THE COMPUTATION OF APPROXIMATE GENERALIZED FEEDBACK NASH EQUILIBRIA*

FORREST LAINE †, DAVID FRIDOVICH-KEIL‡, CHIH-YUAN CHIU†, AND CLAIRE TOMLIN†

Abstract. We present the concept of a Generalized Feedback Nash Equilibrium (GFNE) in dynamic games, extending the Feedback Nash Equilibrium concept to games in which players are subject to state and input constraints. We formalize necessary and sufficient conditions for (local) GFNE solutions at the trajectory level, which enable the development of efficient numerical methods for their computation. Specifically, we propose a Newton-style method for finding game trajectories which satisfy the necessary conditions, which can then be checked against the sufficiency conditions. We show that the evaluation of the necessary conditions in general requires computing a series of nested, implicitly-defined derivatives, which quickly becomes intractable. To this end, we introduce an approximation to the necessary conditions which is amenable to efficient evaluation, and in turn, computation of solutions. We term the solutions to the approximate necessary conditions Generalized Feedback Quasi Nash Equilibria (GFQNE), and we introduce numerical methods for their computation. In particular, we develop a Sequential Linear-Quadratic Game approach, in which a locally approximate LQ game is solved at each iteration. The development of this method relies on the ability to compute a GFNE to inequality- and equality-constrained LQ games, and therefore specific methods for the solution of these special cases are developed in detail. We demonstrate the effectiveness of the proposed solution approach on a dynamic game arising in an autonomous driving application.

Key words. Generalized Feedback Nash Equilibrium, Dynamic Games

1. Introduction.

There has been a recent growing interest in the application of game-theoretic concepts to applications in automated systems, as explored in [12, 19, 3, 2, 8, 9, 4, 7]. Indeed, numerous problems arising in these domains can be modeled as games, and particularly as dynamic or repeated games. For the purposes considered in this paper, we restrict our attention to discrete-time dynamic games, in which players have the ability to influence the state of the game over a finite set of game *stages*. Associated with dynamic/repeated game solutions are game trajectories, which capture the evolution of the game state and player inputs over the sequence of stages.

Solutions to dynamic games have been researched extensively, as in the well-known text [1]. Within the theory of dynamic games, there are a variety of information patterns that can be associated with the game, and each of these patterns results in a fundamentally different solution. Perhaps the simplest information pattern is the *open-loop* pattern, in which the repeated nature of the game is ignored. In this pattern the stages of the game are combined into a single *static* game, and the entire trajectory is chosen at once to satisfy a Nash, Stackelberg, or other type of Equilibrium. When constraints on the state variables are imposed upon the players, a Generalized Equilibrium must be considered. Formulating games with this type of information pattern has many advantages as the resultant static game often admits known methods for analysis and computing solutions, as presented, for example, in [5, 6]. However, by ignoring the dynamic nature of these games, the expressiveness

*This research is supported by an NSF CAREER award, the Air Force Office of Scientific Research (AFOSR), NSF's CPS FORCES and VeHICaL projects, the UC-Philippine-California Advanced Research Institute, the ONR MURI Embedded Humans, a DARPA Assured Autonomy grant, and the SRC CONIX Center.

†Electrical Engineering and Computer Sciences, U.C. Berkeley. (forrest.laine@berkeley.edu).

‡Aeronautics and Astronautics, Stanford University

of the resultant solutions are significantly limited. Intelligent game play in repeated games often involves observing the evolving game state and reacting accordingly.

Reactive game-play can emerge when associating a *closed-loop* information pattern to the dynamic game of interest. Effectively, games with this information pattern are such that the players choose control *policies* which define the control input as a function of the game state at that stage. When those policies at each game stage are chosen to constitute an equilibrium for the dynamic subgame played over the subsequent game stages, the resultant solution is called a feedback equilibrium. This type of solution is capable of capturing strategies for a player which anticipate and account for the reaction of other players. Some advantages of this type of solution are explored in the autonomous driving context in, e.g., [12].

While feedback equilibrium solutions are often desirable over their open-loop counterparts, for all but simple cases, there do not exist well-developed numerical routines for computing them. The unconstrained Linear Quadratic (LQ) setting is perhaps the simplest case, and methods for computing feedback equilibria for these games are well known, as presented in [1]. The extension to the computation of a feedback Nash equilibrium for a class of inequality-constrained LQ games is introduced in [15, 16], although restrictive assumptions are made on the form of the dynamics, constraints and cost terms of the game. Numerous other approaches have considered the computation of feedback equilibria under various special cases, such as those in [17, 18, 11], among others. Methods for computing feedback Nash equilibria have been recently developed in the unconstrained, nonlinear case using a value-iteration based approach [10], and an iterative LQ game approach [8]. Nevertheless, to the best of our knowledge, no methods exist for computing feedback equilibria in games with constraints appearing on both the state and input dimensions, both in the general LQ and nonlinear settings.

Since many emerging applications of dynamic games involve nonlinear dynamics as well as nonlinear constraints on the game states and inputs, we have pursued the development of a robust and efficient method for computing feedback equilibria in this setting. The result of that work is the topic of this paper.

The outline of the paper is the following. In section 2 we introduce the concept of a Generalized Feedback Nash Equilibrium (GFNE), which formally defines the feedback concept in the constrained setting. We discuss pitfalls with a parameterized approach to encoding GFNE problems as a means to motivate and introduce a non-parametric alternative. We then develop necessary and sufficient conditions on game trajectories to satisfy a GFNE using this non-parametric formulation. Challenges associated with the computation of such GFNE are highlighted, and a close approximation is introduced which is amenable to efficient computation. Finally, numerical methods for the computation of such approximate solutions are developed in detail, for the equality-constrained LQ setting (section 3), inequality-constrained LQ setting (section 4), and ultimately, the general nonlinear setting (section 5). We demonstrate our method on an application to autonomous driving in section 6 and conclude the paper in section 7.

2. Formulation. We focus our attention to the class of N-player discrete-time, deterministic, infinite, general-sum dynamic games of discrete stage-length T. Let $\mathbf{N}$ denote the set $\{1, ..., N\}$, and similarly $\mathbf{T}$ the set $\{1, ..., T\}$. We also make use of the sets $\mathbf{T}^+ := \mathbf{T} \cup \{T + 1\}$, $\mathbf{T_t} := \{t, ..., T\}$, and $\mathbf{T}^+_\mathbf{t} := \{t, ..., T + 1\}$. The game state at each discrete time-step t is represented by $x_t \in \mathcal{X} = \mathbb{R}^n$. The game is assumed to start at stage $t = 1$, from a pre-specified initial state $\hat{x}_1$. Throughout this paper we refer to subgames starting at stage t, which refers to the game played over a portion of the original game, on the stages $\{t, ..., T\}$.

The evolution of the game state is described by the dynamic equation:

$$x_{t+1} = f_t(x_t, u_t^1, ..., u_t^N), \quad t \in \mathbf{T}, \tag{2.1}$$

where $u_t^i \in \mathcal{U}_t^i = \mathbb{R}^{m_t^i}$ are the control variables chosen by players $i \in \mathbf{N}$ at time t. Let $m_t := \sum_{i=1}^N m_t^i$, and $m_t^{-i} := m_t - m_t^i$.

To simplify the notation in definitions and derivations, we make use of the following shorthand to refer to various sets of state and control variables:

Notation reference:

$$\begin{aligned}
x &:= (x_1, x_2, ..., x_{T+1}), \\
u^i &:= (u_1^i, u_2^i, ..., u_T^i), \\
u &:= (u^1, ..., u^N), \\
u_t &:= (u_t^1, ..., u_t^N), \\
u_t^{-i} &:= (u_t^1, ..., u_t^{i-1}, u_t^{i+1}, ..., u_t^N), \\
(u_t^i, u_t^{-i}) &:= (u_t^1, ..., u_t^N) = u_t
\end{aligned}$$

Each player in the game is associated with time-separable cost-functionals:

$$L^i(x, u^1, ..., u^N) := \sum_{t=1}^{T} l_t^i(x_t, u_t) + l_{T+1}^i(x_{T+1}) \tag{2.2}$$

Furthermore, each player is assigned stage-wise, non-dynamic, equality and inequality constraints.

$$0 = h_t^i(x_t, u_t), \ t \in \mathbf{T} \qquad 0 = h_{T+1}^i(x_{T+1}) \tag{2.3a}$$

$$0 \leq g_t^i(x_t, u_t) \ t \in \mathbf{T}, \qquad 0 \leq g_{T+1}^i(x_{T+1}) \tag{2.3b}$$

Let the dimension of the constraints h_t^i and g_t^i, for all $t \in \mathbf{T}^+$ and $i \in \mathbf{N}$, be denoted as $a_t^i \geq 0$ and $b_t^i \geq 0$, respectively. Define $V_t^i : \mathcal{X} \to (\mathbb{R} \cup \infty)$ as the *Value-function* for player $i \in \mathbf{N}$ at stage $t \in \mathbf{T}^+$, and $Z_t^i : \mathcal{X} \times \mathcal{U}_t^i \times ... \times \mathcal{U}_t^N \to (\mathbb{R} \cup \infty)$ the *Control-Value-function* for player $i \in \mathbf{N}$ at time $t \in \mathbf{T}$.

A Generalized Feedback Nash Equilibrium is defined in terms of measurable maps $\pi_t^i : \mathcal{X} \to \mathcal{U}_t^i$, for $i \in \mathbf{N}$, $t \in \mathbf{T}$, which we refer to as feedback policies or strategies. The feedback policies, Value-functions, and Control-Value-functions are together defined according to the following recursive relationships (2.4)-(2.7):

$$V_{T+1}^i(x_{T+1}) := \begin{cases} l_{T+1}^i(x_{T+1}), & \begin{array}{l} 0 = h_{T+1}^i(x_{T+1}) \\ 0 \leq g_{T+1}^i(x_{T+1}) \end{array} \\ \infty, & \text{else} \end{cases} \tag{2.4}$$

Given V_{t+1}^i for some $t \in \mathbf{T}$ and $i \in \mathbf{N}$, we define Z_t^i by

$$Z_t^i(x_t, u_t^1, ..., u_t^N) := \begin{cases} l_t^i(x_t, u_t) + V_{t+1}^i(f_t(x_t, u_t)), & \begin{array}{l} 0 = h_t^i(x_t, u_t) \\ 0 \leq g_t^i(x_t, u_t) \end{array} \\ \infty, & \text{else} \end{cases} \tag{2.5}$$

For a particular state x_t at stage t, the feedback policies π_t are defined to return a local Nash equilibrium solution for the static game defined in terms of the N Control-Value-functions evaluated at x_t (one for each player).

$$
\begin{aligned}
&\tilde{u}_t = \pi_t(x_t) \implies \\
&\qquad Z_t^1(x_t, \tilde{u}_t^1, ..., \tilde{u}_t^N) \leq Z_t^1(x_t, u_t^1, \tilde{u}_t^2, ..., \tilde{u}_t^N), \qquad \forall u_t^1 \in \mathcal{N}(\tilde{u}_t^1), \\
&\qquad \vdots \\
&\qquad Z_t^N(x_t, \tilde{u}_t^1, ..., \tilde{u}_t^N) \leq Z_t^N(x_t, \tilde{u}_t^1, ..., \tilde{u}_t^{N-1}, u_t^N), \quad \forall u_t^N \in \mathcal{N}(\tilde{u}_t^N),
\end{aligned} \tag{2.6}
$$

The set $\mathcal{N}(\tilde{u}_t^i)$ is some neighborhood around $\tilde{u}_t^i$. There may exist multiple, potentially non-isolated, local Nash equilibria. For the purposes considered here, we only require that for any state x_t, the policies evaluate to one arbitrarily chosen, yet particular, local equilibrium. A more stringent definition for the policies π_t could require that the inequalities in (2.6) hold over the entire sets $\mathcal{U}_t^i$. In any case, the Value-functions for stages $t \in \mathbf{T}$ are defined as

$$
V_t^i(x_t) := Z_t^i(x_t, \pi_t^1(x_t), ..., \pi_t^N(x_t)). \tag{2.7}
$$

Definition 2.1 (GFNE). *A Generalized Feedback Nash Equilibrium (GFNE) is defined by a set of policies π_t^i, $t \in \mathbf{T}$, $i \in \mathbf{N}$ defined in (2.6), such that the value of $V_1^i(\hat{x}_1)$, defined in (2.7), is finite for all $i \in \mathbf{N}$. Note that in this paper we refer to local GFNE whenever we write GFNE.*

