System Identification for Safe Controllers using Inverse Optimization

Jaskaran Grover **Changliu Liu** **Katia Sycara**
The Robotics Institute, Carnegie Mellon University, USA
(email: {jaskarag,cliu6,sycara}@andrew.cmu.edu)

Abstract: This paper presents algorithms for learning parameters of optimization-based controllers used in multiagent systems based on their position-velocity measurements. The motivation to learn these parameters stems from the need to infer an agent's intent (human or robot) to facilitate accurate predictions of motion as well as efficient interactions in a multiagent system. In this work, we demonstrate how to perform inference using algorithms based on the theory of inverse optimization (IO). We propose QP-based reformulations of IO algorithms for faster processing of batch-data to facilitate quicker inference. In our prior work, we used persistency of excitation analysis for deriving conditions under which conventional estimators such as a Kalman filter can successfully perform such inference. In this work, we demonstrate that whenever these conditions are violated, inference of parameters will fail, be it using IO-based algorithms or a UKF. We provide numerical simulations to infer desired goal locations and controller gains of each robot in a multirobot system and compare performance of IO-based algorithms with a UKF and an adaptive observer. In addition to these, we also conduct experiments with Khepera-4 robots and demonstrate the power of IO-based algorithms in inferring goals in the presence of perception noise.

Keywords: Multiagent systems, Optimization-based control, Robotics

1. INTRODUCTION

There are several applications of planning and control for multiple robots like search and rescue, sensor coverage Cortes et al. (2004) and exploration Burgard et al. (2005). Recently, optimization-based control synthesis techniques are being used extensively for synthesizing task-based controllers for multirobot systems, which also guarantee collision avoidance Wang et al. (2017), Grover et al. (2019), Van Den Berg et al. (2011), Grover et al. (2020b). The inverse problem to task-based control is that of task inference. Characterizing the drive behind the actions of a robot team is useful for both predicting their future behavior as well as imitating a robot team controlled by human users Pierpaoli et al. (2019), Zhang and Shell (2020). This is especially useful in settings where inferring the strategies of an adversarial multirobot team can be used to plan defensive actions.

Many inference algorithms based on the theory of 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 Englert et al. (2017), Majumdar et al. (2017), Johnson et al. (2013), Molloy et al. (2018). However, all these algorithms assume that the agent minimizes a long-term cost function integrated over a horizon. On the other hand, several existing controllers devised for multirobot systems use a reactive collision avoidance mechanism with a task-based controller Van den Berg et al. (2008), Wang et al. (2017). The control inputs are generated at every time step by solving a finite-dimensional optimization problem instead of an infinite dimensional optimal control problem as is assumed in IOC. Examples of such controllers include barrier-function based quadratic programs (CBF-QPs) Ames et al. (2019) and safe-set controller Liu and Tomizuka (2014). This prevents direct application of IOC based approaches for inference of tasks involving multirobot systems. Moreover, these algorithms do not provide any bounds on the reconstruction errors in the parameter estimates they generate. Relatively little work has been done to explore identifiability conditions under which this error will be zero.

Given these limitations, we are interested in addressing how can an observer infer parameters of the tasks being performed by individual robots in a team, by exploiting that the underlying dynamics involve optimization in the loop. Now, our observer can only observe the robots but not intervene physically. Hence, parameter inference must occur based on pure observational monitoring of input-output signals, unlike system identification for manipulators Yang et al. (2018) where manual inputs may be designed to excite the robot to purposefully aid inference. Given this lack of ability to interfere with the system, we want to investigate when can the observer be confident about inference solely based on measuring the robots' states and the decisions made by them in those states.

Recent literature on inverse optimization (IO) has developed approaches to address these questions. These methods focus on estimating cost/constraint parameters of latent parametric optimization problems Ahuja and Orlin (2001). While IO algorithms have been explored in finance Bertsimas et al. (2012) and OR Carr and Lovejoy (2000), they have not been explored as much in robotics. In this work, we consider three previously developed IO algorithms from Aswani et al. (2018),Bertsimas et al. (2012) and Keshavarz et al. (2011) and reformat these algorithms to perform task inference of individual robots in a multirobot system. We assume that robots use CBF-QPs for control synthesis, but our inference technique can be used to perform inference of any other optimization based controller as well Wei and Liu (2019). For these IO algorithms, 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 Esfahani et al. (2018). 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.

Further, by exploiting the structure of the agent's control synthesis optimization, we derive quadratic program based reformulations of the KKT-residual based inference algorithm from Keshavarz et al. (2011) and suboptimality minimization based algorithm from Bertsimas et al. (2012). Lastly, building on our prior work on persistency of excitation analysis Grover et al. (2020a), we give conditions when IO algorithms would fail to recover latent parameters of tasks of robots. These conditions depend on the number of obstacles that are perceived as potential sources of collisions (active obstacles) by the ego robot while it is performing its task. Our prior work derived that inference is guaranteed to fail when the number of such obstacles is more than or equal to two. In this work, we show that this result is valid for IO based inference as well. To substantiate this claim, we consider the task where a robot must reach a goal while navigating around obstacles and the observer's problem is to infer the goal and controller gain of the robot. Through repeated trials, we demonstrate that each of the IO algorithms correctly infers these parameters when the number of active obstacles is less than two, while they produce large errors otherwise, as expected based on our identifiability conditions.

The outline of this paper is as follows. In Sec. 2, we briefly review the multirobot task completion and collision avoidance control approach and pose a mathematical formulation of the task inference problem. The main technical contributions start from Sec. 3. We reformat previously developed IO algorithms for the task inference problem. Additionally, we derive novel QP-based reformulations of these algorithms. In Sec. 4, we present numerical results for inference of controller gains and goal location of each robot in a multirobot system using the presented algorithms. We also report results averaged over ten trials to show that with greater number of active obstacles, inference algorithms will fail. In Sec. 5, we show experimental results of goal inference using these algorithms on Khepera-4 robots and demonstrate that even in the presence of perception noise, these algorithms provide accurate estimates of goals. Finally, we summarize our work in Sec. 6 and conclude with directions for future work.

2. MULTIROBOT SAFE TASK-BASED CONTROL AND TASK INFERENCE

2.1 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. We refer the reader to Wang et al. (2017) for a detailed treatment on this topic. Suppose there are $M+1$ robots in the system. From the perspective of an ego robot, the remaining M robots are "cooperative obstacles" who are all responsible for avoiding collisions amongst one another and the ego robot while performing their own tasks. 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

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

Here, $\boldsymbol{\theta} \in \mathbb{R}^p$ 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 $\boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$. If the observer wishes to infer the goal $\boldsymbol{x}_d$, then $\boldsymbol{\theta} = \boldsymbol{x}_d$, so choosing $C(\boldsymbol{x}) = k_p$, $\boldsymbol{d}(\boldsymbol{x}) = -k_p\boldsymbol{x}$ gives $C(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{d}(\boldsymbol{x}) = -k_p(\boldsymbol{x} - \boldsymbol{x}_d)$.

