Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Inferring Objectives in Continuous Dynamic Games from Noise-Corrupted Partial State Observations

Lasse Peters*, David Fridovich-Keil†, Vicenç Rubies-Royo‡, Claire J. Tomlin‡ and Cyrill Stachniss*
* University of Bonn, Germany † University of Texas, Austin, USA ‡ University of California, Berkeley, USA
Email: {lasse.peters, cyrill.stachniss}@igg.uni-bonn.de, dfk@utexas.edu, {vrubies, tomlin}@berkeley.edu

***Abstract*—Robots and autonomous systems must interact with one another and their environment to provide high-quality services to their users. Dynamic game theory provides an expressive theoretical framework for modeling scenarios involving multiple agents with differing objectives interacting over time. A core challenge when formulating a dynamic game is designing objectives for each agent that capture desired behavior. In this paper, we propose a method for inferring parametric objective models of multiple agents based on observed interactions. Our inverse game solver jointly optimizes player objectives and continuous-state estimates by coupling them through Nash equilibrium constraints. Hence, our method is able to directly maximize the observation likelihood rather than other non-probabilistic surrogate criteria. Our method does not require full observations of game states or player strategies to identify player objectives. Instead, it robustly recovers this information from noisy, partial state observations. As a byproduct of estimating player objectives, our method computes a Nash equilibrium trajectory corresponding to those objectives. Thus, it is suitable for downstream trajectory forecasting tasks. We demonstrate our method in several simulated traffic scenarios. Results show that it reliably estimates player objectives from a short sequence of noise-corrupted partial state observations. Furthermore, using the estimated objectives, our method makes accurate predictions of each player's trajectory.**

I. Introduction

Most robots use motion planning and optimal control methods to select and execute actions when operating in the real world. Commonly used approaches require specifying the objective to optimize. In many real-world applications, however, designing optimal control objectives is challenging. For example, tuning cost parameters, even in the case of a linear-quadratic regulator (LQR), can be a tedious heuristic process when performed manually. As a result, it can be desirable to learn optimal control objectives automatically from demonstrations. To this end, researchers have investigated learning from demonstration and inverse optimal control (IOC). Recent work shows promising results, even for complex problems with large state and observation spaces [10, 23].

Optimal control methods, however, are not directly suitable for interactive settings with multiple agents. For example, consider multiple vehicles engaged in lane changes on a crowded highway. In this setting, each agent has its own objective that naturally depends upon the behavior of others. For instance, agents may wish to maintain a safe distance from others and at the same time travel at a preferred speed. Thus, their interaction is more accurately characterized as a noncooperative game-theoretic equilibrium rather than as the solution to a joint optimal control problem. Despite the added complexity of these noncooperative interactions, recent developments enable computationally-efficient solutions to the dynamic games which arise in multi-agent robotic settings [8, 11, 12, 19].

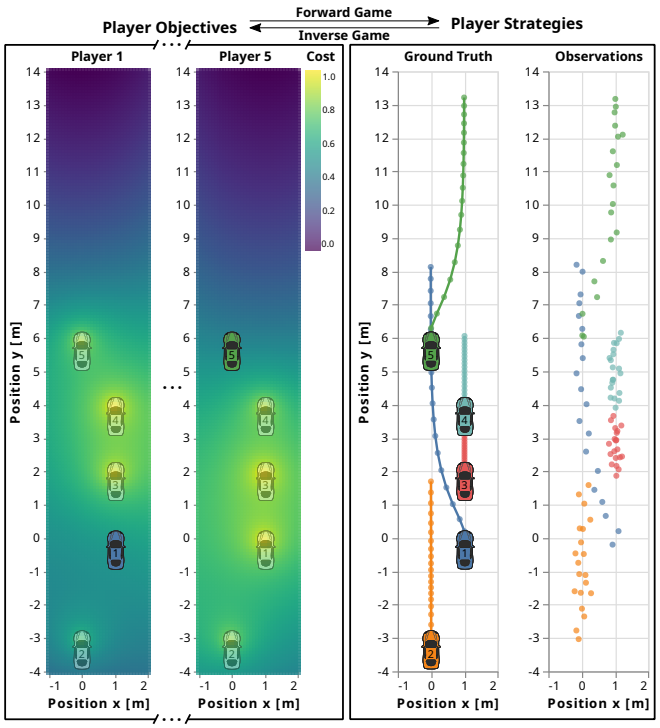


Fig. 1. Inverse and forward versions of a dynamic game modeling a 5-player highway driving scenario. The solution of the forward problem maps the player objectives (left) to the players' optimal strategies (right). Our method solves the inverse problem: it takes noisy, partial state observations of multi-agent interaction as input to recover an objective model for each player that explains the the observed behavior. The visualized slice of the cost landscape shows one important aspect of the recovered objective model, namely, each player's preference to keep a safe distance from others. The inferred objectives defines an abstract game-theoretic behavior model that can be used to predict player strategies for arbitrary agent configurations.

There are similar challenges in designing objectives for dynamic games as for single-player optimal control problems. As in the single-player case, automatic cost learning promises to circumvent this difficulty. However, cost inference takes on an even more important role in multi-agent settings. That is, any individual player must also understand the objectives of other players to interact effectively. In this paper, we study

the problem of identifying such cost functions from noisy, partial state observations of multi-agent interactions. Figure 1 shows an overview of this problem, commonly referred to as an *inverse* dynamic game.

The main contribution of this paper is a novel, noise-robust technique that identifies unknown parameters of each player's cost function in a noncooperative game based on observed multi-agent interactions. Recovering these unknown parameters allows us to infer important aspects of each players preferences. For example, in the highway driving setting of Figure 1 one such aspect is each agent's preference to avoid collisions with other vehicles. We propose a solution approach to the inverse dynamic game problem which estimates all player's states and control inputs jointly with their unknown objective parameters by coupling them through noncooperative equilibrium constraints. Through this formulation, our method is can operate seamlessly with noise-corrupted partial state observations. Based on simulated traffic scenarios, we evaluate our method and provide comparisons to existing methods in a Monte Carlo study. We show that our method is more robust to incomplete state information and observation noise. As a result, our method identifies player objectives more reliably, and predicts player trajectories more accurately.