Consider a collection of policies constituting a GFNE. Let the corresponding *equilibrium trajectory* be denoted by x_t^*, u_t^* such that

$$
\begin{aligned}
x_1^* &:= \hat{x}_1, \\
u_t^{i*} &:= \pi_t^i(x_t^*), && t \in \mathbf{T}, \\
x_{t+1}^* &:= f_t(x_t^*, u_t^{1*}, ..., u_t^{N*}), && t \in \mathbf{T}.
\end{aligned} \tag{2.8}
$$

2.1. Parametric Formulation. To encode a GFNE problem, one approach is to use a parametric representation of the policies π_t^i, and reinterpret the decisions made by players as the parameters of these policies. That is, we could restrict each player to choose from policies $\pi_t^i \equiv \pi_{t,\theta_t^i}^i$ at each time t, parameterized by a real vector θ_t^i of arbitrary finite dimension. For clarity we also define $\theta^i := (\theta_1^i, ..., \theta_T^i)$ and $\theta_t := (\theta_t^1, ..., \theta_t^N)$. A GFNE problem could then be expressed as the following set of coupled optimization problems in which each player i minimizes over policy parameters θ^i, and the trajectory x, u (including the controls of other players, u^{-i}):

$$\min_{\theta^i, x, u} \; L^i(x, u) \tag{2.9a}$$

$$s.t. \quad 0 = x_{t+1} - f_t(x_t, u_t), \quad t \in \mathbf{T} \tag{2.9b}$$

$$0 = u_t - \pi_{t,\theta_t}(x_t), \quad t \in \mathbf{T} \tag{2.9c}$$

$$0 = h_t^i(x_t, u_t), \quad t \in \mathbf{T} \tag{2.9d}$$

$$0 \leq g_t^i(x_t, u_t), \quad t \in \mathbf{T} \tag{2.9e}$$

$$0 = h_{T+1}^i(x_{T+1}) \tag{2.9f}$$

$$0 \leq g_{T+1}^i(x_{T+1}) \tag{2.9g}$$

Because the parameterized control policies are treated as constraints, and the controls u^{-i} are treated as decision variables, the reaction of other players to the decisions of player i are explicitly accounted for in the optimization (2.9). Solutions to this encoding of the game could be found by finding a Generalized Nash Equilibrium for the set of N optimization problems (one corresponding to each player), using a method such as those described in [5].

Although this formulation is theoretically equivalent to a Generalized Nash Equilibrium problem (as in open-loop games), there are several important issues associated with it. Specifically, even if the parameterization of policies is a simple stage-varying affine map, the dimension of decision variables for player i is increased by $m^i \times (n + 1) \times T$ relative to a non-parametric version. Furthermore, this over-parameterization leads to an ambiguity in the choice of policy—for any state x_t and control u_t, there exist an infinite number of linear maps which relate the two. In general, any collection of policy parameters which lead to the same equilibrium trajectory (i.e., equivalent *representations* of one another, [1, Definition 5.12]), are indistinguishable in (2.9). Therefore, without additional regularization, the optimization is under-specified, and this leads to ill-conditioning of the problem.

Perhaps the most important problem that arises with the formulation (2.9) is that despite relationships between the control variables u_t and x_t being accounted for via the policies π_{t,θ_t}, the gradient of this policy is not necessarily meaningful. Specifically, if the dynamics (2.9b) and policies (2.9c) are substituted into the cost functional (2.9a), and the gradient is taken with respect to the parameters θ_t^i, the chain rule relates the affect of the parameters on the cost through the policy gradients $\nabla_x \pi_{t,\theta_t}$. However, the over-representation of the policies π_{t,θ_t} implies that the gradient need not correspond to the true gradient of the optimal policy. To illustrate this, consider a simple example, describing the relationship between a scalar state x_t, and player 1's scalar control u_t^1. Let $\pi_{t,\theta_t}(x_t) = \theta_1 \cdot x_t + \theta_2$. If at the state $x_t = 2$, the optimal value for u_t^1 is 1, then one possible parameterization is $\theta_1 = 0.5$, and $\theta_2 = 0$. However, if at the state $x_t = 2 + \epsilon$, the optimal value for u_t^1 is $1 - \epsilon$, then this implies that the original estimate of $\nabla_x \pi_{t,\theta_t} = 0.5$ is unrelated to the observed gradient of the optimal policy, which is -1.

2.2. Non-parametric Formulation. As discussed above, while it is conceivable to express GFNE problems by use of parameterized policies, the resulting formulation leads to significant numerical and theoretical challenges. The remainder of this work is devoted to an "implicit," non-parametric encoding of the GFNE problem, which does not suffer the same problems associated with the parametric formulation. We begin the presentation of this approach by expressing the policies at each stage t in terms of the subgame starting at that stage.

THEOREM 2.2. *The policies defined in (2.6) can equivalently be expressed in terms of the nested Generalized Nash Equilibrium Problems (2.10).*

$$\pi_t^i(x_t) := \tilde{u}_t^i \in \arg\min_{u_t^i} \min_{\substack{u_{t:T}^i, \\ \tilde{u}_{t+1:T}^{-i}, \\ x_{t+1:T+1}}} \sum_{s=t}^{T} l_s^i(x_s, u_s^i, \tilde{u}_s^{-i}) + l_{T+1}^i(x_{T+1}) \tag{2.10a}$$

$$\text{s.t.} \quad 0 = \tilde{u}_s^{-i} - \pi_s^{-i}(x_s), \qquad s \in \mathbf{T_{t+1}} \tag{2.10b}$$

$$0 = x_{s+1} - f_s(x_s, u_s^i, \tilde{u}_s^{-i}), \qquad s \in \mathbf{T_t} \tag{2.10c}$$

$$0 = h_s^i(x_s, u_s^i, \tilde{u}_s^{-i}), \qquad s \in \mathbf{T_t} \tag{2.10d}$$

$$0 \le g_s^i(x_s, u_s^i, \tilde{u}_s^{-i}), \qquad s \in \mathbf{T_t} \tag{2.10e}$$

$$0 = h_{T+1}^i(x_{T+1}) \tag{2.10f}$$

$$0 \le g_{T+1}^i(x_{T+1}) \tag{2.10g}$$

Here $\pi_t^{-i}(x_t) := (\pi_t^1(x_t), ..., \pi_t^{i-1}(x_t), \pi_t^{i+1}(x_t), ..., \pi_t^N(x_t))$, and the notation $\arg\limits_{a} \min\limits_{a,b}$ is used to indicate that the minimum is taken over a and b, but only the value of a at the minimum is returned. Furthermore, the value of the minimization appearing in (2.10) is considered infinite for any combination of optimization variables violating the constraints. Note that the set $\mathbf{T}_{T+1}$ is empty, so for stage $t = T$, the constraint (2.10b) vanishes.

Proof. Starting with stage T, and substituting x_{T+1} using the dynamics (2.10c), and moving constraints (2.10d)–(2.10g) into the objective by means of infinite-valued indicator functions, observe that the objective of the minimization is equivalent to Z_T^i (2.5) as claimed.

Now, for some other stage $t \in \mathbf{T}$, assuming $\pi_t^i(x_{t+1})$ can be expressed by (2.10), it can be shown that $\pi_t^i(x_t)$ must also be equivalently expressed by (2.10). (2.10) can be re-written as the following:

(2.11a)

$$\pi_t^i(x_t) := \tilde{u}_t^i \in \arg_{u_t^i} \min_{\substack{u_t^i,\\ \tilde{u}_{t+1}^{-i},\\ x_{t+1}}} \left\{ l_t^i(x_t, u_t^i, \tilde{u}_t^{-i}) + \min_{\substack{u_{t+1:T}^i,\\ \tilde{u}_{t+2:T}^{-i},\\ x_{t+2:T+1}}} \sum_{s=t+1}^{T} l_s^i(x_s, u_s^i, \tilde{u}_s^{-i}) + l_{T+1}^i(x_{T+1}) \right\}$$

$$\text{s.t.} \quad 0 = \tilde{u}_s^{-i} - \pi_s^{-i}(x_s), \qquad s \in \mathbf{T_{t+2}} \tag{2.11b}$$

$$0 = x_{s+1} - f_s(x_s, u_s^i, \tilde{u}_s^{-i}), \quad s \in \mathbf{T_{t+1}} \tag{2.11c}$$

$$0 = h_s^i(x_s, u_s^i, \tilde{u}_s^{-i}), \qquad s \in \mathbf{T_{t+1}} \tag{2.11d}$$

$$0 \le g_s^i(x_s, u_s^i, \tilde{u}_s^{-i}), \qquad s \in \mathbf{T_{t+1}} \tag{2.11e}$$

$$0 = h_{T+1}^i(x_{T+1}) \tag{2.11f}$$

$$0 \le g_{T+1}^i(x_{T+1}) \tag{2.11g}$$

$$\text{s.t.} \quad 0 = \tilde{u}_{t+1}^{-i} - \pi_{t+1}^{-i}(x_{t+1}) \tag{2.11h}$$

$$0 = x_{t+1} - f(x_t, u_t^i, \tilde{u}_t^{-i}) \tag{2.11i}$$

$$0 = h_t^i(x_t, u_t^i, \tilde{u}_t^{-i}) \tag{2.11j}$$

$$0 \le g_t^i(x_t, u_t^i, \tilde{u}_t^{-i}) \tag{2.11k}$$

The nested minimum appearing in (2.11) is exactly that appearing in (2.10) for stage $t+1$ (ignoring (2.11b) if $t+1 = T$). Because the controls $\tilde{u}_{t+1}^{-i}$ (for $t+1 \le T$) are constrained by the policies $\pi_{t+1}^{-i}(x_{t+1})$, the value of this nested minimization must equal the value function $V_{t+1}(x_{t+1})$ as defined in (2.7) for any minimizer u_{t+1}^i, regardless of whether or not the minimizer corresponds to the particular one corresponding to $\pi_{t+1}^i(x_{t+1})$. By substituting x_{t+1} using the constraint (2.11i), and using infinite-valued indicator functions to move (2.11j) and (2.11k) into the objective of (2.11), we see that the objective of the minimization is equivalent to (2.5) for stage t. Thus, the alternate definition of $\pi_t^i(x_t)$ in (2.10) is equivalent to that in (2.6) for all stages t. ∎

Here we have defined the GFNE policies in terms of the *nested* equilibrium problems with equilibrium constraints (EPEC) (2.10). These equilibrium constraints arise in these problems because the constraints (2.10b) are defined in terms of equilibrium problems. Critically though, the set of players in all inner-level equilibrium problems are exactly those in the outer-level problems, allowing for the removal of the redundant constraint that $u_s^i = \pi_s^i(x_s),\ s \in \mathbf{T_{t+1}}$ from the player i's problem statement (2.10), as demonstrated in Theorem 2.2. When the necessary conditions of all players are concatenated, the constraints $u_s^i = \pi_s^i(x_s),\ s \in \mathbf{T_{t+1}}$ become redundant for *all* i, as we show in Theorem 2.3. This fact will allow for a compact representation of necessary conditions associated with solutions of a GFNE, and ultimately algorithms for finding such solutions.

