2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Real-time Model Predictive Control with Zonotope-Based Neural Networks for Bipedal Social Navigation

Abdulaziz Shamsah[1,2], Krishanu Agarwal[3], Shreyas Kousik[1,*], and Ye Zhao[1,*]

***Abstract*— This study addresses the challenge of bipedal navigation in a dynamic human-crowded environment, a research area that remains largely underexplored in the field of legged navigation. We propose two cascaded zonotope-based neural networks: a Pedestrian Prediction Network (PPN) for pedestrians' future trajectory prediction and an Ego-agent Social Network (ESN) for ego-agent social path planning. Representing future paths as zonotopes allows for efficient reachability-based planning and collision checking. The ESN is then integrated with a Model Predictive Controller (ESN-MPC) for footstep planning for our bipedal robot Digit designed by Agility Robotics. ESN-MPC solves for a collision-free optimal trajectory by optimizing through the gradients of ESN. ESN-MPC optimal trajectory is sent to the low-level controller for full-order simulation of Digit. The overall proposed framework is validated with extensive simulations on randomly generated initial settings with varying human crowd densities.**

I. Introduction

Bipedal navigation in complex environments has garnered substantial attention in the robotics community [1]–[4]. Social navigation is a particularly challenging problem due to the inherent uncertainty of the environment, unknown pedestrian dynamics, and implicit social behaviours [5]. Recently, there has been an increasing focus on social navigation for mobile robots in human environments [6]–[9]. Nonetheless, the exploration of social navigation for bipedal robots remains largely underexplored. This can be attributed to the intricate hybrid, nonlinear, and high degrees-of-freedom dynamics associated with bipedal locomotion.

In this study, we present an integrated framework for prediction and motion planning for socially acceptable bipedal navigation in human-crowded environments as shown in Fig. 1. We propose a navigation framework composed of two cascaded neural networks: a Pedestrian Prediction Network (PPN) for pedestrians' future trajectory prediction and an Ego-agent Social Network (ESN) for ego-agent social path planning. The ego-agent is aware solely of the neighboring pedestrians within a radius as shown in Fig. 1. Our neural networks output reachable sets for pedestrians and the ego-agent represented as zonotopes, a convex symmetric polytope. Zonotopes provide a balance between geometric complexity and computational efficiency. We specifically take advantage of two facts: (1) the Minkowski sum of two zonotopes is again a zonotope, allowing us to easily augment the zonotopes output by a neural network; and (2) collision checking a pair of zonotopes can be differentiated for use in gradient-based motion planning methods [10]–[13]. In this study, we use zonotopes to detect and avoid collisions by checking for intersections between the zonotopes corresponding to the ego-agent and pedestrians.

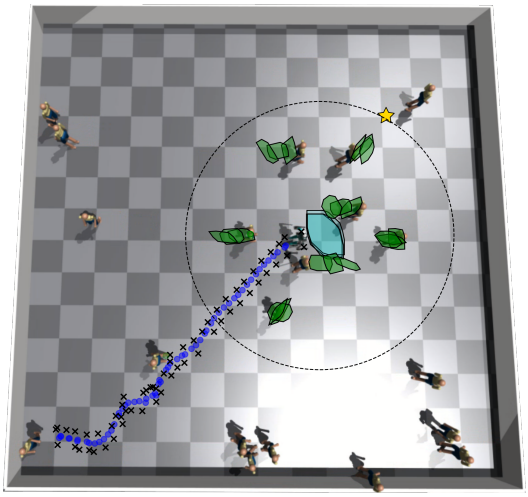

Fig. 1: Snapshot of the simulation environment with superimposed zonotopes for the proposed reachability-based social navigation framework. The environment is a 14 m $\times$ 14 m open space with 20 pedestrians.

Our framework integrates ESN in a model predictive controller (MPC) as shown in Fig. 2. The ESN-MPC optimizes over the output of the neural network, with reachability and collision avoidance constraints. It incorporates a reduced-order model (ROM) for the bipedal locomotion process and then sends optimal commands, i.e., center of mass (CoM) velocity and heading change, to the low-level controller on Digit for full-body joint trajectory design and control.

The main contributions of this study are as follows:

- A zonotope-based prediction and planning framework for bipedal navigation in a social environment.
- Novel loss functions to shape zonotopes that represent the future social trajectory of the ego-agent.
- A framework for hierarchically integrating the neural networks with an MPC and a low-level passivity controller for full-body joint control of Digit.

This article is outlined as follows. Sec. II is a literature review of related work. Sec. III introduces the problem

This work was supported in part by the Office of Naval Research (ONR) Award N000142312223, NSF grants IIS1924978, CMMI-2144309, FRR-2328254, and USDA 2023-67021-41397.

[1]George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0405 USA. ashamsah3@gatech.edu

[2] Mechanical Engineering Department, College of Engineering and Petroleum, Kuwait University, PO Box 5969, Safat, 13060, Kuwait

[3]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA.

*Co-senior authorships.

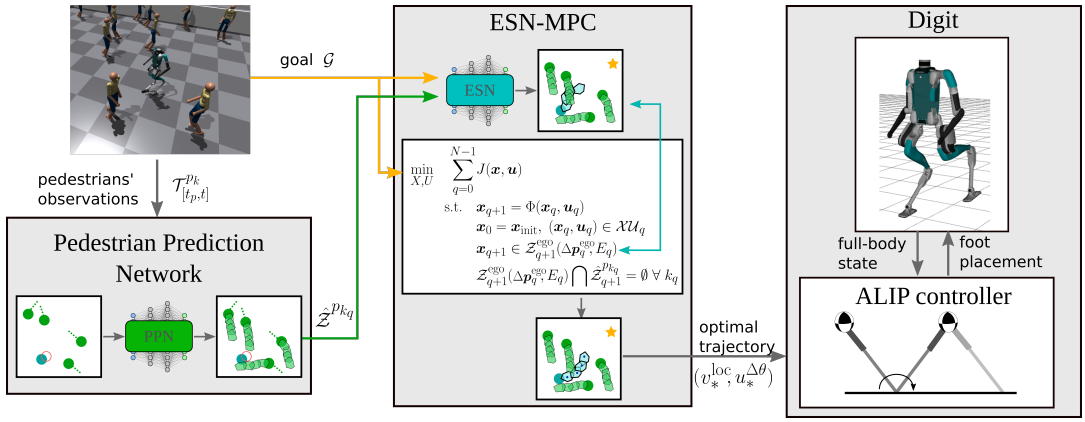

Fig. 2: Block diagram of the proposed framework. The framework is composed of two sub-networks: the Pedestrian Prediction Network (PPN) and the Ego-agent Social Network (ESN) shown in green and cyan, respectively (Sec. V). Given an environment with observed pedestrians and a goal location, PPN predicts the future pedestrians' reachable set. ESN-MPC optimizes through ESN to generate collision-free trajectories for Digit (Sec. VI). The optimal trajectory is then sent to the ALIP controller [14] to generate the desired foot placement for reduced-order optimal trajectory tracking. An ankle-actuated-passivity-based controller [15], [16] is implemented on Digit for full-body trajectory tracking.

we are seeking to solve. Then, the environment setup and zonotope preliminaries are in Sec. IV. Sec. V presents the neural network architecture and loss functions. Sec. VI formulates the problem as an MPC. Implementation details and results are in Sec. VII and conclusion is in Sec. VIII.