II. Related Work

We begin by discussing recent advances in the well-studied area of IOC. While methods from that field address only single-player settings, this body of work exposes many of the important mathematical and algorithmic concepts that appear in games. We discuss how some of these approaches have been applied in the multi-player setting and emphasize the connections between existing approaches and our contributions.

A. Single-Player Inverse Optimal Control

The IOC problem has been extensively studied since the well-known work of Kalman [15]. In the context of inverse reinforcement learning (IRL), early formulations such as that of Ng and Russell [25] and maximum-entropy variants [31] have proven successful in treating problems with discrete state and control sets. In robotic applications, optimal control problems typically involve decision variables in a continuous domain. Hence, recent work in IOC differs from the IRL literature mentioned above as it is explicitly designed for smooth problems.

One common framework for addressing IOC problems with nonlinear dynamics and nonquadratic cost structures is bilevel optimization [1, 24]. Here, the outer problem is a least squares or maximum likelihood estimation (MLE) problem in which demonstrations are matched with a nominal trajectory estimate and decision variables parameterize the objective of the underlying optimal control problem. The inner problem determines the nominal trajectory estimate as the optimizer of the "forward" (i.e., standard) optimal control problem for the outer problem's decision variables. A key benefit of bilevel IOC formulations is that they naturally adapt to settings with noise-corrupted partial state observations [1].

Early bilevel formulations for IOC utilize derivative-free optimization schemes to estimate the unknown objective parameters in order to avoid explicit differentiation of the solution to the inner optimal control problem [24]. That is, the inner solver is treated as a black-box mapping from cost parameters to optimal trajectories which is utilized by the outer solver to identify the unknown parameters using a suitable derivative-free method. While black-box approaches can be simple to implement due to their modularity and lack of reliance on derivative information, they often suffer from a high sampling complexity [26]. Since each sample in the context of black-box IOC methods amounts to solving a full optimal control problem, such approaches remain intractable for scenarios with large state spaces or additional unknown parameters, such as unknown initial conditions.

Other works instead embed the Karush–Kuhn–Tucker (KKT) conditions of the inner problem as constraints on the outer problem. Since these techniques enforce only first-order necessary conditions of optimality, globally optimal observations are unnecessary and locally optimal demonstrations suffice. Yet, a key computational difficulty of KKT-constrained IOC formulations is that they yield a nonconvex optimization problem due to decision variables in the outer problem appearing nonlinearly with inner problem variables in KKT constraints. This occurs even in the relatively benign case of linear-quadratic IOC.

In contrast to bilevel optimization formulations where necessary conditions of optimality are embedded as constraints, recent methods [3, 10, 21, 23] minimize the residual of these conditions directly at the demonstrations. Since the observed demonstration is assumed to satisfy any constraints of the underlying forward optimal control problem, this method can be formulated as fully unconstrained optimization. Additionally, these residual formulations yield a *convex* optimization problem if the class of objective functions is convex in the unknown parameters at the demonstration [10, 16]. This condition holds in the common setting of linearly-parameterized objective functions. Levine and Koltun [21] propose a variant of this approach that utilizes quadratic approximations of the reward model around demonstrations to derive optimality residuals in a maximum entropy framework. Englert and Toussaint [10] present an extensions of this method do accommodate inequality constraints on states and inputs. Much like KKT-constrained formulations, these residual methods operate on locally optimal demonstrations. However, an important limitation of residual methods is that they require observations of full state and input sequences. More recently, Menner and Zeilinger [23] compared IOC techniques based on KKT constraints and residuals and demonstrated inferior performance of the latter even in problems with linear dynamics and quadratic target objectives.

Our work takes inspiration from the KKT-constraint formulation for single-player IOC as discussed by Albrecht et al. [1] and Menner and Zeilinger [23]. While these work apply only to single-player settings, we utilize necessary conditions for open-loop Nash equilibria (OLNEs) to generalize this

approach to noncooperative multi-player scenarios.

B. Multi-Player Inverse Dynamic Games

Many of the IOC techniques discussed above have close analogues in the context of multi-player inverse dynamic games.

As in single-player IOC, methods akin to black-box bilinear optimization have also been studied in the context of inverse games [20, 27]. Peters [27] use a particle-filtering technique for online estimation of human behavior parameters. This work demonstrates the importance of inferring human behavior parameters for accurate prediction in interactive scenarios. However, there, inference is limited to a single parameter and the work highlight the challenges associated with scaling this sampling based approach to high-dimensional latent parameter spaces. Le Cleac'h et al. [20] employ a similar derivative-free filtering technique based on an unscented Kalman filter. While this approach drastically reduces the overall sample complexity, it still relies on exact observations of the state to reduce the required number of solutions to full dynamic games at the inner level.

Another line of research has put forth solution techniques for inverse games that follow from the residual methods outlined in Section II-A [4, 14, 17, 28]. Köpf et al. [17] study a special case of an inverse linear-quadratic game in which the equilibrium feedback strategies of all but one player are known. This assumption reduces the estimation problem to single-player IOC to which the residual methods discussed above can be applied directly. Rothfuß et al. [28] present a more general approach that does not exploit such special structure but instead minimizes the residual of the first-order necessary conditions for a local OLNE. Inga et al. [14] present a variant of this OLNE residual method in a maximum entropy framework, generalizing the single-player IOC algorithm proposed by Levine and Koltun [21]. Recently, Awasthi and Lamperski [4] also extended the OLNE residual method of Rothfuß et al. [28] to inverse games with state and input constraints. This approach extends that of Englert and Toussaint [10] to noncooperative multi-player scenarios.

All of these inverse game KKT residual methods share many properties with their single-player counterparts. In particular, since they rely upon only local equilibrium criteria, they are able to recover player objectives even from local—rather than only global—equilibrium demonstrations. However, as in the single-player case, they rely upon observation of both state and input to evaluate the residuals.

In contrast to KKT residual methods [4, 14, 28], we enforce these conditions as constraints on a jointly estimated trajectory, rather than minimizing the residual of these conditions directly at the observation. Thus, our method can explicitly account for observation noise, partial state observability, and unobserved control inputs. Furthermore, in contrast to black-box approaches to the inverse dynamic game problem [20, 27], our method does not require repeated solutions of the underlying forward game. Moreover, our method returns a full forward game solution in addition to the estimated objective parameters for all players.