THEOREM 2.3 (Necessary Conditions). *For some stage $t \in \mathbf{T}$, consider any set of policies $\pi_s^i,\ s \in \mathbf{T_t},\ i \in \mathbf{N}$, as defined in (2.10). Let the state $\hat{x}_t$ be such that a solution exists to equilibrium problem (2.10) at stage t. Denote the resultant sub-game solution trajectory by $\{x_s^*;\ s \in \mathbf{T_t^+},\ x_t^* = \hat{x}_t\}$, $\{u_s^*;\ s \in \mathbf{T_t}\}$. Furthermore, assume the policies $\pi_s(x_s)$ are differentiable at the point x_s^* for $s \in \mathbf{T_{t+1}}$, and a standard constraint qualification such as the linear independence constraint qualification holds for the optimization problem appearing in (2.10), for each $i \in \mathbf{N}$. Then there exist multipliers $\{\lambda_s^i \in \mathbb{R}^n;\ s \in \mathbf{T_t},\ i \in \mathbf{N}\}$, $\{\mu_s^i \in \mathbb{R}^{a_s^i};\ s \in \mathbf{T_t^+},\ i \in \mathbf{N}\}$, $\{\gamma_s^i \in \mathbb{R}^{b_s^i};\ s \in \mathbf{T_t^+},\ i \in \mathbf{N}\}$, and $\{\psi_s^i \in \mathbb{R}^{m^{-i}};\ s \in \mathbf{T_{t+1}},\ i \in \mathbf{N}\}$ which satisfy:*

(2.12a)
$$0 = \nabla_{u_s^i}\left[l_s^i + f_s^{\intercal}\lambda_s^i - h_s^{i\intercal}\mu_s^i - g_s^{i\intercal}\gamma_s^i\right]_{x_s^*,u_s^*}, \quad i \in \mathbf{N},\ s \in \mathbf{T_t}$$

(2.12b)
$$0 = \nabla_{x_s}\left[l_s^i - \lambda_{s-1}^i + f_s^{\intercal}\lambda_s^i - h_s^{i\intercal}\mu_s^i - g_s^{i\intercal}\gamma_s^i + \pi_s^{-i\intercal}\psi_s^i\right]_{x_s^*,u_s^*}, \quad i \in \mathbf{N},\ s \in \mathbf{T_{t+1}},$$

(2.12c)
$$0 = \nabla_{u_s^{-i}}\left[l_s^i + f_s^{\intercal}\lambda_s^i - h_s^{i\intercal}\mu_s^i - g_s^{i\intercal}\gamma_s^i - \psi_s^i\right]_{x_s^*,u_s^*}, \quad i \in \mathbf{N},\ s \in \mathbf{T_{t+1}},$$

(2.12d)
$$0 = \nabla_{x_{T+1}}\left[l_{T+1}^i - \lambda_T^i - h_{T+1}^{i\intercal}\mu_{T+1}^i - g_{T+1}^{i\intercal}\gamma_{T+1}^i\right]_{x_{T+1}^*}, \quad i \in \mathbf{N},$$

(2.12e)
$$0 = x_{s+1}^* - f_s(x_s^*, u_s^*), \quad s \in \mathbf{T_t},$$

(2.12f)
$$0 = h_s^i(x_s^*, u_s^*), \quad i \in \mathbf{N},\ s \in \mathbf{T_t},$$

(2.12g)
$$0 \le g_s^i(x_s^*, u_s^*) \perp \gamma_s^i \ge 0, \quad i \in \mathbf{N},\ s \in \mathbf{T_t},$$

(2.12h)
$$0 = h_{T+1}^i(x_{T+1}^*), \quad i \in \mathbf{N},$$

(2.12i)
$$0 \le g_{T+1}^i(x_{T+1}^*) \perp \gamma_{T+1}^i \ge 0, \quad i \in \mathbf{N}.$$

Furthermore, let $L_t(z_t, x_t^*) = 0$ denote the entire set of conditions (2.12) formed by treating active inequalities (2.12g) and (2.12i) as equalities, and ignoring the inactive inequalities. Here, z_t is the set of all variables appearing in (2.12) other than x_t^* and *all* *multipliers γ_s^i associated with inactive inequality constraints. If strict complementarity holds, and the matrix $\nabla_{z_t} L_t$ is non-singular, then the policy $\pi_t(x_t)$ is also differentiable at the point $\hat{x}_t$, and $\nabla_x \pi_t(x_t) := -[[\nabla_{z_t} L_t]^{-1} \nabla_{x_t} L_t]_{u_t}$.*

The notation $[\cdot]_{u_t}$ implies that if u_t appears in the j_1 through j_2 indices of z_t, then the j_1 through j_2 rows of the matrix argument are selected. The notation in the first equation of (2.12) is used to indicate that the gradients of the functions $l_s^i(x_s, u_s)$, $f(x_s, u_s)$, $h_s^i(x_s, u_s)$, and $g_s^i(x_s, u_s)$ are evaluated at x_s^* and u_s^*. The symbol $\perp$ is used to indicate complementarity of the left- and right-hand-side conditions. For example, if the k-th element of $g_s^i(x_s^*, u_s^*) > 0$ then the k-th element of γ_s^i must be 0, and vice-versa. As before, for the final stage $t = T$, the set of conditions (2.12b) and (2.12c) is empty, as the set $\mathbf{T_{T+1}} = \emptyset$.

Proof. By the assumption that the optimization problems (2.10) satisfy a standard constraint qualification, the Lagrange Multiplier theorem states that there must exist multipliers associated with each player $i \in \mathbf{N}$'s optimization problem at stage t, such that both the conditions in (2.12) (evaluated for the particular $i \in \mathbf{N}$) and the constraints $u_s^{-i*} = \pi_s^{-i}(x_s)$ hold. Concatenating these conditions for all of the N players gives rise to the conditions (2.12), with the addition of the constraints (2.10b) for each $i \in \mathbf{N}$. Since there must exist multipliers satisfying those conditions, the conditions (2.12) must also be satisfied, as (2.12) are formed by simply removing the constraints $u_s^* = \pi_s(x_s)$.

After removing all inactive inequality constraints and associated multipliers from (2.12), it is straightforward to verify that the dimension of the system $L_t(z_t, x_t^*)$ and z_t are the same, and therefore that $\nabla_{z_t} L_t$ is a square matrix. If z_t^* is comprised of $x_{t+1:T+1}^*$, $u_{t:T}^*$ and multipliers satisfying (2.12) such that $L_t(z_t^*, x_t^*) = 0$, then assuming this matrix is non-singular, the Implicit Function Theorem states that there must exist a unique function $\Pi_t(x_t)$ and open set $\mathcal{X}_t^* \subset \mathcal{X}$ containing x_t^* such that $L_t(\Pi_t(x_t), x_t)) = 0$ for all $x_t \in \mathcal{X}_t^*$. By the uniqueness of this function, we must have that for all $x_t \in \mathcal{X}_t^*$, $[\Pi_t(x_t)]_{u_t} = \pi_t(x_t)$, where $[\Pi_t(x_t)]_{u_t}$ selects the components of the function Π_t corresponding to the subset of z_t containing u_t. Therefore, for all $x_t \in \mathcal{X}_t^*$, $\nabla_{x_t} \pi_t(x_t) = \nabla_{x_t}([\Pi_t(x_t)]_{u_t}) = -[[\nabla_{z_t} L_t]^{-1} \nabla_{x_t} L_t]_{u_t}$. ∎

Observe that the conditions $L_t(z_t, x_t) = 0$ contain as a subset the conditions $L_s(z_s, x_s) = 0$, $s \in \mathbf{T_{t+1}}$. If the matrices $\nabla_{x_s} L_s(z_s, x_s)$, $s \in \mathbf{T_{t+1}}$ are also non-singular, then in some neighborhood of z_t^*, the constraints $u_s^* = \pi_s(x_s^*)$, $s \in \mathbf{T_{t+1}}$ are equivalent to $L_s(z_s^*, x_s^*) = 0$, motivating the removal of the constraints (2.10b) from the conditions (2.12).

For games and corresponding solutions satisfying the stated assumptions, Theorem 2.3 suggests a method for computing those solutions. Evaluating the conditions (2.12) only requires the evaluation of the policy gradients $\nabla_{x_s} \pi_s(x_s)$, and not the policies themselves. This is important for computational reasons, since while in general an explicit representation of the policies is unavailable, the evaluation of the policy gradients can be made.

The procedure we propose for computing GFNE trajectories is to find a solution to the conditions (2.12), which can then be checked against a sufficiency condition to ensure that the solution is indeed constitutes a GFNE.

THEOREM 2.4 (Sufficient Conditions). *Consider any set of states $\{x_s^*,\ s \in \mathbf{T_t^+}\}$ and controls $\{u_s^*,\ s \in \mathbf{T_t}\}$, which together with multipliers $\{\lambda_s^i \in \mathbb{R}^n;\ s \in \mathbf{T_t},\ i \in \mathbf{N}\}$, $\{\mu_s^i \in \mathbb{R}^{a_s^i};\ s \in \mathbf{T_t^+},\ i \in \mathbf{N}\}$, $\{\gamma_s^i \in \mathbb{R}^{b_s^i};\ s \in \mathbf{T_t^+},\ i \in \mathbf{N}\}$, and $\{\psi_s^i \in \mathbb{R}^{m^{-i}};\ s \in \mathbf{T_{t+1}},\ i \in \mathbf{N}\}$ satisfy the conditions (2.12), with the matrix $\nabla_{z_s} L_s$ non-singular for* *all $s \in \mathbf{T}$. If additionally, for all $i \in \mathbf{N}$,*

$$\sum_{s=t}^{T} \begin{bmatrix} d_{x,s} \\ d_{u,s} \end{bmatrix}^\intercal \nabla^2 l_s^i(x_s^*, u_s^*) \begin{bmatrix} d_{x,s} \\ d_{u,s} \end{bmatrix} + d_{x,T+1}^\intercal \nabla^2 l_{T+1}^i(x_{T+1}^*) d_{x,T+1} > 0, \tag{2.13a}$$

$$\forall \{d_{x,s}, d_{u,s}\} \text{ s.t. } 0 = d_{u_s} - \nabla_x \pi_s(x_s^*) d_{x,s}, \qquad s \in \mathbf{T_t}, \tag{2.13b}$$

$$0 = d_{x,s+1} - \nabla_x f_s(x_s^*, u_s^*) d_{x,s} - \nabla_u f(x_s^*, u_s^*) d_{u_s}, \quad s \in \mathbf{T_t}, \tag{2.13c}$$

then the trajectory x_s^*, u_s^* constitutes a GFNE trajectory.

Proof. The set of $d_{x,s}$, $d_{u,s}$ satisfying (2.13b) and (2.13c) is a super-set of the critical constraint cone [14, Eq. 12.53], therefore the trajectory x_s^*, u_s^* must constitute a true local minimum of the problems (2.10) for stage t and each player $i \in \mathbf{N}$, as stated in [14, Theorem 12.6]. ∎

The sufficiency condition outlined in Theorem 2.4 is stricter than necessary since it ignores other active constraints which reduce the volume of the critical constraint cone, and could be relaxed by considering the linearization of other active constraints.

Although Theorems 2.3 and 2.4 together outline a procedure for computing GFNE trajectories, there remain some difficulties which must be addressed if such a procedure is to be practical. It is important to note that while the conditions (2.12) do not require the evaluation of the policies $\pi_s(x_s), s \in \mathbf{T_{t+1}}$ explicitly, evaluating the policy gradients $\nabla_x \pi_s(x_s)$ is required, and their evaluation involves inverting a matrix which depends on second-order derivatives of all policies $\pi_r(x_r)$, $r \in \mathbf{T_{s+1}}$. Furthermore, evaluating second-order derivatives of any policy requires evaluating third-order derivatives of all subsequent policies, and so forth. The effect of this is that T^{th}-order derivatives of dynamic and constraint functions appearing at the late stages of the game must be computed to evaluate the conditions (2.12) when $t = 1$. While technically possible, this requirement is impractical for many games. We therefore introduce a reasonable approximation to the computation of policy gradients $\nabla_x \pi_t(x_t)$ which do not require the evaluation of any higher-order derivatives of policies $\pi_s(x_s)$, $s \in \mathbf{T_{t+1}}$.

Definition 2.5 (Policy Quasi-Gradients). *We approximate $\nabla_x \pi_t(x_t)$ by what is termed the policy quasi-gradient, K_t, defined implicitly by the following conditions:*

$$
\begin{aligned}
&0 = \nabla_{u_s^i}\Big[l_s^i + f_s^\intercal \lambda_s^i - h_s^{i\intercal} \mu_s^i - g_s^{i\intercal} \gamma_s^i\Big]_{x_s^*, u_s^*}, && i \in \mathbf{N},\ s \in \mathbf{T_t} \\
&0 = \nabla_{x_s}\Big[l_s^i - \lambda_{s-1}^i + f_s^\intercal \lambda_s^i - h_s^{i\intercal} \mu_s^i - g_s^{i\intercal} \gamma_s^i\Big]_{x_s^*, u_s^*} + K_s^{-i\intercal} \psi_s^i, && i \in \mathbf{N},\ s \in \mathbf{T_{t+1}}, \\
&0 = \nabla_{u_s^{-i}}\Big[l_s^i + f_s^\intercal \lambda_s^i - h_s^{i\intercal} \mu_s^i - g_s^{i\intercal} \gamma_s^i - \psi_s^i\Big]_{x_s^*, u_s^*}, && i \in \mathbf{N},\ s \in \mathbf{T_{t+1}}, \\
&0 = \nabla_{x_{T+1}}\Big[l_{T+1}^i - \lambda_T^i - h_{T+1}^{i\intercal} \mu_{T+1}^i - g_{T+1}^{i\intercal} \gamma_{T+1}^i\Big]_{x_{T+1}^*}, && i \in \mathbf{N}, \\
&0 = x_{s+1}^* - f_s(x_s^*, u_s^*), && s \in \mathbf{T_t}, \\
&0 = h_s^i(x_s^*, u_s^*), && i \in \mathbf{N},\ s \in \mathbf{T_t}, \\
&0 \le g_s^i(x_s^*, u_s^*) \perp \gamma_s^i \ge 0, && i \in \mathbf{N},\ s \in \mathbf{T_t}, \\
&0 = h_{T+1}^i(x_{T+1}^*), && i \in \mathbf{N}, \\
&0 \le g_{T+1}^i(x_{T+1}^*) \perp \gamma_{T+1}^i \ge 0, && i \in \mathbf{N}.
\end{aligned}
\tag{2.14}
$$

The conditions (2.14) are nearly the same as (2.12), although the actual policy gradients $\nabla_x \pi_s(x_s)$ have been replaced with the quasi-gradients K_s. Letting $\hat{L}_t(z_t, x_t^*)$ represent the concatenation of active conditions in (2.14) (analogous to $L_t(z_t, x_t^*)$ in Theorem 2.3), then

$$K_t := -[[\nabla_{z_t} \hat{L}_t]^{-1} \nabla_{x_t} \hat{L}_t]_{u_t}, \tag{2.15}$$

where $K_s,\ s \in \mathbf{T_{t+1}}$ are treated as constants ($\nabla_{z_t} K_s = 0, s \in \mathbf{T_{t+1}}$). If the matrix $[\nabla_{z_t} \hat{L}_t]$ is singular at some (z_t, x_t), we say that the policy quasi-gradient does not exist at that point.

