Social Navigation in Crowded Environments with Model Predictive Control and Deep Learning-Based Human Trajectory Prediction

Viet-Anh Le[1], Behdad Chalaki[2*], Vaishnav Tadiparthi[2*],
Hossein Nourkhiz Mahjoub[2], Jovin D'sa[2], Ehsan Moradi-Pari[2]

***Abstract*— This paper presents a model predictive control (MPC) approach for robot navigation in crowded environments. Our proposed approach couples agent motion prediction and planning to avoid the freezing robot problem while capturing multi-agent social interactions by utilizing a state-of-the-art trajectory prediction model i.e., social long short term memory model (Social-LSTM). Leveraging the output of Social-LSTM for the prediction of future trajectories of pedestrians at each time step given the robot's possible actions, our framework computes the optimal control action for the robot to navigate among pedestrians. We demonstrate the effectiveness of our proposed approach in multiple scenarios of simulated crowd navigation and compare it against several state-of-the-art reinforcement learning-based methods.**

I. Introduction

Robot navigation in environments with humans, also known as crowd navigation or social navigation, remains a challenging problem due to the uncertainty of human intentions and the reciprocal interactions between the robot's and humans' motions. As a result, crowd navigation has received increasing attention over the last few decades, and a variety of approaches have been proposed to date to address this problem [1]. Current research on robot navigation can be divided into three categories: 1) Reactive-based [2]; 2) Reinforcement Learning (RL)-based; and 3) Optimization-based. Modern approaches have sought to solve the freezing robot problem [3] by coupling the prediction of humans' future trajectories with robot planning in an interaction-aware manner. Several studies have utilized RL framework with deep neural networks - e.g., collision avoidance with deep RL (CADRL) [4], LSTM-RL [5] for handling arbitrary numbers of agents, SARL [6] for obtaining the collective impact of crowd through a self-attention mechanism, and recurrent graph neural network with attention mechanisms [7]. All of these efforts seek to train navigation policies for a single robot that maximize a specially designed reward function while minimizing the possibility of collisions with other agents. On the other hand, optimization-based methods such as model predictive control (MPC) can be typically used to optimize the behavior of a robot over a finite control horizon given certain prediction models of human trajectories. Brito *et al*. [8] combined RL with an optimization-based method in which a learned policy provides long-term guidance to a local MPC planner. Several other studies have recently proposed MPC with various human prediction models, including constant velocity [9], intention-enhanced optimal reciprocal collision avoidance (iORCA) [10], social generative adversarial networks (GAN) [11], long short term memory (LSTM) [12], and Kalman filters [13]. Recently, machine learning models such as Social-LSTM [14], Social-GAN [15], Social-NCE [16], sparse Gaussian processes [17] have shown better performance in human trajectory prediction compared to solely domain knowledge-based models. Machine learning prediction models can thus be combined with MPC to enhance the planning performance.

In this paper, we present our framework for robot navigation in crowded environments in which we integrate a machine-learning based trajectory prediction model i.e, Social-LSTM [14] into an optimization-based planning in an MPC fashion. We couple the prediction and planning to avoid freezing robot problem while capturing multi-agent social interactions. In our framework, we leverage a Social-LSTM model trained on a real human-trajectory data-set to predict the future behavior of human pedestrians and their interactions with the robot's possible actions. The solution of MPC framework coupled with the Social-LSTM model is the optimal control action for the robot to navigate among the crowd. To numerically solve the MPC problem coupled with the Social-LSTM , we utilize an iterative best-response (IBR) approach [18] inspired by the Nash equilibrium [19]. At each time-step, the method sequentially computes the neural network prediction and solves for the optimal control action of the robot. The performance of the proposed method is evaluated in simulations with different scenarios in comparison with baseline RL techniques to demonstrate the potency and the domain-invariant nature of the MPC approach.

The remainder of the paper is organized as follows. In Section II, we present our problem statement for crowd navigation, while the details of proposed control method is given in Section III. We show some simulation results in multiple scenarios with analysis in Section IV before concluding the paper in Section V.

