Pedestrian Dynamics Inference Using Control Barrier Functions and Mixed-Integer Quadratic Programming

Jaskaran Grover[1], Yiwei Lyu[2], Wenhao Luo[3], Changliu Liu[1], John Dolan[1], Katia Sycara[1]

***Abstract*— We develop algorithms for inferring long-term intentions and safety constraints of agents in a multiagent system. By safety constraints, we refer to the parameters of agents' collision-avoidance with other entities in the environment which could either be other agents or obstacles/walls. We model the agent's dynamics using a reactive optimization whose objective function captures the agent's desire to follow a nominal task-oriented control while its constraints capture collision-avoidance behavior. Given this model, we develop two robust mixed-integer programming algorithms that infer the task and safety parameters using measurements that may contain noise. To evaluate these algorithms, exhaustive numerical simulations are performed on synthetic datasets using metrics such as parameter estimation errors, average and final displacement errors and computation time. These algorithms are also tested and validated on a pedestrian dataset of human trajectories. We infer each human's desired velocity, its safety margins as well as a conservativeness parameter that models the willingness of a human to sacrifice optimality to maintain safety. We show that the learned parameters capture the true underlying model by rolling out the learned model and showing similarity between the ground truth trajectories and the reconstructed trajectories.**

I. INTRODUCTION

As robots are envisioned to co-exist with humans, frequent close-proximity interactions amongst robots and humans are inevitable. One critical challenge in programming the behavior of robots around humans is ensuring that the robots' motions are safe. This is all the more pressing when the underlying dynamic models of humans are unknown. Hence, it is necessary to develop methods for inferring the dynamics of these agents so that the control engineer can use those models to predict future behaviors of other agents for generating safe robot motions.

In this paper, we consider the problem of dynamics inference for a group of heterogeneous agents such as pedestrians. The objective is to identify a set of behavior-related parameters for each agent's dynamics model that describes their (i) long-term intention (such as goal, desired velocity etc.) and (ii) collision avoidance behavior around other agents or walls (e.g. underlying safety margin, aggressiveness etc). From the perspective of an observer watching these agents, this inference problem is challenging because the motion that the observer watches comes through the filters of goal-directed behavior and safety combined. Because of this, the observer cannot tell the extent to which safety constraints manifest in the agent's motion. Therefore, the observer must learn the safety margins of each agent as well to perform this dynamics disaggregation.

To address this challenge, we take recourse to control barrier functions based quadratic programs to model the dynamics of each agent [1]. This model captures each agent's intent to follow a nominal task-oriented plan as closely as possible unless this plan causes the agent to come too close to another agent or a wall. Additionally, it also captures the aggressiveness of an agent *i.e.* the extent of its unwillingness to sacrifice optimality for attaining collision-free motions. Finally, it also models cooperation among different agents to achieve collision-free behavior. We develop two mixed-integer quadratic program (MIQP) algorithms that learn the parameters of this model from the agents' trajectories. We explicitly consider safety constraints with other agents and obstacles/walls. Our proposed algorithms are decentralized, robust to model mismatch and do not require long training times (we show empirical results to support these claims). After validating these algorithms on several simulated datasets, we evaluate them on a pedestrian dataset called THöR [2] that contains human motion trajectories recorded in a controlled indoor experiment at Örebro University. These trajectories exhibit social interactions that occur in populated spaces like offices, thus making it suitable for evaluating our algorithms. Our results show that the learned parameters capture pedestrian dynamics accurately which we demonstrate by showing low ADE and FDE values.