III. Background: Open-Loop Nash Games

This section offers a concise background on *forward* open-loop Nash games. In this work, we use the term forward to disambiguate this class of problems from that of learning costs in games (i.e., inverse games). For a thorough treatment, refer to Başar and Olsder [5]. Note that OLNE differ from noncooperative equilibrium concepts in other information structures including feedback Nash equilibria [5, Chapter 3]. Recent algorithms for open-loop games are those by Di and Lamperski [8], Le Cleac'h et al. [19], and for feedback games we refer to Fridovich-Keil et al. [11] and Laine et al. [18].

An open-loop (infinite) Nash game with N players is characterized by *state* $x \in \mathbb{R}^n$ and control inputs for each player $u^i \in \mathbb{R}^{m^i}$ which follow *dynamics* $x_{t+1} = f_t(x_t, u_t^1, \dots, u_t^N)$ at each discrete time $t \in [T] := \{1, \dots, T\}$. Each player has a cost function[1] $J^i := \sum_{t=1}^{T} g_t^i(x_t, u_t^1, \dots, u_t^N)$, which is implicitly a function of the initial condition x_1 and explicitly of both the control inputs for each player $\mathbf{u}^i := (u_1^i, \dots, u_T^i)$ and the state trajectory $\mathbf{x} := (x_1, \dots, x_T)$. The tuple of initial state, joint dynamics, and player objectives which fully characterizes a game is denoted $\Gamma := \left(x_1, f, \{J^i\}_{i \in [N]}\right)$ throughout this work.

Given a sequence of control inputs for all players $\mathbf{u} := (\mathbf{u}^1, \dots, \mathbf{u}^N)$ the states are determined by the dynamics and initial condition. Note that for clarity we use bold variables to indicate aggregation over time and omit player indices to further aggregate a quantity over all players. Hence, for shorthand, we will overload cost notation to define $J^i(\mathbf{u}; x_1) \equiv J^i(\mathbf{u}^1, \dots, \mathbf{u}^N; x_1) \equiv J^i(\mathbf{x}, \mathbf{u}^1, \dots, \mathbf{u}^N)$.

Nash equilibria are solutions to the coupled optimization problems, one for each player Pi:

$$\forall i \in [N] \begin{cases} \min\limits_{\mathbf{x}, \mathbf{u}^i} \; J^i(\mathbf{u}; x_1) & \text{(1a)} \\ \text{s.t. } x_{t+1} = f_t(x_t, u_t^1, \dots, u_t^N), \forall t \in [T-1]. & \text{(1b)} \end{cases}$$

Nash equilibrium strategies $\mathbf{u}^* := (\mathbf{u}^{1*}, \dots, \mathbf{u}^{N*})$ satisfy the inequality $J^1(\mathbf{u}^1, \mathbf{u}^{2*}, \dots, \mathbf{u}^{N*}; x_1) \geq J^1(\mathbf{u}^*; x_1)$ for the first player (P1) and likewise for all other players. Intuitively, at equilibrium no player wishes to unilaterally deviate from their respective strategy $\mathbf{u}^{i*}$. Note that this solution concept differs from a formulation as joint optimal control problem. In particular, players' objectives may conflict in which case the resulting equilibrium is *noncooperative*.

Running example: To make these concepts concrete, we introduce the following running example. Consider $N = 2$ vehicles avoiding collision. Each vehicle has its own state x^i such that the global game state is concatenated as $x = (x^1, x^2)$. Further, each vehicle follows unicycle dynamics at

[1]This setup readily extends to the constrained case as in [18, 19]—as our own work does. We ignore such constraints here for clarity.

time discretization Δt:

$$x^i_{t+1} = \begin{cases} \textit{(x-position)} & p^i_{x,t+1} = p^i_{x,t} + \Delta t\ v^i_t \cos\psi^i_t \\ \textit{(y-position)} & p^i_{y,t+1} = p^i_{y,t} + \Delta t\ v^i_t \sin\psi^i_t \\ \textit{(heading)} & \psi^i_{t+1} = \psi^i_t + \Delta t\ \omega^i_t \\ \textit{(speed)} & v^i_{t+1} = v^i_t + \Delta t\ a^i_t, \end{cases} \tag{2}$$

where $u^i_t = (\omega_t, a_t)$ is the yaw rate and longitudinal acceleration. Finally, each player's objective is characterized by a running cost g^i_t defined as a combination of multiple basis functions:

$$g^i_t = \sum_{j=1}^{5} w^i_j g^i_{j,t} \begin{cases} g^i_{1,t} = \mathbf{1}(t \geq T - t_{\text{goal}})\mathrm{d}(x^i_t, x^i_{\text{goal}}) & \text{(3a)} \\ g^i_{2,t} = -\log(\|p_i - p_{-i}\|_2^2) & \text{(3b)} \\ g^i_{3,t} = (v^i)^2 & \text{(3c)} \\ g^i_{4,t} = (\omega^i_t)^2 & \text{(3d)} \\ g^i_{5,t} = (a^i_t)^2, & \text{(3e)} \end{cases}$$

where $w^i_j \in \mathbb{R}_+$ are non-negative weights for each cost component, p_i and p_{-i} denote the position of player Pi and its opponent, and $\mathrm{d}(\cdot,\cdot)$ is a distance mapping. For this example we choose $\mathrm{d}(x^i_t, x^i_{\text{goal}}) = \|p^i_t - p^i_{\text{goal}}\|_2^2$ to compute squared distance from a goal position. The basis functions encode the following aspects of each player's preferences:

1) be close to the goal state in the last t_{goal} time steps (3a),
2) avoid close proximity to the other vehicle (3b),
3) avoid high speed (3c) and large control effort (3d, 3e).

This game is inherently noncooperative since players must compete to reach their own goals safely and efficiently: No player wishes to deviate from a direct path to the goal, yet all players also wish to avoid collision. Hence, they must negotiate these conflicting objectives and thereby find an equilibrium of the underlying game. Note that the cost structure in (3) can also be used to encode more complex problems such as the highway driving scenario depicted in Figure 1. For clarity, we limit discussion to a simplified 2-player scenario in this running example, and present a 5-player example later in the paper.