II. Problem Statement

We consider an environment $\mathcal{W} \subset \mathbb{R}^2$ where a single robot navigates among $N \in \mathbb{N}$ human pedestrians, as can be illustrated in Fig. 1. Let 0 be the index of the robot, while $\mathcal{H} = \{1, \ldots, N\}$ denotes the set of human pedestrians in the environment.

At time step $k \in \mathbb{N}$, let $\boldsymbol{s}_{0,k} = [s^x_{0,k}, s^y_{0,k}]^\top \in \mathcal{W}$, $\boldsymbol{v}_{0,k} = [v^x_{0,k}, v^y_{0,k}]^\top \in \mathbb{R}^2$, and $\boldsymbol{a}_{0,k} = [a^x_{0,k}, a^y_{0,k}]^\top \in \mathbb{R}^2$ be the

*Both authors contributed equally. [1]V.-A. Le is with the Department of Mechanical Engineering, University of Delaware, Newark, USA (email: vietale@udel.edu)
[2]B. Chalaki, V. Tadiparthi, H. Nourkhiz Mahjoub, E. Moradi-Pari, and Jovin D'sa are with Honda Research Institute, USA Inc. (email:behdad_chalaki; vaishnav_tadiparthi; hossein_nourkhizmahjoub; emoradipari; jovin_dsa@honda-ri.com)

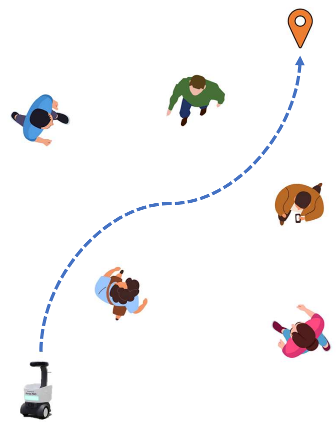


Fig. 1: An example of robot navigating in a crowded environment.

vectors corresponding to position, velocity, and acceleration of the robot in Cartesian coordinates, respectively, where each vector consists of two components for $x-$ and $y-$axis. Additionally, the robot needs to navigate from an initial position $\boldsymbol{s}_0^{\text{orig}} := [s_{0,0}^x, s_{0,0}^y]^\top$ called origin to a final goal position $\boldsymbol{s}_0^{\text{goal}} \in \mathcal{W}$ while avoiding collisions and any potential discomfort to other pedestrians $i \in \mathcal{H}$. Discomfort is a social conformity metric and is defined to be present if the robot's projected path intersects with a human's predicted path [20]. Let $\boldsymbol{x}_{0,k}^\top = [\boldsymbol{s}_{0,k}^\top, \boldsymbol{v}_{0,k}^\top]$ and $\boldsymbol{u}_{0,k} = \boldsymbol{a}_{0,k}$ be the state vector and control action for the robot at time step k, respectively. Likewise, let $\boldsymbol{s}_{i,k} = [s_{i,k}^x, s_{i,k}^y]^\top \in \mathcal{W}$ be the position of human $i \in \mathcal{H}$ at time step k in a vector form.

Assumption 1. *We assume that each pedestrian's real-time position can be measured by either onboard sensors or obtained through a positioning system.*

III. Robot Navigation with Model Predictive Control

Our framework consists of two main components: (1) a Social-LSTM model [14] which learns the social interaction and predicts the future behavior of human pedestrians, and (2) an MPC to find the optimal control action for the robot.

A. Human Motion Prediction using Social-LSTM

Let $t \in \mathbb{N}$ be the current time step, $H \in \mathbb{Z}^+$ is the control/prediction horizon length (with both equal to each other), and $\mathcal{I}_t = \{t, t+1, \dots, t+H-1\}$ be the set of time steps in the control horizon. The human prediction model aims at predicting the trajectories of human pedestrians over a prediction horizon of length H given the current and past observations over $L \in \mathbb{Z}^+$ previous time steps of all agents' trajectories including the robot's. Social-LSTM was developed in [14] for jointly predicting multi-step trajectory of multiple agents. Social-LSTM uses a separate LSTM network for each trajectory, then the LSTMs are connected to each other through a social pooling (S-pooling) layer.