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 $\boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x})$ and satisfies M safety constraints as follows:

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

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 $\boldsymbol{b}$ 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 and γ is a hyperparameter that governs the size of the safe-control set. Typically γ is set at 1. The ego robot solves this QP at every time step to determine its optimal control $\boldsymbol{u}^*$, which ensures safety while encouraging satisfaction of the task parametrized by $\boldsymbol{\theta}$. This control depends on $\boldsymbol{\theta}$, through the cost function of (3) since $\boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x}) = C(\boldsymbol{x})\boldsymbol{\theta} + \boldsymbol{d}(\boldsymbol{x})$. To emphasize this, let us denote it as $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x})$.

Let us point out why this inference is non-trivial. The dependence of $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x})$ on $\boldsymbol{\theta}$ comes through the filter of constraints and the objective function in (3). In a situation where $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x})$ is solely determined by constraints, $\boldsymbol{\theta}$ will not be inferable because it is the cost function that depends on $\boldsymbol{\theta}$ not the constraints. Naturally, the observer may not have means to deduce the extent of the dependence of $\boldsymbol{u}_{\boldsymbol{\theta}}^*(\boldsymbol{x})$ on the constraints, and this is precisely what makes this problem challenging.

2.2 The Inverse Problem-Task 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 the task parameter $\boldsymbol{\theta}$ 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 $\{\boldsymbol{x}^o_j(t)\}_{j=1}^M$ (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* $\left\|\boldsymbol{u} - \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x})\right\|^2$

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

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

Most IO algorithms operate in the batch-setting. In the context of our problem, this implies that the observer will sample K signal-response pairs over some duration. By signal-response pairs, we refer to tuples of the form $\left((\boldsymbol{x}(k), \{\boldsymbol{x}^o_j(k)\}_{j=1}^M), \boldsymbol{u}^*_{\boldsymbol{\theta}}(k)\right)$ $\forall k \in \{1, 2, \cdots, K\}$. Then, the observer uses all of these K measurements in one step to compute $\boldsymbol{\theta}$ by using an IO algorithm, described next.

3. IO-BASED ALGORITHMS FOR INFERENCE

We consider three prominent IO algorithms. Since different authors follow different notations, we have used their ideas but reformatted their notations and language to be compatible with our robot task inference problem.

3.1 Predictability Loss Minimization

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 pose a parallel surrogate problem akin to the one being solved by the robot *i.e.* (3). In the surrogate problem, the observer treats the unknown parameter $\boldsymbol{\theta}$ as a tunable knob. The observer modulates this knob until the observer's *predicted* controls computed by the solving the surrogate problem in response to state $\boldsymbol{x}$, match with the controls measured from the robot's motion when the robot is also in state $\boldsymbol{x}$. To do this tuning, the observer poses the following problem:

$$\hat{\boldsymbol{\theta}}, \{\hat{\boldsymbol{u}}_k\}_{k=1}^K = \underset{\boldsymbol{\theta}, \{\boldsymbol{u}_k\}_{k=1}^K}{\arg\min} \ \frac{1}{K}\sum_{k=1}^K \|\boldsymbol{u}_k - \boldsymbol{u}^*_{\boldsymbol{\theta}}(k)\|^2 \quad \text{such that } \boldsymbol{u}_k \text{ solves (3) } \forall k \in \{1, \cdots, K\}. \tag{5}$$

In this problem, the observer is learning both the parameter $\boldsymbol{\theta} \in \mathbb{R}^p$ as well as predictions of the optimal control $\hat{\boldsymbol{u}}_k \in \mathbb{R}^2$ $\forall k \in \{1, 2, \cdots, K\}$. The cost function in (5) is the empirical average of the deviations of the predicted controls $\hat{\boldsymbol{u}}_k$ from the observed optimal controls $\boldsymbol{u}^*_{\boldsymbol{\theta}}(k)$. This is known as the *predictability loss* and was proposed in Aswani et al. (2018). Naturally, it makes sense to minimize this loss only if the observer's predicted controls solve the forward problem (3) which is posed as a constraint in (5). Since (3) is in itself an optimization problem, problem (5) is a bi-level optimization which is known to be computationally difficult to solve. Aswani et al. (2018) proposed a duality based reformulation of a bi-level optimization to a single level problem. Applying their technique to our robot task inference, we replace the constraint in (5) with the optimality conditions of (3) *i.e.*:

$\boldsymbol{u}_k$ solves (3) $\iff \exists\ \boldsymbol{\lambda}_k \in \mathbb{R}^M$ such that

(1) $\left\|\boldsymbol{u}_k - \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x}(k))\right\|^2 \le h(\boldsymbol{\lambda}_k, \boldsymbol{x}(k), \boldsymbol{\theta})$
(2) $\boldsymbol{\lambda}_k \ge \boldsymbol{0}$
(3) $A(\boldsymbol{x}(k))\boldsymbol{u}_k \le \boldsymbol{b}(\boldsymbol{x}(k))$

Here $\boldsymbol{\lambda}_k \in \mathbb{R}^M$ for each time instant k, are the M Lagrange multipliers corresponding to the M collision avoidance constraints of the ego robot in (3). $h(\boldsymbol{\lambda}_k, \boldsymbol{x}(k), \boldsymbol{\theta})$ is the Lagrange dual function of (3) and is given by

$$h = \left\|\tilde{\boldsymbol{u}} - \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x})\right\|^2 + \boldsymbol{\lambda}_k^T(A(\boldsymbol{x}(k))\tilde{\boldsymbol{u}} - \boldsymbol{b}(\boldsymbol{x}(k)))$$
$$\text{where } \tilde{\boldsymbol{u}} = \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x}(k)) - \frac{1}{2}A^T(\boldsymbol{x}(k))\boldsymbol{\lambda}_k \tag{6}$$

Given these three conditions, (5) can be re-posed as follows

$$\begin{aligned} &\hat{\boldsymbol{\theta}}, \{\hat{\boldsymbol{u}}_k\}_{k=1}^K, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K = \\ &\underset{\boldsymbol{\theta}, \{\boldsymbol{u}_k\}_{k=1}^K, \{\boldsymbol{\lambda}_k\}_{k=1}^K}{\arg\min} \quad \frac{1}{K}\sum_{k=1}^K \|\boldsymbol{u}_k - \boldsymbol{u}^*_{\boldsymbol{\theta}}(k)\|^2, \\ &\text{subject to } \ \left\|\boldsymbol{u}_k - \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x})\right\|^2 \le h(\boldsymbol{u}_k, \boldsymbol{\lambda}_k, \boldsymbol{\theta}) \\ &\qquad \boldsymbol{\lambda}_k \ge \boldsymbol{0} \\ &\qquad A(\boldsymbol{x}(k))\boldsymbol{u}_k \le \boldsymbol{b}(\boldsymbol{x}(k)) \ \forall k \in \{1, \cdots, K\} \end{aligned} \tag{7}$$

Even though (7) is a single-level reformulation of (5), yet it is non-convex because of the first constraint in (7). Therefore, it can only be solved using generic nonlinear programming solvers which tend to be slow, especially when the number of measurements K is large. In Sec. 4, we present numerical results using (7).