IV. Problem Formulation

A Nash game requires finding optimal strategies for each player, given their objectives. In contrast, this work is concerned with the inverse problem that requires finding players' objectives for which the observed behavior is a Nash equilibrium. In short, it seeks an answer to the question: *Which player objectives explain the observed interaction?*

We cast this question as an estimation problem. To that end, we assume that each player's cost function is parameterized by a vector $\theta^i \in \mathbb{R}^{k^i}$, i.e., $J^i(\cdot;\theta^i) \equiv \sum_{t=1}^{T} g^i_t(x_t, u^1_t, \ldots, u^N_t; \theta^i)$.

This formulation includes arbitrary smooth and potentially nonlinear parameterizations. Hence, a player's objective may also be parameterized by a differentiable function approximator such as an artificial neural network. While such a parameterization is very flexible, it may also reduce the interpretability of the resulting parameters. The parameterization of player objectives may further be designed to exploit domain knowledge for a specific application.

Running example: For clarity of presentation, our running example throughout this paper considers a linear parameterization $\theta^i = (w^i_1, \ldots, w^i_5)$ which weights individual cost functions from (3) that comprise the overall objective for each player. As we will see in Section VI, this parametrization is able to capture a variety of interactive traffic scenarios.

Thus equipped, we seek to estimate those parameter values that maximize the likelihood of a given sequence of partial state observations $\mathbf{y} := (y_1, \ldots, y_T)$ for the induced parametric family of games $\Gamma(\theta) = \big(x_1, f, \{J^{(i)}(\,\cdot\,;\theta^{(i)})\}_{i\in[N]}\big)$:

$$\begin{aligned} \max_{\theta,\mathbf{x},\mathbf{u}} \quad & p(\mathbf{y} \mid \mathbf{x}, \mathbf{u}) && \text{(4a)} \\ \text{s.t.} \quad & (\mathbf{x}, \mathbf{u}) \text{ is an OLNE of } \Gamma(\theta) && \text{(4b)} \\ & (\mathbf{x}, \mathbf{u}) \text{ is dynamically feasible under } f, && \text{(4c)} \end{aligned}$$

where, θ is the vector of aggregated parameters over all players, i.e., $\theta := (\theta^1, \ldots, \theta^N)$, and $p(\mathbf{y} \mid \mathbf{x}, \mathbf{u})$ denotes a known observation likelihood model.

In the simplest case, the inverse planner receives an exact observation of the full state and input sequence and the observation model is a Dirac delta function. In general, however, the observation model $p(\mathbf{y} \mid \mathbf{x}, \mathbf{u})$ allows modelling noise-corrupted partial state observations. Thus, our formulation is amenable to more realistic scenarios and real sensors, for example, range/bearing measurements from a LiDAR.

In summary, the above formulation of the inverse dynamic game problem attempts a *joint* estimation of states, control inputs, and player objectives by tightly coupling them through Nash equilibrium constraints. Note that this is an important difference to existing formulations [4, 28] which treat these estimation problems separately and do not exploit the strong Nash priors which couple them. We discuss these methods in further detail below and compare to them as a baseline.

V. Our Approach

This section describes our main contribution: a novel solution technique for identifying objective parameters of players in a continuous game. Our formulation is directly expressed in the standard format of a constrained optimization problem. That is, our method yields a mathematical program which can be encoded using well-established modeling languages (e.g., CasADi [2], JuMP [9], and YALMIP [22]) and solved by a number of off-the-shelf methods (e.g., IPOPT [29], KNITRO [7], and SNOPT [13]).

A. Encoding Nash Equilibrium Constraints

A key challenge to solving the estimation problem in (4) is posed by the requirement to encode the equilibrium constraint in (4b) in order to couple the estimates of game trajectory $(\mathbf{x}, \mathbf{u})$ and objective parameters θ. In this work, akin to the bilevel optimization approach to single-player IOC of Albrecht et al. [1], we encode this forward optimality constraint via the corresponding first-order necessary conditions.

For an OLNE, the first-order necessary conditions are given by the union of the individual players' KKT conditions, i.e.,

$$\mathbf{G}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) := \begin{bmatrix} \left.\begin{matrix} \nabla_{\mathbf{x}} J^i + \boldsymbol{\lambda}^{i\top} \nabla_{\mathbf{x}} \mathbf{F}(\mathbf{x},\mathbf{u}) \\ \nabla_{\mathbf{u}^i} J^i + \boldsymbol{\lambda}^{i\top} \nabla_{\mathbf{u}^i} \mathbf{F}(\mathbf{x},\mathbf{u}) \end{matrix}\right\} \forall i \in [N] \\ \mathbf{F}(\mathbf{x},\mathbf{u}) \end{bmatrix} = \mathbf{0}. \tag{5}$$

The first two blocks of this equation are repeated for all players Pi and $\mathbf{F}(\mathbf{x},\mathbf{u})$ collects the dynamics constraint error from (1a) with t^{th} block of $x_{t+1} - f_t(x_t, u_t^1, \dots, u_t^N)$. Here, we introduce costates $\boldsymbol{\lambda}^i := (\lambda_1^i, \dots, \lambda_{T-1}^i)$ for all players, where $\lambda_t^i \in \mathbb{R}^n$ is the Lagrange multiplier associated with the constraint between decision variables at time step t and $t+1$ in (1a).

Incorporating (5) as constraints, we cast the inverse dynamic game problem of (4) as

$$\max_{\theta,\mathbf{x},\mathbf{u},\boldsymbol{\lambda}} \quad p(\mathbf{y} \mid \mathbf{x},\mathbf{u}) \tag{6a}$$

$$\text{s.t.} \quad \mathbf{G}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda};\theta) = \mathbf{0}. \tag{6b}$$

Here, the costates $\boldsymbol{\lambda}$ of (5) appear as *additional primal decision variables*. Further, $\mathbf{G}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda};\theta)$ is the KKT residual from (5), with added explicit dependency on the cost parameters θ. In practice, it can also be important to regularize or otherwise constrain parameters.

Running example: j To ensure that the problem remains well-defined, we constrain all parameters to sum to unity for all players, and we also require positive control cost weights, i.e., $\theta_{4,5}^i > \epsilon \geq 0$.

Note that (6b) does not explicitly depend upon observations $\mathbf{y}$ but instead utilizes the trajectory $(\mathbf{x},\mathbf{u})$ which we optimize simultaneously to maximize observation likelihood. Thus, our method does not rely on complete observation of states, or even inputs. Rather, we reconstruct this missing information by exploiting knowledge of dynamics and objective model structure.