We consider recursive prediction for the pedestrians' positions over the next control horizon using the single-step Social-LSTM model denoted by $\phi(\cdot) : \mathbb{R}^{2(N+1)(L)} \to \mathbb{R}^{2N}$ as follows: [21]

$$\boldsymbol{s}_{1:N,k+1} = \phi(\boldsymbol{s}_{0:N,k-L+1:k}), \ \forall k \in \mathcal{I}_t. \tag{1}$$

In (1), at each time step, predicted positions of pedestrians computed from the previous time steps are used recursively as the inputs of the Social-LSTM model. Furthermore, the Social-LSTM-based predicted positions of the robot are disregarded as they are computed using the solution of the MPC problem. For further details on the architecture design and implementation of Social-LSTM, the readers are referred to [14]. It should be noted that while in this work we employ the Social-LSTM model [14] as a human prediction model, our framework can be integrated with alternative deep learning models such as [15], [16], [22].

B. Model Predictive Control for Crowd Navigation

In this section, we formulate an MPC problem to navigate the robots while taking into account the trajectory prediction model of surrounding pedestrians. For ease of notation, henceforth, we use $\boldsymbol{u}_0$, $\boldsymbol{x}_0$, and $\boldsymbol{s}_i$, $\forall i \in \mathcal{H}$ instead of $\boldsymbol{u}_{0,t:t+H-1}$, $\boldsymbol{x}_{0,t+1:t+H}$ and $\boldsymbol{s}_{i,t+1:t+H}$, respectively, to denote the vectors concatenating the variables over the control horizon.

The system dynamics of the robot for all $k \in \mathcal{I}_t$ is given by the following discrete-time double-integrator model

$$\begin{aligned} \boldsymbol{s}_{0,k+1} &= \boldsymbol{s}_{0,k} + \tau \boldsymbol{v}_{0,k} + \frac{1}{2}\tau^2 \boldsymbol{a}_{0,k}, \\ \boldsymbol{v}_{0,k+1} &= \boldsymbol{v}_{0,k} + \tau \boldsymbol{a}_{0,k}, \end{aligned} \tag{2}$$

where $\tau \in \mathbb{R}^+$ is the sampling time period.

The speed and control input of the robot at each time step k are bounded by:

$$\begin{aligned} -v_{\max} &\le v_{0,k}^x, v_{0,k}^y \le v_{\max}, \\ -a_{\max} &\le a_{0,k}^x, a_{0,k}^y \le a_{\max}, \end{aligned} \tag{3}$$

where $v_{\max} \in \mathbb{R}^+$ and $a_{\max} \in \mathbb{R}^+$ are the maximum velocity and maximum acceleration, respectively. We formulate the objective function in MPC by a weighted sum of multiple features. In particular, to navigate the robot to the goal, we include tracking minimization to the desired trajectory

$$J^{\text{goal}}(\boldsymbol{s}_0) = \sum_{k=t}^{t+H-1} (\boldsymbol{s}_{0,k+1} - \boldsymbol{s}_{0,k+1}^{\text{ref}})^\top (\boldsymbol{s}_{0,k+1} - \boldsymbol{s}_{0,k+1}^{\text{ref}}), \tag{4}$$

where $\boldsymbol{s}_{0,k+1}^{\text{ref}}$ is the desired position at time $k+1$. We compute the desired trajectory based on the straight line to the robot's goal as follows

$$\boldsymbol{s}_{0,k+1}^{\text{ref}} = \boldsymbol{s}_{0,k}^{\text{ref}} + \min\Big\{\tau v_{\max}, \Big\|\boldsymbol{s}_0^{\text{goal}} - \boldsymbol{s}_{0,k}^{\text{ref}}\Big\|\Big\} \frac{\boldsymbol{s}_0^{\text{goal}} - \boldsymbol{s}_{0,t}^{\text{ref}}}{\Big\|\boldsymbol{s}_0^{\text{goal}} - \boldsymbol{s}_{0,t}^{\text{ref}}\Big\|}, \tag{5}$$