The conditions (2.14) can be evaluated without the need for computing any third- or higher-order derivatives of any constraint or objective terms of the game, and also be evaluated efficiently, as we will show. Solutions satisfying the conditions (2.14) will not satisfy the conditions (2.12) in general. Since the solutions to (2.14) will be distinct from solutions to (2.12), we introduce the notion of a Generalized Feedback Quasi-Nash Equilibrium (GFQNE), to characterize these solutions. We believe GFQNE solutions closely approximate GFNE solutions.

Definition 2.6 (GFQNE). *Let $\{x_s^*;\ s \in \mathbf{T_t^+},\ x_t^* = \hat{x}_t\}$, $\{u_s^*;\ s \in \mathbf{T_t}\}$, $\{\lambda_s^i \in \mathbb{R}^n;\ s \in \mathbf{T_t},\ i \in \mathbf{N}\}$, $\{\mu_s^i \in \mathbb{R}^{a_s^i};\ s \in \mathbf{T_t^+},\ i \in \mathbf{N}\}$, $\{\gamma_s^i \in \mathbb{R}^{b_s^i};\ s \in \mathbf{T_t^+},\ i \in \mathbf{N}\}$, and $\{\psi_s^i \in \mathbb{R}^{m^{-i}};\ s \in \mathbf{T_{t+1}},\ i \in \mathbf{N}\}$ be such that the conditions (2.14) are satisfied. Furthermore, let*

$$\begin{aligned}
&\sum_{s=t}^{T} \begin{bmatrix} d_{x,s} \\ d_{u,s} \end{bmatrix}^{\intercal} \nabla^2 l_s^i(x_s^*, u_s^*) \begin{bmatrix} d_{x,s} \\ d_{u,s} \end{bmatrix} + d_{x,T+1}^{\intercal} \nabla^2 l_{T+1}^i(x_{T+1}^*) d_{x,T+1} > 0, \\
&\forall \{d_{x,s}, d_{u,s}\} \text{ s.t. } 0 = d_{u_s} - K_s d_{x,s}, \qquad\qquad\qquad\qquad s \in \mathbf{T_t}, \\
&\qquad\qquad\qquad\quad\ 0 = d_{x,s+1} - \nabla_x f_s(x_s^*, u_s^*) d_{x,s} - \nabla_u f(x_s^*, u_s^*) d_{u_s}, \quad s \in \mathbf{T_t},
\end{aligned} \tag{2.16}$$

then we say that the trajectory x_s^*, u_s^* constitutes a Generalized Feedback Quasi-Nash Equilibrium (GFQNE) trajectory. Note that the condition (2.16) differs from the condition in Theorem 2.4 in the definition of the critical cone.

If all cost functionals (2.2) in the game are quadratic, and all dynamic (2.1) and non-dynamic (2.3a)–(2.3b) constraints are linear, then the policy quasi-gradients are equivalent to the policy gradients, and all GFQNE are therefore GFNE.

In the general setting, the policy quasi-gradients do not exactly match the policy gradients, which potentially introduces a different type of computational difficulty. Using Newton-type methods to solve for solutions to (2.14), we will ideally be able to evaluate $[\nabla_{z_1} \hat{L}_1]^{-1}$ exactly, without treating $K_s,\ s \in \mathbf{T_2}$ as constants (since indeed, they depend on z_1). If we are unwilling or unable to compute derivatives of sub-game policy quasi-gradients, we will be forced to use a quasi-newton method at best. Because we are working in the game setting, and the matrix $\nabla_{z_t} \hat{L}_t$ will in general be asymmetric, it is difficult to provide guarantees that such a quasi-newton method will converge. Nevertheless, we find that in practice, such an approach does in fact converge and is useful in interesting settings.

So far we have also made an important, limiting assumption, which is that the matrices $\nabla_{z_t} L_t$ and $\nabla_{z_t} \hat{L}_t$ are non-singular for all $t \in \mathbf{T}$. For many common forms of constraints (2.3a), (2.3b), this assumption cannot hold. This will occur, for example, when there is a terminal constraint on the entire game state, such as $h_{T+1}^i(x_{T+1}) :=$

x_{T+1}. If $m_t^i < n$, then the matrices $\nabla_{z_t} \hat{L}_t$ necessarily must be singular. Since many games involve constraints of this form, we handle them in the following way.

If at any stage t, the matrix $\nabla_{z_t} \hat{L}_t$ is found to be singular, and the game is otherwise well-posed[1], then this is likely due to an over-constrained sub-game. In this situation, we can combine the stage t with the preceding stage $t-1$, and define new combined-stage dynamics and constraint functions accordingly. For example, assume at stage t the matrix $\nabla_{z_t} \hat{L}_t$ is singular. We then define $\hat{u}_{t-1} := [u_{t-1}^\intercal \ u_t^\intercal]^\intercal$, $\hat{\mathcal{U}}_{t-1}^i := \mathcal{U}_{t-1}^i \times \mathcal{U}_t^i$, and the updated dynamic, constraint, and stage-wise cost functionals as

$$\hat{f}_{t-1}(x_{t-1}, \hat{u}_{t-1}) := f_t(f_{t-1}(x_{t-1}, u_{t-1}), u_t), \tag{2.17a}$$

$$\hat{g}_{t-1}^i(x_{t-1}, \hat{u}_{t-1}) := [g_{t-1}^i(x_{t-1}, u_{t-1})^\intercal \ g_t^i(f(x_{t-1}, u_{t-1}), u_t)^\intercal]^\intercal, \tag{2.17b}$$

$$\hat{h}_{t-1}^i(x_{t-1}, \hat{u}_{t-1}) := [h_{t-1}^i(x_{t-1}, u_{t-1})^\intercal \ h_t^i(f(x_{t-1}, u_{t-1}), u_t)^\intercal]^\intercal, \tag{2.17c}$$

$$\hat{l}_{t-1}^i(x_{t-1}, \hat{u}_{t-1}) := l_{t-1}^i(x_{t-1}, u_{t-1}) + l_t^i(f(x_{t-1}, u_{t-1}), u_t). \tag{2.17d}$$

In this procedure, we effectively reduce the number of stages of the game by 1, but the dimension of all controls input to the game and the cost and constraints imposed upon each player are unchanged.

Throughout the remainder of this paper, we will assume that game stages are combined as necessary to ensure the subgame policy quasi-gradients are well-defined, and the game horizon T, dynamics, constraints, and cost-functionals all reflect any such modifications. In what follows we focus on the derivation of numerical methods for computing Generalized Feedback Quasi-Nash Equilibria. We begin our presentation by considering a special-case, which will serve as a building block for more general methods.

3. Equality-Constrained LQ Games. We consider the case in which the dynamics equation describing the game evolution (2.1) is linear in its arguments, the cost-functionals (2.2) for each player are quadratic functions of the state and control variables, and each player is subject only to linear equality constraints.

In particular, let

$$x_{t+1} = A_t x_t + B_t^1 u_t^1 + ... + B_t^N u_t^N + c_t, \quad t \in \mathbf{T}, \tag{3.1}$$

for (possibly time-varying) matrices $A_t \in \mathbb{R}^{n \times n}$, $B_t^i \in \mathbb{R}^{m_t^i \times n}$, and vectors $c_t \in \mathbb{R}^n$. Associated with the dynamic constraints are multipliers λ_t^i for each player $i \in \mathbf{N}$. Let

$$\begin{aligned} &B_t := \begin{bmatrix} B_t^1 & ... & B_t^N \end{bmatrix}, \quad \hat{B}_t := \begin{bmatrix} B_t^1 & & \\ & \ddots & \\ & & B_t^N \end{bmatrix}, \quad \lambda_t := \begin{bmatrix} \lambda_t^1 \\ \vdots \\ \lambda_t^N \end{bmatrix}, \\ &\tilde{B}_t := \begin{bmatrix} B_t^2 & \dots & B_t^N & & & \\ & & & \ddots & & \\ & & & & B_t^1 & \dots & B_t^{N-1} \end{bmatrix}. \end{aligned} \tag{3.2}$$

[1] For example, the quadratic cost functionals of every player have sufficient curvature in the tangent cone of the game.

In this setting, the cost functionals for each player can be expressed as:

$$
\begin{aligned}
l_t^i(x_t,u_t) &:= \frac{1}{2}\left(\begin{bmatrix} x_t \\ u_t \end{bmatrix}^\intercal \begin{bmatrix} Q_t^i & S_t^{i\intercal} \\ S_t^i & R_t^i \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + 2 \begin{bmatrix} x_t \\ u_t \end{bmatrix}^\intercal \begin{bmatrix} q_t^i \\ r_t^i \end{bmatrix} \right), \\
l_{T+1}^i(x_{T+1}) &:= \frac{1}{2}\left(x_{T+1}^\intercal Q_{T+1}^i x_{T+1} + 2 x_{T+1}^\intercal q_{T+1}^i \right),
\end{aligned} \tag{3.3}
$$

for (possibly stage-varying) matrices $Q_t^i \in \mathbb{R}^{n\times n}$, $S_t^i \in \mathbb{R}^{m_t\times n}$, $R_t^i \in \mathbb{R}^{m_t\times m}$, and vectors $q_t^i \in \mathbb{R}^n$, $r_t^i \in \mathbb{R}^{m_t}$. For notational purposes, let the terms R_t^i, S_t^i, and r_t^i be comprised of sub-matrices, $R_t^{i,j,k} \in \mathbb{R}^{m_t^j\times m_t^k}$, $S_t^{i,j} \in \mathbb{R}^{m_t^j\times n}$, and sub-vectors $r_t^{i,j} \in \mathbb{R}^{m_t^j}$, for $j,k \in \mathbf{N}$:

$$
R_t^i := \begin{bmatrix} R_t^{i,1,1} & \dots & R_t^{i,1,N} \\ \vdots & \ddots & \vdots \\ R_t^{i,N,1} & \dots & R_t^{i,N,N} \end{bmatrix}, \quad S_t^i := \begin{bmatrix} S_t^{i,1} \\ \vdots \\ S_t^{i,N} \end{bmatrix}, \quad r_t^i := \begin{bmatrix} r_t^{i,1} \\ \vdots \\ r_t^{i,N} \end{bmatrix} \tag{3.4}
$$

We additionally make use of the following matrix terms for brevity, which combine components from the cost functionals of all players:

$$
\begin{aligned}
&R_t := \begin{bmatrix} R_t^{1,1,1} & \dots & R_t^{1,1,N} \\ R_t^{2,2,1} & \dots & R_t^{2,2,N} \\ \vdots & \ddots & \vdots \\ R_t^{N,N,1} & \dots & R_t^{N,N,N} \end{bmatrix}, \quad S_{x_t} := \begin{bmatrix} S_t^{1,1} \\ S_t^{2,2} \\ \vdots \\ S_t^{N,N} \end{bmatrix}, \quad r_t := \begin{bmatrix} r_t^{1,1} \\ r_t^{2,2} \\ \vdots \\ r_t^{N,N} \end{bmatrix}, \\
&Q_t := \begin{bmatrix} Q_t^1 \\ \vdots \\ Q_t^N \end{bmatrix}, \quad S_{u_t} := \begin{bmatrix} S_t^{1\intercal} \\ \vdots \\ S_t^{N\intercal} \end{bmatrix}, \quad q_t := \begin{bmatrix} q_t^1 \\ \vdots \\ q_t^N \end{bmatrix}, \\
&\tilde{R}_t := \begin{bmatrix} \begin{matrix} R_t^{1,2,1} & \dots & R_t^{1,2,N} \\ \vdots & \ddots & \vdots \\ R_t^{1,N,1} & \dots & R_t^{1,N,N} \end{matrix} & & \\ & \ddots & \\ & & \begin{matrix} R_t^{N,1,1} & \dots & R_t^{N,1,N} \\ \vdots & \ddots & \vdots \\ R_t^{N,N-1,1} & \dots & R_t^{N,N-1,N} \end{matrix} \end{bmatrix}, \tilde{S}_{x_t} := \begin{bmatrix} S_t^{1,2} \\ \vdots \\ S_t^{1,N} \\ S_t^{2,1} \\ S_t^{2,3} \\ \vdots \\ S_t^{2,N} \\ \vdots \\ S_t^{N,N-1} \end{bmatrix}
\end{aligned} \tag{3.5}
$$

We impose the regularity assumptions

$$
R_t^{i,i,i} \succ 0, \quad R_t^i \succeq 0, \quad Q_t^i \succeq 0 \tag{3.6}
$$

to ensure that the objective of each player is strictly convex. These conditions are sufficient for any solution to the conditions (2.12) to constitute a GFNE, as stated in Theorem 2.4, but not necessary.

