Simultaneously learning safety margins and task parameters of multirobot systems

Jaskaran Grover, Changliu Liu, Katia Sycara

***Abstract*—We present an algorithm for learning constraint and objective function parameters of optimization-based controllers used in multirobot systems. Our proposed approach uses position-velocity measurements of each robot in the team to perform this inference. The motivation to learn these parameters stems from the need to infer an agent's intent for accurate predictions of motion in a multiagent system. We develop an extension of our prior work in which we performed task learning assuming constraint parameters were known. In this work, we perform simultaneous learning of constraint and cost function parameters by posing it as a constrained nonconvex optimization problem. The cost function parameters that we learn encode information of the task being performed by each robot in the team whereas the constraint parameters encode information about individual safety margin distances and size of the safe control set for each robot. Our simulation results show the accurate reconstruction of both the constraint and cost function parameters and we analyze some failure cases.**

I. Introduction

Optimization-based control synthesis techniques have become very useful for synthesizing task-based controllers for multirobot systems, which also guarantee collision avoidance [7, 16]. While there is a lot of work on control synthesis for various multirobot tasks, the problem of inferring tasks from observations has only begun to receive attention in the context of multirobot teams [14, 6, 8, 9]. Characterizing the drive behind the actions of a robot team has several applications; for example, an untrained multirobot team can learn by imitating an expert team or one controlled by a human user [18]. Furthermore, addressing how easy it is for the intent of a team to be inferred, reveals the vulnerability of the team to attacks by an adversarial observer.

Algorithms based on inverse optimal control (IOC) can be used to reverse engineer the reward function parameters of a robot using measurements of its states and control inputs [5, 10, 13]. However, these algorithms assume that the robot minimizes a long-term cost function integrated over a horizon. On the other hand, several existing controllers devised for multirobot systems try to solve a reactive optimization problem that minimizes deviation from a task-based controller while satisfying safety constraints [15, 17]. The control inputs are generated at every time step by solving a finite-dimensional optimization problem instead of an infinite dimensional optimal control problem. Some common examples of such controllers include include barrier-function based quadratic programs (CBF-QPs) [1], safe-set controller [12] and recirpocal velocity obstacles [15]. This prevents direct application of IOC based approaches for inference of tasks involving multirobot systems. Furthermore, even in the IOC literature, relatively little work has been done on constraint inference. Objective function parameters are inferred assuming constraints on the states are known. Existing optimization based controllers used in multirobot systems encode task information in the objective and collision avoidance as constraints which depend on safety margins. Thus, for the purpose of task inference, assuming that the safety margins are known, is an unrealistic assumption since the observed dynamics of the robots come through the filter of both objective function and constraints.

Given these limitations, we are interested in addressing how can an observer simultaneously infer the parameters of the tasks as well as safety constraints by observing each robot in a multirobot team given that the robots use optimization-based controllers. Recent literature on inverse optimization (IO) has developed approaches to address these questions. While IO algorithms have been explored in finance [2] and OR [4], they have not been explored as much in robotics. In this work, we develop an extension to the algorithm proposed in [11] and reformulate it to perform simultaneous task and constraint inference of individual robots in a multirobot system. We assume that robots use CBF-QPs for control synthesis. The training data set needed to perform inference consists of pairs of exogenous signals to the agent's forward optimization problem, and the agent's decisions made in response to those signals. In our context, we treat the positions of robots as the exogenous signals and the velocities computed by CBF-QPs as decisions, to perform task inference.

The outline of this paper is as follows. In Sec. II, we briefly review the multirobot task completion and collision avoidance control approach and pose a mathematical formulation of the task and constraint inference problem. The main technical contributions start from Sec. III. We develop an extension to a previously proposed IO algorithm and develop a nonconvex optimization algorithm for task + constraint inference inference. In Sec. IV, we present numerical results for inference of controller gains, goal locations, safety margin distances and size of safe control set of each robot in a multirobot system using our proposed algorithm. We summarize our work in Sec. V and conclude with directions for future work.