II. Related Work

Navigating an environment with humans in a socially compliant manner requires a proactive approach to motion planning [6]–[8]. In [8], the authors use opinion dynamics to proactively design motion plans for a mobile robot, without the need for human prediction models. It relies only on the observation of the approaching human position and orientation to form an opinion that alters the neutral path and avoids collisions with pedestrians. Gradient-based trajectory optimization is introduced in [7] to minimize the difference between the humans' future path prediction conditioned on the robot's plan and the nominal prediction. The studies of [6], [7] both assume that a minimally-invasive robot trajectory, with minimal effect on surrounding humans' nominal trajectory, is socially acceptable. In contrast, our work aims to learn the socially acceptable trajectory from human crowd datasets to minimize any heuristic biases on what a socially acceptable trajectory is.

Our framework is inspired by the human trajectory prediction community [17]–[20], where we aim to design a socially acceptable trajectory for the ego-agent that mimics the path learned from human crowd datasets. The work in [21] proposes an obstacle avoidance learning method that uses a Conditional Variational Autoencoder (CVAE) framework to learn a temporary target distribution to avoid pedestrians actively. However, during the learning phase, the temporary targets are selected heuristically. In contrast, we aim to learn such temporary waypoints from human crowd datasets to capture a heuristic-free socially acceptable path. In [17], the authors develop a simple yet, accurate CVAE architecture based on Multi-Layer Perceptrons (MLP) networks to predict crowd trajectories conditioned on past observations and intermediate endpoints. Our ESN follows a similar MLP-based CVAE architecture, where the ego-agent path is conditioned on the final goal location, surrounding pedestrians' future trajectories, and immediate change in the ego-agent state. Utilizing a non-complex network architecture is pivotal for enabling real-time planning and prediction when integrated into gradient-based motion planning for the ego-agent.

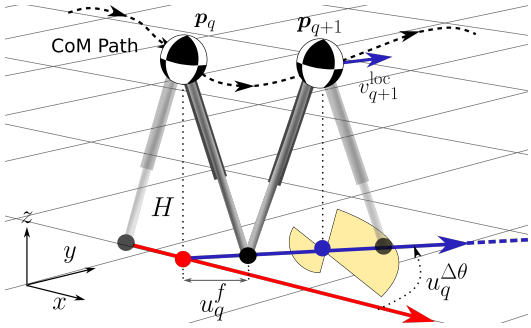

Fig. 3: Illustration of the Linear Inverted Pendulum model for two consecutive walking steps, with discrete states $\boldsymbol{p}_q$ and $\boldsymbol{p}_{q+1}$ at the contact switching time. The shaded yellow regions indicate the kinematics constraint on the control input $\boldsymbol{u}$ detailed in Sec. VI-A.

The authors in [10] present a Zonotope Alignment of Prediction and Planning (ZAPP) that relies on zonotopes to enable continuous-time reasoning for planning. They use trajectron++ [18] to predict obstacle trajectories as a Gaussian distribution. They construct a zonotope over these distributions, which leads to an overapproximation of the uncertainties. We propose learning these distributions directly as zonotopes, bypassing the initial step of predicting Gaussian distributions for pedestrian motion. This approach is computationally efficient and facilitates real-time integration with an MPC.

III. Problem Formulation

A. Robot Model

Consider a bipedal ego-agent with discrete step-by-step dynamics $\boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q)$, where $\boldsymbol{x}_q$ and $\boldsymbol{u}_q$ are the state

and control input respectively at the contact switching time of the q^{th} walking step. The robot's state $\boldsymbol{x} = (\boldsymbol{p}, v^{\text{loc}}, \theta)$, where $\boldsymbol{p} = (x, y)$ is the 2-D location in the global coordinate, v^{loc} is the local sagittal velocity, and θ the heading. The control input is $\boldsymbol{u}_q = (u_q^f \quad u_q^{\Delta\theta})$, where u_q^f is the local sagittal foot position relative to the CoM, and $u_q^{\Delta\theta}$ is the heading angle change as shown in Fig. 3.

The reduced-order model (ROM) used to design the walking motion of Digit is the Linear Inverted Pendulum (LIP) model [22]. For the LIP model, we assume that each step has a fixed duration T[1] [2], [23]. Then we build our model on the discrete local sagittal dynamics $(\Delta x_q^{\text{loc}}, v_q^{\text{loc}})$ [2] , where $\Delta x^{\text{loc}} = x_{q+1}^{\text{loc}} - x_q^{\text{loc}}$ and v_q^{loc} is the sagittal velocity at the local coordinate for the q^{th} walking step (see Fig. 3):

$$\Delta x^{\text{loc}}(u_q^f) = \left(v_q^{\text{loc}} \frac{\sinh(\omega T)}{\omega} + (1 - \cosh(\omega T)) u_q^f \right) \quad (1)$$

$$v_{q+1}^{\text{loc}}(u_q^f) = \cosh(\omega T) v_q^{\text{loc}} - \omega \sinh(\omega T) u_q^f \quad (2)$$

where $\omega = \sqrt{g/H}$, where g is the gravitational constant and H is the CoM height. Based on the local sagittal dynamics (1) and (2), we add heading angle θ_q to control the LIP dynamics in 2-D Euclidean space. The heading angle change is governed by $\theta_{q+1} = \theta_q + u_q^{\Delta\theta}$ across walking steps. Therefore the full LIP dynamics in 2-D Euclidean space become:

$$x_{q+1} = x_q + \Delta x^{\text{loc}}(u_q^f) \cos(\theta_q) \quad (3a)$$

$$y_{q+1} = y_q + \Delta x^{\text{loc}}(u_q^f) \sin(\theta_q) \quad (3b)$$

$$v_{q+1}^{\text{loc}} = \cosh(\omega T) v_q^{\text{loc}} - \omega \sinh(\omega T) u_q^f \quad (3c)$$

$$\theta_{q+1} = \theta_q + u_q^{\Delta\theta} \quad (3d)$$

For notation simplicity, hereafter, we refer to (3) as:

$$\boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q) \quad (4)$$

B. Environment Setup and Problem Statement

The ego-agent is tasked to navigate to a known goal location $\mathcal{G}$ in an open environment with $m \in \mathbb{N}$ observed pedestrians treated as dynamic obstacles. The pedestrian state $\mathcal{T}_{[t_p,t]}^{p_k}$ is the 2-D trajectory of pedestrian k observed over the discrete time interval $[t_p, t]$. The environment is partially observable as only the pedestrians in a pre-specified sensory radius of the ego-agent are observed. The path the ego-agent takes should ensure navigation safety, and promote social acceptability.