The constraints imposed upon each player take the form

$$
\begin{aligned}
0 &= H_{x_t}^i x_t + H_{u_t^1}^i u_t^1 + \dots + H_{u_t^N}^i u_t^N + h_t^i, \quad t \in \mathbf{T} \\
0 &= H_{x_{T+1}}^i x_{T+1} + h_{T+1}^i,
\end{aligned} \tag{3.7}
$$

for matrices $H^i_{x_t} \in \mathbb{R}^{a^i_t \times n}$, $H^i_{u^j_t} \in \mathbb{R}^{a^i_t \times m^j_t}$, and vectors $h^i_t \in \mathbb{R}^{a^i_t}$, where a^i_t is the dimension of the equality constraint imposed on player i at stage $t \in \mathbf{T}^+$. As in subsection 2.2, we associate multipliers $\mu^i_t \in \mathbb{R}^{a^i_t}$, $t \in \mathbf{T}^+$ with these constraints for each player $i \in \mathbf{N}$. Let

(3.8)

$$
\begin{aligned}
H_{u_t} &:= \begin{bmatrix} H^1_{u^1_t} & \dots & H^1_{u^N_t} \\ \vdots & \ddots & \vdots \\ H^N_{u^1_t} & \dots & H^N_{u^N_t} \end{bmatrix}, &
\hat{H}_{u_t} &:= \begin{bmatrix} H^1_{u^1_t} & & \\ & \ddots & \\ & & H^N_{u^N_t} \end{bmatrix}, \\
H_{x_t} &:= \begin{bmatrix} H^1_{x_t} \\ \vdots \\ H^N_{x_t} \end{bmatrix}, &
\hat{H}_{x_t} &:= \begin{bmatrix} H^1_{x_t} & & \\ & \ddots & \\ & & H^N_{x_t} \end{bmatrix}, \\
h_t^\intercal &:= \begin{bmatrix} (h^1_t)^\intercal & \dots & (h^N_t)^\intercal \end{bmatrix}, &
\mu_t^\intercal &:= \begin{bmatrix} (\mu^1_t)^\intercal & \dots & (\mu^N_t)^\intercal \end{bmatrix}, \\
\tilde{H}_{u_t} &:= \begin{bmatrix} H^1_{u^2_t} & \dots & H^1_{u^N_t} & & & \\ & & & \ddots & & \\ & & & & H^N_{u^1_t} & \dots & H^N_{u^{N-1}_t} \end{bmatrix}
\end{aligned}
$$

Due to the linearity of all dynamic and non-dynamic constraint functions appearing in the game, and the quadratic cost functionals, the solutions of the conditions (2.12) and (2.14) will be identical, as stated in subsection 2.2. Therefore we will use the terms K_t and $\nabla_{x_t}\pi_t(x_t)$ interchangeably in this section.

Using the above-defined dynamic, constraint and cost terms, we are able to proceed with development of numerical methods for computing GFNE solutions to this game. Instead of taking a dynamic programming perspective as is, for example, taken in the classic derivation of Feedback Nash for unconstrained LQ games in [1], we derive our method using what we refer to as a *dynamic matrix factorization.* The primary idea behind this derivation is simply that the computation used to evaluate K_{t+1} can be reused to compute K_t efficiently.

To start, note that the conditions (2.14) for stage $t = T$ can be expressed in terms of the following matrix system:

(3.9)

$$
\begin{bmatrix} R_T & -\hat{H}^\intercal_{u_T} & \hat{B}^\intercal_T & & & \\ H_{u_T} & & & & & \\ -B_T & & & & I_n & \\ & & & -I_{N*n} & Q_{T+1} & \hat{H}^\intercal_{x_{T+1}} \\ & & & & H_{x_{T+1}} & \end{bmatrix}
\begin{bmatrix} u_T \\ \mu_T \\ \lambda_T \\ x_{T+1} \\ \mu_{T+1} \end{bmatrix}
+ \begin{bmatrix} S_{x_t} \\ H_{x_T} \\ -A_T \\ 0 \\ 0 \end{bmatrix} x_T
+ \begin{bmatrix} r_t \\ h_T \\ -c_T \\ q_{T+1} \\ h_{T+1} \end{bmatrix} = 0
$$

In (3.9), the matrices $I_\square$ denote the $\square \times \square$-dimensional identity matrix. Letting the system in (3.9) be denoted in shorthand as

$$
M_T z_T + N_T x_T + n_T = 0, \tag{3.10}
$$

where $z_T := [u_T^\top \; \mu_T^\top \; \lambda_T^\top \; x_{T+1}^\top \; \mu_{T+1}^\top]^\top$, we have that

$$
\begin{aligned}
\pi(x_T) &:= K_T x_T + k_T, \\
\lambda_T &:= K_{\lambda_T} x_T + k_{\lambda_T}, \\
\mu_T &:= K_{\mu_T} x_T + k_{\mu_T}, \\
\mu_{T+1} &:= K_{\mu_{T+1}} x_T + k_{\mu_{T+1}}, \\
K_T &:= -[M_T^{-1} N_T]_{u_T}, & k_T &:= -[M_T^{-1} n_T]_{u_T}, \\
K_{\lambda_T} &:= -[M_T^{-1} N_T]_{\lambda_T}, & k_{\lambda_T} &:= -[M_T^{-1} n_T]_{\lambda_T}, \\
K_{\mu_T} &:= -[M_T^{-1} N_T]_{\mu_T}, & k_{\mu_T} &:= -[M_T^{-1} n_T]_{\mu_T}, \\
K_{\mu_{T+1}} &:= -[M_T^{-1} N_T]_{\mu_{T+1}}, & k_{\mu_{T+1}} &:= -[M_T^{-1} n_T]_{\mu_{T+1}},
\end{aligned}
\tag{3.11}
$$

For any stage t, we also make use of the matrix

$$
\Pi_t^\top := \begin{bmatrix}
K_t^{2\top} & \cdots & K_t^{N\top} & & & & & & & \\
& & & K_t^{1\top} & K_t^{3\top} & \cdots & K_t^{N\top} & & & \\
& & & & & & & \ddots & & \\
& & & & & & & & K_t^{1\top} & \cdots & K_t^{(N-1)\top}
\end{bmatrix}^\top
\tag{3.12}
$$

Notice that if we denote the conditions (2.14) for some stage $t+1$ as

$$
M_{t+1} z_{t+1} + N_{t+1} x_{t+1} + n_{t+1} = 0,
\tag{3.13}
$$

as we did for $t+1 = T$, then we have also that the conditions (2.14) for stage t can be denoted as

$$
M_t z_t + N_t x_t + n_t = 0,
\tag{3.14}
$$

where $z_t := [u_t^\top \; \mu_t^\top \; \lambda_t^\top \; \psi_{t+1}^\top \; x_{t+1}^\top \; z_{t+1}^\top]^\top$, and the matrices M_t, N_t, and n_t are defined as follows:

$$
\begin{aligned}
M_t &:= \begin{bmatrix} D_t^1 & D_t^2 \\ \begin{bmatrix} 0 & N_{t+1} \end{bmatrix} & M_{t+1} \end{bmatrix} \\
D_t^1 &:= \begin{bmatrix}
R_t & -\hat{H}_{u_t}^\top & \hat{B}_t^\top & & \\
H_{u_t} & & & & \\
-B_t & & & & I_n \\
& & -I_{N*n} & \Pi_t^\top & Q_{t+1} \\
& & & -I_{(N-1)*m_{t+1}} & \tilde{S}_{x_{t+1}}
\end{bmatrix}, \\
D_t^2 &:= \begin{bmatrix}
& & & & \\
& & & & \\
& & & & \\
S_{u_{t+1}} & -\hat{H}_{x_{t+1}}^\top & \hat{A}_{t+1}^\top & & \\
\tilde{R}_{t+1} & -\tilde{H}_{u_{t+1}}^\top & \tilde{B}_{t+1}^\top & &
\end{bmatrix}, \\
N_t^\top &:= \begin{bmatrix} S_{x_t}^\top & H_{x_t}^\top & -A_t^\top & \end{bmatrix}, \quad n_t^\top := \begin{bmatrix} r_t^\top & h_t^\top & -c_t^\top & 0 & 0 & n_{t+1}^\top \end{bmatrix}.
\end{aligned}
\tag{3.15}
$$

From this form, we have as before, that

$$
\begin{aligned}
\pi(x_t) &:= K_t x_t + k_t, \\
\lambda_t &:= K_{\lambda_t} x_t + k_{\lambda_t}, \\
\mu_t &:= K_{\mu_t} x_t + k_{\mu_t}, \\
\psi_{t+1} &:= K_{\psi_{t+1}} x_t + k_{\psi_{t+1}}, \\
K_t &:= -[M_t^{-1} N_t]_{u_t}, \quad k_t := -[M_t^{-1} n_t]_{u_t}, \\
K_{\lambda_t} &:= -[M_t^{-1} N_t]_{\lambda_t}, \quad k_{\lambda_t} := -[M_t^{-1} n_t]_{\lambda_t}, \\
K_{\psi_{t+1}} &:= -[M_t^{-1} N_t]_{\psi_{t+1}}, \quad k_{\psi_{t+1}} := -[M_t^{-1} n_t]_{\psi_{t+1}}, \\
K_{\mu_t} &:= -[M_t^{-1} N_t]_{\mu_t}, \quad k_{\mu_t} := -[M_t^{-1} n_t]_{\mu_t},
\end{aligned}
\tag{3.16}
$$

The advantage of expressing our system in the form (3.15) is that the computation performed to solve K_{t+1} and k_{t+1} can be reused to solve K_t and k_t. Using the block form of M_t defined in (3.15), and that M_{t+1} is non-singular, we have from [13]:

$$
\begin{aligned}
[M_t^{-1}[N_t \; n_t]]_{u_t,\mu_t,\lambda_t,\psi_t,x_{t+1}} &= \begin{bmatrix} P_t^1 & P_t^2 \end{bmatrix} \begin{bmatrix} N_t & n_t \end{bmatrix}, \\
&= \begin{bmatrix} P_t^1 N_t, & P_t^1 \begin{bmatrix} r_t \\ h_t \\ -c_t \\ \\ \end{bmatrix} + P_t^2 n_{t+1} \end{bmatrix}
\end{aligned}
\tag{3.17}
$$

where the matrices P_t^1 and P_t^2 are defined as

$$
\begin{aligned}
P_t^1 &:= \left(D_t^1 - D_t^2 M_{t+1}^{-1} [0 \; N_{t+1}]\right)^{-1} \\
P_t^2 &:= -P_t^1 D_t^2 M_{t+1}^{-1}
\end{aligned}
\tag{3.18}
$$

Substituting in the form of the matrices in (3.15), we have that

$$
\begin{aligned}
P_t^1 &:= \begin{bmatrix} R_t & -\hat{H}_{u_t}^\intercal & \hat{B}_t^\intercal & & \\ H_{u_t} & & & & \\ -B_t & & & & I_n \\ & & -I_{N*n} & \Pi_t^\intercal & \hat{P}_t^{1,a} \\ & & & -I_{(N-1)*m_{t+1}} & \hat{P}_t^{1,b} \end{bmatrix}^{-1}, \\
\hat{P}_t^{1,a} &:= Q_{t+1} - S_{u_{t+1}} K_{t+1} + \hat{H}_{x_{t+1}}^\intercal K_{\mu_{t+1}} - \hat{A}_{t+1}^\intercal K_{\lambda_{t+1}}, \\
\hat{P}_t^{1,b} &:= \tilde{S}_{x_{t+1}} - \tilde{R}_{t+1} K_{t+1} + \tilde{H}_{u_{t+1}}^\intercal K_{\mu_{t+1}} - \tilde{B}_{t+1}^\intercal K_{\lambda_{t+1}}, \\
P_t^2 n_{t+1} &:= -P_t^1 \begin{bmatrix} S_{u_{t+1}} k_{t+1} - \hat{H}_{x_{t+1}}^\intercal k_{\mu_{t+1}} + \hat{A}_{t+1}^\intercal k_{\mu_{t+1}} \\ \tilde{R}_{t+1} k_{t+1} - \tilde{H}_{u_{t+1}}^\intercal k_{\mu_{t+1}} + \tilde{B}_{t+1}^\intercal k_{\lambda_{t+1}} \end{bmatrix}.
\end{aligned}
\tag{3.19}
$$

From the above, it can be seen that the entire matrix inverse M_t^{-1} does not need to be computed for any stage t (other than the terminal stage T), and the factorization presented here allows computation of the entire GFNE trajectory and associated multipliers very efficiently. In particular, the overall computational complexity of solving this system is $O(T \cdot ((N+1) \cdot (n+m))^2 \cdot (n+1))$ time due to the dominating cost of at each stage solving the system of equations of the form $P_t^1 W_t$, where $(P_t^1)^{-1}$

is a square matrix of width no greater than $(N+1)\cdot(n+m)$, and W_t is some matrix with $n+1$ columns.

