

# The explicit game-theoretic linear quadratic regulator for constrained multi-agent systems

Emilio Benenati\*, and Giuseppe Belgioioso\*

\*Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden

**Abstract**—We present an efficient algorithm to compute the explicit open-loop solution to both finite and infinite-horizon dynamic games subject to state and input constraints. Our approach relies on a multiparametric affine variational inequality characterization of the open-loop Nash equilibria and extends the classical explicit constrained LQR and MPC frameworks to multi-agent non-cooperative settings. A key practical implication is that linear-quadratic game-theoretic MPC becomes viable even at very high sampling rates for multi-agent systems of moderate size. Extensive numerical experiments demonstrate order-of-magnitude improvements in online computation time and solution accuracy compared with state-of-the-art game-theoretic solvers.

## I. INTRODUCTION

Dynamic games model strategic interactions among self-interested rational decision makers whose actions influence and are influenced by the evolution of a shared dynamical system [1]. From a control perspective, dynamic games can be viewed as a multi-agent generalization of optimal control, in which each agent, the decision maker, faces an individual optimal control problem that is coupled to the others through the shared dynamics and the performance objective.

The most celebrated solution concept for dynamic games is the Nash equilibrium (NE), namely, a collection of control sequences (open-loop NE) or feedback laws (closed-loop NE) which are simultaneously optimal for all decision makers. In contrast to standard optimal control problems, dynamic games exhibit several distinctive theoretical and computational challenges. For instance, the NE may be non-unique even in simple linear quadratic (LQ) settings with strongly convex objectives [2], [3], and it can yield time-varying feedback laws that fail to stabilize the dynamics [4]. Moreover, available numerical methods are largely restricted to special classes of games. For LQ dynamic games, the NE can be computed by integrating the coupled state–costate dynamics on an invariant subspace [5], [6]. Another approach formulates the problem as a system of coupled algebraic Riccati equations, solved using iterative algorithms [7]. However, these techniques do not naturally extend beyond the unconstrained LQ setting.

In dynamic games subject to state and input constraints, the problem of computing a NE can instead be reformulated as a finite-dimensional variational inequality [8] and solved using existing iterative algorithms [9]. While this approach does not directly extend to infinite-horizon settings, one can approximate the infinite-horizon NE solution by repeatedly solving a finite-horizon dynamic game and applying the first element of the resulting control sequence. This control paradigm, known as Receding Horizon Games [10] or Game Theoretic

Planning [11] generalizes Model Predictive Control (MPC) to multi-agent non-cooperative settings, and has recently gained considerable research momentum in several research areas, including swarm robotics [12], energy management in smart grids [13], smart mobility systems [14], autonomous driving [15], [16], [17], and racing, involving drones [11], various ground vehicles [18], [19] and even Formula 1 cars [20].

Motivated by the growing interest in applications, several recent works have investigated theoretical performance and stability guarantees for game-theoretic MPC under various structural assumptions. For instance, [13], [21] establish closed-loop stability for dynamic games with a potential structure, while [10] extends these results to general-form dynamic games with stable dynamics. The work in [22] proves closed-loop stability based on an approximate NE computation combined with a suitable terminal cost. Finally, [23] shows that infinite-horizon performance and stability can be achieved through an appropriate choice of terminal ingredients.

From a practical perspective, a major challenge lies in the computational burden of reliably solving a finite-horizon constrained dynamic game at each sampling time. To address this issue, several dedicated numerical solvers have been proposed, including methods based on Newton’s type iterations [24], [25], ADMM [26], interior-point methods [27], successive LQ approximations [15], and active-set strategies [28].

The practical applicability of these solvers in on-line settings is nevertheless limited by their convergence speed. For example, in an autonomous driving scenario with four vehicles, ALGAMES [24] and iLQGames [15], two state-of-the-art solvers, report an average computation time of  $860 \pm 251$  ms and  $705 \pm 209$  ms, respectively [24, § 5]. These computation times correspond to a control frequency of roughly 1 Hz, which is already borderline for autonomous driving and clearly inadequate for more time-critical applications such as autonomous racing. In the latter case, trajectory-level planning typically operates at 5–20 Hz, while low-level vehicle dynamics are controlled at 20–100 Hz [29], [30].

An approach to mitigate this computational burden is proposed in [31], where the authors study the closed-loop stability of game-theoretic MPC under inexact computations, namely when only an approximate NE is available at each sampling instant. The limitations of this approach lie in the inherent trade-off between closed-loop performance and the available computational-time budget, as well as in the lack of analytical bounds on the iteration number to preserve nominal stability.

In this paper, we address this computational bottleneck in

the LQ settings by developing an algorithm that constructs the explicit feedback policy of game-theoretic MPC, namely, the mapping from the initial state of the system to the resulting open-loop NE. This can be interpreted as an extension of the classical explicit MPC framework [32] to multi-agent game-theoretic settings. Our contributions are listed below.