II. Multirobot Safe Control and Task + Constraint Inference

A. The Forward Problem - Control for Task Completion

In our formulation, we assume that each robot uses CBF-QPs to synthesize safe controllers for task completion [17]. Suppose there are $M+1$ robots in the system. From the perspective of an ego robot, the remaining M robots are "obstacles" who all cooperate with the ego robot to avoid collisions. In the following, the focus is on the ego robot. This robot follows single-integrator dynamics *i.e.*

$$\dot{\boldsymbol{x}} = \boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} = (p_x, p_y) \in \mathbb{R}^2$ is its position and $\boldsymbol{u} \in \mathbb{R}^2$ is its velocity (*i.e.* the control input). Suppose there is a nominal controller for performing the primary task given by

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

Here, $\boldsymbol{\theta}_{task} \in \mathbb{R}^p$ is the task-parameter the observer wishes to infer and $C(\boldsymbol{x}), \boldsymbol{d}(\boldsymbol{x})$ are known functions. For example, if the ego robot's task is to reach a goal at $\boldsymbol{x}_d$, the robot can use $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$. If the observer wishes to infer the goal $\boldsymbol{x}_d$ (assuming the gain k_p is known), then $\boldsymbol{\theta}_{task} = \boldsymbol{x}_d$, so choosing $C(\boldsymbol{x}) = k_p$, $\boldsymbol{d}(\boldsymbol{x}) = -k_p\boldsymbol{x}$ gives $C(\boldsymbol{x})\boldsymbol{\theta}_{task} + \boldsymbol{d}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$.

In addition to performing the task, the ego robot must have a mechanism to maintain a safe distance, say D_s with the remaining robots to avoid collisions. 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 M safety constraints as follows:

$$\begin{aligned} \boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \quad & \|\boldsymbol{u} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x})\|^2 \\ \text{subject to} \quad & A(\boldsymbol{x})\boldsymbol{u} \le \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x}). \end{aligned} \tag{3}$$

Here $A(\boldsymbol{x}) \in \mathbb{R}^{M\times 2}$, $\boldsymbol{b}(\boldsymbol{x}) \in \mathbb{R}^M$ are defined such that the j^{th} row of A is $\boldsymbol{a}_j^T$ and the j^{th} element of $\boldsymbol{b}_{\boldsymbol{\theta}_{const.}}$ is b_j. For CBF-QP based formulation, these are given by:

$$\begin{aligned} \boldsymbol{a}_j^T(\boldsymbol{x}) &:= -\Delta\boldsymbol{x}_j^T = -(\boldsymbol{x} - \boldsymbol{x}_j^o)^T \\ b_j(\boldsymbol{x}) &:= \frac{\gamma}{2}(\|\Delta\boldsymbol{x}_j\|^2 - D_s^2)\ \forall j \in \{1, 2, \dots, M\}. \end{aligned} \tag{4}$$

Here $\{\boldsymbol{x}_j^o\}_{j=1}^M$ are the positions of the remaining M robots. $\gamma > 0$ is a parameter that describes the size of the set of feasible controls that ensure collision-free motions. A larger γ makes a larger set of controls available for safe collision free operation to this QP. γ is typically a hyper-parameter that is set by the control engineer much like the safety margin distance D_s. As an observer, it is reasonable to assume that the robots have this form of safety mechanism for control synthesis, but assuming that the safety margins of each robot are known is unrealistic. Thus, we define $\boldsymbol{\theta}_{const.} = (\gamma, D_s)$ as the constraint parameters that are unknown to the observer and must be inferred.

The ego robot solves this QP at every time step to determine its optimal control $\boldsymbol{u}^*$ which ensures safety parametrized by $\boldsymbol{\theta}_{const.}$ while encouraging satisfaction of the task parametrized by $\boldsymbol{\theta}_{task}$. This control depends on both the task and safety parammeters because the cost function of (3) depends on $\boldsymbol{\theta}_{task}$ and constraints depend on $\boldsymbol{\theta}_{const.}$. To emphasize this dependence, let us denote it as $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x})$ where $\boldsymbol{\theta} = (\boldsymbol{\theta}_{task}, \boldsymbol{\theta}_{const.})$ are the parameters the observer wishes to infer.