Definition III.1 (Navigation safety)**.** *Navigation safety is defined as maneuvering in human crowded environments while avoiding collisions with pedestrians, i.e., $\|\boldsymbol{p}_t - \mathcal{T}_t^{p_k}\| > d,\ \forall t, k$, where d represent the minimum allowable distance between the ego-agent and the pedestrians.*

[1]set to be equal to the timestep between frames in the dataset (0.4 s)

[2]the lateral dynamics are only considered in the ALIP model at the low level since they are periodic with a constant desired lateral foot placement

Definition III.2 (Socially acceptable path for bipedal systems)**.** *A path that a bipedal ego-agent takes in a human-crowded environment is deemed socially acceptable if it has an Average Displacement Error (ADE) $< \epsilon$ [3] when compared to ground truth data in the same environment.*

Based on the aforementioned definitions and environment setup the problem we aim to solve is as follows:

Problem III.1. *Given the discrete dynamics of the bipedal robot $\boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q)$ and an environment state $E = (\mathcal{T}_{[t_p,t]}^{p_k}, \mathcal{G})$, find a motion plan that promotes social acceptability for the bipedal ego-agent in a partially observable environment containing pedestrians while ensuring navigation safety.*

IV. Preliminaries

To solve the social navigation problem defined above, we propose a learning framework to learn socially acceptable reachable sets parameterized as zonotopes (Sec. V-A). Problem.III.1 is then reformulated as a step-by-step MPC problem with navigation safety constraints and implemented in real time on our Digit humanoid robot [24] (Sec. VI-D). This section begins by introducing the learning and environment assumptions, and zonotope preliminaries.

1) Environment Assumptions and Observations: In this work, we hypothesize that in a social setting, the information accessible by the ego-agent that is used to determine its future path $\mathcal{T}_{[t,t_f]}^{\text{ego}} = \{x_q^{\text{ego}}, y_q^{\text{ego}}\}_{q=t}^{t_f}$ [4] are three fold: (i) its final destination $\mathcal{G} = (x^{\text{dest}}, y^{\text{dest}})$ (ego-agent intent), (ii) the surrounding pedestrians' past trajectory $\mathcal{T}_{[t_p,t]}^{p_k} = \{x_q^{p_k}, y_q^{p_k}\}_{q=t_p}^{t}$ for the k^{th} pedestrian, and (iii) the ego-agent's social experience, i.e., its assumptions on how to navigate the environment in a socially-acceptable manner. We treat the social experience as latent information that is not readily available in human crowd datasets. Therefore we make the following assumption.

Assumption IV.1. *Learning the future trajectory of an ego-agent $\mathcal{T}_{[t,t_f]}^{\text{ego}}$ based on its final goal $\mathcal{G}$ and surrounding pedestrians' past trajectories $\mathcal{T}_{[t_p,t]}^{p_k}$, will learn the ego-agent's social experience.*

2) Zonotopes Preliminaries: A zonotope $\mathcal{Z} \in \mathbb{R}^n$ is a convex, symmetrical polytope paramterized by a center $\boldsymbol{c} \in \mathbb{R}^n$ and a generator matrix $G \in \mathbb{R}^{n \times n_G}$ (see Fig. 5).

$$\mathcal{Z} = \mathscr{Z}(\boldsymbol{c}, G) = \{\boldsymbol{c} + G\beta \mid \|\beta\|_\infty \leq 1\} \quad (5)$$

The Minkowski sum of $\mathcal{Z}_1 = \mathscr{Z}(\boldsymbol{c}_1, G_1)$ and $\mathcal{Z}_2 = \mathscr{Z}(\boldsymbol{c}_2, G_2)$ is $\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \mathscr{Z}(\boldsymbol{c}_1 + \boldsymbol{c}_2, [G_1\ G_2])$. To Check collisions between two zonotopes, [25, Lemma 5.1] is used:

Proposition IV.2. *([25, Lemma 5.1]) $\mathcal{Z}_1 \cap \mathcal{Z}_2 = \emptyset$ iff $\boldsymbol{c}_1 \notin \mathscr{Z}(\boldsymbol{c}_2, [G_1\ G_2])$.*

[3]ϵ represents the allowable deviation from the socially acceptable path. The Average Displacement Error denotes the average error between the planned path and the ground-truth path.

[4]the subscripts t_p, t, and t_f represent a discrete time indices denoting the past, current and future trajectories, respectively, where $t_p < t < t_f$.

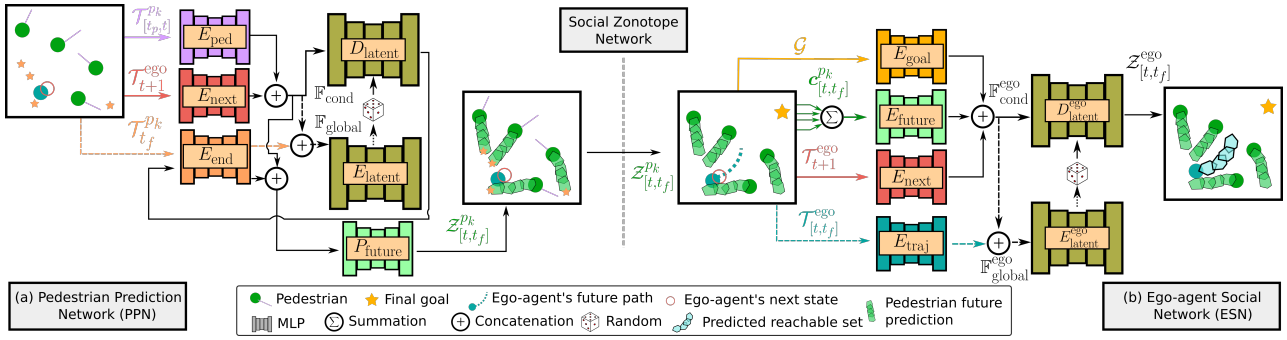

Fig. 4: (a) shows the pedestrian prediction network, conditioned on the pedestrian endpoints and the immediate change in the ego-agent's state. (b) shows the ego-agent social network conditioned on the pedestrians' future prediction, the immediate change in the ego-agent's state, and the ego-agent's goal location. Dashed connections are used during training only.