Our work is most closely related to physics-based approaches [3] for modeling interactions between heterogeneous agents in addition to each agent's self-motivated dynamics. In this domain, the social force (SF) model [4], [5] is widely used to describe attractive forces from a goal with repulsive forces from other agents and obstacles. In [6] a sparse topological map of the dynamic environment is summarized consisting of varying state-destination pairs. The goal is inference by comparing the maximum likelihood to the state-destination pairs. In [7] SF-based interaction models are parameterized, and the parameters are learned offline. The newly acquired observations are then classified to the closest behavior. For human-agent interactions in a crowded environment, [8] proposed a method to classify each human as aware or not aware to the agent based on visual cues, to describe the sources of the repulsive force the agent receives from the dynamic environments. While social forces approaches consider perpetual and long-range repulsions

[1]J. Grover, J. Dolan, C. Liu, K. Sycara are with Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. {jaskarag,jdolan, cliu6,sycara}@andrew.cmu.edu. [2]Y. Lyu is with Electrical and Computer Engineering Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA, [3]W. Luo is with University of North Carolina, Charlotte

from other agents, our modeling choice automatically filters out agents far away from the ego agent in the ego-agent's dynamics. Additionally, our chosen model adds repulsions from only those agents/walls in the ego agent's dynamics that fall in the way of ego agent en-route to its goal, *i.e.* it implicitly captures directional 'local gaze'.

The outline of this paper is as follows. In section II, we describe our model of the dynamics each agent and pose the task + safety constraint inference problem. In section III, we propose our MIQP inference algorithms based on stationarity-residual and predictability-loss minimization. In section IV, we validate these algorithms on synthetic data sets, compare them using parameter inference and trajectory reconstruction errors and evaluate their robustness to measurement noise. In section V, we show the results of our algorithms on the THöR dataset. Finally, we conclude in section VI with directions for future work.

II. Multiagent Safe Task-Based Control and Inference Problem

A. Agents' dynamic model

We model each agent in the system as a single integrator that is velocity-controlled. Suppose there are a total of N_A+1 agents in the system. Additionally, let their be N_O obstacles (assumed polytopic). Let the position of the agents be $\boldsymbol{x}_i \in \mathbb{R}^2$ and their velocities be $\boldsymbol{u}_i \in \mathbb{R}^2$ $\forall i \in \{1, \cdots, N_A+1\}$. We assume that their dynamics are given by

$$\dot{\boldsymbol{x}}_i = \boldsymbol{u}_i \qquad \forall i \in \{1, \cdots, N_A+1\} \tag{1}$$

To keep the discussion simple, we focus on one agent located at $\boldsymbol{x}$. This agent has a primary task and we assume it can accomplish this task by using a reference control $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}$ in (1). We we represent this control as

$$\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}) = C(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{d}(\boldsymbol{x}). \tag{2}$$

Here $\boldsymbol{\theta}$ are the parameters capturing the task of that agent and are in general different for each agent. $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$ are some task oriented basis functions. This representation is general enough to capture elementary tasks such as (a) reaching a goal position and (b) maintaining a constant velocity.

In addition to performing the task, the ego agent must have a mechanism to maintain a safe distance, say D_s with the remaining N_A agents and N_O obstacles. To combine this safety requirement with task-satisfaction, the ego robot solves a QP that computes a controller closest to $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x})$ and satisfies $N_A + N_O$ safety constraints as follows:

$$\boxed{\begin{aligned} \dot{\boldsymbol{x}} = \boldsymbol{u}^* \ &= \arg\min_{\boldsymbol{u}} \|\boldsymbol{u} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})\|^2 \\ \text{subject to} \ &\underbrace{\begin{bmatrix} A^A \\ A^O \end{bmatrix}}_{A} \boldsymbol{u} \leq \underbrace{\begin{bmatrix} \boldsymbol{b}^A_{\gamma, D_s} \\ \boldsymbol{b}^O_{\gamma, D_s} \end{bmatrix}}_{\boldsymbol{b}} \end{aligned}} \tag{3}$$

Here $A^A \in \mathbb{R}^{N_A \times 2}$, $\boldsymbol{b}^A \in \mathbb{R}^{N_A}$ are defined so that the j^{th} row of A^A and the j^{th} entry of $\boldsymbol{b}^A_{\gamma, D_s}$ are

$$\boldsymbol{a}_j^T := -\Delta \boldsymbol{x}_j^T = -(\boldsymbol{x} - \boldsymbol{x}_j^o)^T \tag{4}$$

$$b_j := \gamma(\left\|\boldsymbol{x} - \boldsymbol{x}_j^o\right\|^2 - {D_s}^2) \ \forall j \in \{1, 2, \ldots, N_A+1\} \backslash i \tag{5}$$