3.2 KKT Loss Minimization

Another candidate loss that can be used to compute an estimate of risk is the KKT loss Keshavarz et al. (2011). In our context, 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

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

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

(1) Stationarity: $\nabla_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{\lambda})|_{(\boldsymbol{u}^*, \boldsymbol{\lambda}^*)} = 0$ which gives
$$\boldsymbol{u}^* = \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \frac{1}{2}A^T(\boldsymbol{x})\boldsymbol{\lambda}^*. \tag{8}$$

(2) Primal Feasibility
$$A(\boldsymbol{x})\boldsymbol{u}^* \le \boldsymbol{b}(\boldsymbol{x}) \tag{9}$$

(3) Dual Feasibility
$$\boldsymbol{\lambda}^* \ge \boldsymbol{0} \tag{10}$$

(4) Complementary Slackness
$$\boldsymbol{\lambda}^* \odot \left(A(\boldsymbol{x})\boldsymbol{u}^* - \boldsymbol{b}(\boldsymbol{x})\right) = \boldsymbol{0} \tag{11}$$

Using (8) and (11), the KKT loss is defined as follows

$$\begin{aligned} &l^{KKT} = l^{stat.} + l^{comp.\ slack.} \text{ where,} \\ &l^{stat.} = \left\|\boldsymbol{u}^* - \boldsymbol{u}^{task}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \frac{1}{2}A^T(\boldsymbol{x})\boldsymbol{\lambda}\right\|^2 \text{ (from (8))} \\ &l^{comp-slack.} = \left\|\boldsymbol{\lambda} \odot \left(A(\boldsymbol{x})\boldsymbol{u}^* - \boldsymbol{b}(\boldsymbol{x})\right)\right\|^2 \text{ (from(11))} \end{aligned} \tag{12}$$

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 Lagrange multipliers $\{\boldsymbol{\lambda}_k\}_{k=1}^K \in \mathbb{R}^M$ which minimize the total KKT loss:

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \{\hat{\boldsymbol{\lambda}}_k\}_{k=1}^K = \underset{\boldsymbol{\theta}, \{\boldsymbol{\lambda}_k\}_{k=1}^K}{\arg\min} \quad & \sum_{k=1}^K l^{KKT}(\boldsymbol{\theta}, \boldsymbol{\lambda}_k, \boldsymbol{\Omega}(k)) \\ \text{subject to} \quad & \boldsymbol{\lambda}_k \geq \mathbf{0} \ \forall k \in \{1, \cdots, K\}. \end{aligned} \tag{13}$$

In this problem, the decision variables are the task parameter $\boldsymbol{\theta}$ and the Lagrange multipliers $\boldsymbol{\lambda}_k$. The objective function in (13) 'softens' the stationarity (8) and complementary slackness (11) conditions. The constraints in (13) capture the dual feasibility condition from (10). Given the complicated nature of the loss function in (12), the first instinct is to use a generic solver such as `fmincon` to perform inference. However, these solvers tend to be computationally slow, especially when the number of constraints and decision variables is large. We reformulate this problem to a QP for faster inference.

Reformulating (13) as a QP

Define a vector $\boldsymbol{\mu} = (\boldsymbol{\theta}^T, \boldsymbol{\lambda}_1^T, \cdots, \boldsymbol{\lambda}_K^T)^T \in \mathbb{R}^{p+MK}$ which is the decision variable of (13). Define matrices $E^{\boldsymbol{\theta}}$ and $E_k^{\boldsymbol{\lambda}}$ appropriately to extract $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}_k$ from $\boldsymbol{\mu}$ as follows:

$$\begin{aligned} \boldsymbol{\theta} &= E^{\boldsymbol{\theta}} \boldsymbol{\mu} \\ \boldsymbol{\lambda}_k &= E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} \end{aligned} \tag{14}$$

We can already re-pose the constraints in (13) as follows:

$$\begin{aligned} & \boldsymbol{\lambda}_k \geq \mathbf{0} \ \forall k \in \{1, \cdots, K\} \\ \iff & E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} \geq \mathbf{0} \ \forall k \in \{1, \cdots, K\}, \end{aligned} \tag{15}$$

which are convex by construction. Next, we reformulate the cost function of (13). Recall that $l^{KKT} = l^{stat.} + l^{comp.slack.}$. First we focus on $l_k^{stat.}$ from (12). We have

$$\begin{aligned} l_k^{stat.} &= \left\| \boldsymbol{u}^*(k) - \boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x}(k)) + \frac{1}{2} A^T(\boldsymbol{x}(k)) \boldsymbol{\lambda}_k \right\|^2 \\ &= \left\| \underbrace{\boldsymbol{u}^*(k) - \boldsymbol{d}(\boldsymbol{x}(k))}_{\boldsymbol{r}_k} - C(\boldsymbol{x}(k))\boldsymbol{\theta} + \frac{1}{2} A^T(\boldsymbol{x}(k)) \boldsymbol{\lambda}_k \right\|^2 \\ &= \left\| \boldsymbol{r}_k - C(\boldsymbol{x}(k)) E^{\boldsymbol{\theta}} \boldsymbol{\mu} + \frac{1}{2} A^T(\boldsymbol{x}(k)) E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} \right\|^2 \\ &= \left\| \boldsymbol{r}_k - \tilde{F}_k \boldsymbol{\mu} \right\|^2 \\ &= \boldsymbol{\mu}^T \tilde{F}_k^T \tilde{F}_k \boldsymbol{\mu} - 2\boldsymbol{r}_k^T \tilde{F}_k \boldsymbol{\mu} + \boldsymbol{r}_k^T \boldsymbol{r}_k \end{aligned} \tag{16}$$

In the equation above,

$$\tilde{F}_k = C(\boldsymbol{x}(k)) E^{\boldsymbol{\theta}} - \frac{1}{2} A^T(\boldsymbol{x}(k)) E_k^{\boldsymbol{\lambda}} \tag{17}$$