B. The Inverse Problem-Task + Constraint Inference

We focus here on the ego robot and pose the inference problem for this robot. The inference approach we propose can be easily extended to perform inference for multiple robots in parallel, so the focus is on the ego robot. The observer monitors this robot *i.e.* tracks its position $\boldsymbol{x}(t)$, its velocity *i.e.* $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x}(t))$ and additionally, tracks the positions of other robots *i.e.* $\{\boldsymbol{x}_j^o(t)\}_{j=1}^M$. The observer's problem is to infer both the task parameter $\boldsymbol{\theta}_{task}$ and constraint parameters $\boldsymbol{\theta}_{const.} = (\gamma, D_s)$ based on the knowledge that the optimal control of the ego robot, $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x}(t))$, is computed using (3) in response to the ego robot's position at $\boldsymbol{x}(t)$ and obstacles' positions at $\{\boldsymbol{x}_j^o(t)\}_{j=1}^M$ (the exogenous signals). Let us state all the assumptions on the observer's knowledge.

Assumption 1. *The observer knows that the ego robot's cost function is of the form* $\|\boldsymbol{u} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x})\|^2$

Assumption 2. *The observer knows the task functions* $C(\boldsymbol{x}), d(\boldsymbol{x})$ *of* $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x}) = C(\boldsymbol{x})\boldsymbol{\theta}_{task} + d(\boldsymbol{x})$ *in the cost.*

Assumption 3. *The observer knows the form of safety constraints* $A(\boldsymbol{x}), \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x})$ *in* (3) *except for* $\boldsymbol{\theta}_{const.}$.

We will operate in the batch setting *i.e.* our observer will sample K signal-response pairs over some duration and perform inference using this data. By signal-response pairs, we refer to tuples of the form $\Big(\big(\boldsymbol{x}(k), \{\boldsymbol{x}_j^o(k)\}_{j=1}^M\big), \boldsymbol{u}_{\boldsymbol{\theta}}^*(k)\Big)$ $\forall k \in \{1, 2, \cdots, K\}$. Then, the observer uses all of these K measurements in one step to compute $\boldsymbol{\theta}$ by using an IO algorithm, described next.

III. Inverse Optimization based Inference

We know that the observer has access to state-control measurements of the ego robot. Given assumptions 1-3 and these measurements, the observer can develop an empirical risk minimization algorithm that uses each sample of available measurement to compute parameters $\boldsymbol{\theta}$. One candidate loss that can be used to compute an estimate of risk is the KKT loss [11]. In the context of our problem, this loss quantifies the extent to which the observed optimal control violates the KKT conditions of the robot's optimization problem (3). Let's recall these conditions. The Lagrangian for (3) is

$$L(\boldsymbol{u}, \boldsymbol{\lambda}) = \|\boldsymbol{u} - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x})\|_2^2 + \boldsymbol{\lambda}^T(A(\boldsymbol{x})\boldsymbol{u} - \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x})).$$

Let $(\boldsymbol{u}^*, \boldsymbol{\lambda}^*)$ be the optimal primal-dual solution to (3). The KKT conditions are [3]:

1) Stationarity: $\nabla_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{\lambda})|_{(\boldsymbol{u}^*, \boldsymbol{\lambda}^*)} = 0$ which gives

$$\boldsymbol{u}^* = \hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x}) - \frac{1}{2}A^T(\boldsymbol{x})\boldsymbol{\lambda}^*. \tag{5}$$

2) Primal Feasibility

$$A(\boldsymbol{x})\boldsymbol{u}^* \le \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x}) \tag{6}$$

3) Dual Feasibility

$$\boldsymbol{\lambda}^* \ge \boldsymbol{0} \tag{7}$$

4) Complementary Slackness

$$\boldsymbol{\lambda}^* \odot \left(A(\boldsymbol{x})\boldsymbol{u}^* - \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x})\right) = \boldsymbol{0} \tag{8}$$