for $k \in \mathcal{I}_t$ and $\boldsymbol{s}_{0,t}^{\text{ref}} = \boldsymbol{s}_{0,t}$. In addition, we minimize the acceleration and jerk rates of the robot's motion by the following objectives

$$J^{\text{acce}}(\boldsymbol{u}_0) = \sum_{k=t}^{t+H-1} \boldsymbol{u}_{0,k}^\top \boldsymbol{u}_{0,k}, \tag{6}$$

and

$$J^{\text{jerk}}(\boldsymbol{u}_0) = \sum_{k=t}^{t+H-1} (\boldsymbol{u}_{0,k} - \boldsymbol{u}_{0,k-1})^\top (\boldsymbol{u}_{0,k} - \boldsymbol{u}_{0,k-1}). \quad (7)$$

To encourage safety between the robot and the pedestrians, we impose the following constraint that the distance between the robot and each pedestrian $i \in \mathcal{H}$ be greater than a safe speed-dependent distance

$$\|\boldsymbol{s}_{0,k+1} - \boldsymbol{s}_{i,k+1}\|_2^2 \geq d_{\min}^2 + \rho \,\|\boldsymbol{v}_{0,k+1}\|_2^2, \quad (8)$$

where $d_{\min} \in \mathbb{R}^+$ is the minimum allowed distance and $\rho \in \mathbb{R}^+$ is a scaling factor. The above constraint implies that the robot should keep further distances to the humans while moving with higher speed. We include the collision avoidance constraint as a soft constraint in the objective function by using a smoothed max penalty function as follows

$$J^{\text{coll}}(\boldsymbol{x}_0, \boldsymbol{s}_i) = \sum_{k=t}^{t+H-1} \operatorname{smax}\Big(d_{\min}^2 + \rho \,\|\boldsymbol{v}_{0,k+1}\|_2^2 - \|\boldsymbol{s}_{0,k+1} - \boldsymbol{s}_{i,k+1}\|_2^2\Big), \quad (9)$$

where the smoothed max penalty function is defined as

$$\operatorname{smax}(x) = \frac{1}{\mu} \log\big(\exp(\mu x) + 1\big),$$

with $\mu \in \mathbb{R}^+$ as a parameter that manipulates the smoothness of the penalty function.

The MPC objective function can be given by a weighted sum of those features as follows

$$\begin{aligned} J(\boldsymbol{u}_0, \boldsymbol{x}_0, \boldsymbol{s}_{1:N}) &= \omega^{\text{goal}} J^{\text{goal}}(\boldsymbol{x}_0) + \omega^{\text{acce}} J^{\text{acce}}(\boldsymbol{u}_0) \\ &\quad + \omega^{\text{jerk}} J^{\text{jerk}}(\boldsymbol{u}_0) + \sum_{i \in \mathcal{H}} \omega^{\text{coll}} J^{\text{coll}}(\boldsymbol{x}_0, \boldsymbol{s}_i), \end{aligned} \quad (10)$$

where $\omega^{\text{goal}}, \omega^{\text{acce}}, \omega^{\text{jerk}}$, and $\omega^{\text{coll}} \in \mathbb{R}^+$ are positive weights. Note that the penalty weight ω^{coll} chosen should be sufficiently large. Hence, the MPC formulation for each time step t is formulated as follows

$$\underset{\boldsymbol{u}_0, \boldsymbol{x}_0}{\text{minimize}} \; J(\boldsymbol{u}_0, \boldsymbol{x}_0, \boldsymbol{s}_{1:N}), \quad (11a)$$

$$\text{subject to: } (1), (2), \text{and } (3), \; \forall k \in \mathcal{I}_t, \quad (11b)$$