Similarly, using (12), we reformulate $l_k^{comp.\ slack.}$:

$$\begin{aligned} l_k^{comp.\ slack.} &= \left\| \boldsymbol{\lambda}_k \odot \underbrace{\left(A(\boldsymbol{x}(k)) \boldsymbol{u}^*(k) - \boldsymbol{b}(\boldsymbol{x}(k)) \right)}_{\boldsymbol{w}_k} \right\|^2 \\ &= \left\| E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} \odot \boldsymbol{w}_k \right\|^2 \\ &= \boldsymbol{\mu}^T E_k^{\boldsymbol{\lambda} T} W_k E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} \end{aligned} \tag{18}$$

where $W_k = \text{diag}\big([\boldsymbol{w}_k^2(1), \boldsymbol{w}_k^2(2), \cdots, \boldsymbol{w}_k^2(M)]\big)$. Adding (16) and (18) gives

$$\begin{aligned} l_k^{KKT} &= \boldsymbol{\mu}^T \tilde{F}_k^T \tilde{F}_k \boldsymbol{\mu} + \boldsymbol{\mu}^T E_k^{\boldsymbol{\lambda} T} W_k E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} - 2\boldsymbol{r}_k^T \tilde{F}_k \boldsymbol{\mu} + \boldsymbol{r}_k^T \boldsymbol{r}_k \\ &= \boldsymbol{\mu}^T \underbrace{\left(\tilde{F}_k^T \tilde{F}_k + E_k^{\boldsymbol{\lambda} T} W_k E_k^{\boldsymbol{\lambda}} \right)}_{Q_k} \boldsymbol{\mu} + \underbrace{\left(-2\boldsymbol{r}_k^T \tilde{F}_k \right)}_{\boldsymbol{v}_k^T} \boldsymbol{\mu} + \underbrace{\boldsymbol{r}_k^T \boldsymbol{r}_k}_{s_k} \\ &= \boldsymbol{\mu}^T Q_k \boldsymbol{\mu} + \boldsymbol{v}_k^T \boldsymbol{\mu} + s_k \end{aligned} \tag{19}$$

Thus, the total loss over all K measurements from the cost function of (13) is obtained by summing (19) as follows:

$$\begin{aligned} \sum_{k=1}^K l_k^{KKT} &= \sum_{k=1}^K \left(\boldsymbol{\mu}^T Q_k \boldsymbol{\mu} + \boldsymbol{v}_k^T \boldsymbol{\mu} + s_k \right) \\ &= \boldsymbol{\mu}^T \underbrace{\left(\sum_{k=1}^K Q_k \right)}_{Q} \boldsymbol{\mu} + \underbrace{\left(\sum_{k=1}^K \boldsymbol{v}_k^T \right)}_{\boldsymbol{v}^T} \boldsymbol{\mu} + \underbrace{\left(\sum_{k=1}^K s_k \right)}_{s} \\ &= \boldsymbol{\mu}^T Q \boldsymbol{\mu} + \boldsymbol{v}^T \boldsymbol{\mu} + s \end{aligned} \tag{20}$$

From (20), it is evident that the total KKT loss in (13) is indeed in quadratic in the decision variables $\boldsymbol{\mu}$. Hence, we can re-pose (13) using the reformulated cost in (20) and constraints in (15) as the following QP:

$$\begin{aligned} \hat{\boldsymbol{\mu}} = \underset{\boldsymbol{\mu}}{\arg\min} \quad & \boldsymbol{\mu}^T Q \boldsymbol{\mu} + \boldsymbol{v}^T \boldsymbol{\mu} \\ \text{subject to} \quad & E_k^{\boldsymbol{\lambda}} \boldsymbol{\mu} \geq \mathbf{0} \ \forall k \in \{1, \cdots, K\}. \end{aligned} \tag{21}$$

(21) is thus a QP-based reformulation of (13), and is amenable to faster solutions using existing QP-solvers.

3.3 Sub-optimality Minimization

Bertsimas et al. (2015) proposed data-driven techniques to infer unobservable parameters of models describing Nash equilibria in game theory. They combined ideas from inverse optimization with variational inequalities to develop data-driven techniques for estimating the parameters of these models from observed equilibria. Following their approach, we show how to reformat (3) so that we can leverage their approach for inferring robot task parameters $\boldsymbol{\theta}$. Consider a general optimization problem

$$\begin{aligned} & \text{minimize } F_{\boldsymbol{\theta}}(\boldsymbol{\xi}) \\ & \text{subject to } \boldsymbol{\xi} \in \mathcal{X} \end{aligned} \tag{22}$$

Here $\boldsymbol{\theta}$ are parameters of the convex cost function F known to the agent solving (22) and $\mathcal{X} \subset \mathbb{R}^n$ is a convex set.

Assumption 4. *$\mathcal{X}$ can be represented as the intersection of a small number of conic inequalities in standard form, $\mathcal{X} = \{\boldsymbol{\xi} \in \mathbb{R}^n | G\boldsymbol{\xi} = \boldsymbol{h}, \boldsymbol{\xi} \geq \mathbf{0}\}$.*

Assumption 5. *$\mathcal{X}$ satisfies a Slater's condition.*

The following result from Bertsimas et al. (2015) characterizes necessary and sufficient conditions for $\hat{\boldsymbol{\xi}}$ to be an ϵ-optimal solution to (22):

Theorem 1. *Bertsimas et al. (2015) Assuming $\mathcal{X}$ satisfies 4-5, an observed decision $\hat{\boldsymbol{\xi}}$ is an ϵ-optimal solution to (22) if and only if $\exists$ $\boldsymbol{y}$ such that $G^T\boldsymbol{y} \leq \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi})|_{\hat{\boldsymbol{\xi}}}$ and $\hat{\boldsymbol{\xi}}^T \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi})|_{\hat{\boldsymbol{\xi}}} - \boldsymbol{h}^T\boldsymbol{y} \leq \epsilon$*

The inverse problem requires an observer to infer $\boldsymbol{\theta}$ based on the knowledge that the agent solves (22) using K samples of $\hat{\boldsymbol{\xi}}$ which are known to be ϵ-optimal solutions to (22). Since the observer does not know the suboptimality of a decision ϵ, the observer poses a suboptimality minimization problem querying for $\epsilon_k, \boldsymbol{y}_k, \boldsymbol{\theta}$ as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}}, \{\hat{\epsilon}\}_{k=1}^K, \{\hat{\boldsymbol{y}}_k\}_{k=1}^K = & \underset{\boldsymbol{\theta}, \{\epsilon\}_{k=1}^K, \{\boldsymbol{y}_k\}_{k=1}^K}{\arg\min} \sum_{k=1}^K \epsilon_k^2 \\ \text{subject to} \quad & G^T \boldsymbol{y}_k \leq \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi})|_{\hat{\boldsymbol{\xi}}_k} \\ & \hat{\boldsymbol{\xi}}_k^T \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi})|_{\hat{\boldsymbol{\xi}}} - \boldsymbol{h}^T \boldsymbol{y}_k \leq \epsilon_k \ \forall k \in \{1, \cdots, K\} \end{aligned} \tag{23}$$

Notice that the cost function in (23) penalizes the suboptimality of observed solutions $\boldsymbol{\xi}_k$ whereas the constraints are necessary and sufficient conditions for observed decisions $\boldsymbol{\xi}_k$ to be ϵ_k-optimal solutions of (22) based on Theorem 1.