Finally, we also note that our method applies coherently when there are multiple observed trajectories; our development here treats the single-trajectory observation case for clarity.

B. Structure of Constraints

Consider the t^{th} term in the first block of $\mathbf{G}$ in (5)

$$0 = \nabla_{x_t} J^i(\mathbf{x},\mathbf{u};\theta^i) + \boldsymbol{\lambda}^{i\top} \nabla_{x_t} \mathbf{F}(\mathbf{x},\mathbf{u}) \tag{7a}$$

$$= \nabla_{x_t} g_t^i(x_t,u_t;\theta^i) + \lambda_{t-1}^i - \lambda_t^{i\top} \nabla_{x_t} f_t(x_t,u_t), \tag{7b}$$

with aggregated player inputs $u_t = (u_t^1, \dots, u_t^N)$. The right hand side contains two potential sources of nonlinearities that may render (6) nonconvex. First, for any non-trivial player objectives, the gradient of the running cost $\nabla_{x_t} g$ must couple state x_t and inputs u_t with parameters θ. Second, costate λ_t^i multiplies the potentially state-dependent Jacobian of dynamics $\nabla_{x_t} f_t$.

When the dynamics are affine, the Jacobian is a constant and the latter term remains linear in decision variables λ_t^i and x_t. Furthermore, in a forward dynamic game, the parameter vector θ is given in the problem description and thus also the former nonlinearity vanishes if player objectives are quadratic and include no mixed terms in states and inputs. As a result of this structure, linear-quadratic OLNE problems admit an analytic solution [5, Chapter 6] and only more complex problems require the use of iterative solution techniques, e.g. a Newton method as employed by Le Cleac'h et al. [19].

In an inverse game, however, objective parameters θ necessarily appear as decision variables. Since our method additionally estimates the game trajectory $(\mathbf{x},\mathbf{u})$ to account for noise-corrupted partial state observations, the equilibrium constraints in (6b) remain at least bilinear even for linear-quadratic games and the optimization problem is inevitably nonconvex. Therefore, our approach inherently relies on an iterative method to identify solutions of (6) and the ability to solve this problem can depend on suitable initialization of the decision variables.

To this end, we leverage the observation sequence $\mathbf{y}$ to initialize the decision variables $\mathbf{x}$ and $\mathbf{u}$ by solving a relaxed version of (6) without equilibrium constraints. That is, we compute the initialization of the state-input trajectory as the solution of

$$\tilde{\mathbf{x}},\tilde{\mathbf{u}} := \arg\max_{\mathbf{x},\mathbf{u}} \quad p(\mathbf{y} \mid \mathbf{x},\mathbf{u}) \tag{8a}$$

$$\text{s.t.} \quad \mathbf{F}(\mathbf{x},\mathbf{u}) = \mathbf{0}. \tag{8b}$$

This pre-solve step can be interpreted as sequentially activating the different components of the KKT constraints in (6b). First, we enforce only dynamics constraints to recover a trajectory that maximizes observation likelihood while remaining dynamically feasible, regardless of Nash equilibrium constraints. Subsequently, we activate the Nash equilibrium constraints encoded by the first two blocks of $\mathbf{G}$ and solve the full problem in (6). Note that in this second step, the state $\mathbf{x}$ and input $\mathbf{u}$ still remain decision variables and thus the game trajectory is further refined during optimization of the objective parameters θ.

C. Maximum Likelihood Objective

A common yet effective class of observation models assumes additive white Gaussian noise. In this case, each observation depends only upon the state and input at the current time step, i.e.,

$$y_t = h_t(x_t, u_t^1, \dots, u_t^N) + n_t, \tag{9}$$

where h_t is a deterministic mapping from the current state and input to the *expected* observation, and n_t is a zero-mean white noise-process, i.e., $n_t \sim \mathcal{N}(0,\Sigma_t)$. For this class of observation models the inverse OLNE problem can be equivalently treated as a constrained nonlinear least-squares problem.

Running example: We presume isotropic additive white Gaussian noise and minimize the corresponding negative log-likelihood objective $\sum_t \|y_t - h_t(x_t)\|_2^2$ in (4a). In summary, the inverse problem to the collision avoidance game entails the following task: Find those weights to the basis functions in (3) for which the corresponding game solution generates expected observations near the observed data.

VI. Experiments

This section analyzes the performance of the proposed inverse game solution approach and compares it to a state-of-the-art baseline in a Monte Carlo study.

A. Baseline: Minimizing KKT Residuals

We use as a baseline the KKT residual approach presented in Rothfuß et al. [28] and Awasthi and Lamperski [4]. Other approaches exist as described in Section II-B, e.g., [20, 27], however these black-box approaches do not utilize derivative information. Algorithmic differences are sufficiently extensive that they render a direct comparison difficult to interpret.

Like our method, the KKT residual approach uses the first-order necessary conditions in (5) to encode forward optimality. However, it does not jointly optimize a trajectory estimate for the problem. Instead, these method assume access to a *preset* trajectory along which they minimize the violation of the optimality constraint. That is, the baseline solves

$$\min_{\theta,\boldsymbol{\lambda}} \left\|\mathbf{G}(\tilde{\mathbf{x}},\tilde{\mathbf{u}},\boldsymbol{\lambda};\theta)\right\|_2^2, \tag{10}$$

where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{u}}$ are assumed to be given as part of the observation. Thus, only the objective parameters θ and the costates $\boldsymbol{\lambda}$ are decision variables in the problem.

In scenarios with incomplete information due to unobserved inputs, noise, or partial state observations, the solution to the optimization problem in (10) is not meaningful and not always well-defined. Instead, the state-input trajectory must first be estimated from the observation sequence $\mathbf{y}$ in order to evaluate the constraint residual. To this end, we extend the technique of [4, 28] with a pre-processing step to estimate the state-input trajectory as the solution of (8). That is, we recover the dynamically feasible state-input sequence that maximizes the likelihood of the observation. This baseline can be thought of as a sequential, loosely coupled version of our approach. Instead of optimizing a trajectory estimate jointly with the objective parameters θ, they are estimated one at a time in a two-stage procedure.

B. Experimental Setup

We implement our proposed approach as well as the KKT residual baseline [28] in Julia [6] using the algebraic modeling language JuMP [9]. Due to the abstraction provided by the modeling language our implementation is agnostic to the algorithm used to solve the synthesized problem description. In this work, we use the open source COIN-OR IPOPT algorithm [29]. The source code is publicly available at https://github.com/PRBonn/PartiallyObservedInverseGames.jl.

