Cost Inference in Smooth 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. We demonstrate our method in a simulated highway driving scenario. 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. These methods perform well in settings with a single robot operating in a well-characterized environment. However, such techniques formalize optimal decision-making for a *single* agent and are not directly suitable for interactive settings with multiple strategic agents. For example, consider multiple vehicles engaged in lane changes on a crowded highway as shown in Figure 1. In this setting, standard motion planning techniques may still be applied by first predicting the future trajectories of other agents and then planing one's own reactions. However, each agent's optimal behavior in the future depends upon the decisions of everyone else; sequential prediction and planning schemes fail to capture this coupling. Instead, this setting is more accurately characterized as a dynamic game. Despite the added complexity of modeling these interactions, recent developments enable computationally-efficient solutions to the noncooperative dynamic games which arise in multi-agent robotic settings [8, 10, 11, 14].

Like single-player optimal control techniques, dynamic games recover optimal behavior from given objectives. In contrast to their single-player counterparts, however, each player can have an *individual* objective, and objectives of different players may conflict. Hence, the solution of a dynamic game does not generally optimize a single utility function for all agents simultaneously. Instead, solutions are *equilibria* in which each player optimize their own utility while accounting for the possibly noncooperative behavior of other agents. In general, there can be multiple equilibria in a game, each of which corresponds to a potential mode of interaction. Recent work demonstrates the importance of planning one's own decisions to align with the equilibrium preferences of others [16].

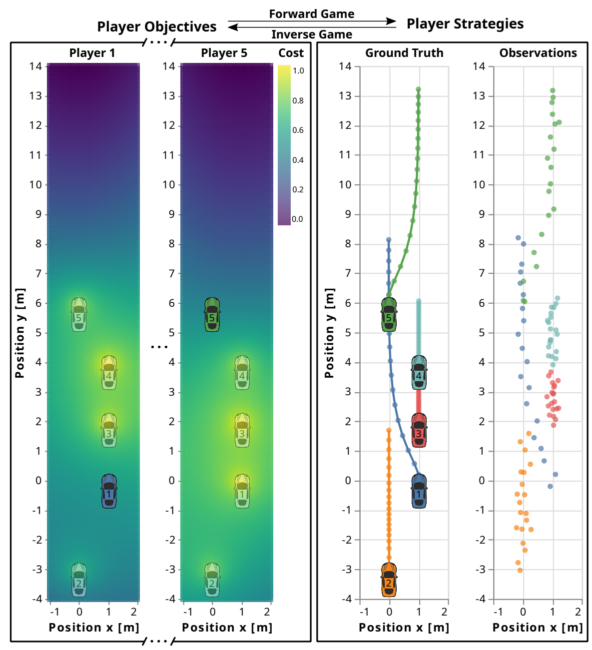


Fig. 1. Inverse and forward versions of a dynamic game modeling a 5-player highway driving scenario. The solution of the forward problem maps the player objectives (left) to the players' optimal strategies (right). Our method solves the inverse problem: it takes noisy, partial state observations of multi-agent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

While games are an expressive mathematical framework for modeling multi-agent interactions, they assume knowledge of each players objectives. For example, in Figure 1 the vehicle objectives define a dynamic game whose solution can be used to predict vehicle strategies for arbitrary initial configurations. That is, a (forward) game solver defines a mapping from player objectives to player strategies; left to right in Figure 1. However, if player objectives are unknown, the game is underspecified and unusable for prediction. In that case, it may be desirable to learn the player objectives from observed behavior; right to left in Figure 1.

In this paper and its full-length version [17], we study this *inverse* dynamic game problem of identifying objectives from noisy, partial state observations of multi-agent interactions. Recovering these unknown parameters allows us to infer important aspects of each player's preferences. For example, in the highway driving setting of Figure 1, one such preference is each agent's desire to avoid collisions with other vehicles. Our proposed inverse game solution estimates all player's states and control inputs jointly with their unknown objective parameters by coupling them through noncooperative equilibrium constraints. 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. 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 [4].

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 $\Gamma := \big(x_1, f, \{J^i\}_{i \in [N]}\big)$ 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*.

III. 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 which 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., $J^i(\cdot; \theta^i) \equiv \sum_{t=1}^{T} g_t^i(x_t, u_t^1, \dots, u_t^N; \theta^i)$.

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

$$\max_{\theta, \mathbf{x}, \mathbf{u}} \quad p(\mathbf{y} \mid \mathbf{x}, \mathbf{u}) \tag{2a}$$

$$\text{s.t.} \quad (\mathbf{x}, \mathbf{u}) \text{ is an OLNE of } \Gamma(\theta) \tag{2b}$$

$$(\mathbf{x}, \mathbf{u}) \text{ is dynamically feasible under } f, \tag{2c}$$

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

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 [3, 18] 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.