Robot task inference using (23)

In the context of inferring task parameters of a robot, recall that our ego robot solves (3) where the task parameters are involved in the cost function. This optimization problem is a specific instance of the general problem in (22), therefore we need to reformat the ego-robot's optimization problem (3) to (22) to facilitate inference of $\boldsymbol{\theta}$ using (23). Recall that controls in (3) are required to satisfy safety constraints

$$A(\boldsymbol{x})\boldsymbol{u} \leq \boldsymbol{b}(\boldsymbol{x}). \tag{24}$$

On the other hand, the feasible set $\mathcal{X}$ in (22) is required to satisfy assumptions 4-5. To reformat (24) so that these assumptions are satisfied, define $\boldsymbol{u}_1 \geq \boldsymbol{0}$ and $\boldsymbol{u}_2 \geq \boldsymbol{0}$ such that $\boldsymbol{u} = \boldsymbol{u}_1 - \boldsymbol{u}_2$. One choice satisfying these requirements is $\boldsymbol{u}_1 = \boldsymbol{u} + |\boldsymbol{u}|, \boldsymbol{u}_2 = |\boldsymbol{u}|$. Define the following variables,

$$\boldsymbol{z} = (\boldsymbol{b}(\boldsymbol{x}) - A(\boldsymbol{x})\boldsymbol{u}) \in \mathbb{R}^M \tag{25}$$

$$\boldsymbol{\xi} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}) \tag{26}$$

Then, it is evident that

$$\begin{aligned} A(\boldsymbol{x})\boldsymbol{u} \leq \boldsymbol{b}(\boldsymbol{x}) \iff A(\boldsymbol{x})(\boldsymbol{u}_1 - \boldsymbol{u}_2) \leq \boldsymbol{b}(\boldsymbol{x}) \\ \iff A(\boldsymbol{x})(\boldsymbol{u}_1 - \boldsymbol{u}_2) + \boldsymbol{z} = \boldsymbol{b}(\boldsymbol{x}), \boldsymbol{z} \geq \boldsymbol{0} \end{aligned} \tag{27}$$

Define G and $\boldsymbol{h}$ required in assumption 4 as follows

$$\begin{aligned} G(\boldsymbol{x}) &:= \left[A(\boldsymbol{x}), -A(\boldsymbol{x}), I^M\right] \\ \boldsymbol{h}(\boldsymbol{x}) &:= \boldsymbol{b}(\boldsymbol{x}) \end{aligned} \tag{28}$$

where I^M is the $M \times M$ identity matrix. Then, it is easy to verify that $G(\boldsymbol{x})\boldsymbol{\xi} = \boldsymbol{h}(\boldsymbol{x})$. We have

$$\begin{aligned} & A(\boldsymbol{x})\boldsymbol{u} \leq \boldsymbol{b}(\boldsymbol{x}) \\ \iff & \boldsymbol{u}_1 \geq \boldsymbol{0}, \boldsymbol{u}_2 \geq \boldsymbol{0}, \boldsymbol{z} \geq \boldsymbol{0}, A(\boldsymbol{x})(\boldsymbol{u}_1 - \boldsymbol{u}_2) + \boldsymbol{z} = \boldsymbol{b}(\boldsymbol{x}) \\ \iff & \boldsymbol{\xi} \geq 0 \text{ and } G(\boldsymbol{x})\boldsymbol{\xi} = \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{29}$$

Recall that the cost function in (3) is

$$F_{\boldsymbol{\theta}}(\boldsymbol{u}) = \left\|\boldsymbol{u} - \boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x})\right\|^2 \tag{30}$$

So we define a matrix $E^{\boldsymbol{u}}$ appropriately so that $\boldsymbol{u} = E^{\boldsymbol{u}}\boldsymbol{\xi}$. The cost then becomes a function of $\boldsymbol{\xi}$:

$$\begin{aligned} F_{\boldsymbol{\theta}}(\boldsymbol{\xi}) &= \left\|E^{\boldsymbol{u}}\boldsymbol{\xi} - \boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x})\right\|^2 \\ \implies \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi}) &= 2E^{\boldsymbol{u}T}(E^{\boldsymbol{u}}\boldsymbol{\xi} - \boldsymbol{u}_{\boldsymbol{\theta}}^{task}(\boldsymbol{x})) \\ &= 2E^{\boldsymbol{u}T}(E^{\boldsymbol{u}}\boldsymbol{\xi} - C(\boldsymbol{x})\boldsymbol{\theta} - \boldsymbol{d}(\boldsymbol{x})) \end{aligned}$$

Thus, with (3) re-posed as (22), the signal-response pairs for inference are no longer $(\boldsymbol{x}(k), \boldsymbol{u}^*(k))$, rather they are $(\boldsymbol{x}(k), \boldsymbol{\xi}^*(k))$. To construct $\boldsymbol{\xi}^*(k)$ from $\boldsymbol{u}^*(k)$, define

$$\begin{aligned} \boldsymbol{u}_1(k) &:= \boldsymbol{u}^*(k) + |\boldsymbol{u}^*(k)| \\ \boldsymbol{u}_2(k) &:= |\boldsymbol{u}^*(k)| \\ \boldsymbol{z}(k) &:= \boldsymbol{b}(\boldsymbol{x}(k)) - A(\boldsymbol{x}(k))\boldsymbol{u}^*(k) \\ \implies \boldsymbol{\xi}^*(k) &:= (\boldsymbol{u}_1(k), \boldsymbol{u}_2(k), \boldsymbol{z}(k)). \end{aligned} \tag{31}$$

Note that $\boldsymbol{u}_1(k), \boldsymbol{u}_2(k) \geq \boldsymbol{0}$ because of the way we defined them. $|\boldsymbol{u}^*(k)|$ is the absolute value of $\boldsymbol{u}^*(k)$ taken element wise. $\boldsymbol{z}^k \geq \boldsymbol{0}$ because the measured controls $\boldsymbol{u}^*(k)$ must satisfy the safety constraints (24). Thus, $\boldsymbol{\xi}(k) \geq \boldsymbol{0}$ as is required by assumption 4. Assumption 5 is automatically satisfied by all QPs (Boyd and Vandenberghe (2004)), and by extension, by (3). Once again, (23), in the form presented, can only be solved by general NLP solvers. In the next section, we show how to reformulate this as a QP.

