang, and J. Wang, "Simulating crowd evacuation in a social force model with iterative extended state observer," *Journal of Advanced Transportation*, vol. 2020, 2020.
- [26] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012; 7th German Conference on Robotics*, pp. 1–6, 2012.
- [27] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *2013 European Conference on Mobile Robots*, pp. 138–143, 2013.
- [28] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: a general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.
- [29] A. Wang, C. Mavrogiannis, and A. Steinfield, "Group-based motion prediction for navigation in crowded environments," *arXiv preprint arXiv:2107.11637*, 2021.
- [30] C. Rösmann, A. Makarow, and T. Bertram, "Online motion planning based on nonlinear model predictive control with non-euclidean rotation groups," 2020.
- [31] H. Nishimura, B. Ivanovic, A. Gaidon, M. Pavone, and M. Schwager, "Risk-sensitive sequential action control with multi-modal human trajectory forecasting for safe crowd-robot interaction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11205–11212, 2020.
- [32] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajetcron++: dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision – ECCV 2020*, (Cham), pp. 683–700, Springer International Publishing, 2020.
- [33] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.
- [34] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha, "RobustTP: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs," *ACM Computer Science in Cars Symposium*, 2019.
- [35] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: a deep learning perspective," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2021.
- [36] K. D. Katyal, G. D. Hager, and C.-M. Huang, "Intent-aware pedestrian prediction for adaptive crowd navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3277–3283, 2020.
- [37] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, "DenseCAvoid: real-time navigation in dense crowds using anticipatory behaviors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11345–11352, 2020.
- [38] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester, "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," *Machine Learning*, 2021.
- [39] A. Liniger and J. Lygeros, "A noncooperative game approach to autonomous racing," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 884–897, 2020.
- [40] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, "A real-time game theoretic planner for autonomous two-player drone racing," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1389–1403, 2020.
- [41] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game theoretic planning for self-driving cars in competitive scenarios," in *Robotics: Science and Systems*, 2019.
- [42] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best response model predictive control for agile interactions between autonomous ground vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2403–2410, 2018.
- [43] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd Edition. Society for Industrial and Applied Mathematics, 1998.
- [44] "Jueming Mini." https://www.deeprobotics.cn/en/products_jy_301.html. Accessed: 2021-08-30.