When $n = 2$ zonotopes can be represented as polytopes using the half-space representation $\mathcal{P} = \{x \mid Ax \leq b\}$, where $x \in \mathcal{P} \iff \max(Ax - b) \leq 0$ and $x \notin \mathcal{P} \iff \max(Ax - b) > 0$. To convert a 2-D zonotope from the center-generator representation to the half-space representation, we use the following proposition:

Proposition IV.3. *([13, Theorem 2.1]) Let* $C = \begin{bmatrix} -G[2,:] & G[1,:] \end{bmatrix}$ *and* $l_G[i] = \|G[:,i]\|_2$ *the half-space representation of a 2-D zonotope:*

$$A[i,:] = \frac{1}{l_G[i]} \cdot \begin{bmatrix} C \\ -C \end{bmatrix} \in \mathbb{R}^{2n_G \times 2} \tag{6}$$

$$b = A \cdot c + |AG| \; 1_{m \times 1} \in \mathbb{R}^{2n_G} \tag{7}$$

In this work, zonotopes are used to describe the social reachable set for the ego-agent. We seek to learn a sequence of social zonotopes $\mathcal{Z}_q^{\text{ego}}$, each of which contains two consecutive waypoints of the ego-agent's future social trajectory $\mathcal{T}_{[t,t_f]}^{\text{ego}}$.

Definition IV.1 (Social Zonotope $\mathcal{Z}_q^{\text{ego}}$)**.** *A social zonotope for the ego-agent's* q^{th} *walking step is* $\mathcal{Z}_q^{\text{ego}} = \mathscr{L}(\boldsymbol{c}_q, G_q)$, *such that* $\mathcal{T}_{[t,t_f]}^{\text{ego}} \in \bigcup_{q=t}^{t_f-1} \mathcal{Z}_q^{\text{ego}}$.

V. Social Zonotope Network

A. Learning Architecture

We set up a conditional variational autoencoder (CVAE) architecture to learn the ego-agent's future trajectory conditioned on the final destination goal, the immediate change in the ego-agent's state, and the surrounding pedestrians' past trajectories. The proposed architecture incorporates Multi-Layer Perceptrons (MLP) with ReLU non-linearity for all the sub-networks.

1) Pedestrian Prediction Network (PPN): The pedestrian prediction network (shown in Fig. 4(a)) is inspired by PECNet [17], where the endpoint of the pedestrian trajectory $\mathcal{T}_{t_f}^{p_k}$ is learned first, and then the future trajectory is predicted. Our proposed network deviates from PECNet in three ways. First, the pedestrian future trajectory is also conditioned on the immediate change in the ego-agent's state $\mathcal{T}_{t+1}^{\text{ego}}$ (shown in red in Fig. 4(a)). This coupling of the pedestrian prediction and ego-agent planning networks is intended to capture the effect of the robot's control on the future trajectories of the surrounding pedestrians [5], [7], and enable bidirectional influence for the entire ego-agent-pedestrian team. Second, the output of the network is the pedestrian's future reachable set parameterized as zonotopes $\mathcal{Z}_{[t,t_f]}^{p_k}$ rather than trajectories for robust collision checking and uncertainty parameterization [10]–[12]. Third, we replace the social pooling module with a simple ego-agent sensory radius threshold for computational efficiency.

The pedestrians' past trajectories $\mathcal{T}_{[t_p,t]}^{p_k}$ are encoded in E_{ped} as seen by the purple arrow in Fig. 4(a), while the incremental change in the ego-agent state representing the ego-agent control is encoded in E_{next} as seen by the red arrow in Fig. 4(a). This allows us to condition the prediction of the pedestrians' trajectory on the ego-agent's control. The resultant latent features $E_{\text{ped}}(\mathcal{T}_{[t_p,t]}^{p_k})$ and $E_{\text{next}}(\mathcal{T}_{t+1}^{\text{ego}})$ are then concatenated and used as the condition features $\mathbb{F}_{\text{cond}}$. The pedestrian's endpoint locations are encoded in E_{end} as seen by the orange arrows in Fig. 4(a). The resultant latent features $E_{\text{end}}(\mathcal{T}_{t_f}^{p_k})$ are then concatenated with $\mathbb{F}_{\text{cond}}$ as global features $\mathbb{F}_{\text{global}}$ and encoded in the latent encoder E_{latent}. We randomly sample features from a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ generated by the E_{latent} module, and concatenate them with $\mathbb{F}_{\text{cond}}$. This concatenated information is then passed into the latent decoder D_{latent}. Then D_{latent} outputs the predicted endpoint that is passed again through E_{end}. The output is concatenated again with $\mathbb{F}_{\text{cond}}$ and passed to P_{future} to output the predicted zonotopes of the pedestrians $\mathcal{Z}_{[t,t_f]}^{p_k}$.

2) Ego-agent Social Network (ESN): ESN architecture is shown in Fig. 4(b). The surrounding pedestrians' future zonotope centers $\boldsymbol{c}_{[t,t_f]}^{p_k}$ are aggregated through summation to take into account the collective effect of surrounding pedestrians while keeping a fixed architecture[5] [18]. The summed pedestrian features are then encoded in E_{future} as seen by the green arrows in Fig. 4(b). The goal location for the ego-agent is encoded in E_{goal}, while the incremental change in the ego-agent state is encoded in E_{next} as seen

[5]Other human trajectory learning modules include a social module to take into account the surrounding pedestrians effect such as social non-local pooling mask [17], max-pooling [19], and sorting [20].