To compare robustness and performance of our method with the baseline, we perform a Monte Carlo study. For this purpose, we fix a set of cost weights in (3) for each player, find corresponding OLNE trajectories as roots of (5) using the well-known iterated best response (IBR) algorithm [30], and corrupt them with additive white Gaussian noise as described in (9). We generate 40 random observation sequences at each of 22 different levels of isotropic observation noise. For each of the resulting 880 observation sequences we run both our method and the baseline to recover estimates of weights $w_j^i, j \in \{1,\dots,5\}$ for each player. Note that all methods infer objective parameters from observation of a single trajectory. That is, each estimate in the Monte Carlo study relies only upon $25\,\mathrm{s}$ of interaction history of a single scenario instead of batches of multiple demonstrations. This evaluation setup is designed to benchmark the methods in a realistic setting where an estimator typically cannot observe the same scene multiple times.

C. Simulation Experiments

1) 2-Player Running Example: First, we evaluate our method and the residual baseline in a Monte Carlo study using the running example of collision-avoidance with $N = 2$ players. This experiment aims to demonstrate the performance gap of both methods in a conceptually simple and more easily interpretable scenario.

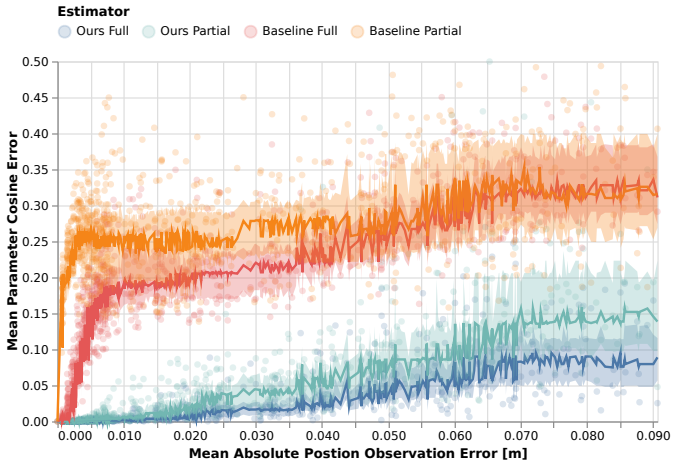
Figure 2 shows the estimator performance for varying levels of observation noise in two different metrics. Figure 2(a) reports the mean cosine error of the objective parameter estimates. That is, we measure dissimilarity between the unobserved true model parameters θ_{true} and the estimate θ_{est} by

$$D_{\cos}(\theta_{\text{true}},\theta_{\text{est}}) = 1 - \frac{1}{N}\sum_{i\in[N]} \frac{\theta_{\text{true}}^{i\top}\theta_{\text{est}}^{i}}{\left\|\theta_{\text{true}}^{i}\right\|_2 \left\|\theta_{\text{est}}^{i}\right\|_2}, \tag{11}$$

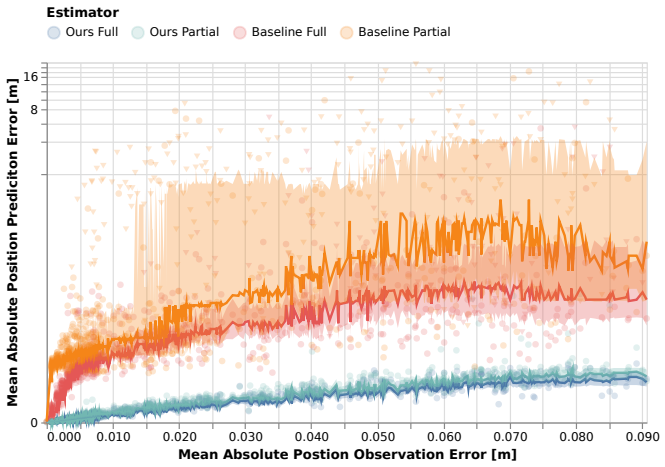
where the mean is taken over the N players. The normalization of the parameter vectors in (11) reflects the fact that the absolute scaling of the cost weights within each player's objective does not effect their optimal behavior. Hence, this metric measures the estimator performance in model parameter space.

Figure 2(b) shows the mean absolute position error for trajectory predictions computed by finding a root of (5) using the estimated objective parameters. We anticipate our method will ultimately be used to forecast the behavior of agents using the estimated objective model, e.g., in a model-predictive control scheme. For such settings, this metric gives a more tangible sense of algorithmic quality. In both plots, we evaluate the approaches for two different noisy observation models: in one, estimators observe the *full* state, and in another, estimators observe the position and heading but not the speed of each agent; i.e., they receive a *partial* state observation. In addition to the raw data, we highlight the median as well as the IQR of the estimation error over a rolling window of 60 data points.

Figure 2(a) shows that both methods recover the true cost parameters θ if observations are not corrupted by noise. However, the performance of the baseline degrades rapidly with increasing observation noise variance. This performance degradation is particularly pronounced if the baseline receives only partial state observations. Our estimator recovers the unknown cost parameters more accurately and with a smaller IQR. In contrast to the baseline, the performance of our method degrades gracefully when observations are corrupted