Reformulating (23) as a QP

In this section, we reformulate this problem as a QP. Define a vector $\boldsymbol{\mu} = (\boldsymbol{\theta}^T, \epsilon_1, \cdots, \epsilon_K, \boldsymbol{y}_1^T, \cdots, \boldsymbol{y}_K^T)^T$ which is the decision variable of (23). Define matrices $E^{\boldsymbol{\theta}}$,E_k^{ϵ} $E_k^{\boldsymbol{y}}$ appropriately to extract $\boldsymbol{\theta}$, ϵ_k and $\boldsymbol{\lambda}_k$ from $\boldsymbol{\mu}$ as follows:

$$\begin{aligned} \boldsymbol{\theta} &= E^{\boldsymbol{\theta}}\boldsymbol{\mu} \\ \boldsymbol{\epsilon}_k &= E_k^{\epsilon}\boldsymbol{\mu} \\ \boldsymbol{y}_k &= E_k^{\boldsymbol{y}}\boldsymbol{\mu} \end{aligned} \tag{32}$$

Based on these definitions, we can already re-pose the cost function in (23) as follows:

$$\sum_{k=1}^K \epsilon_k^2 = \sum_{k=1}^K \|E_k^{\epsilon}\boldsymbol{\mu}\|^2 = \boldsymbol{\mu}^T \underbrace{\left(\sum_{k=1}^K E_k^{\epsilon T} E_k^{\epsilon}\right)}_{Q} \boldsymbol{\mu} \tag{33}$$

The constraints can be re-posed as

$$\begin{aligned} & G^T \boldsymbol{y}_k \leq \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi})|_{\boldsymbol{\xi}^*(k)} \iff \\ & G^T(\boldsymbol{x}(k)) E_k^{\boldsymbol{y}}\boldsymbol{\mu} \leq 2E^{\boldsymbol{u}T}(E^{\boldsymbol{u}}\boldsymbol{\xi}^*(k) - C(\boldsymbol{x}(k))E^{\boldsymbol{\theta}}\boldsymbol{\mu} - \boldsymbol{d}(\boldsymbol{x}(k)) \end{aligned}$$

After some rearrangement, this can be re-posed as an inequality in $\boldsymbol{\mu}$:

$$H_1\boldsymbol{\mu} \leq \boldsymbol{g}_1 \tag{34}$$

Similarly the second constraint in (23) can be re-posed as

$$\begin{aligned} & \boldsymbol{\xi}^{*T} \nabla_{\boldsymbol{\xi}} F_{\boldsymbol{\theta}}(\boldsymbol{\xi})|_{\boldsymbol{\xi}^*(k)} - \boldsymbol{h}^T \boldsymbol{y}_k \leq \epsilon_k \iff \\ & 2\boldsymbol{\xi}^{*T}(k) E^{\boldsymbol{u}T}\Big(E^{\boldsymbol{u}}\boldsymbol{\xi}^*(k) - C(\boldsymbol{x}(k))E^{\boldsymbol{\theta}}\boldsymbol{\mu} - \boldsymbol{d}(\boldsymbol{x}(k)\Big) \\ & - \boldsymbol{h}^T(\boldsymbol{x}(k)) E_k^{\boldsymbol{y}}\boldsymbol{\mu} \leq E_k^{\epsilon}\boldsymbol{\mu} \end{aligned} \tag{35}$$

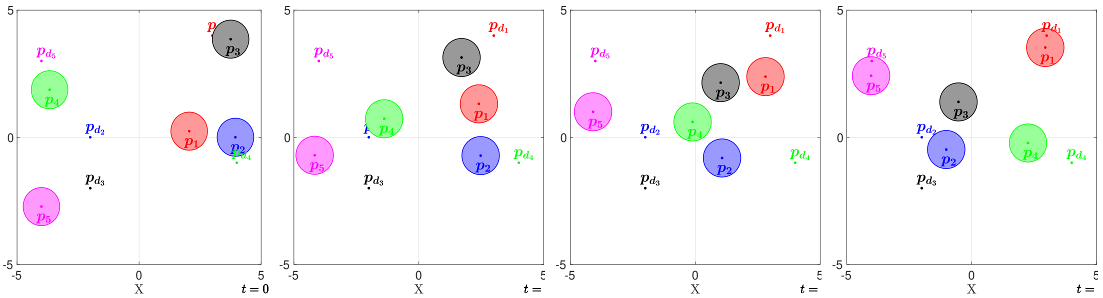
which after some rearrangement gives

$$H_2\boldsymbol{\mu} \leq \boldsymbol{g}_2 \tag{36}$$

The expressions for $H_1, \boldsymbol{g}_1, H_2, \boldsymbol{g}_2$ can be found in the appendix. Combining the quadratic cost in (33) with the affine constraints (34) and (36), we arrive at the following QP-based formulation:

$$\begin{aligned} \hat{\boldsymbol{\mu}} = & \arg\min_{\boldsymbol{\mu}} \quad \boldsymbol{\mu}^T Q \boldsymbol{\mu} \\ \text{subject to} \quad & H_1\boldsymbol{\mu} \leq \boldsymbol{g}_1 \ \forall k \in \{1, 2, \cdots, K\} \\ & H_2\boldsymbol{\mu} \leq \boldsymbol{g}_2 \ \forall k \in \{1, 2, \cdots, K\}. \end{aligned} \tag{37}$$

Thus, the convexity of this formulation allows for faster inference on batch-data by exploiting existing QP solvers.



(a) $t = 0.06s$ (b) $t = 0.6s$ (c) $t = 1.2s$ (d) $t = 2.4s$

Fig. 1. Each robot is navigating towards its goal while avoiding collisions with other robots.

4. SIMULATION RESULTS

We provide numerical results for inference of task parameters in a multirobot system using (7), (21) and (37). 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 estimate the desired goal $\boldsymbol{x}_d$ and proportional gain k_p for each robot by using their positions and velocities collected over some time. We also describe situations where these estimators will fail to identify these parameters, which occur when the measurements fail to satisfy certain "richness" criteria.

4.1 Gain and Goal inference in a multirobot setting

In Fig. 1, we have five robots located in a 5m × 5m area. Each robot has a unique color and is required to reach a goal position denoted with the same color while staying safe. The robots use $\boldsymbol{u}^{task}_{\boldsymbol{p}^*_d} = -k_p(\boldsymbol{p} - \boldsymbol{p}^*_d)$ as a nominal task-based controller in (3). The inference algorithm must compute estimates of $\hat{\boldsymbol{p}}_d$ and gain $\boldsymbol{k}^*_p$ for each robot. Table 1 shows the gain reconstruction errors for different algorithms. and Table 2 presents the goal-reconstruction errors. As is evident from the table, all three algorithms produce 0 error, meaning that they are able to correctly estimate both the correct goal and gain for each robot.