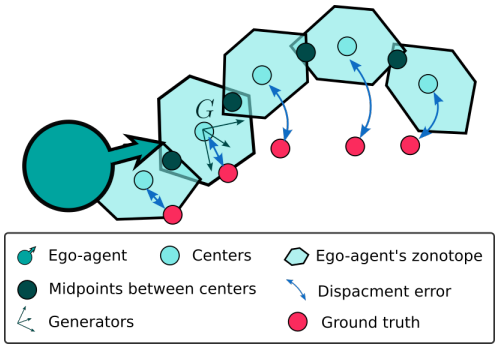

Fig. 5: Our zonotope shaping loss functions. The loss aims to learn interconnected zonotopes that engulf the ground truth path.

by the orange and red arrows respectively in Fig. 4(b). The resultant latent features $E_{\text{future}}(\sum_{k=1}^{m} \boldsymbol{c}^{p_k}_{[t,t_f]})$, $E_{\text{goal}}(\mathcal{G})$ and $E_{\text{next}}(\mathcal{T}^{\text{ego}}_{t+1})$ are then concatenated and used as the condition features $\mathbb{F}^{\text{ego}}_{\text{cond}}$ for the CVAE. The ground truth of the ego-agent's future trajectory $\mathcal{T}^{\text{ego}}_{[t,t_f]}$ is encoded in E_{traj} as shown by the cyan arrows in Fig. 4(b). The resultant latent features $E_{\text{traj}}(\mathcal{T}^{\text{ego}}_{[t,t_f]})$ are then concatenated with $\mathbb{F}^{\text{ego}}_{\text{cond}}$ as global features $\mathbb{F}^{\text{ego}}_{\text{global}}$ and encoded in the latent encoder E_{latent}. Similarly, we randomly sample features from a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ generated by the E_{latent} module, and concatenate them with $\mathbb{F}^{\text{ego}}_{\text{cond}}$. This concatenated information is then passed into the latent decoder D_{latent}, resulting in our prediction of the ego-agent's future reachable set $\mathcal{Z}^{\text{ego}}_{[t,t_f]}$.

Remark 1. *Including E_{next} in both neural networks facilitates seamless integration with a step-by-step MPC, as the MPC's decision variables ($\Delta \boldsymbol{p}^{\text{ego}}$) will be used as inputs to E_{next} as detailed in Sec. VI.*

B. Zonotope Shaping Loss Functions

The zonotope shaping loss functions are used for both PPN and ESN, where both outputs are parameterized as zonotopes. The goal of these loss functions is three folds: (i) penalize deviation of the centers of the zonotopes from the ground truth future trajectory; (ii) generate intersecting zonotopes for consecutive walking steps; and (iii) reduce the size of the zonotopes to avoid unnecessary, excessively large zonotopes. Based on these goals, the following is a list of loss functions to shape the zonotopes (Fig. 5):

1) Average displacement error between the predicted centers and midpoint of the ground truth trajectory $\mathcal{T}_{\text{mid},i}$:

$$\mathcal{L}_{ADE} = \frac{\sum_{i=1}^{t_f-1} \|\mathcal{T}_{\text{mid},i} - \boldsymbol{c}_i\|}{t_f - 1}$$

2) Final displacement error between the last predicted center and the final midpoint of the ground truth trajectory:

$$\mathcal{L}_{FDE} = \|\mathcal{T}_{\text{mid},t_f-1} - \boldsymbol{c}_{t_f-1}\|$$

3) The midpoint between the current center and previous center $\boldsymbol{c}^{p}_{\text{mid},i}$ is contained in the current zonotope:

$$\mathcal{L}_{\text{prev}} = \sum_{i=0}^{t_f-1} \text{ReLU}(A_i \cdot \boldsymbol{c}^{p}_{\text{mid},i} - b_i)$$

4) The midpoint between the current center and the next center $\boldsymbol{c}^{n}_{\text{mid},i}$ is contained in the current zonotope:

$$\mathcal{L}_{\text{nxt}} = \sum_{i=0}^{t_f-1} \text{ReLU}(A_i \cdot \boldsymbol{c}^{n}_{\text{mid},i} - b_i)$$

5) Regulating the size of the zonotope, by penalizing the norm of the generators such the neural network does not produce excessively large zonotopes that contain the ground truth trajectory:

$$\mathcal{L}_G = \|l_G[1] - d_1\| + \|l_G[1:] - d_2\|$$

where d_1 and d_2 are the desired lengths for the generators. We sum the zonotope shaping losses listed above in a single term $\mathcal{L}_{\mathcal{Z}}$. Similar to PECNet [17], we use Kullback–Leibler divergence to train the output of the latent encoder, aiming to regulate the divergence between the encoded distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$and the standard normal distribution $\mathcal{N}(0, \boldsymbol{I})$:

$$\mathcal{L}_{KL} = D_{KL}\big(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \| \mathcal{N}(0, \boldsymbol{I})\big)$$

The network is trained end to end using the following loss function: $\mathcal{L} = \mathcal{L}_{KL} + \mathcal{L}_{\mathcal{Z}}$.

VI. Social MPC

To enable safe navigation in the human-crowded environment, we propose to solve the following optimization problem:

$$\min_{X,U} \quad \sum_{q=0}^{N-1} J(\boldsymbol{x}, \boldsymbol{u}) \tag{8a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q) \tag{8b}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}}, \ (\boldsymbol{x}_q, \boldsymbol{u}_q) \in \mathcal{XU}_q \tag{8c}$$

$$\boldsymbol{x}_{q+1} \in \mathcal{Z}^{\text{ego}}_{q+1}(\Delta \boldsymbol{p}^{\text{ego}}_q, E_q) \tag{8d}$$

$$\mathcal{Z}^{\text{ego}}_{q+1}(\Delta \boldsymbol{p}^{\text{ego}}_q, E_q) \bigcap \mathcal{Z}^{p_{k_q}}_{q+1} = \emptyset, \ \forall \ k_q \tag{8e}$$

where the cost (8a) is designed to reach the goal and promote social acceptability, subject to the ROM dynamics (8b) (Sec. III-A). Constraint (8d) requires the ego-agent at the next $(q+1)^{\text{th}}$ walking step to stay within the reachable set, while constraint (8e) requires the ego-agent to avoid collision with the pedestrians. Next, we introduce the kinematics, reachability, and navigation constraints (Sec. VI-A-VI-B), and finally reformulate the MPC in (8) with a detailed version for implementation (Sec. VI-D).

A. Kinematics Constraints

To prevent the LIP dynamics from taking a step that is kinematically infeasible by the Digit robot the following constraint is implemented

$$\mathcal{XU}_q = \{(\boldsymbol{x}_q, \boldsymbol{u}_q) \mid \boldsymbol{x}_{\text{lb}} \leq \boldsymbol{x}_q \leq \boldsymbol{x}_{\text{ub}} \text{ and } \boldsymbol{u}_{\text{lb}} \leq \boldsymbol{u}_q \leq \boldsymbol{u}_{\text{ub}}\} \tag{9}$$

where $\boldsymbol{x}_{\text{lb}}$ and $\boldsymbol{x}_{\text{ub}}$ are the lower and upper bounds of $\boldsymbol{x}_q$ respectively, and $\boldsymbol{u}_{\text{lb}}$ and $\boldsymbol{u}_{\text{ub}}$ are the bounds for $\boldsymbol{u}_q$ (See yellow shaded region in Fig. 3). The detailed parameters in our implementation are specified in Table II.