Similarly, $A^O \in \mathbb{R}^{N_O \times 2}$, $\boldsymbol{b}^O \in \mathbb{R}^{N_O}$ are defined such that the i^{th} row of A^O and the i^{th} entry of $\boldsymbol{b}_{\gamma, D_s}$ are

$$\boldsymbol{a}_i^T := -(\boldsymbol{x} - \boldsymbol{y}_i^O)^T \tag{6}$$

$$b_i := \gamma(\left\|\boldsymbol{x} - \boldsymbol{y}_i^O\right\|^2 - {D_s}^2) \ \forall i \in \{1, 2, \ldots, N_O\} \tag{7}$$

where y_i^O is the point on obstacle O_i closest to the agent's instantaneous position $\boldsymbol{x}(t)$. Lastly, $\gamma > 0$ is a hyperparameter unique to each agent. Fig. 1 shows example trajectories computed by this model for an agent with one collision avoidance constraint relative to the rectangular obstacle. The reference control is $\hat{\boldsymbol{u}}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$. To get an intuitive understanding of γ, we conducted several simulations with increasing values of γ keeping $D_s, \boldsymbol{x}_0, \boldsymbol{x}_d$ and k_p fixed. It is evident that increasing γ makes the trajectory follow the nominal plan $\hat{\boldsymbol{u}}(\boldsymbol{x})$ more closely while still maintaining D_s relative to the obstacle.

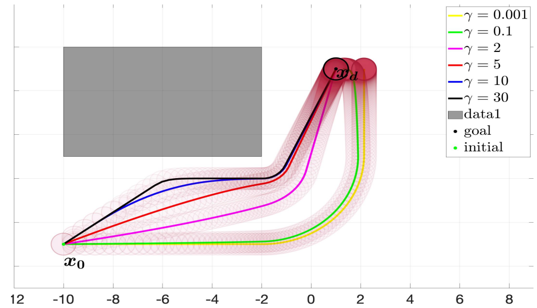

Fig. 1: Sample trajectories produced by (3) with one rectangular obstacle for different values of γ. Increasing γ makes the trajectory follow the reference controller more closely.

Going forward, we will assume that each agent in the multiagent system uses (3) as its underlying dynamic model. The task+constraint parameters $\{\boldsymbol{\theta}, \gamma, D_s\}$ are what distinguish one agent from another. There are several reasons in favor of using (3) as the model:

1) The nominal control $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})$ represents the agent's preferred plan should there be no other agents. This represents the self-motivated dynamics of that agent. Thus, having an optimization model the dynamics automatically encourages the agent's intent to follow this plan as much as possible while also ensuring that safety is respected if and when other agents show up.
2) This model implicitly captures the 'local gaze' of the ego agent. In our prior work [9], we showed that for the goal reaching task using (3), agents that do not lie in the way of the ego agent en-route to its goal do not influence its dynamics *i.e.* they lie outside its 'local gaze'. Thus, this model then automatically filters out agents far away from the ego agent that don't interfere with the ego's task.

B. Task + Safety Behavior Inference Problem Formulation

Before stating the inference problem for the multiagent system, we first state all assumptions on the observer's knowledge:

Assumption 1. *The observer knows the task functions* $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$ *of* $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x})$.

Assumption 2. *The observer knows the form of safety constraints* $A^A, \boldsymbol{b}^A_{\gamma,D_s}, A^O, \boldsymbol{b}^O_{\gamma,D_s}$.