Table 1. Gain Estimation Errors $|k_p - k^*_p|$

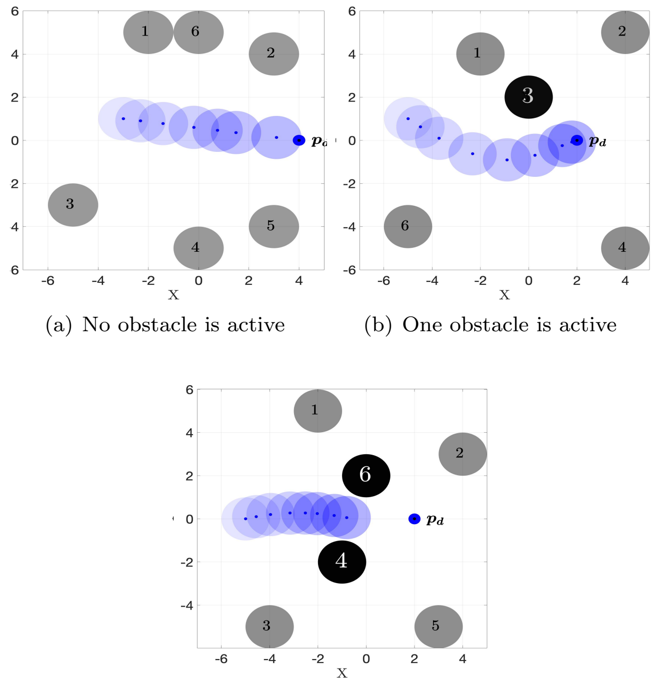
Robot ID	Predict. loss algorithm	KKT loss algorithm	Suboptimality loss algorithm
1	0.0000	0.0000	0.0000
2	0.0001	0.0000	0.0000
3	0.0002	0.0000	0.0000
4	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000

Table 2. Goal Estimation Errors $\|\hat{\boldsymbol{p}}_d - \boldsymbol{p}^*_d\|$ in [m]

Robot ID	Predict. loss algorithm	KKT loss algorithm	Suboptimality loss algorithm
1	0.0000	0.0000	0.0000
2	0.0005	0.0000	0.0000
3	0.0012	0.0000	0.0000
4	0.0002	0.0000	0.0000
5	0.0001	0.0000	0.0000

4.2 Inference when Identifiability Conditions are Violated

In our prior work Grover et al. (2020b), we used persistency of excitation analysis to identify conditions where conventional estimators such as a Kalman filter would fail to infer task parameters $\boldsymbol{\theta}$ using position and velocity measurements of a robot. Given the new algorithms considered in the current work, we want to stress test these conditions against these new algorithms. To keep this paper self-contained, we present these conditions only at a high-level. For an intuitive understanding, we present results for a single robot navigating amongst static obstacles. The inference problem is to determine the goal of this robot and its controller gain using its position and velocity measurements. The first condition, stated in Theorem 2 identifies a situation when inference will be successful.



(a) No obstacle is active (b) One obstacle is active

(c) Two obstacles are active

Fig. 2. As the robot navigates to its goal, the success of inference depends on the number of obstacles that it actively interacts with. In Fig (c) there are two active interactions (dark), so per Theorem 3, goal/gain inference will fail.

Theorem 2. *[Grover et al. (2020b)] If* $\forall t \in [0, T]$*, no constraint is active, then the observer can always estimate the goal and gain using* $\boldsymbol{x}(t), \boldsymbol{u}^*(\boldsymbol{x}(t))$ $\forall t \in [0, T]$ *(unless the robot is not already at goal).*

No. of Active Obstacles	Predictability loss algorithm (7)	KKT loss algorithm (21)	Suboptimality minmization algorithm (37)	Unscented Kalman Filter	Adaptive Observer
0	0.0003 ± 0.0006	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0241 ± 0.0000	0.0220 ± 0.0018
1	0.0311 ± 0.0456	0.0000 ± 0.0000	0.0000 ± 0.0000	0.2897± 0.3070	0.0214 ± 0.0180
2	6.3704 ± 2.4477	5.8848 ± 3.7777	4.3371 ± 2.2358	5.6569 ± 0.3070	5.6252 ± 0.0180

Table 3. Goal inference errors when the number of active obstacles is 0 (top row), 1 (middle row) and 2 (bottom row)

By an active constraint, we refer to the constraint of (3) which is active (for which equality holds). Intuitively, this refers to an active interaction with an obstacle, because that obstacle is a potential risk of collision. This result says that if the robot does not have active interactions with any obstacles, then the position-velocity measurements of the robot will be so rich that goal/gain inference will always be successful. One situation when this occurs is when the obstacles are far enough from the robot such that the robot can freely use $\boldsymbol{u}_{\boldsymbol{\theta}}^{task}$ as shown in Fig. 2(a)). We evaluate IO algorithms presented in this paper and compare them with a UKF and an Adaptive observer (from Grover et al. (2020b)) to stress-test this theorem for inferring the location of the goal. Table 3 shows the numerical results for this situation. We show the mean and standard deviation of goal estimation errors averaged over ten trials. In each trial, we varied the final goal position of the robot and locations of obstacles while keeping the initial position of the robot identical. As is evident from the errors in the first row, goal inference is always successful for all estimators. Let us look at a situation where inference will fail.

Theorem 3. *[Grover et al. (2020b)] If $\forall t \in [0,T]$, two or more than two constraints are active, then the observer **cannot** estimate either the goal or the gain, using $\boldsymbol{x}(t), \boldsymbol{u}^*(\boldsymbol{x}(t))\ \forall t \in [0,T]$.*

In this situation, the robot has active interactions with either two or more than two obstacles while navigating towards its goal. Consequently, this theorem says that the position-velocity measurements are not rich enough to facilitate correct inference of either the goal or the gain. Intuitively, this occurs because of the following. We know that the robot has two degrees of freedom in its control and when there are two or more obstacles to actively avoid, both of those degrees of freedom are exhausted in repelling the obstacles leaving no freedom dedicated for the task. Therefore, the motion that the observer measures does not have any explicit information about the task, which is why task inference will fail. Figure 2(c) shows an example simulation where the robot is moving towards its goal. There are several obstacles, two of these, shown in black, are the ones active during the robot's motion. Hence, for this simulation, the inference algorithm will not be able to deduce the goal or the gain. This is indeed evident from the large errors in the third row in Table 3.

In the middle of the spectrum is the situation where there is exactly one active obstacle to avoid. Figure 2(b) shows a simulation where the robot is navigating towards its goal, and obstacle 3 remains active during the robot's journey to the goal. Since the robot can use one degree of freedom to avoid that obstacle, and use the remaining degree of freedom to perform the task, its position-velocity measurements **will** have information about the task. Therefore, it **is** possible to infer either the goal or the gain. The second row in Table 3 shows the errors in estimating the goal. These errors are negligible, and comparable to those in the first row, demonstrating that goal inference is successful.