Using (5) and (8), the KKT loss is defined as follows

$$\begin{aligned} & l^{KKT} = l^{stat.} + l^{comp.\ slack.} \text{ where,} \\ & l^{stat.} = \left\| \boldsymbol{u}^* - \hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}}(\boldsymbol{x}) + \frac{1}{2}A^T(\boldsymbol{x})\boldsymbol{\lambda} \right\|^2 \text{ (from (5))} \\ & l^{comp-slack.} = \left\| \boldsymbol{\lambda} \odot \left(A(\boldsymbol{x})\boldsymbol{u}^* - \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x})\right) \right\|^2 \text{ (from(8))} \end{aligned} \tag{9}$$

Using K observed signal-response pairs $\boldsymbol{\Omega}(k) = (\boldsymbol{x}(k), \boldsymbol{u}^*_{\boldsymbol{\theta}}(k))$, the observer poses an empirical risk minimization problem that queries for $\boldsymbol{\theta}$ and K multipliers $\{\boldsymbol{\lambda}_k\}_{k=1}^K \in \mathbb{R}^M$ which minimize the total KKT loss:

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K = \underset{\boldsymbol{\theta}, \{\boldsymbol{\lambda}_k\}_{k=1}^K}{\arg\min} \quad & \sum_{k=1}^K l^{KKT}(\boldsymbol{\theta}, \boldsymbol{\lambda}_k, \boldsymbol{\Omega}(k)) \\ \text{subject to} \quad & \boldsymbol{\lambda}_k \geq \boldsymbol{0} \ \forall k \in \{1, \cdots, K\} \\ & A(\boldsymbol{x}(k))\boldsymbol{u}^*_{\boldsymbol{\theta}}(k) \leq \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x}) \\ & \boldsymbol{\theta} \in \boldsymbol{\Theta}_0 \end{aligned} \tag{10}$$

In this problem, the decision variables are the task parameter $\boldsymbol{\theta}$ and the Lagrange multipliers $\boldsymbol{\lambda}_k$. The objective function in (10) 'softens' the stationarity (5) and complementary slackness (8) conditions. The constraints in (10) capture the primal and (6) and dual feasibility condition from (7). The constraint $\boldsymbol{\theta} \in \boldsymbol{\Theta}_0$ captures our prior knowledge on $\boldsymbol{\theta}$. For example, the constraint parameters γ, D_s are non-negative so $\boldsymbol{\theta}_{const.} \geq 0$. Likewise, the proportional controller gains are non-negative so that can be incorporated in $\boldsymbol{\Theta}_0$. The loss function in (9) is nonconvex in $\boldsymbol{\theta}_{const.}$ albeit convex in $\boldsymbol{\theta}_{task}$, making it nonconvex in $\boldsymbol{\theta}$. Therefore, the only option to solve (10) is to use a generic nonlinear programming solver such as `fmincon` to perform inference. Note that [11] developed a KKT loss minimization algorithm for objective function inference. We extend their approach for inferring constraint parameters in addition to objective function parameters, which comes at the cost of losing convexity of the resulting inference problem.

IV. Results

We provide numerical results for inference of task parameters in a multirobot system using (10). We consider the task where each robot is trying to reach a goal position while avoiding collisions with every other robot. We will use these algorithms to simultaneously estimate the desired goal $\boldsymbol{x}_d$, the proportional gain k_p which represent the task parameters along with γ and D_s which are the constraint parameters. The robots use $\hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}} = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$ as a nominal task-based controller in (3). We write this as

$$\hat{\boldsymbol{u}}_{\boldsymbol{\theta}_{task}} = \underbrace{\begin{bmatrix} -\boldsymbol{x}_x & 1 & 0 \\ -\boldsymbol{x}_y & 0 & 1 \end{bmatrix}}_{C(\boldsymbol{x})} \underbrace{\begin{bmatrix} k_p \\ k_p\boldsymbol{x}_{d_x} \\ k_p\boldsymbol{x}_{d_y} \end{bmatrix}}_{\boldsymbol{\theta}_{task}} + \underbrace{\boldsymbol{0}}_{\boldsymbol{d}(\boldsymbol{x})} \tag{11}$$

The constraint parameters $\boldsymbol{\theta}_{const.} = (\gamma, D_s)$. To evaluate the repeatability of our approach, we conduct simulations for five different arrangements of initial conditions and goals and report the mean and standard deviations of reconstruction errors. Further, we talk about failure cases where our algorithm converged to local minima of (10) yet did not reconstruct the true parameters.