Definition 1 (Multiagent Behavior Inference)**.** *The observer's problem is to infer parameters* $\{\boldsymbol{\theta}, \gamma, D_s\}$ *for each agent by monitoring its position* $\boldsymbol{x}(t)$ *and the positions of other agents* $\boldsymbol{x}^o_j(t)\ \forall j \in \{1, 2, \cdots, N_A\}$ *over some time.*

The observer will use a batch of K signal-response pairs *i.e.* $\left(\underbrace{\left(\boldsymbol{x}(k), \{\boldsymbol{x}^o_j(k)\}_{j=1}^{N_A}\right)}_{signal}, \underbrace{\boldsymbol{u}^*(k)}_{response} \right)\ \forall k \in \{1, 2, \cdots, K\}$ to compute an estimate of $\boldsymbol{\theta}, \gamma, D_s$. Next, we propose our algorithms to solve this problem using MIQP.

III. MIQP-based Robust Inference Algorithms

The general approach for inferring $\boldsymbol{\theta}, \gamma, D_s$ is to pose an empirical risk minimization algorithm that uses a reasonable heuristic as a loss. We propose two algorithms: the algorithm in section III-A considers the prediction error as a heuristic while the algorithm in section III-B considers a variant of the KKT loss proposed in [10] as a heuristic. Both these algorithms rely on the KKT conditions of (3). Thus, we state these conditions first before presenting these algorithms. Let $(\boldsymbol{u}^*, \boldsymbol{\lambda}^*)$ be the optimal primal-dual solution to (3). The KKT conditions are [11]:

1) Stationarity: $\boldsymbol{u}^*_{\boldsymbol{\theta}} = \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \frac{1}{2}A^T(\boldsymbol{x})\boldsymbol{\lambda}^*$
2) Primal Feasibility: $A(\boldsymbol{x})\boldsymbol{u}^* \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x})$
3) Dual Feasibility: $\boldsymbol{\lambda}^* \geq \boldsymbol{0}$
4) Complementary Slackness:
$\boldsymbol{\lambda}^* \odot \left(A(\boldsymbol{x})\boldsymbol{u}^* - \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x})\right) = \boldsymbol{0}$

Complementary slackness can be re-posed with an equivalent formulation by using the big-M approach [12]. This is done by augmenting the lower bounds $\boldsymbol{0} \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}) - A(\boldsymbol{x})\boldsymbol{u}^*$ and $\boldsymbol{0} \leq \boldsymbol{\lambda}^*$ with artificial upper bounds as follows:

$$\begin{aligned} \boldsymbol{0} \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}) - A(\boldsymbol{x})\boldsymbol{u}^* &\leq M\boldsymbol{z} \\ \boldsymbol{0} \leq \boldsymbol{\lambda}^* \qquad\qquad &\leq M(\boldsymbol{1} - \boldsymbol{z}) \end{aligned} \tag{8}$$

Here $\boldsymbol{z} \in \{0, 1\}^{N_A+N_O}$ are Boolean variables and M is a large number chosen as a hyperparameter. The Boolean variables $\boldsymbol{z}$ are also unknown and will be learned as part of the inference problem in the next section. Given these conditions, we are ready to develop the first inference algorithm.

A. Predictability Loss MIQP

The observer assumes that each agent uses (3) as the underlying model. Akin to this model, the observer poses a copy problem in which he treats $\boldsymbol{\theta}, \gamma, D_s$ as tunable knobs. These can be tuned until the *predicted* velocities computed by the solving the copy problem match with the measured velocities. This can be done by solving:

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \hat{\gamma}, \hat{D}_s, \{\hat{\boldsymbol{u}}_k\}_{k=1}^K = &\underset{\boldsymbol{\theta},\gamma,D_s,\{\boldsymbol{u}_k\}_{k=1}^K}{\arg\min} \sum_{k=1}^K \|\boldsymbol{u}_k - \boldsymbol{u}_k^{meas}\|^2 \\ \text{such that} \qquad &\boldsymbol{u}_k \text{ solves (3)}\ \ \forall k \in \{1, \cdots, K\} \end{aligned} \tag{9}$$