5. EXPERIMENTAL RESULTS

We also performed physical experiments with three Khepera robots to analyze the effect of noise in hardware and perception sensing on the estimation of goals. We conducted experiments with three robots and performed a total of eight runs in which the initial positions of the robots, their gains and safety margins were varied, but their individual respective goal locations were kept same through the runs. Table 4 reports the errors of these individual runs for the three algorithms along with the mean and standard deviations. The videos of these experiments can be found at `https://bit.ly/3s7yovL`. From the numerical values of the errors, it is evident that all algorithms are able to obtain estimates of the goals that are within a maximum of $5cms$ of the true goals.

Table 4. Goal Errors in Experiments $\|\hat{\boldsymbol{p}}_d - \boldsymbol{p}_d^*\|$ in [m]

Run Number	Predict. loss algorithm	KKT loss algorithm	Suboptimality loss algorithm
1	0.0166	0.0073	0.0566
2	0.0502	0.0185	0.1394
3	0.0171	0.0079	0.0133
4	0.0292	0.0218	0.0178
5	0.0047	0.0046	0.0332
6	0.0057	0.0058	0.0408
7	0.0043	0.0041	0.0951
8	0.0053	0.0050	0.0281
Mean	**0.0166**	**0.0094**	**0.0530**
Std.	**0.0151**	**0.0064**	**0.0406**

6. CONCLUSIONS

We considered the problem of inference of parameters of tasks being performed by robots in 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 several IO algorithms to solve this problem in a batch setting and demonstrated how accurate estimates of underlying parameters can be reconstructed. Furthermore, we derived QP based reformulations of the KKT-loss minimization and suboptimality minimization algorithms. Finally, using our previously derived criteria for successful inference, we demonstrated that these IO algorithms may fail to identify the correct underlying task parameters whenever the ego robot interacts with two or more obstacles. In future, we plan to extend this work to simultaneously learn parameters of the cost function as well as constraints in the ego-robot's forward problem. We also plan to consider robust estimation in the presence of model mismatch and measurement uncertainty.

7. APPENDIX

The expressions for $H_1, \boldsymbol{g}_1, H_2, \boldsymbol{g}_2$ in (34) and (36) are

$$
\begin{aligned}
H_1 &= G^T(\boldsymbol{x}(k))E_k^{\boldsymbol{y}} + 2E^{\boldsymbol{u}T}C(\boldsymbol{x}(k))E^{\boldsymbol{\theta}} \\
\boldsymbol{g}_1 &= 2E^{\boldsymbol{u}T}(E^{\boldsymbol{u}}\boldsymbol{\xi}^*(k) - \boldsymbol{d}(\boldsymbol{x}(k))) \\
H_2 &= -\bigg(2\boldsymbol{\xi}^{*T}(k)E^{\boldsymbol{u}T}C(\boldsymbol{x}(k))E^{\boldsymbol{\theta}} + 2\boldsymbol{\xi}^{*T}(k)E^{\boldsymbol{u}T}\boldsymbol{h}^T(\boldsymbol{x}(k))E_k^{\boldsymbol{y}} \\
&\quad + E_k^{\epsilon}\bigg) \\
\boldsymbol{g}_2 &= -2\boldsymbol{\xi}^{*T}(k)E^{\boldsymbol{u}T}E^{\boldsymbol{u}}\boldsymbol{\xi}^* + 2\boldsymbol{\xi}^{*T}(k)E^{\boldsymbol{u}T}\boldsymbol{d}(\boldsymbol{x}(k))
\end{aligned} \tag{38}
$$

REFERENCES

Ahuja, R.K. and Orlin, J.B. (2001). Inverse optimization. *Operations Research*, 49(5), 771–783.

Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, 3420–3431. IEEE.

Aswani, A., Shen, Z.J., and Siddiq, A. (2018). Inverse optimization with noisy data. *Operations Research*, 66(3), 870–892.

Bertsimas, D., Gupta, V., and Paschalidis, I.C. (2012). Inverse optimization: A new perspective on the black-litterman model. *Operations research*, 60(6), 1389–1403.

Bertsimas, D., Gupta, V., and Paschalidis, I.C. (2015). Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2), 595–633.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F.E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3), 376–386.

Carr, S. and Lovejoy, W. (2000). The inverse newsvendor problem: Choosing an optimal demand portfolio for capacitated resources. *Management Science*, 46(7), 912–927.

Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2), 243–255.

Englert, P., Vien, N.A., and Toussaint, M. (2017). Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 36(13-14), 1474–1488.

Esfahani, P.M., Shafieezadeh-Abadeh, S., Hanasusanto, G.A., and Kuhn, D. (2018). Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167(1), 191–234.

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

Grover, J., Liu, C., and Sycara, K. (2020a). Feasible region-based identification using duality (extended version). *arXiv preprint arXiv:2011.04904*.

Grover, J.S., Liu, C., and Sycara, K. (2020b). Parameter identification for multirobot systems using optimization based controllers (extended version). *arXiv preprint arXiv:2009.13817*.

Johnson, M., Aghasadeghi, N., and Bretl, T. (2013). Inverse optimal control for deterministic continuous-time nonlinear systems. In *52nd IEEE Conference on Decision and Control*, 2906–2913. IEEE.

Keshavarz, A., Wang, Y., and Boyd, S. (2011). Imputing a convex objective function. In *2011 IEEE international symposium on intelligent control*, 613–619. IEEE.

Liu, C. and Tomizuka, M. (2014). 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.

Majumdar, A., Singh, S., Mandlekar, A., and Pavone, M. (2017). Risk-sensitive inverse reinforcement learning via coherent risk models. In *Robotics: Science and Systems*, volume 16, 117.

Molloy, T.L., Ford, J.J., and Perez, T. (2018). Online inverse optimal control on infinite horizons. In *2018 IEEE Conference on Decision and Control (CDC)*, 1663–1668. IEEE.

Pierpaoli, P., Ravichandar, H., Waytowich, N., Li, A., Asher, D., and Egerstedt, M. (2019). Inferring and learning multi-robot policies by observing an expert. *arXiv preprint arXiv:1909.07887*.

Van Den Berg, J., Guy, S.J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In *Robotics research*, 3–19. Springer.

Van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, 1928–1935. IEEE.

Wang, L., Ames, A.D., and Egerstedt, M. (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3), 661–674.

Wei, T. and Liu, C. (2019). Safe control algorithms using energy functions: A uni ed framework, benchmark, and new directions. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 238–243. IEEE.

Yang, C., Jiang, Y., He, W., Na, J., Li, Z., and Xu, B. (2018). Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Transactions on Industrial Electronics*, 65(10), 8112–8123.

Zhang, Y. and Shell, D.A. (2020). Plans that remain private, even in hindsight.