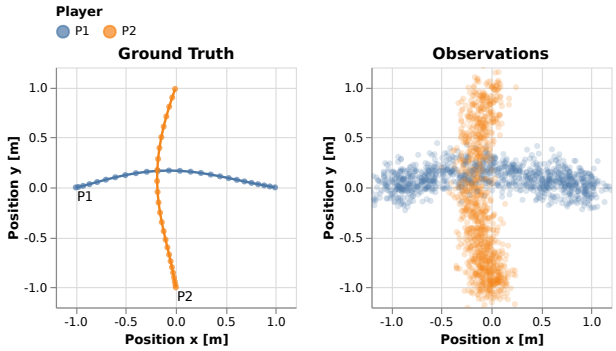


(a) Parameter estimation

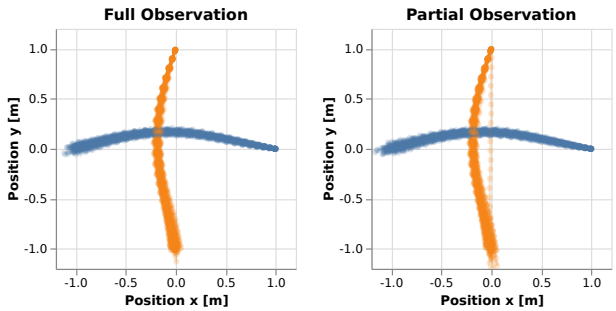


(b) Position prediction

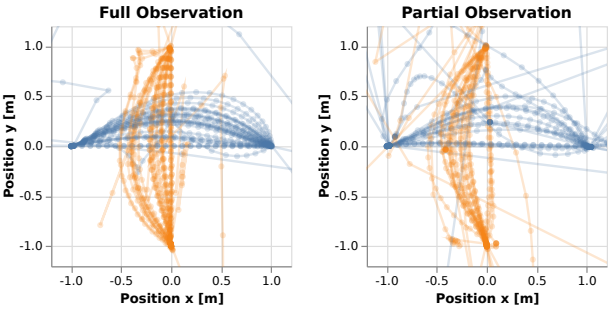
Fig. 2. Estimation performance of our method and the baseline for the 2-player running example, with noisy full and partial state observations. (a) Error measured directly in parameter space using (11). (b) Error measured in position space. Triangular data markers in (b) highlight objective estimates which lead to ill-conditioned games. Solid lines and ribbons indicate the median and IQR of the error for each case.



(a) Demonstrations



(b) Ours



(c) Baseline

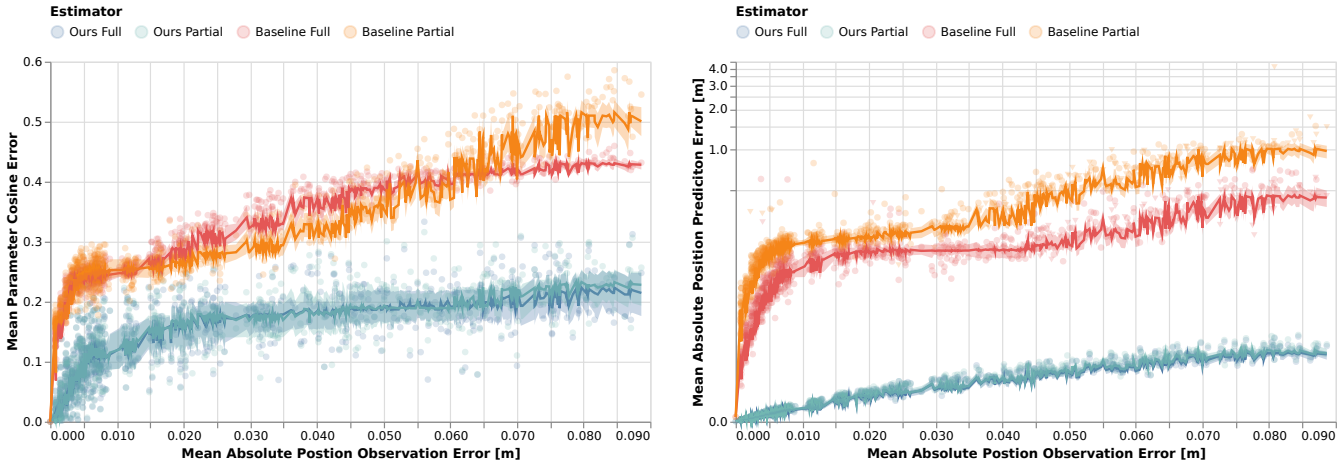
Fig. 3. Qualitative prediction performance for the 2-player running example at noise level $\sigma = 0.1$ for 40 different observation sequences. (a) Ground truth trajectory and observations, where each player wishes to reach a goal location opposite their initial position. (b, c) Trajectories recovered by solving the game at the estimated parameters for our method and the baseline using noisy full and partial state observations.

by noise. This finding also holds if the estimator receives only partial state observations.

Figure 2(b) displays qualitatively similar patterns, though here the vertical axis measures position prediction error rather than parameter error. Here, note that some data markers for the baseline estimator are triangles. These denote instances when the estimated parameters specify ill-conditioned objectives which prevent us from recovering roots of (5). For example, this can happen when proximity costs dominate control input costs. Thus, these data points correspond to a complete failure of the estimator. For the baseline, a total of 104 out of 880 estimates result in an ill-conditioned forward game when states are fully observed. In the case of partial observations, the number of estimator failures increases to 218. In contrast, our method recovers well-conditioned player objectives for all demonstrations and allows for accurate trajectory prediction.

For additional intuition of the performance gap, Figure 3 visualizes the prediction results in trajectory space for a fixed initial condition. Figure 3(a) shows the noise corrupted demonstrations generated for isotropic Gaussian noise with standard deviation $\sigma = 0.1$. Figure 3(b) and Figure 3(c) show the corresponding trajectories predicted by solving the game at the recovered objective estimates of our method and the baseline, respectively. Note that our method generates a far smaller fraction of outliers than the baseline. Further, the performance of our method is only marginally effected by partial state observability.

2) 5-Player Highway Overtaking: We also replicate the Monte Carlo study in a larger 5-player highway driving



(a) Parameter estimation

(b) Position prediction

Fig. 4. Estimation performance of our method and the baseline for the 5-player highway overtaking example, with noisy full and partial state observations. (a) Error measured directly in parameter space using (11). (b) Error measured in position space. Triangular data markers in (b) highlight objective estimates which lead to ill-conditioned games. Solid lines and ribbons indicate the median and IQR of the error for each case.

scenario in order to demonstrate scalability of the approach. This scenario is depicted in Figure 1. In this highway scenario, each player does not seek to reach a specific goal location at the end of the game horizon. Instead, a player's objective in this game is more accurately characterized by the desire to make forward progress at an unknown nominal speed. Therefore, ground-truth objectives use a quadratic penalty on deviation from a desired state that encodes each player's target lane and preferred travel speed. Note, that this objective can still be modeled by the cost structure in (3).