The cost in (9) is the sum of the deviations of the predicted velocities $\boldsymbol{u}_k$ from the measured velocities $\boldsymbol{u}_k^{meas}$. This is known as the *predictability loss* [13]. Naturally, it makes sense to minimize this loss only if the predicted velocities solve (3) which is posed as a constraint in (9). This is a bi-level problem which can be computationally difficult to solve. We convert this to a single level problem by replacing the inner problem with its KKT conditions:

$$\begin{aligned} &\hat{\boldsymbol{\theta}}, \hat{\gamma}, \hat{D}_s, \{\hat{\boldsymbol{u}}_k\}_{k=1}^K, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K, \{\hat{\boldsymbol{z}}_k\}_{k=1}^K, \{\hat{\boldsymbol{\delta}}_k\}_{k=1}^K = \\ &\underset{\substack{\boldsymbol{\theta},\gamma,D_s,\{\boldsymbol{u}_k\}_{k=1}^K, \\ \{\boldsymbol{\lambda}_k\}_{k=1}^K,\{\boldsymbol{z}_k\}_{k=1}^K,\{\boldsymbol{\delta}_k\}_{k=1}^K}}{\arg\min} \sum_{k=1}^K \|\boldsymbol{u}_k - \boldsymbol{u}_k^{meas}\|^2 + \rho \sum_{k=1}^K \|\boldsymbol{\delta}_k\|^2 \\ &\text{subject to} \\ &\quad \boldsymbol{0} \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}_k) - A(\boldsymbol{x}_k)\boldsymbol{u}_k \leq M\boldsymbol{z}_k \\ &\quad \boldsymbol{0} \leq \boldsymbol{\lambda}_k \qquad\qquad\qquad\quad \leq M(\boldsymbol{1} - \boldsymbol{z}_k) \\ &\quad \boldsymbol{z}_k \in \{0, 1\}^{N_A+N_O} \\ &\quad -\boldsymbol{\delta}_k \leq \boldsymbol{u}_k - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}_k) + \frac{1}{2}A^T(\boldsymbol{x}_k)\boldsymbol{\lambda}_k \leq \boldsymbol{\delta}_k \\ &\quad \boldsymbol{\theta}_L \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_U, \gamma_L \leq \gamma \leq \gamma_U, D_{sL} \leq D_s \leq D_{sU} \end{aligned} \tag{10}$$

The first term in the cost is the predictability loss and the second penalizes the slack variables which allow for violations in the stationarity. The constraints are linear in $\boldsymbol{\theta}, \gamma, \gamma D_s^2, \{\boldsymbol{u}_k, \boldsymbol{\lambda}, \boldsymbol{z}_k, \boldsymbol{\delta}_k\}_{k=1}^K$. Since $\{\boldsymbol{z}_k\}_{k=1}^K$ are Boolean, the overall problem is an MIQP.

B. Stationarity Loss MIQP

Another heuristic that can be used to solve the inference problem is the stationarity loss. This loss quantifies the residual of the stationarity condition evaluated on observed positions and velocities:

$$l_k^{Stat.} = \left\| \boldsymbol{u}_k^{meas} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}}(\boldsymbol{x}_k) + \frac{1}{2}A^T(\boldsymbol{x}_k)\boldsymbol{\lambda}_k \right\|^2 \tag{11}$$

This residual is quadratic in both $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}_k$. Using K observed signal-response pairs, the observer poses an empirical risk minimization problem that queries for $\boldsymbol{\theta}, \gamma, D_s$, $\{\boldsymbol{\lambda}_k, \boldsymbol{z}_k\}_{k=1}^K$ which minimize the total stationarity loss:

$$\begin{aligned} &\hat{\boldsymbol{\theta}}, \hat{\gamma}, \hat{D}_s, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K, \{\hat{\boldsymbol{z}}_k\}_{k=1}^K = \underset{\substack{\boldsymbol{\theta},\gamma,D_s, \\ \{\boldsymbol{\lambda}_k\}_{k=1}^K,\{\boldsymbol{z}_k\}_{k=1}^K}}{\arg\min} \sum_{k=1}^K l_k^{Stat.} \\ &\text{subject to} \\ &\quad \boldsymbol{0} \ \leq \boldsymbol{b}_{\gamma,D_s}(\boldsymbol{x}_k) - A(\boldsymbol{x}_k)\boldsymbol{u}^* \leq M\boldsymbol{z}_k \\ &\quad \boldsymbol{0} \ \leq \boldsymbol{\lambda}_k \leq M(\boldsymbol{1} - \boldsymbol{z}_k) \\ &\quad \boldsymbol{z}_k \ \in \{0, 1\}^{N_A+N_O} \\ &\quad \boldsymbol{\theta}_L \ \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_U, \gamma_L \leq \gamma \leq \gamma_U, D_{sL} \leq D_s \leq D_{sU} \end{aligned} \tag{12}$$

TABLE I: Evaluation with perfect measurements

Algorithm \ Metric	ADE [m]	FDE [m]	Time [s]
Stat. Loss MIQP	0.0012 ± 0.0014	0.044± 0.0534	8.17 ± 0.49
Pred. Loss MIQP	0.0012 ± 0.0014	0.044± 0.0539	1.63 ± 0.067
Pred. Loss (warm start)	0.0012 ± 0.0014	0.044± 0.0534	1.61 ± 0.06

TABLE II: Evaluation with noisy measurements

Algorithm \ Metric	ADE [m]	FDE [m]	Time [s]
Stat. Loss MIQP	0.0476 ± 0.1047	0.806± 1.432	8.008± 0.0827
Pred. Loss MIQP	0.3734 ± 0.3718	12.90± 12.01	1.6 ± 0.082
Pred. Loss (warm start)	0.3317 ± 0.3461	9.67 ± 10.281	1.592± 0.048

Since the cost is quadratic and constraints are linear in $(\boldsymbol{\theta}, \gamma, \gamma D_s^2, \{\boldsymbol{u}_k, \boldsymbol{\lambda}_k, \boldsymbol{z}_k\}_{k=1}^{K})$, this problem is also an MIQP.

IV. Results: Validation on Synthetic Datasets

Before testing these algorithms on the human trajectory dataset, we evaluate them by on a simulated dataset. The nominal task for the agent is to follow a constant velocity $\boldsymbol{v}_d$. We generated trajectory data of a single agent using (3). We modeled the environment after the map in the THöR dataset by manually converting the walls and obstacles into polytopes. The observer's problem is to infer $\boldsymbol{v}_d, \gamma, D_s$ of this agent from the obtained trajectories using (9) and (12). If we model the task reference control as fixed-speed goal-directed behavior, the resultant task control is $\hat{\boldsymbol{u}} = -s_d \frac{(\boldsymbol{x}-\boldsymbol{x}_d)}{\|(\boldsymbol{x}-\boldsymbol{x}_d)\|}$ because this control is not linear in $s_d, \boldsymbol{x}_d$ as required in (2). Hence, we chose to model it as a constant velocity control $\hat{\boldsymbol{u}} = \boldsymbol{v}_d$.

To test the robustness of these algorithms, we consider two scenarios: scenario (1) with no noise in the measured velocities and in scenario (2) we add zero mean Gaussian noise with 2m/s standard deviation to the velocities. This level of noise reflects the maximum noise we expect in real human motions. We compare the performance of (9) and (12) using the ADEs and FDEs of the trajectories generated using the inferred parameters relative to the ground truth trajectories. We also try to see if there is any benefit of warm-starting the predictability loss MIQP (9) with the solution returned by the stationarity loss MIQP (12). To assess repeatability, we conduct ten simulations with a randomly chosen initial position of the agent. The hyperparameter ρ for the predictability loss MIQP was chosen systematically by tuning performance on a validation dataset. Table I shows the ADEs and FDEs averaged over ten runs with noiseless demonstrations while Table II shows these errors with noisy demonstrations. It is evident that while both solvers give accurate reconstruction when the demonstrations are noiseless, it is the stationarity loss MIQP (12) which exhibits a great amount of robustness to measurement noise.