$$\text{given: } \; \boldsymbol{s}_{0:N,t-L+1:t}. \quad (11c)$$

C. Iterative Best-Response Implementation

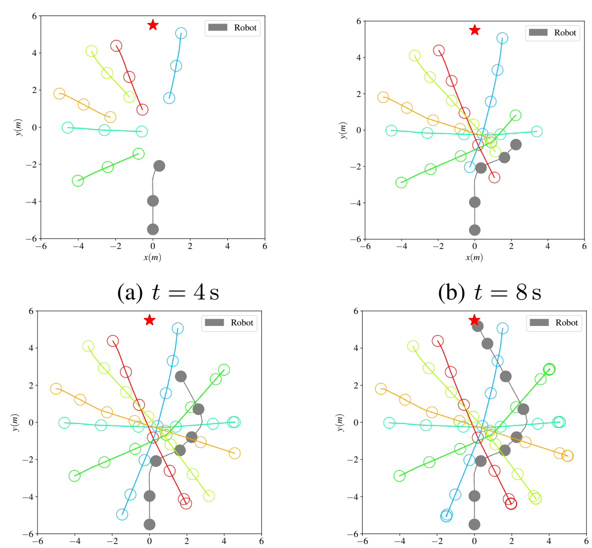
To solve the MPC problem (11) coupled with the Social-LSTM model, one can use gradient-based methods that requires gradient computation by back-propagating the LSTM's gradients [21]. However, due to the complexity of the neural network model, solving the MPC problem would be computationally intractable. Therefore, in this section, we present an iterative best-response approach [18], [23] inspired by the Nash equilibrium which at each time-step sequentially computes the neural network prediction and solves the MPC problem, for several iterations or until convergence. We use superscript $j \in \mathbb{N}$ in $\boldsymbol{u}_0^{(j)}$ and $\boldsymbol{x}_0^{(j)}$ to denote the results at the j'th iteration. If the algorithm converges, the resulting equilibrium is Nash equilibrium [18], [23]. The iterative best-response algorithm for solving MPC problem with the recursive prediction model is detailed in Algorithm 1. At $t = 0$, we initialize $\boldsymbol{u}_0^{(0)} = \boldsymbol{0}$, and at every time-step $t > 1$, the optimization is warm-started with the solution of the previous time-step.

Algorithm 1 Iterative Best-Response MPC Implementation

Require: $t,\ H,\ j_{\max} \in \mathbb{N},\ \epsilon \in \mathbb{R}^+,\ \boldsymbol{u}_0^{(0)} := \boldsymbol{u}_{0,t:t+H-1}^{(0)},\ \boldsymbol{s}_0^{(0)} := \boldsymbol{s}_{0,t+1:t+H}^{(0)},\ \boldsymbol{s}_{1:N,t-L:t}^{(0)}$

for $j = 1, 2, \ldots, j_{\max}$ **do**
  Predict $\boldsymbol{s}_{1:N}^{(j)} := \boldsymbol{s}_{1:N,t+1:t+H}^{(j)}$ recursively by (1) given $\boldsymbol{s}_0^{(j-1)}$.
  Solve (11) given $\boldsymbol{s}_{1:N}^{(j)}$ to obtain $\boldsymbol{u}_0^{(j)}$ and $\boldsymbol{x}_0^{(j)}$.
  if $\left\|\boldsymbol{u}_0^{(j)} - \boldsymbol{u}_0^{(j-1)}\right\| \leq \epsilon$ **then**
    return $\boldsymbol{u}_0^{(j)}$
  end if
end for
return $\boldsymbol{u}_0^{(j_{\max})}$

IV. Simulation Results



(a) $t = 4\,\text{s}$ (b) $t = 8\,\text{s}$ (c) $t = 12\,\text{s}$ (d) $t = 15.2\,\text{s}$

Fig. 2: Trajectories of the robot (proposed framework) and human pedestrians at several time steps in a circle crossing simulation with the robot visible to the humans. The destination of the robot is marked by a red star.