A. Simulations where inference succeeds

In Fig. 1, we have five robots located in a 5m $\times$ 5m area. Each robot has a unique color and is required to reach a goal position denoted with the same color while staying safe. Table I shows the gain reconstruction errors for different parameters using (10). We conducted five simulations with varying goal locations, initial conditions and parameters to test the repeatability of our approach. As is evident from the table, all these errors are very small, thus demonstrating the effectiveness of our method.

B. Analysis of failure cases

We give intuitive arguments justifying the cases where (10) will fail, referring the reader to [9] for a formal analysis. Looking at the structure of (3), one can notice that it is **not** necessary that optimal control $\boldsymbol{u}^*$ will **always explicitly** depend on both $\boldsymbol{\theta}_{task}$ and $\boldsymbol{\theta}_{const.}$. For example, if there are two robots in the system and their positions are very far away from one another, then the safety constraints $A(\boldsymbol{x})\boldsymbol{u} \leq \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x})$ will most likely not get active, and thus inferring exact values of $\boldsymbol{\theta}_{const.}$ would be difficult. However, the NLP in (10) will still infer a feasible estimate of $\boldsymbol{\theta}_{const.}$ since they must satisfy $A(\boldsymbol{x}(k))\boldsymbol{u}^*(k) \leq \boldsymbol{b}_{\boldsymbol{\theta}_{const.}}(\boldsymbol{x}(k)) \ \forall k$ measurements. Furthermore, in this situation, the task parameters $\boldsymbol{\theta}_{task}$ will be inferrable. On the other hand, when the the geometric arrangement of robots is crowded such that a given robot interacts with many robots, then in this duration, safety constraints will dominate (3) so that $\boldsymbol{u}^*$ will not depend on $\boldsymbol{\theta}_{task}$. In this case, inferring $\boldsymbol{\theta}_{task}$ is difficult.

V. Conclusions

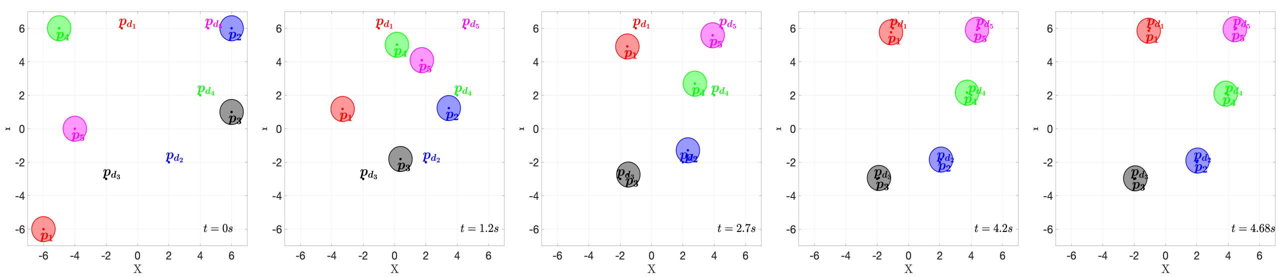
We considered the problem of inference of task and safety constraint parameters of a multirobot system. In such a system, robots use optimization based controllers to mediate between task satisfaction and collision avoidance, thus the trajectories they take, reflect how a purely task-based motion is warped to ensure safety. This makes inference of task parameters non-trivial. We considered the KKT loss minimization algorithm to solve this problem in a batch setting and demonstrated how accurate estimates of underlying parameters can be reconstructed. In future, we plan to consider robust estimation in the presence of model mismatch and measurement uncertainty.