IV. 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 [15]) and solved by a number of off-the-shelf methods (e.g., IPOPT [19], KNITRO [6], and SNOPT [12]).

A. Encoding Nash Equilibrium Constraints

A key challenge to solving the estimation problem in (2) is posed by the requirement to encode the equilibrium constraint in (2b) 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 inverse optimal control (IOC) of Albrecht et al. [1], we encode this forward optimality constraint via the corresponding first-order necessary conditions. For an open-loop

[1] This setup readily extends to the constrained case as in [13, 14]—as our own work does. We ignore such constraints here for clarity.

Nash equilibrium (OLNE), the first-order necessary conditions are given by the union of the individual players' Karush–Kuhn–Tucker (KKT) conditions, i.e.,

$$\mathbf{G}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) := \begin{bmatrix} \left.\begin{matrix} \nabla_{\mathbf{x}} J^i + \boldsymbol{\lambda}^{i\top}\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x},\mathbf{u}) \\ \nabla_{\mathbf{u}^i} J^i + \boldsymbol{\lambda}^{i\top}\nabla_{\mathbf{u}^i}\mathbf{F}(\mathbf{x},\mathbf{u}) \end{matrix}\right\} \forall i \in [N] \\ \mathbf{F}(\mathbf{x},\mathbf{u}) \end{bmatrix} = \mathbf{0}. \tag{3}$$

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 (3) as constraints, we cast the inverse dynamic game problem of (2) as

$$\max_{\theta,\mathbf{x},\mathbf{u},\boldsymbol{\lambda}} \quad p(\mathbf{y} \mid \mathbf{x},\mathbf{u}) \tag{4a}$$

$$\text{s.t.} \quad \mathbf{G}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda};\theta) = \mathbf{0}. \tag{4b}$$

Here, the costates $\boldsymbol{\lambda}$ of (3) appear as *additional primal decision variables*. Further, $\mathbf{G}(\mathbf{x},\mathbf{u},\boldsymbol{\lambda};\theta)$ is the KKT residual from (3), with added explicit dependency on the cost parameters θ.

Note that (4b) 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 (3)

$$0 = \nabla_{x_t} J^i(\mathbf{x},\mathbf{u};\theta^i) + \boldsymbol{\lambda}^{i\top}\nabla_{x_t}\mathbf{F}(\mathbf{x},\mathbf{u}) \tag{5a}$$

$$= \nabla_{x_t} g_t^i(x_t,u_t;\theta^i) + \lambda_{t-1}^i - \lambda_t^{i\top}\nabla_{x_t} f_t(x_t,u_t), \tag{5b}$$

with aggregated player inputs $u_t = (u_t^1, \dots, u_t^N)$. In the inverse game, 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 (4b) 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 (4) 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 (4) without equilibrium constraints. That is, we compute the initialization of the state-input trajectory as the solution of

$$\tilde{\mathbf{x}},\tilde{\mathbf{u}} := \arg\max_{\mathbf{x},\mathbf{u}} \quad p(\mathbf{y} \mid \mathbf{x},\mathbf{u}) \tag{6a}$$

$$\text{s.t.} \quad \mathbf{F}(\mathbf{x},\mathbf{u}) = \mathbf{0}. \tag{6b}$$

This pre-solve step can be interpreted as sequentially activating the different components of the KKT constraints in (4b).

V. 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. [18]. Like our method, the KKT residual approach uses the first-order necessary conditions in (3) 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, i.e.,

$$\min_{\theta,\boldsymbol{\lambda}} \|\mathbf{G}(\tilde{\mathbf{x}},\tilde{\mathbf{u}},\boldsymbol{\lambda};\theta)\|_2^2, \tag{7}$$

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 (7) is not always well-defined. To this end, we extend the technique of [3, 18] with a pre-processing step that recovers a dynamically feasible state-input sequence by maximizing the likelihood of the observations via (6).

B. Implementation

We implement our proposed approach as well as the KKT residual baseline [18] in Julia [5] using the algebraic modeling language JuMP [9]. The source code is publicly available at https://github.com/PRBonn/PartiallyObservedInverseGames.jl.

C. Results

To compare robustness and performance of our method with the baseline, we study a simulated highway driving scenario as shown in Figure 1. More extensive results are presented in the full version of our paper [17]. In this scenario, each player wishes to make forward progress in a preferred lane at a specific travel speed, without colliding. At the same time, they wish to do so without turning or accelerating significantly. We model each vehicle's dynamics using a standard nonlinear 4D unicycle model whose state includes planar position, speed, and heading. To express players' costs, we use a weighted sum of basis functions that encode their aforementioned preferences following [10]. In this cost structure, the unknown parameters θ are the weights of each basis function. However, linear parameterization is not a strict requirement and our method applies more generally, e.g., to time-varying cost structures [7] and parametrizations via neural networks.