Figure 4 shows the estimator performance of our method and the baseline for this highway driving problem. Here, we use the same metrics as in the previous experiment to measure estimator performance in parameter space—Figure 4(a)—and position space—Figure 4(b). Again, our method clearly outperforms the baseline for both fully and partially observed demonstrations. Furthermore, the baseline performance is not consistent across the two metrics. That is, while the performance of the baseline measured in parameter space is not much effected by partial state observations, the observation model has a decisive impact on the trajectory prediction accuracy. This performance inconsistency of the baseline can be attributed the fact that certain objective parameters are more critical for accurate prediction of the game trajectory than others. Since our method's objective is data-fidelity, here measured by observation likelihood (4a), it directly accounts for these effects. The baseline, however, greedily optimizes the KKT residual irrespective of the downstream trajectory prediction task.

VII. Conclusion & Future Work

We have proposed a novel method for estimating player objectives from noise-corrupted partial state observations of non-cooperative multi-agent interactions—a task referred to as the *inverse* dynamic game problem. The proposed solution technique estimates the trajectory to recover unobserved states and inputs, and optimizes this trajectory *simultaneously* with an objective model estimate in order to maximize data-fidelity. The estimated trajectory is a forward game solution of the observed game including each players' strategy, and may be used for trajectory prediction.

Numerical simulations show that the resulting algorithm is more robust to observation noise and partial state observability than existing methods [4, 28], which require estimating states and inputs a priori. Our method recovers model parameters that closely match the unobserved true objectives and accurately predicts the state trajectory; even for high levels of observation noise.

Despite these encouraging results, there is ample room for future improvement. In the present work, we study the utility of our method for *offline* scenarios in which an external observer recovers the objectives of players *post hoc*. Our method, however, yields not only the estimated objective model, but also the forward game solution, including each players' strategy. This property makes our technique particularly suitable for *online* filtering applications in which an autonomous agent must estimate the objectives of other players for safe and efficient closed-loop interaction. In such a setting, the proposed estimator could be used on a fixed-lag buffer of past observations to simultaneously estimate each opponent's objective while generating the optimal response for the ego-agent over a receding prediction horizon.

Another exciting direction lies in the extension of the proposed method to information structures beyond OLNE. Recent work has put forth efficient solution techniques for the more expressive class of feedback Nash equilibria [11, 18]. While our proposed framework is generally agnostic to the information structure of the observed game, future work should investigate efficient techniques for encoding forward optimality for these equilibrium concepts.

References

[1] Sebastian Albrecht, Karinne Ramirez-Amaro, Federico Ruiz-Ugalde, David Weikersdorfer, Marion Leibold, Michael Ulbrich, and Michael Beetz. Imitating human reaching motions using physically inspired optimization principles. In *Proc. of the IEEE Intl. Conf. on Humanoid Robots*. IEEE, 2011.

[2] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[3] Chaitanya Awasthi. Forward and inverse methods in optimal control and dynamic game theory. Master's thesis, University of Minnesota, 2019.

[4] Chaitanya Awasthi and Andrew Lamperski. Inverse differential games with mixed inequality constraints. In *Proc. of the IEEE American Control Conference (ACC)*. IEEE, 2020.

[5] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*, volume 23. Society for Industrial and Applied Mathematics (SIAM), 1999.

[6] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review (SIREV)*, 59(1):65–98, 2017.

[7] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. Knitro: An integrated package for nonlinear optimization. *Large-Scale Nonlinear Optimization*, pages 35–59, 2006.

[8] Bolei Di and Andrew Lamperski. Newton's method and differential dynamic programming for unconstrained nonlinear dynamic games. In *Proceedings of the Conference on Decision Making and Control (CDC)*. IEEE, 2019.

[9] Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review (SIREV)*, 59(2):295–320, 2017.

[10] Peter Englert and Marc Toussaint. Inverse KKT: Learning cost functions of manipulation tasks from demonstrations. *Intl. Journal of Robotics Research (IJRR)*, pages 57–72, 2018.

[11] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D. Dragan, and Claire J. Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2020.

[12] David Fridovich-Keil, Vicenc Rubies-Royo, and Claire J. Tomlin. An iterative quadratic method for general-sum differential games with feedback linearizable dynamics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2020.

[13] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review (SIREV)*, 47:99–131, 2005.

[14] Jairo Inga, Esther Bischoff, Florian Köpf, and Sören Hohmann. Inverse dynamic games based on maximum entropy inverse reinforcement learning. *arXiv preprint arXiv:1911.07503*, 2019.

[15] Rudolf E. Kalman. When Is a Linear Control System Optimal? *ASME Journal of Basic Engineering*, 86(1): 51–60, 1964. doi: 10.1115/1.3653115.

[16] Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In *Proc. of the Intl. Symp. on Intelligent Control (ISIC)*. IEEE, 2011.

[17] Florian Köpf, Jairo Inga, Simon Rothfuß, Michael Flad, and Sören Hohmann. Inverse reinforcement learning for identification in linear-quadratic dynamic games. *IFAC-PapersOnLine*, 50(1):14902–14908, 2017.

[18] Forrest Laine, David Fridovich-Keil, Chih-Yuan Chiu, and Claire Tomlin. The computation of approximate generalized feedback nash equilibria. *arXiv preprint arXiv:2101.02900*, 2021.

[19] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. ALGAMES: A fast solver for constrained dynamic games. In *Proc. of Robotics: Science and Systems (RSS)*, 2020.

[20] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. LUCIDGames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning. *IEEE Robotics and Automation Letters (RA-L)*, 6(3): 5485–5492, 2021.

[21] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2012.

[22] Johan Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2004.

[23] Marcel Menner and Melanie N. Zeilinger. Maximum likelihood methods for inverse learning of optimal controllers. *arXiv preprint arXiv:2005.02767*, 2020.

[24] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion—an inverse optimal control approach. *Autonomous Robots*, 28(3):369–383, 2010.

[25] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2000.

[26] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Verlag, 2006.

[27] Lasse Peters. Accommodating intention uncertainty in general-sum games for human-robot interaction. Master's thesis, Hamburg University of Technology, 2020.

[28] Simon Rothfuß, Jairo Inga, Florian Köpf, Michael Flad, and Sören Hohmann. Inverse optimal control for identification in non-cooperative differential games. *IFAC-PapersOnLine*, 50(1):14909–14915, 2017.

[29] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[30] Zijian Wang, Riccardo Spica, and Mac Schwager. Game

theoretic motion planning for multi-robot racing. *Distributed Autonomous Robotic Systems*, pages 225–238, 2019.

[31] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proc. of the Conference on Advancements of Artificial Intelligence (AAAI)*, 2008.