The planning algorithm was implemented in Python in which CasADi [24] and the built-in IPOPT solver [25] are used for formulating and solving the MPC problem, respectively. The parameters of MPC were chosen as: $\tau = 0.4\,\text{s}$, $H = 8$, $L = 8$, $v_{\max} = 1.0\,\text{m/s}$, $a_{\max} = 2.0\,\text{m/s}^2$, $d_{\min} = 0.8\,\text{m}$, $\rho = 0.5\,\text{s}^2$, $\mu = 30$, $\omega^{\text{goal}} = 10.0$, $\omega^{\text{acce}} = 10^{-1}$, $\omega^{\text{jerk}} = 10^{-1}$, $\omega^{\text{coll}} = 10^7$.

For social navigation simulations, we used the CrowdNav environment[1] [6] in which the human pedestrians are simu-

Method ↓	Success Rate (%)								Collision Rate (%)							
# Humans	5		6		7		8		5		6		7		8	
Scenario	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
MPC	**98.9**	**100**	**98.3**	**99.3**	**98.5**	**98.3**	**97.6**	**98.6**	1.1	**0.0**	1.5	0.5	1.5	1.1	2.3	1.0
CADRL	**98.9**	32.1	97.6	32.4	93.8	29.5	94.0	28.9	0.1	0.2	**0.0**	0.3	**0.0**	0.3	**0.0**	0.6
SARL	96.8	47.5	97.4	41.3	97.0	35.5	96.7	33.0	**0.0**	**0.0**	0.1	**0.0**	0.2	**0.0**	0.2	**0.0**

TABLE I: Statistical results in Circle and Square Crossing Scenarios - Success and Collision Rates

Method ↓	Discomfort Rate (%)								Average Travel Time (s)							
# Humans	5		6		7		8		5		6		7		8	
Scenario	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
MPC	0.9	**0.9**	**0.4**	**0.6**	1.4	1.3	**0.8**	**1.2**	**13.1**	**11.6**	**13.6**	**11.7**	**14.2**	**12.0**	14.8	**12.3**
CADRL	1.4	4.3	1.6	4.7	**1.0**	4.9	2.1	6.4	14.0	16.0	14.4	16.0	14.9	16.1	15.4	16.8
SARL	**0.6**	1.1	0.8	1.4	2.2	**1.1**	3.1	1.3	13.8	14.9	14.0	15.3	14.3	15.8	**14.7**	16.2

TABLE II: Statistical results in Circle and Square Crossing Scenarios - Discomfort Rates and Average Travel Time

lated using ORCA [2]. We utilize Trajnet++ benchmark[2] [22] for training Social-LSTM models using the ETH dataset [26]. We demonstrate the effectiveness of the proposed method by the trajectories of the robot and human pedestrians in a circle crossing simulation with 6 human pedestrians in Fig. 2. As can be seen from the figure, the robot is able to navigate among the humans and reach the goal in $15.2\,\mathrm{s}$ without any collisions. The robot is visible to the humans during motion to ensure that the interactive behaviors are captured by the prediction module.

To further validate the performance of the proposed method in comparison with different navigation algorithms, we collect and compare the following metrics:

- **Success rate**: the percentage of simulations in which all agents reach their individual destinations.
- **Collision rate**: the percentage of simulations that the minimum distance between the robot and the pedestrians is less than $0.8\,\mathrm{m}$ (violation of personal space).
- **Discomfort rate**: the percentage of simulations that the robot's projected path intersects with a pedestrian's projected path [20]. The projected path is defined as a line segment from the current position along with the direction of the velocity and the length proportional to the speed.
- **Average travel time**: Time to the destination in seconds (for the simulations with success).

Among the four metrics, the success rate and average time to the destination describe the path quality of the navigation algorithms. On the other hand, collision rate and discomfort rate are related to social conformity [20]. If the simulation reports neither a success nor a collision, it means a timeout has occurred wherein the robot has not been able to traverse to its destination. We compare our proposed MPC with two different RL algorithms including CADRL [4] and SARL [6]. This evaluation was based on 8000 simulations with varying numbers of human agents and randomized initial conditions. Both RL algorithms are trained using the circle-crossing scenario. The statistical results are shown in tables I and II.