B. Reachability and Navigation Safety Constraints

To enforce navigation safety (i.e., collision avoidance), we require that Digit remains in the social zonotope $\mathcal{Z}^{\text{ego}}$ and outside of the surrounding pedestrians reachable set $\hat{\mathcal{Z}}^{p_k}$.

1) Reachability constrains: For the robot's CoM to remain inside the desired zonotope for the next walking step $\mathcal{Z}^{\text{ego}}_{q+1}$, we represent the zonotope using half-space representation as shown in Prop. IV.3. The constraint is reformulated:

$$\max(A^{\text{ego}}\boldsymbol{p}^{\text{ego}} - b^{\text{ego}}) \leq 0 \tag{10}$$

2) Navigation safety constraint: For pedestrian collision avoidance, we require that the reachable set of the ego-agent does not intersect with that of the pedestrians for the corresponding step. Therefore, we create a new zonotope for the ego-agent as Minkowski sum of the ego-agent's zonotope and the pedestrian's zonotope centered around the ego-agent $\mathcal{Z}^{\text{mink}} = \mathscr{Z}(\boldsymbol{c}^{\text{ego}}, [G^{\text{ego}}\ G^{p_k}])$ to check for collision with the pedestrians zonotope following Prop. IV.2. We then represent $\mathcal{Z}^{\text{mink}}$ using half-space representation and require that the pedestrian is outside the combined set:

$$\max(A^{\text{mink}}\boldsymbol{p}_k - b^{\text{mink}}) > 0 \tag{11}$$

C. Cost Function

The MPC cost function is designed to drive the ROM state to a goal location $\mathcal{G}$. The terminal cost penalizes the distance between the current ROM state and the global goal state $\mathcal{G}$.

$$J_N(\boldsymbol{x}_N) = \|\boldsymbol{x}_N - \boldsymbol{x}_{\mathcal{G}}\|^2_{W_1} + \|\theta_N - \theta_{\mathcal{G}}\|^2_{W_2} \tag{12}$$

where $\boldsymbol{x}_{\mathcal{G}} = (\mathcal{G}, v_{\text{terminal}})$, and $\theta_{\mathcal{G}}$ is the angle between the ego-agent's current position and the final goal location.

D. MPC Reformulation with Ego-agent Social Network

According to the aforementioned costs and constraints for implementation, we reformulate our Ego-agent Social Network MPC (ESN-MPC) shown in (8) as follows:

$$\min_{X,U} \quad \sum_{\mathfrak{q}=0}^{N-1} J_N(\boldsymbol{x}_N) \tag{13a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{q+1} = \Phi(\boldsymbol{x}_q, \boldsymbol{u}_q) \tag{13b}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}},\ (\boldsymbol{x}_q, \boldsymbol{u}_q) \in \mathcal{XU}_q \tag{13c}$$

$$\max(A^{\text{ego}}_{q+1}\boldsymbol{p}^{\text{ego}}_{q+1} - b^{\text{ego}}_{q+1}) \leq 0 \tag{13d}$$

$$\max(A^{\text{mink}}_{q+1}\boldsymbol{p}_{k_{q+1}} - b^{\text{mink}}_{q+1}) > 0,\ \forall\ k_q \tag{13e}$$

VII. Implementation and Results

A. Training

The social path planner module introduced in Sec. V was trained on the UCY [26] and ETH [27] crowd datasets with the common leave-one-out approach, reminiscent of prior studies [17]–[19]. The models were trained on a data set that excludes UNIV from the training examples. We employ a historical trajectory observation $\mathcal{T}^{p_k}_{[-8,0]}$ and a prediction horizon $\hat{\mathcal{T}}^{\text{ego}}_{[0,8]}$, each spanning 8 timesteps (3.2 s) and only consider neighboring pedestrians that are within a radius of 4 m. The network architecture details are shown in Table I.

TABLE I: Network architecture parameters

Pedestrian Prediction Network	
E_{ped}	$16 \to 32 \to 16$
E_{end}	$2 \to 8 \to 16$
E_{nxt}	$2 \to 32 \to 16$
P_{future}	$50 \to 32 \to 16 \to 32 \to 70$
E_{latent}	$48 \to 8 \to 16 \to 32$
D_{latent}	$48 \to 32 \to 16 \to 32 \to 2$
Ego-agent Social Network	
E_{goal}	$2 \to 8 \to 16 \to 2$
E_{future}	$16 \to 64 \to 32 \to 16$
E_{nxt}	$2 \to 64 \to 32 \to 2$
E_{traj}	$16 \to 64 \to 32 \to 16$
E_{latent}	$36 \to 8 \to 50 \to 16$
D_{latent}	$36 \to 128 \to 64 \to 128 \to 70$

B. Pedestrian Simulation

We use SGAN (Social Generative Adversarial Network), a state-of-the-art human trajectory model, for simulating pedestrians [19]. SGAN is specifically designed to grasp social interactions and dependencies among pedestrians. It considers social context, including how people influence each other and move in groups. This is important for creating realistic simulations of pedestrian motion. Employing a different prediction model ensures a fair evaluation by eliminating any inherent advantages of our proposed method [7].

In our simulation framework, SGAN incorporates both the historical trajectories of pedestrians and the trajectory of the ego agent. This approach enhances the realism of the simulation by accounting for the interaction between the ego-agent and pedestrians within the environment.

C. Testing Environment Setup

The environment for all the following tests is an open space of 14×14 m^2 as shown in Fig. 1 and Fig. 8, with randomly generated pedestrians' initial trajectory. We test with 5, 15, and 30 pedestrians in the environment. The goal location is $\mathcal{G} = (10, 10)$ m, and the ego-agent starting position is uniformly sampled along the y-axis as such $\boldsymbol{x}_0 = (0, \mathcal{U}_{[0,13]}, 0)$ with $\theta_0 = 0$. The MPC is solved with a planning horizon of $N = 4$, and ESN-MPC parameters are included in Table. II. Simulations and training are conducted using a 16-core Intel Xeon W-2245 CPU and an RTX-5000 GPU with 64 GB of memory. The ESN-MPC is implemented using the do-mpc Python libraray [28]. Digit is simulated using the MuJoCo simulator provided by Agility Robotics [24] and visualized using Nvidia Isaac Gym [29] for a realistic human crowded environment set-up.