After computing the terms (3.16) for all stages t using the procedure above, the resultant GFNE trajectory and associated multipliers can be extracted:

$$\begin{aligned}
x_1^* &:= \hat{x}_1, && \\
u_s^* &:= K_s x_s^* + k_s, && s \in \mathbf{T}, \\
\mu_s &:= K_{\mu_s} x_s^* + k_{\mu_s}, && s \in \mathbf{T}, \\
\lambda_s &:= K_{\lambda_s} x_s^* + k_{\lambda_s}, && s \in \mathbf{T}, \\
\psi_s &:= K_{\psi_s} x_{s-1}^* + k_{\psi_s}, && s \in \mathbf{T_2}, \\
x_{s+1}^* &:= A_s x_s^* + B_s u_s^* + c_s, && s \in \mathbf{T}, \\
\mu_{T+1} &:= K_{\mu_{T+1}} x_T + k_{\mu_{T+1}}. &&
\end{aligned} \tag{3.20}$$

4. Inequality-Constrained LQ Games. We now extend the basic results presented in section 3 on the computation of GFNE for equality-constrained LQ games, to the computation of GFNE for inequality-constrained LQ games. The approach we take here is that of an active-set method, analogous to active-set methods for quadratic programming (see, e.g. [14]).

Consider a dynamic game among N players over T stages, with linear dynamics described by (3.1), and quadratic cost functionals (3.3). Assume that each player is also subject to linear equality constraints of the form (3.7), along with linear inequality constraints of the form

$$\begin{aligned}
0 &\leq G^i_{x_t} x_t + G^i_{u_t^1} u_t^1 + \ldots + G^i_{u_t^N} u_t^N + g_t^i, \quad t \in \mathbf{T} \\
0 &\leq G^i_{x_{T+1}} x_{T+1} + g^i_{T+1},
\end{aligned} \tag{4.1}$$

for matrices $G^i_{x_t} \in \mathbb{R}^{b_t^i \times n}$, $G^i_{u_t^j} \in \mathbb{R}^{b_t^i \times m_t^j}$, and vectors $g_t^i \in \mathbb{R}^{b_t^i}$, where b_t^i is the dimension of the inequality constraint imposed on player i at stage $t \in \mathbf{T}^+$. As in subsection 2.2, we associate multipliers $\gamma_t^i \in \mathbb{R}^{b_t^i}$, $t \in \mathbf{T}^+$ with these constraints for each player $i \in \mathbf{N}$. Assume that a solution to the system (2.12) exists for this game at stage $t = 1$, and that strict complementarity holds for the conditions (2.12) for the subgame starting at every $t \in \mathbf{T}$ along any solution.

The method we presented for computing a GFNE of this game is an adaptation of Algorithm 16.3 in [14] to the current setting. Under the strict complementarity assumption (which ensures differentiability of the policies π_t along the solution), we have that at any GFNE solution, some subset of the constraints (4.1) associated with strictly positive multipliers hold with equality at the solution. If the set of active constraints along some solution were known in advance, we could consider all active constraints as equality constraints, ignore all inactive constraints, and solve for the resultant equality-constrained game using the method presented in section 3. In general the set of active constraints along a solution is obviously unknown in advance. The active-set method we propose accounts for this by iteratively solving for the unique GFNE solution for different guesses of the active constraint set, and uses dual variable information to update the guess of the active set. In the remainder of this section we describe the proposed method. The presentation of this section is based off of section 16.5 in [14], with necessary modifications made to account for the multiplayer feedback setting considered here.

The method begins with a feasible initialization for the game (defined by the linear dynamics (3.1), equality constraints (3.7), and inequality constraints (4.1), and the quadratic cost functionals (3.3)). We denote the set of all primal variables associated with the game at the kth iteration of the method by $\mathbf{X}_k := [x_{(1:T+1),k}, u_{(1:T),k}]$. Also associated with the kth iteration of the algorithm is the working set $\mathcal{W}_k$ which denotes the set of constraints which are treated with equality at the kth iteration. Note that the working set $\mathcal{W}_k$ always contains all of the equality constraints (3.7). The working set $\mathcal{W}_1$ is taken to be a subset of the constraints active along the initialization $\mathbf{X}_1$.

Given an iterate $\mathbf{X}_k$ and working set $\mathcal{W}_k$, a step $\mathbf{P}_k := [p_{x_{1:T+1}}, p_{u_{1:T}}]$ is found which moves $\mathbf{X}_k$ to the GFNE associated with the working set of equality constraints in $\mathcal{W}_k$. Specifically, the problem to be solved at each iteration is the GFNE problem for the equality-constrained LQ game defined by the stage-wise cost functionals for each player

$$
\begin{aligned}
l^i_{t,k}(p_{x_t}, p_{u_t}) &:= \frac{1}{2}\left(\begin{bmatrix} p_{x_t} \\ p_{u_t} \end{bmatrix}^\intercal \begin{bmatrix} Q^i_t & S^{i\intercal}_t \\ S^i_t & R^i_t \end{bmatrix} \begin{bmatrix} p_{x_t} \\ p_{u_t} \end{bmatrix} + 2 \begin{bmatrix} p_{x_t} \\ p_{u_t} \end{bmatrix}^\intercal \begin{bmatrix} q^i_{t,k} \\ r^i_{t,k} \end{bmatrix} \right), \quad t \in \mathbf{T} \\
q^i_{t,k} &:= Q^i_t x_{t,k} + S^{i\intercal}_t u_{t,k} + q^i_t, \quad t \in \mathbf{T}, \\
r^i_{t,k} &:= S^i_t x_{t,k} + R^i_t u_{t,k} + r^i_t, \quad t \in \mathbf{T}, \\
l^i_{T+1,k}(p_{x_{T+1}}) &:= \frac{1}{2}\left(p^\intercal_{x_{T+1}} Q^i_{T+1} p_{x_{T+1}} + 2 p^\intercal_{x_{T+1}} q^i_{T+1,k} \right), \\
q^i_{T+1,k} &:= Q^i_{T+1} x_{T+1,k} + q^i_{T+1},
\end{aligned}
\tag{4.2}
$$

the dynamics

$$
p_{x_{t+1}} = A_t p_{x_t} + B^1_t p_{u^1_t} + \ldots + B^N_t p_{u^N_t}, \quad t \in \mathbf{T}, \tag{4.3}
$$

and the linear equality constraints

$$
\begin{aligned}
0 &= H^i_{(x_t,k)} p_{x_t} + H^i_{(u^1_t,k)} p_{u^1_t} + \ldots + H^i_{(u^N_t,k)} p_{u^N_t}, \quad t \in \mathbf{T} \\
0 &= H^i_{(x_{T+1},k)} p_{x_{T+1}}.
\end{aligned}
\tag{4.4}
$$

Above, the matrices $H^i_{(x_t,k)}$ and $H^i_{(u^j_t,k)}$ are defined to be the set of active equality constraint coefficients corresponding to $\mathcal{W}_k$:

$$
H^i_{(x_t,k)} := \left[\begin{array}{c} H^i_{x_t} \\ \hline \vdots \\ \{G^{i,j}_{x_t}\}_{(t,i,j)\in\mathcal{W}_k\cap\mathcal{I}} \\ \vdots \end{array} \right], H^i_{(u_t,k)} := \left[\begin{array}{c} H^i_{u_t} \\ \hline \vdots \\ \{G^{i,j}_{u_t}\}_{(t,i,j)\in\mathcal{W}_k\cap\mathcal{I}} \\ \vdots \end{array} \right]. \tag{4.5}
$$

The set $\mathcal{W}_k \cap \mathcal{I}$ is the index set of all active inequality constraints, and $G^{i,j}_{x_t}$ is the jth row of the matrix $G^i_{x_t}$.

Associated with the constraints (4.4) are multipliers $\mu^i_{(t,k)}$, defined as

$$
\mu^i_{(t,k)} := \left[\begin{array}{c} \mu^i_t \\ \hline \vdots \\ \{\gamma^{i,j}_t\}_{(t,i,j)\in\mathcal{W}_k\cap\mathcal{I}} \\ \vdots \end{array} \right] \tag{4.6}
$$

where $\gamma_t^{i,j}$ is the jth element of γ_t^i.

After solving for $\mathbf{P}_k$, the GFNE of the resultant equality-constrained LQ game, $\mathbf{X}_k + \mathbf{P}_k$ is the GFNE for the equality-constrained LQ game defined by (3.1), (3.3), (3.7), and the active constraints (4.1) in $\mathcal{W}_k \cap \mathcal{I}$. However, it may be that $\mathbf{X}_k + \mathbf{P}_k$ is infeasible with respect to the entire set of inequality constraints (4.1). Therefore we instead find the point $\mathbf{X}_k + \beta_k \mathbf{P}_k$, where

$$\begin{aligned} \beta_k &:= \max_{\beta \in [0,1]} \beta \\ \text{s.t. } & \mathbf{X}_k + \beta \mathbf{P}_k \text{ feasible w.r.t. (4.1)}. \end{aligned} \tag{4.7}$$

The optimization in (4.7) is a linear program, and can be computed exactly and efficiently. When $\beta_k < 1$, it implies that there is an inequality constraint not considered in the working set which must be accounted for. When this is the case, the iterate $\mathcal{X}_{k+1}$ is updated to the point $\mathbf{X}_k + \beta_k \mathbf{P}_k$, and the working set is updated to include the blocking constraint. If instead, $\beta_k = 1$, then the point $\mathbf{X}_k + \mathbf{P}_k$ both is a GFNE solution for the working set and is feasible with respect to all equality and inequality constraints. All that is left to check is whether the constraints $\mu_{(t,k)}^{i,j} > 0$ for all $j > a_t^i$, meaning the complementarity conditions in (2.12) are satisfied, and therefore a solution satisfying the entire set of conditions (2.12) for the inequality-constrained problem has been found. If some multiplier associated with the inequality constraints in the working set $\mathcal{W}_k$ is negative[2], then the corresponding constraint is to be dropped from the working set at the next iteration. Unlike in the convex quadratic programming setting, forming $\mathcal{W}_{k+1}$ by dropping a constraint associated with a negative multiplier in the set $\mathcal{W}_k \cap \mathcal{I}$ and setting $\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{P}_k$, the update $\mathbf{P}_{k+1}$ does not necessarily move away from the dropped constraint boundary. In such situations, the procedure fails to make progress, since in such cases the value of β_{k+1} in (4.7) will be 0. In practice such situations can be overcome by dropping a different constraint (also associated with a negative multiplier) from the working set. If no other constraints are associated with negative multipliers, a GFNE does not exist in the vicinity of the iterate, and failure is declared. The full procedure is stated in Algorithm 4.1.

5. Nonlinear Games. We now present a method for computing GFQNE solutions to general games. In particular, we outline a procedure for finding a solutions to the conditions (2.14) for games defined by the dynamics (2.1), cost functionals (2.2), and constraints (2.3a) and (2.3b). We assume that all functions appearing in the conditions (2.14) are continuously twice differentiable, with the exception of the implicitly defined policies. The procedure leverages the method for computing solutions to inequality-constrained LQ games presented in section 4 and Algorithm 4.1, and generally is inspired by Sequential Quadratic Programming methods for non-convex numerical optimization (see e.g. [14], Chapter 18).