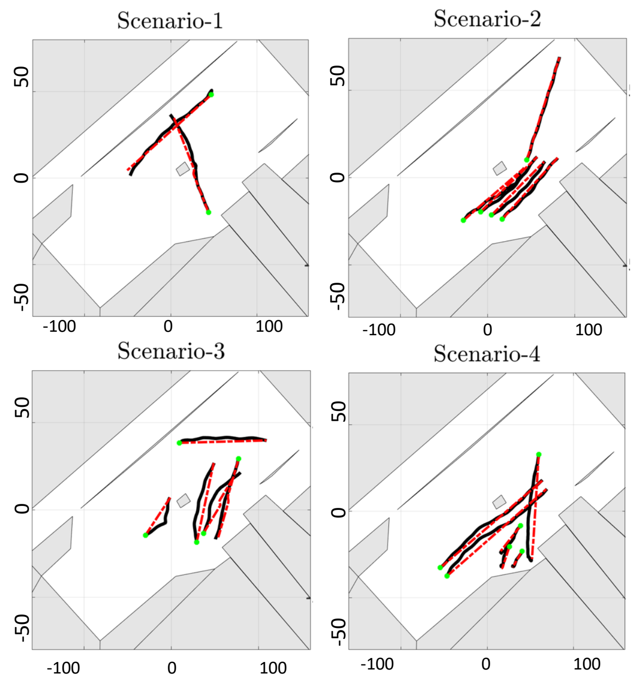

Fig. 2: Comparison between reconstructed trajectories with learned parameters (red) and ground truth trajectories (black) on four scenarios in THöR dataset. X,Y axis in meters.

V. Results: Validation on Human Datasets

Next, we evaluate these algorithms on the trajectories from the THöR dataset [2]. Due to the constant velocity assumption, we chose 15 scenarios from the dataset with selected time intervals of trajectories over which the constant velocity assumption was valid for most of the pedestrians. Parameter inference is then conducted on each pedestrian independently using our proposed algorithms.

In the interest of space, we only report results obtained using the stationarity loss MIQP. Fig. 2 shows these results. The black lines are ground truth demonstrations for four sample scenarios involving upto seven humans. We ran (12) on them to infer $\gamma, D_s, \boldsymbol{v}_d$ of each human individually. Then, we used these parameters to roll out (3) and obtained trajectories shown in red. These trajectories almost overlap with the given demonstrations showing that the learned parameters capture the underlying behavior. In future work, we will consider richer tasks as well as allow for γ, D_s to be time-varying to capture time-varying behavior of the humans.

VI. Conclusions

In this paper, we considered the problem of simultaneously inferring task and safety constraint parameters of individual agents of a multiagent system. We modeled the agents using control barrier functions and developed the predictability loss MIQP and the stationarity loss MIQP to solve the inference problem. We demonstrated how accurate estimates of underlying parameters can be reconstructed using these algorithms on a both a single-agent scenario and validated them on a real human dataset.

References

[1] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[2] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020.

[3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[4] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier, "Intentional motion on-line learning and prediction," *Mach. Vision Appl.*, vol. 19, no. 5–6, p. 411–425, sep 2008. [Online]. Available: https://doi.org/10.1007/s00138-007-0116-9

[5] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 464–469.

[6] J. Elfring, R. Van De Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 591–602, 2014.

[7] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5940–5945.

[8] S. Oli, B. L'Esperance, and K. Gupta, "Human motion behaviour aware planner (hmbap) for path planning in dynamic human environments," in *2013 16th International Conference on Advanced Robotics (ICAR)*, 2013, pp. 1–7.

[9] J. Grover, C. Liu, and K. Sycara, "Deadlock analysis and resolution in multi-robot systems (extended version)," 2019.

[10] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *2011 IEEE international symposium on intelligent control*. IEEE, 2011, pp. 613–619.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] C. Dong, Y. Chen, and B. Zeng, "Generalized inverse optimization through online learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[13] A. Aswani, Z.-J. Shen, and A. Siddiq, "Inverse optimization with noisy data," *Operations Research*, vol. 66, no. 3, pp. 870–892, 2018.