Overall, the performance of MPC and RL algorithms in circle crossing simulations are highly comparable. Unlike the RL algorithms, the control policy in the proposed MPC formulation does not need any pre-training. Furthermore, RL algorithms are highly susceptible to domain shift. In this set of evaluations, they have been trained in the circle-crossing scenario but tested in both the circle-crossing and a square-crossing scenario. The success rate of RL algorithms is comparable to that of the MPC in the circle-crossing scenario, but drops dramatically in the square-crossing environment. The simulations don't show collision but report a high timeout rate, indicating the incapability of the policies to generate a feasible control path. This clearly illustrates the training domain-dependent nature of the RL techniques.

The CADRL policy encounters growing discomfort rates with increasing crowd densities. MPC on the other hand, shows a lower discomfort rate, which implies MPC approach can cause less discomfort to the human pedestrians. We also observe higher travel times for the RL techniques. In addition, the MPC approach presents significantly lower acceleration and jerk rates, which indicates more natural motion of the robot. However, we have chosen not to include those comparison results in this manuscript since it would be unfair to compare against RL techniques on metrics that don't account for them in their respective control design approaches.

V. Conclusions

This work presented a control method for navigating a robot in crowded environments. The control method combines MPC and a human trajectory prediction model based on Social-LSTM. We conducted extensive simulations to evaluate the performance of the proposed methods in comparison with RL algorithms. Our future work will focus on an extension of the control framework dealing with multiple robots navigating in a coordinated manner through crowds of humans. We also want to examine diversity in human behavior and introduce more sociability metrics in our evaluation to lower the gap between simulation and reality.

[1] https://github.com/vita-epfl/CrowdNav

[2] https://github.com/vita-epfl/trajnetplusplusbaselines

REFERENCES

[1] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.

[2] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE international conference on robotics and automation*. Ieee, 2008, pp. 1928–1935.

[3] 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.

[4] 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)*. IEEE, 2017, pp. 1343–1350.

[5] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[6] 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)*. IEEE, 2019, pp. 6015–6022.

[7] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 015–12 021.

[8] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4616–4623, 2021.

[9] T. Akhtyamov, A. Kashirin, A. Postnikov, and G. Ferrer, "Social robot navigation through constrained optimization: a comparative study of uncertainty-based objectives and constraints," *arXiv preprint arXiv:2305.02859*, 2023.

[10] 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*, vol. 52, no. 4, pp. 2289–2301, 2021.

[11] S. Poddar, C. Mavrogiannis, and S. S. Srinivasa, "From crowd motion prediction to robot navigation in crowds," *arXiv preprint arXiv:2303.01424*, 2023.

[12] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robotics and Automation Letters*, 2023.

[13] V. Vulcano, S. G. Tarantos, P. Ferrari, and G. Oriolo, "Safe robot navigation in a crowd combining nmpc and control barrier functions," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 3321–3328.

[14] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[15] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.

[16] Y. Liu, Q. Yan, and A. Alahi, "Social nce: Contrastive learning of socially-aware motion representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 118–15 129.

[17] P. Trautman, "Sparse interacting gaussian processes: Efficiency and optimality theorems of autonomous crowd navigation," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 327–334.

[18] J. L. V. Espinoza, A. Liniger, W. Schwarting, D. Rus, and L. Van Gool, "Deep interactive motion prediction and planning: Playing games with motion prediction models," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 1006–1019.

[19] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[20] J. Wang, W. P. Chan, P. Carreno-Medrano, A. Cosgun, and E. Croft, "Metrics for evaluating social conformity of crowd navigation algorithms," in *2022 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*. IEEE, 2022, pp. 1–6.

[21] P. Gupta, D. Isele, D. Lee, and S. Bae, "Interaction-aware trajectory planning for autonomous vehicles with analytic integration of neural networks into model predictive control," *arXiv preprint arXiv:2301.05393*, 2023.

[22] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7386–7400, 2021.

[23] 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)*. IEEE, 2018, pp. 2403–2410.

[24] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[25] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[26] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 261–268.