D. Low-level Full-Body Control

At the low level we use the Angular momentum LIP (dubbed as ALIP) planner introduced in [14], and a Digit's passivity controller [15] with ankle actuation which we have previously shown to exhibit desirable ROM tracking

TABLE II: ESN-MPC Parameters

parameter	value	parameter	value
$u_{\text{ub}}^{\Delta\theta}$	15°	$u_{\text{lb}}^{\Delta\theta}$	-15°
u_{ub}^{f}	0.4 m	u_{lb}^{f}	-0.1 m
d_1	0.1	d_2	0.005
v_{terminal}	0 m/s	n_G	4
W_1	3	W_2	1

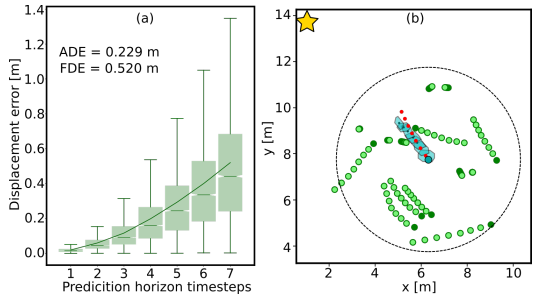

Fig. 6: Quantitative (a) and qualitative (b) results of ESN. (a) Shows the displacement error between the prediction of ESN $\boldsymbol{c}^{\text{ego}}$ and the ground truth data $\mathcal{T}_{\text{mid}}^{\text{ego}}$ (red dots in (b)). (b) shows a snapshot of ESN output, where the ego-agent's predicted zonotopes (cyan) contain the ground truth ego-agent data (red). ESN is conditioned on the goal position (yellow $\star$) and surrounding pedestrian future trajectories (green). The data is collected based on the UNIV dataset with 7831 unique frames. The solid line in (a) shows the average displacement error at each prediction horizon.

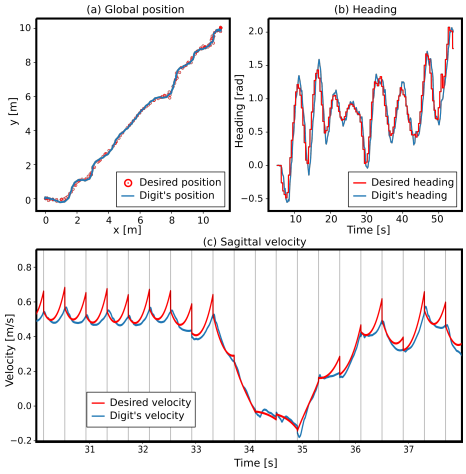

Fig. 7: Full-order simulation results of Digit tracking the desired trajectory from ESN-MPC. (a) shows Euclidean position tracking, (b) shows heading tracking, and (c) shows the sagittal velocity tracking in local coordinates.

results [16]. The ALIP controller used here is based on velocity rather than angular momentum, though the formulation remains similar to [14]. In this case, the ALIP model is equivalent to the LIP model used during planning. Here we set the desired walking step time and the desired lateral step width to be fixed at 0.4 s and 0.4 m, respectively.

E. Results and Discussion

In Fig. 6(a) we show that ESN produces an ADE$=0.229$ m over the prediction horizon of 7 timesteps[6], and a Final Displacement Error (FDE)$=0.52$ m. Fig. 6(b) shows a snapshot of the ESN social zonotope output $\mathcal{Z}^{\text{ego}}$ (cyan) compared to the ground truth data $\mathcal{T}_{\text{mid}}^{\text{ego}}$ shown in red.

Fig. 7 shows the tracking performance of integrating ESN-MPC with the low-level full-body controller [14]–[16]. We show the global Euclidean position tracking Fig. 7(a), heading angle tracking in Fig. 7(b), and local sagittal velocity tracking in Fig. 7(c). Fig. 1 and Fig. 8 show snapshots of the resultant trajectory at different walking steps.

Snapshots of ESN-MPC results at different walking steps are shown in Fig. 8. In Fig. 9(a), all three crowd densities produce relatively similar median velocities. At lower crowd density the velocity is more consistent. As expected, Fig. 9(b) shows that in less crowded areas, the ego-agent can reach the goal in fewer steps. With 30 pedestrians in the environment, it took more steps on average to reach the goal while maintaining a relatively similar velocity to the environments with fewer crowds (see Fig. 9(a)). This indicates that our framework can predict the future trajectory of the surrounding pedestrians, and is not required to come to a sudden stop.

[6]The prediction horizon timesteps is 7 and not 8, since the displacement error is calculated based on the middle points $\mathcal{T}_{\text{mid}}^{\text{ego}}$ of $\mathcal{T}_{[0,8]}^{\text{ego}}$

ESN-MPC produces a consistent and predictable behavior for the ego-agent. Predictability of the ego-agent behavior in a social context is desirable by pedestrians as it is perceived to be less disruptive. With 5 and 15 pedestrians in the environment our framework produced a 100% success rate by reaching the goal in 100 walking steps, while it managed a 90% success rate with 30 pedestrians as shown in Fig. 9(c). Due to the larger number of constraints, the time it takes to solve ESN-MPC decreases with increasing the number of pedestrians (See Fig. 9(d)). However, even with 30 pedestrians, the median of the frequency is higher than the required minimum for Digit implementation[7] as indicated by the dashed red line in Fig. 9(d). Finally, ESN-MPC can maintain a safe distance to the pedestrians in all three testing environments as indicated in Fig. 9(e).

VIII. Conclusion

This study introduced a novel framework for bipedal robot navigation in human environments, addressing a significant gap in the field of locomotion navigation. The proposed framework, which comprises the Pedestrian Prediction Network (PPN) and the Ego-agent Social Network (ESN), leverages zonotopes for efficient reachability-based planning and collision checking. Integrating ESN with MPC for step planning for Digit showed promising results for safe navigation in social environments.

References

[1] J.-K. Huang and J. W. Grizzle, "Efficient anytime clf reactive planning system for a bipedal robot on undulating terrain," *IEEE Trans. Robot.*, 2023.

[2] K. S. Narkhede, A. M. Kulkarni, D. A. Thanki, and I. Poulakakis, "A sequential mpc approach to reactive planning for bipedal robots using safe corridors in highly cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 831–11 838, 2022.

[3] J. Warnke, A. Shamsah, Y. Li, and Y. Zhao, "Towards safe locomotion navigation in partially observable environments with uneven terrain," in *IEEE Conference on Decision and Control*, 2020, pp. 958–965.