The foundation of this approach is in the observation that a Newton-style method can be used to finding a solutions to the conditions (2.14), where each iteration involves solving for a GFNE for the locally approximate LQ game formed around the current iterate. Considering first the case in which the game does not include any inequality constraints, computing a search direction using Newton's method on the conditions (2.14) at $t = 1$ can be seen to be equivalent to computing a search direction by solving an LQ approximation of the game. In the inequality-constrained case,

[2]Note that the multipliers cannot be zero by the strict complementarity assumption made.

Algorithm 4.1 Active Set Inequality Constrained LQ Game GFNE Solver

Start with $\mathbf{X}_1$ feasible w.r.t. (3.1), (3.7), (4.1)
Initialize $\mathcal{W}_1$ to be subset of active inequality constraints along $\mathbf{X}_1$
for k=1,2,3,... **do**
  Solve equality-constrained LQ GFNE defined by (4.2), (4.3), (4.4), and denote the solution as $\mathbf{P}_k$
  if $\mathbf{P}_k == \mathbf{0}$ **then**
    Extract multipliers $\mu^i_{(t,k)}$, ψ^i_t, λ^i_t, using (3.20)
    if $\mu^{i,j}_{(t,k)} > 0,\ \forall j > a^i_t$ {Inequality constraint multipliers} **then**
      return $\mathbf{X}_k$,
    else
      $(t_m, i_m, j_m) \leftarrow \underset{(t,i,j)\in\mathcal{W}_k\cap\mathcal{I}}{\operatorname{argmin}} \mu^{i,j}_{(t,k)}$
      $\mathbf{X}_{k+1} \leftarrow \mathbf{X}_k$
      $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \setminus \{(t_m, i_m, j_m)\}$
    end if
  else
    Find largest $\beta_k \in [0,1]$ such that $\mathbf{X}_k + \beta_k \mathbf{P}_k$ is feasible w.r.t. (4.1)
    $\mathbf{X}_{k+1} \leftarrow \mathbf{X}_k + \beta_k \mathbf{P}_k$
    if $\beta_k < 1$ **then**
      $(t_b, i_b, j_b) \leftarrow$ index of blocking inequality constraint not already in $\mathcal{W}_k \cap \mathcal{I}$
      if $(t_b, i_b, j_b) == (t_m, i_m, j_m)$ {Blocking index is previously dropped index} **then**
        Choose other blocking constraint or declare failure
      else
        $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k \cup (t_b, i_b, j_b)$
      end if
    else
      $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k$
    end if
  end if
end for

search directions can be found by solving an inequality-constrained LQ approximation, analogous to the method in [14].

In particular, we propose an iterative method for finding solutions to the conditions (2.14). Note that throughout this section, iterations are again indexed by $k = 1, 2, 3, ...$, as the were in the method for computing inequality-constrained LQ games in section 4. Here, let the current iterate of the primal and dual game variables at iteration k be denoted as

$$\begin{aligned} \mathbf{X}_k &:= [x_{(1:T+1),k}, u_{(1:T),k}], \\ \mathbf{\Lambda}_k &:= [\lambda_{(1:T),k}, \mu_{(1:T+1),k}, \gamma_{(1:T+1),k}, \psi_{(2:T),k}] \end{aligned} \tag{5.1}$$

Assume some initialization of all variables $\mathbf{X}_1$ and $\mathbf{\Lambda}_1$. Then for each iteration k, a search direction $\mathbf{P}_k$ is found by solving the inequality-constrained LQ game formed in the following way.

Let the dynamics (3.1) for the approximate game at the kth iteration be defined by the terms

$$
\begin{aligned}
A_{t,k} &:= \nabla_x f_t(x_{t,k}, u_{t,k}), \\
B^i_{t,k} &:= \nabla_{u^i} f_t(x_{t,k}, u_{t,k}), \\
c_{t,k} &:= f_t(x_{t,k}, u_{t,k}) - x_{t+1,k}.
\end{aligned}
\tag{5.2}
$$

Similarly, let the equality and inequality constraint terms in (3.7) and (4.1) be defined as

$$
\begin{aligned}
H^i_{x_t,k} &:= \nabla_x h^i_t(x_{t,k}, u_{t,k}), & G^i_{x_t,k} &:= \nabla_x g^i_t(x_{t,k}, u_{t,k}), \\
H^i_{u^i_t,k} &:= \nabla_{u^i} h^i_t(x_{t,k}, u_{t,k}), & G^i_{u^i_t,k} &:= \nabla_{u^i} g^i_t(x_{t,k}, u_{t,k}), \\
h^i_{t,k} &:= h^i_t(x_{t,k}, u_{t,k}), & g^i_{t,k} &:= g^i_t(x_{t,k}, u_{t,k}).
\end{aligned}
\tag{5.3}
$$

Finally, let the cost functional coefficients in (3.3) for stages $t \in \mathbf{T}$ be defined as

$$
\begin{aligned}
Q^i_{t,k} &:= \nabla^2_{x,x} l^i_t(x_{t,k}, u_{t,k}) + (\nabla^2_{x,x} f_t)^\intercal \lambda^i_{t,k} - (\nabla^2_{x,x} h^i_t)^\intercal \mu^i_{t,k} - (\nabla^2_{x,x} g^i_t)^\intercal \gamma^i_{t,k}, \\
S^i_{t,k} &:= \nabla^2_{u,x} l^i_t(x_{t,k}, u_{t,k}) + (\nabla^2_{u,x} f_t)^\intercal \lambda^i_{t,k} - (\nabla^2_{u,x} h^i_t)^\intercal \mu^i_{t,k} - (\nabla^2_{u,x} g^i_t)^\intercal \gamma^i_{t,k}, \\
R^i_{t,k} &:= \nabla^2_{u,u} l^i_t(x_{t,k}, u_{t,k}) + (\nabla^2_{u,u} f_t)^\intercal \lambda^i_{t,k} - (\nabla^2_{u,u} h^i_t)^\intercal \mu^i_{t,k} - (\nabla^2_{u,u} g^i_t)^\intercal \gamma^i_{t,k}, \\
q^i_{t,k} &:= \nabla_x l^i_t(x_{t,k}, u_{t,k}), \\
r^i_{t,k} &:= \nabla_u l^i_t(x_{t,k}, u_{t,k}).
\end{aligned}
\tag{5.4}
$$

The solution to this inequality-constrained LQ game at each iteration k yields the search direction $\mathbf{P}_k$ and multipliers $\bar{\mathbf{\Lambda}}_{k+1}$. To ensure progress towards a solution of the conditions (2.14), a line-search procedure is evoked. We seek a parameter $\alpha_k \in [0, 1]$ such that the iterates

$$
\begin{aligned}
\mathbf{X}_{k+1} &:= \mathbf{X}_k + \alpha_k \mathbf{P}_k, \\
\mathbf{\Lambda}_{k+1} &:= \mathbf{\Lambda}_k + \alpha_k (\bar{\mathbf{\Lambda}}_{k+1} - \mathbf{\Lambda}_k)
\end{aligned}
\tag{5.5}
$$

make maximal progress towards satisfying the conditions (2.14), with respect to an appropriate merit function. Because the game setting involves multiple players, a decrease in the objective of all players' objectives can not be guaranteed. Therefore the merit function we consider is simply the residual squared norm of the conditions

(2.14). In particular, define the merit function to search over as the following:

(5.6)

$$
\begin{aligned}
\mathbf{M}(\mathbf{X},\boldsymbol{\Lambda}) := \sum_{i\in\mathbf{N}}\sum_{s\in\mathbf{T}} \quad & \left\| \nabla_{u_s^i}\left[l_s^i + f_s^\intercal \lambda_s^i - h_s^{i^\intercal}\mu_s^i - g_s^{i^\intercal}\gamma_s^i \right]\right\|_2^2 + \\
\sum_{i\in\mathbf{N}}\sum_{s\in\mathbf{T_2}} \quad & \left\| \nabla_{x_s}\left[l_s^i - \lambda_{s-1}^i + f_s^\intercal \lambda_s^i - h_s^{i^\intercal}\mu_s^i - g_s^{i\intercal}\gamma_s^i + K_s^{-i\intercal}\psi_s^i \right]\right\|_2^2 + \\
\sum_{i\in\mathbf{N}}\sum_{s\in\mathbf{T_2}} \quad & \left\| \nabla_{u_s^{-i}}\left[l_s^i + f_s^\intercal \lambda_s^i - h_s^{i^\intercal}\mu_s^i - g_s^{i\intercal}\gamma_s^i - \psi_s^i \right]\right\|_2^2 + \\
\sum_{i\in\mathbf{N}} \quad & \left\| \nabla_{x_{T+1}}\left[l_{T+1}^i - \lambda_T^i - h_{T+1}^{i\intercal}\mu_{T+1}^i - g_{T+1}^{i\intercal}\gamma_{T+1}^i \right]\right\|_2^2 + \\
\sum_{s\in\mathbf{T}} \quad & \|x_{s+1} - f_s(x_s,u_s)\|_2^2 + \\
\sum_{i\in\mathbf{N}}\Bigg(\sum_{s\in\mathbf{T}} \quad & \left\|h_s^i(x_s,u_s)\right\|_2^2 + \left\|h_{T+1}^i(x_{T+1})\right\|_2^2\Bigg) + \\
\sum_{i\in\mathbf{N}}\Bigg(\sum_{s\in\mathbf{T}} \quad & \left\|\min(g_s^i(x_s,u_s),0)\right\|_2^2 + \left\|\min(g_{T+1}^i(x_{T+1}),0)\right\|_2^2\Bigg) + \\
\sum_{i\in\mathbf{N}}\sum_{s\in\mathbf{T}^+} \quad & \left\|\min(\mu_s^i,0)\right\|_2^2 + |(g_s^i)^\intercal\mu_s^i|.
\end{aligned}
$$

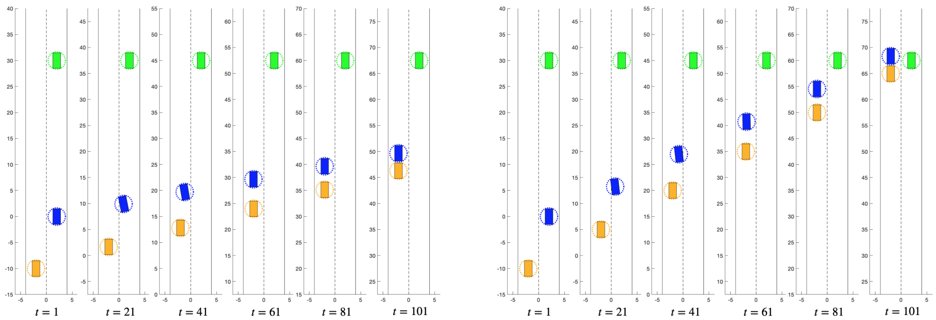
Recall that $\mathbf{T}_2 := \{2,...,T\}$. Then α_k is defined as:

$$
\alpha_k := \min_{\alpha\in[0,1]} \mathbf{M}(\mathbf{X}_k + \alpha\mathbf{P}_k, \boldsymbol{\Lambda}_k + \alpha(\bar{\boldsymbol{\Lambda}}_{k+1} - \boldsymbol{\Lambda}_k)). \tag{5.7}
$$

Algorithm 5.1 GFQNE Solver for Nonlinear Games

Set convergence tolerance $\epsilon > 0$
Start with initial $\mathbf{X}_1$, $\boldsymbol{\Lambda}_1$
for k=1,2,3,... **do**
  Solve inequality-constrained LQ GFNE defined by (5.2), (5.3), (5.4), and denote the solution and corresponding multipliers as $\mathbf{P}_k$, $\bar{\boldsymbol{\Lambda}}_{k+1}$
  Find α_k according to (5.7) and (5.6) via backtracking line-search
  $\mathbf{X}_{k+1} \leftarrow \mathbf{X}_k + \alpha_k\mathbf{P}_k$
  $\boldsymbol{\Lambda}_{k+1} \leftarrow \boldsymbol{\Lambda}_k + \alpha_k(\bar{\boldsymbol{\Lambda}}_{k+1} - \boldsymbol{\Lambda}_k)$
  if $\mathbf{M}(\mathbf{X}_{k+1}, \boldsymbol{\Lambda}_{k+1}) < \epsilon$ **then**
    Return $\mathbf{X}_{k+1}, \boldsymbol{\Lambda}_{k+1}$ and break
  end if
end for

The choice of merit function need not be as defined in (5.6). Any positive-valued function which evaluates to 0 if and only if the arguments constitute a solution to (2.14) is acceptable. Note that in the line-search procedure (5.7) using any such merit function, it is necessary in general to evaluate the policy quasi-gradients K_t at the candidate point $\mathbf{X}_k + \alpha\mathbf{P}_k$, $\boldsymbol{\Lambda}_k + \alpha(\bar{\boldsymbol{\Lambda}}_{k+1} - \boldsymbol{\Lambda}_k))$ for general α. However, recall that these terms are implicitly defined, and in general require solving the approximate LQ game defined at the candidate point to evaluate them. This makes the evaluation



(a) $\sigma_{polite} = 0$ (b) $\sigma_{polite} = 5$

Fig. 1: Snapshots of the resultant GFQNE solutions found to the example in section 6, when excluding (a) and including (b) the politeness term in player 1's objective.

of (5.7) very expensive. Therefore, in practice, we find it acceptable in most cases to replace the policy quasi-gradients K_t appearing in (5.6) corresponding to the candidate point, with the quasi-gradients corresponding to the point $\mathbf{X}_k, \mathbf{\Lambda}_k$, which are evaluated in the computation of the search direction $\mathbf{P}_k$.