References

[1] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In 2019 18th European Control Conference (ECC), pages 3420–3431. IEEE, 2019.

TABLE I: Parameter Estimation Errors $\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right\|$ Averaged over Five Simulations

Robot ID	$\|\hat{k}_p - k_p\|$[1/s]	$\|\|\hat{\boldsymbol{x}}_d - \boldsymbol{x}_d\|\|$[m]	$\|\hat{\gamma} - \gamma\|$[1/s]	$\|\hat{D}_s - D_s\|$[m]
1	0.0007±0.00002	0.0006±0.00006	0.2±0.274	0.0007±0.0013
2	0.00015±0.00018	0.0011±0.0013	0.22±0.312	0.166±0.37
3	0.00178±0.0024	0.0168±0.023	0.12±0.212	0.46±0.646
4	0.00006±0.00004	0.0003±0.0003	0.20±0.273	0.0006±0.0005
5	0.00009±0.00007	0.0005±0.0006	0.20±0.274	0.00024±0.00027



(a) $t = 0.06s$ (b) $t = 1.2s$ (c) $t = 2.7s$ (d) $t = 4.2s$ (e) $t = 4.68s$

Fig. 1: Each robot is navigating towards its goal while avoiding collisions with other robots https://youtu.be/Cuw5qZUylmg

[2] Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Inverse optimization: A new perspective on the black-litterman model. Operations research, 60(6): 1389–1403, 2012.

[3] Stephen Boyd and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.

[4] Scott Carr and William Lovejoy. The inverse newsvendor problem: Choosing an optimal demand portfolio for capacitated resources. Management Science, 46(7):912–927, 2000.

[5] Peter Englert, Ngo Anh Vien, and Marc Toussaint. Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. The International Journal of Robotics Research, 36(13-14):1474–1488, 2017.

[6] Qi Gong, Wei Kang, Claire Walton, Isaac Kaminer, and Hyeongjun Park. Partial observability analysis of an adversarial swarm model. Journal of Guidance, Control, and Dynamics, 43(2):250–261, 2020.

[7] Jaskaran Grover, Changliu Liu, and Katia Sycara. Deadlock analysis and resolution in multi-robot systems (extended version), 2019.

[8] Jaskaran Grover, Changliu Liu, and Katia Sycara. Feasible region-based identification using duality (extended version). arXiv preprint arXiv:2011.04904, 2020.

[9] Jaskaran Singh Grover, Changliu Liu, and Katia Sycara. Parameter identification for multirobot systems using optimization based controllers (extended version). arXiv preprint arXiv:2009.13817, 2020.

[10] Miles Johnson, Navid Aghasadeghi, and Timothy Bretl. Inverse optimal control for deterministic continuous-time nonlinear systems. In 52nd IEEE Conference on Decision and Control, pages 2906–2913. IEEE, 2013.

[11] Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In 2011 IEEE international symposium on intelligent control, pages 613–619. IEEE, 2011.

[12] Changliu Liu and Masayoshi Tomizuka. Control in a safe set: Addressing safety in human-robot interactions. In ASME 2014 Dynamic Systems and Control Conference. American Society of Mechanical Engineers Digital Collection, 2014.

[13] Timothy L Molloy, Jason J Ford, and Tristan Perez. Online inverse optimal control on infinite horizons. In 2018 IEEE Conference on Decision and Control (CDC), pages 1663–1668. IEEE, 2018.

[14] Pietro Pierpaoli, Harish Ravichandar, Nicholas Waytowich, Anqi Li, Derrik Asher, and Magnus Egerstedt. Inferring and learning multi-robot policies by observing an expert. arXiv preprint arXiv:1909.07887, 2019.

[15] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE International Conference on Robotics and Automation, pages 1928–1935. IEEE, 2008.

[16] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In Robotics research, pages 3–19. Springer, 2011.

[17] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. IEEE Transactions on Robotics, 33(3):661–674, 2017.

[18] Yulin Zhang and Dylan A Shell. Plans that remain private, even in hindsight. 2020.