[7]The MPC is used as a step planner for Digit, and it only needs to be solved once per walking step before the swing phase ends.

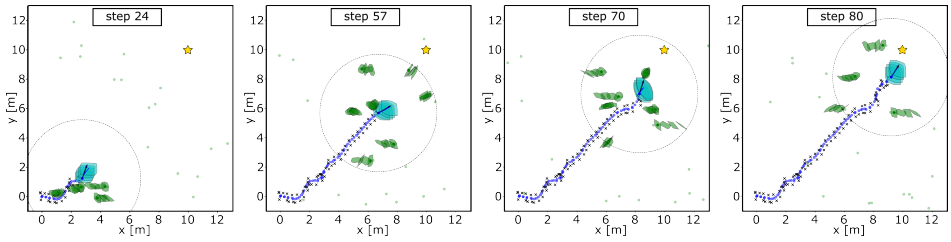

Fig. 8: Snapshots of ESN-MPC trajectory. The ego-agent (cyan) successfully reaches the goal (yellow ⋆) while avoiding pedestrians (green). The dashed circle is the sensory radius. Cayan dots represent the CoM of ROM, and black × is the desired foot placement. Green dots are unobserved pedestrians.

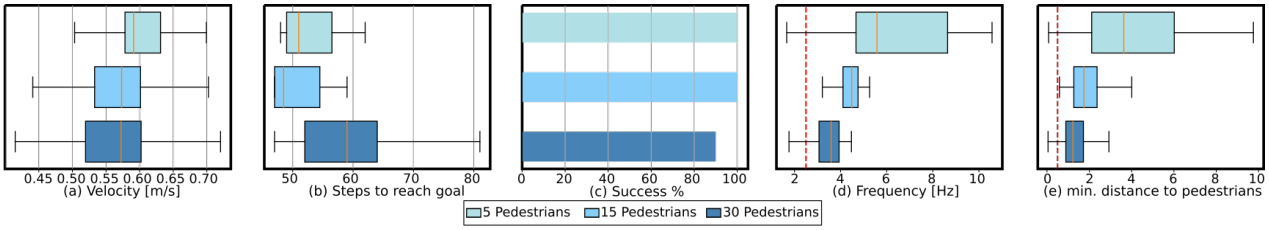

Fig. 9: ESN-MPC results: (a) velocity, (b) number of walking steps to reach within 1 m of the goal, (c) success rate, (d) frequency, and (e) minimum distance to pedestrians. The data consists of 10 different trials with random initial conditions and a fixed goal location (Sec. VII-C). Each trail is 100 walking steps. The velocity data is collected before reaching the goal, to avoid collecting a stop velocity. Success represents reaching within 1 m of the goal in 100 steps. The dashed red line in (d) is the minimum required frequency for full-order Digit simulation, and in (e) it represents d in Def. III.1.

[4] Y. Zhao, Y. Li, L. Sentis, U. Topcu, and J. Liu, "Reactive task and motion planning for robust whole-body dynamic locomotion in constrained environments," *International Journal of Robotics Research*, vol. 41, no. 8, pp. 812–847, 2022.

[5] 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.

[6] M. Moder and J. Pauli, "Proactive robot movements in a crowd by predicting and considering the social influence," in *IEEE International Conference on Robot and Human Interactive Communication*, 2022.

[7] S. Schaefer, K. Leung, B. Ivanovic, and M. Pavone, "Leveraging neural network gradients within trajectory optimization for proactive human-robot interactions," in *IEEE Int. Conf. Robot. Automat.* IEEE, 2021, pp. 9673–9679.

[8] C. Cathcart, M. Santos, S. Park, and N. E. Leonard, "Proactive opinion-driven robot navigation around human movers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2023, pp. 4052–4058.

[9] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe navigation in human occupied environments using sampling and control barrier functions," in *International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 5794–5800.

[10] L. Paparusso, S. Kousik, E. Schmerling, F. Braghin, and M. Pavone, "Zapp! zonotope agreement of prediction and planning for continuous-time collision avoidance with discrete-time dynamics."

[11] M. Selim, A. Alanwar, S. Kousik, G. Gao, M. Pavone, and K. H. Johansson, "Safe reinforcement learning using black-box reachability analysis," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 665–10 672, 2022.

[12] S. Kousik, P. Holmes, and R. Vasudevan, "Safe, aggressive quadrotor flight via reachability-based trajectory design," in *Dynamic Systems and Control Conference*, vol. 59162. American Society of Mechanical Engineers, 2019, p. V003T19A010.

[13] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[14] Y. Gong and J. W. Grizzle, "Zero Dynamics, Pendulum Models, and Angular Momentum in Feedback Control of Bipedal Locomotion," *Journal of Dynamic Systems, Measurement, and Control*, vol. 144, no. 12, 10 2022, 121006.

[15] H. Sadeghian, C. Ott, G. Garofalo, and G. Cheng, "Passivity-based control of underactuated biped robots within hybrid zero dynamics approach," in *International Conference on Robotics and Automation*. IEEE, 2017, pp. 4096–4101.

[16] A. Shamsah, Z. Gu, J. Warnke, S. Hutchinson, and Y. Zhao, "Integrated task and motion planning for safe legged navigation in partially observable environments," *IEEE Trans. Robot.*, pp. 1–22, 2023.

[17] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV: 16th European Conference, Part II 16*. Springer, 2020, pp. 759–776.

[18] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*, 2020.

[19] 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.

[20] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1349–1358.

[21] Y. Hong, Z. Ding, Y. Yuan, W. Chi, and L. Sun, "Obstacle avoidance learning for robot motion planning in human-robot integration environments," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[22] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, vol. 1. IEEE, 2001, pp. 239–246.

[23] S. Teng, Y. Gong, J. W. Grizzle, and M. Ghaffari, "Toward safety-aware informative motion planning for legged robots," *arXiv preprint arXiv:2103.14252*, 2021.

[24] A. Robotics, "Digit robot." [Online]. Available: agilityrobotics.com

[25] L. J. Guibas, A. T. Nguyen, and L. Zhang, "Zonotopes as bounding volumes." in *SODA*, vol. 3. Citeseer, 2003, pp. 803–812.

[26] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[27] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *IEEE International Conference on Computer Vision*, 2009.

[28] F. Fiedler, B. Karg, L. Lüken, D. Brandner, M. Heinlein, F. Brabender, and S. Lucia, "do-mpc: Towards fair nonlinear and robust model predictive control," *Control Engineering Practice*, vol. 140, 2023.

[29] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.