Even with the reuse of the policy quasi-gradients, the minimization in (5.7) cannot be carried out exactly. In practice, a backtracking line-search satisfying a sufficient decrease condition is used instead.

The complete algorithm for computing solutions to (2.14) for nonlinear games is given in Algorithm 5.1.

6. Example. We now demonstrate the methodologies so far presented on a practical example.

Consider a game describing an driving scenario between an autonomous vehicle and two other vehicles on a freeway. Here $N = 3$, and let $T = 100$ denote the number of discrete time-points in the trajectory game. Let the game dynamics be defined as the concatenation of the independent dynamics of each vehicle in the game. We assume that each vehicle follows simple unicycle dynamics. Specifically, for each $i \in \mathbf{N}$, let

$$x^i_{t+1} = \begin{bmatrix} x^{i,1}_{t+1} \\ x^{i,2}_{t+1} \\ x^{i,3}_{t+1} \\ x^{i,4}_{t+1} \end{bmatrix} = f^i_t(x^i_t, u^i_t) = \begin{bmatrix} x^{i,1}_t + \Delta \cdot x^{i,3}_t \cos(x^{i,4}_t) \\ x^{i,2}_t + \Delta \cdot x^{i,3}_t \sin(x^{i,4}_t) \\ x^{i,3}_t + \Delta \cdot u^{i,1}_t \\ x^{i,4}_t + \Delta \cdot u^{i,2}_t \end{bmatrix}. \tag{6.1}$$

Here Δ represents some small time constant. The interpretation of the vehicles states and controls are the following: the first dimension is the longitudinal (along-lane) position, the second is the lateral (across-lane) position. The third dimension is the velocity in the direction of the vehicle heading, and the fourth dimension is the vehicle heading in radians. The first control dimension is the vehicle acceleration, and the second dimension is the angular velocity.

Major Iteration	Minor Iteration	Working Set Indices	Comment	α_k	$\mathbf{M}(\mathbf{X}_{k+1}, \mathbf{\Lambda}_{k+1})$
1	F1	{}			
	F2	{(101,2)}			
	F3	{(101,1),(101,2)}			
	1	{(101,1),(101,2)}			
	2	{(101,2)}			
	3	{(101,1),(101,2)}	Cycle	1	96.25
2	F1	{(101,2)}			
	F2	{(101,1),(101,2)}			
	1	{(101,1),(101,2)}			
	2	{(101,2)}			
	3	{(101,1),(101,2)}	Cycle	1	55.65
3	F1	{(101,2)}			
	1	{(101,2)}	Solution	1	0.221
4	F1	{(101,2)}			
	1	{(101,2)}	Solution	1	2.3e-5
Total Time	Total LQ Solves	Solve Time	Function Eval Time		
5.14	17	0.97	4.17		

Table 1: Algorithm iterate information when using Algorithm 5.1 (Major Iterations) and Algorithm 4.1 (Minor Iterations) to compute a GFQNE to the example in section 6, when $\sigma_{polite} = 5$ (Figure 1b). Here $\mathbf{M}(\mathbf{X}_{k+1}, \mathbf{\Lambda}_{k+1})$ is the merit function value after performing a line search in the direction of $\mathbf{P}_k$ in Algorithm 5.1. In each major iteration of this solve, $\alpha_k = 1$, meaning no backtracking was necessary in the line search procedure. Here the minor iterations labeled "F□" indicate equality-constrained LQ solves used in the search of a feasible initial solution to the inequality-constrained LQ game associated with the major iteration. The detection of a cycle indicates that the removal of a constraint associated with a negative multiplier did not move the iterate associated with Algorithm 4.1 away from the dropped constraint boundary (Line 20). In these cases the iterate is accepted the algorithm continues with the next major iteration.

The dynamics for the entire game state are then given as

$$x_{t+1} := \begin{bmatrix} f_t^1(x_t^1, u_t^1) \\ f_t^2(x_t^2, u_t^2) \\ f_t^3(x_t^3, u_t^3) \end{bmatrix} = f_t(x_t, u_t),\ t \in \mathbf{T}. \tag{6.2}$$

The goal of player 2 is to minimize its acceleration, angular velocity, and deviation from desired speed, while staying in-lane and avoiding collision with player 1. Similarly, the goal of player 3 is to minimize its acceleration, angular velocity, and deviation from desired speed, while staying in-lane.

The goal of player 1 is to complete a lateral lane change while minimizing its own acceleration, angular velocity, and deviation from its desired speed, and avoiding collision with player 3. Player 1 also attempts to minimize the objective of player 2 in addition to its own objective.

A depiction of this game is given in Figure 1. The functions describing the objective and constraints of each player are the following:

$$
\begin{aligned}
0 &= h^1_{T+1}(x_{T+1}) := x^{1,2}_{T+1} + 2,\\
0 &= h^2_{T+1}(x_{T+1}) := x^{2,2}_{T+1} + 2,\\
0 &= h^3_{T+1}(x_{T+1}) := x^{3,2}_{T+1} - 2,\\
0 &\le g^1_t(x_t) := \left\| \begin{matrix} x^{1,1}_t - x^{3,1}_t \\ x^{1,2}_t - x^{3,2}_t \end{matrix} \right\|_2 - d_{min}, \quad t \in \mathbf{T}^+,\\
0 &\le g^2_t(x_t) := \left\| \begin{matrix} x^{2,1}_t - x^{1,1}_t \\ x^{2,2}_t - x^{1,2}_t \end{matrix} \right\|_2 - d_{min}, \quad t \in \mathbf{T}^+,
\end{aligned}
\tag{6.3}
$$

$$
\begin{aligned}
L^1(x,u) &= \Big(\sum_{t=1}^{T} \sigma_1 \left\| u^1_t \right\|_2^2 + \sigma_2 (x^{1,2}_t + 2)^2 + \sigma_3 (x^{1,3}_t - v^1_{goal})^2 \Big) + \sigma_{polite} L^2(x,u),\\
L^2(x,u) &= \sum_{t=1}^{T} \sigma_1 \left\| u^2_t \right\|_2^2 + \sigma_2 (x^{2,2}_t + 2)^2 + \sigma_3 (x^{2,3}_t - v^2_{goal})^2,\\
L^3(x,u) &= \sum_{t=1}^{T} \sigma_1 \left\| u^3_t \right\|_2^2 + \sigma_2 (x^{3,2}_t - 2)^2 + \sigma_3 (x^{3,3}_t - v^3_{goal})^2,
\end{aligned}
\tag{6.4}
$$

The initial state $\hat{x}_1$ is defined to be $\hat{x}_1 := [0, 2, 1, 0, -10, -2, 1.5, 0, 30, 2, 0.75, 0]^\intercal$. The lateral center of the left lane is -2, and the lateral center of the right lane is 2. The constant d_{min} is the minimum separation distance from vehicle centers needed to avoid collision, which in this example is 3.3. The parameters $\sigma_{\{1,2,3\}}$ scale the relative weights between objective terms, and are $\sigma_1 = 10$, $\sigma_2 = 0.2$, and $\sigma_3 = 10$. The desired speeds of the three players are $v^1_{goal} = 1$, $v^2_{goal} = 1.5$, and $v^3_{goal} = 0.75$. The term σ_{polite} is the politeness coefficient and weights how much player 1 cares about interfering with player 2's objective. We consider two variants, with $\sigma_{polite} = 0$ and $\sigma_{polite} = 5$. This example is similar to games explored in [12]. Visualizations of the GFQNE solutions for the two variants are given in Figure 1, and computation details for the $\sigma_{polite} = 5$ case are given in Table 1 (details for the case $\sigma_{polite} = 0$ are omitted for brevity).

7. Conclusion. In this paper, we have presented a non-parametric, implicit policy formulation for generalized feedback Nash equilibrium problems. We developed efficient solution methods for the equality-constrained Linear Quadratic (LQ) case, the inequality-constrained LQ case, and the general nonlinear case. To the best of our knowledge, these constitute the first solution methods for both LQ and nonlinear games with general constraints. Dynamic games have numerous applications; we demonstrate the utility of our method in a trajectory planning setting for a lane-changing autonomous vehicle.

Future work should consider other solution methods for the general game which build upon our solution to the equality-constrained LQ game. In particular, penalty methods and interior point methods may also be competitive. Furthermore, the results presented should be extended to cases in which strict complementarity does not hold.

Necessary conditions based on sub-differentials could be used in such cases. It is also important to further develop a deeper theoretical understanding of the "policy quasi-gradient" approximation and its implications on convergence to local solutions. Finally, a high-performance, optimized implementation of our method will facilitate its use in practical applications by other researchers.

REFERENCES

[1] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*, SIAM, 1998.

[2] S. L. Cleac'h, M. Schwager, and Z. Manchester, *Algames: A fast solver for constrained dynamic games*, arXiv preprint arXiv:1910.09713, (2019).

[3] B. Di and A. Lamperski, *Newton's method and differential dynamic programming for unconstrained nonlinear dynamic games*, in 2019 IEEE 58th Conference on Decision and Control (CDC), IEEE, 2019, pp. 4073–4078.

[4] B. Di and A. Lamperski, *Local first-order algorithms for constrained nonlinear dynamic games*, in 2020 American Control Conference (ACC), IEEE, 2020, pp. 5358–5363.

[5] F. Facchinei, A. Fischer, and V. Piccialli, *Generalized nash equilibrium problems and newton methods*, Mathematical Programming, 117 (2009), pp. 163–194.

[6] F. Facchinei and C. Kanzow, *Generalized nash equilibrium problems*, Annals of Operations Research, 175 (2010), pp. 177–211.

[7] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, *Hierarchical game-theoretic planning for autonomous vehicles*, in International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 9590–9596.

[8] D. Fridovich-Keil, E. Ratner, A. D. Dragan, and C. J. Tomlin, *Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games*, arXiv preprint arXiv:1909.04694, (2019).

[9] D. Fridovich-Keil, V. Rubies-Royo, and C. J. Tomlin, *An iterative quadratic method for general-sum differential games with feedback linearizable dynamics*, arXiv preprint arXiv:1910.00681, (2019).

[10] J. Herrera de la Cruz, B. Ivorra, and Á. M. Ramos, *An algorithm for solving a class of multiplayer feedback-nash differential games*, Mathematical Problems in Engineering, 2019 (2019).

[11] G. Kossioris, M. Plexousakis, A. Xepapadeas, A. de Zeeuw, and K.-G. Mäler, *Feedback nash equilibria for non-linear differential games in pollution control*, Journal of Economic Dynamics and Control, 32 (2008), pp. 1312–1331.

[12] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, *Multi-hypothesis interactions in game-theoretic motion planning*, 2020, https://arxiv.org/abs/2011.06047.

[13] T.-T. Lu and S.-H. Shiou, *Inverses of 2× 2 block matrices*, Computers & Mathematics with Applications, 43 (2002), pp. 119–129.

[14] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[15] P. V. Reddy and G. Zaccour, *Feedback nash equilibria in linear-quadratic difference games with constraints*, IEEE Transactions on Automatic Control, 62 (2017), pp. 590–604, https://doi.org/10.1109/TAC.2016.2555879.

[16] P. V. Reddy and G. Zaccour, *Open-loop and feedback nash equilibria in constrained linear–quadratic dynamic games played over event trees*, Automatica, 107 (2019), pp. 162–174.

[17] A. Tanwani and Q. Zhu, *Feedback nash equilibrium for markov jump games under differential-algebraic constraints with application to robust control*, in 2018 IEEE Conference on Decision and Control (CDC), 2018, pp. 1124–1129, https://doi.org/10.1109/CDC.2018.8619628.

[18] A. Tanwani and Q. Zhu, *Feedback nash equilibrium for randomly switching differential-algebraic games*, IEEE Transactions on Automatic Control, (2019).

[19] Z. Wang, R. Spica, and M. Schwager, *Game theoretic motion planning for multi-robot racing*, (2018).