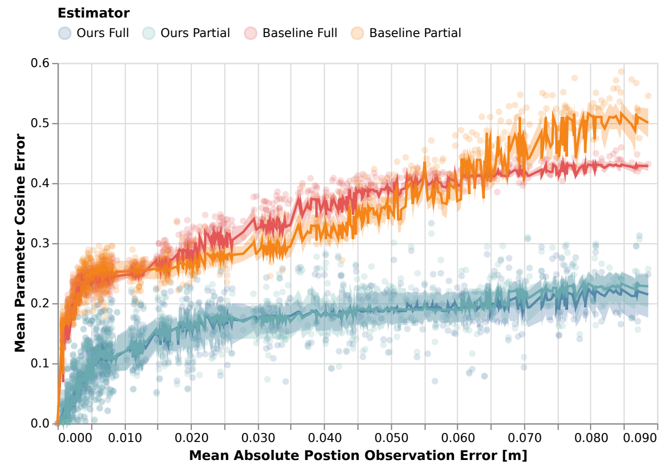
We generate ground-truth behavior by fixing parameters of a cost model for each player and finding the corresponding OLNE trajectory as the root of (3) using the well-known iterated best response (IBR) algorithm [20]. Observations are obtained by corrupting this solution with different levels of isotropic additive white Gaussian noise. For each simulated observation sequence, we run both our method and the baseline to recover estimates of the cost parameters θ. We replicate this Monte Carlo study for two different 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.

To measure estimation performance in parameters space, we use a cosine similarity measure:

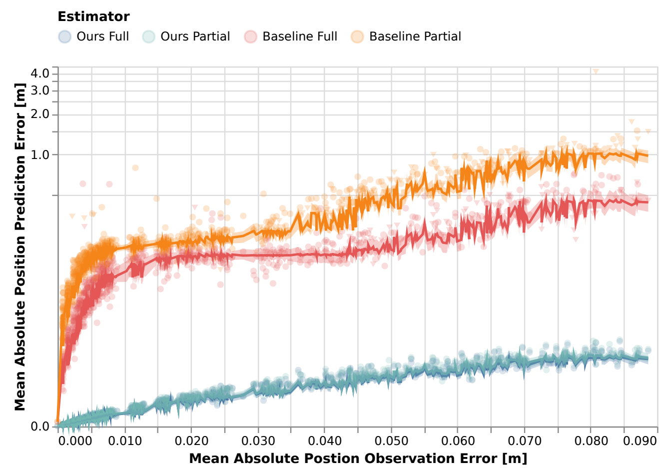
$$D_{\cos}(\theta_{\text{true}}, \theta_{\text{est}}) = 1 - \frac{1}{N} \sum_{i \in [N]} \frac{\theta^{i\top}_{\text{true}} \theta^{i}_{\text{est}}}{\left\|\theta^{i}_{\text{true}}\right\|_2 \left\|\theta^{i}_{\text{est}}\right\|_2}. \tag{8}$$

In order to give a more tangible sense of algorithmic quality, we also report raw position prediction errors, computed by finding a root of (3) using the estimated objective parameters.

Figure 2 shows the performance of our method and the baseline for this highway driving problem. We measure parameter estimation error using (8), with results in Figure 2(a). Figure 2(b) shows the corresponding position prediction errors. In both cases, our method outperforms the baseline. Furthermore, note that 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 trajectory than others. Since our method's objective is data-fidelity, here measured by observation likelihood (2a), it directly accounts for these effects. The baseline, however, greedily optimizes the KKT residual irrespective of the downstream trajectory prediction task.



(a) Parameter estimation



(b) Position prediction

Fig. 2. 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 (8). (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.

VI. 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 [3, 18], that 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. Here, the proposed estimator could be used on a 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. Finally, aspects of active learning and modeling of reputation effects on the dynamics of opponent behavior are exciting avenues for future work.

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

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

[5] 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.

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

[7] Chih-Yuan Chiu, David Fridovich-Keil, and Claire J. Tomlin. Encoding defensive driving as a dynamic Nash game. *arXiv preprint arXiv:2011.04815*, 2020.

[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] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] Lasse Peters, David Fridovich-Keil, Claire J. Tomlin, and Zachary N. Sunberg. Inference-based strategy alignment for general-sum differential games. *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2020.

[17] Lasse Peters, David Fridovich-Keil, Vicenç Rubies-Royo, Claire J. Tomlin, and Cyrill Stachniss. Inferring objectives in continuous dynamic games from noise-corrupted partial state observations. In *Proc. of Robotics: Science and Systems (RSS)*, 2021.

[18] 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.

[19] 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.

[20] Zijian Wang, Riccardo Spica, and Mac Schwager. Game theoretic motion planning for multi-robot racing. *Distributed Autonomous Robotic Systems*, pages 225–238, 2019.