- (i) We derive an explicit characterization of the solution mapping of LQ dynamic games subject to state and input constraints as a piecewise affine (PWA) function of the initial state of the shared system. This directly yields an explicit PWA representation of the state-feedback policy in LQ game-theoretic MPC. Moreover, we extend this result to infinite-horizon dynamic games by leveraging their receding-horizon characterization derived in [23].
- (ii) We develop an efficient algorithm to compute this PWA state-feedback policy based on the state-space exploration technique proposed in [33]. Unlike multi-parameter QP approaches, this method avoids the exponential branching typical of parameter-space partitioning, yielding improved scalability. To the best of our knowledge, this is the first algorithm to compute the explicit solution mapping of a constrained LQ dynamic game.
- (iii) We conduct extensive numerical studies to assess the limitations of the proposed approach and compare its performance against state-of-the-art solvers [34], [26], [25]. In scenarios with up to 8 states and a 10-step horizon, our approach achieves order-of-magnitude speedups in online computation time. Moreover, under a strict computational budget of 0.1s, it yields a worst-case solution accuracy improvement of up to  $10^{10}$  in terms of natural residuals.

Finally, we illustrate the real-time capabilities of the proposed approach on an autonomous driving scenario, showing that it enables the implementation of an overtaking maneuver while meeting a strict sampling rate requirement of 10 Hz.

#### A. Notation and Preliminaries

**Basic notation:** For a matrix  $M$ , we define the weighted norm  $\|x\|_M^2 := x^\top M x$ . For a collection of matrices (or vectors)  $\{M_i\}_{i \in \mathcal{I}}$ , we denote by  $\text{col}(M_i)_{i \in \mathcal{I}}$  their column-wise stacking. The symbol  $\otimes$  denotes the Kronecker product. We write  $\mathbf{0}_{m,n}$  and  $\mathbf{1}_{m,n}$  for the  $m \times n$  matrices of all zeros and all ones, respectively, and omit the dimensions when they are clear from context. Finally, we use  $\mathbb{Z}_T := \{0, \dots, T-1\}$  to denote the discrete-time index set of length  $T$ .

**Operators theory:** We adopt standard operator-theoretic notation and definitions from [35]. For a closed convex set  $\mathcal{Y} \subseteq \mathbb{R}^n$ , we denote the projection of  $x \in \mathbb{R}^n$  onto  $\mathcal{Y}$  by  $\text{proj}_{\mathcal{Y}}(x) = \arg\min_{y \in \mathcal{Y}} \|y - x\|$ . The normal cone of  $\mathcal{Y}$  is the operator  $N_{\mathcal{Y}} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$  defined by  $N_{\mathcal{Y}}(x) = \{v \in \mathbb{R}^n : \sup_{y \in \mathcal{Y}} \langle v, y - x \rangle \leq 0\}$  if  $x \in \mathcal{Y}$ , and  $N_{\mathcal{Y}}(x) = \emptyset$  otherwise. We denote the graph of an operator  $A : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$  by  $\text{gra}_A = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n : y \in A(x)\}$ . We say that  $A$  is  $(\mu - \text{strongly})$  monotone if  $\langle x - y, u - v \rangle \geq 0$  ( $\mu \|x - y\|^2$ ) for any  $(x, u) \in \text{gra}_A$  and  $(y, v) \in \text{gra}_A$ . For a single-valued operator  $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and a set  $\mathcal{B} \subseteq \mathbb{R}^n$ , the variational inequality  $\text{VI}(A, \mathcal{B})$  is the problem of finding a vector  $z^* \in \mathcal{B}$

such that  $A(z^*)^\top (y - z^*) \geq 0$  for all  $y \in \mathcal{B}$ , or, equivalently [8, Eq. 1.1.3], such that  $0 \in \mathcal{N}_{\mathcal{B}}(z^*) + A(z^*)$ . We denote as  $\text{SOL}(A, \mathcal{B}) := \{z \in \mathbb{R}^n \mid 0 \in A(z) + \mathcal{B}(z)\}$  the set of solutions to  $\text{VI}(A, \mathcal{B})$ . If  $A$  is affine and  $\mathcal{B}$  polyhedral, then we refer to  $\text{VI}(\mathcal{F}, \mathcal{C})$  as an affine variational inequality (AVI).

## II. PROBLEM FORMULATION

We consider the multi-agent linear time-invariant system

$$x^{t+1} = Ax^t + \sum_{i \in \mathcal{I}} B_i u_i^t, \quad (1)$$

where  $u_i^t \in \mathbb{R}^{m_i}$  is a control input determined by agent  $i$ , with  $i \in \mathcal{I} := \{1, \dots, N\}$  denoting the set of agents, and  $x^t \in \mathbb{R}^n$  is a shared state. The goal of each agent  $i \in \mathcal{I}$  is to minimize an individual quadratic performance criterion, defined over a control horizon  $\mathbb{Z}_T = \{0, 1, \dots, T-1\}$ , namely,

$$J_i(x, u_i) = \sum_{t=0}^{T-1} \frac{1}{2} \|x^t\|_{Q_i}^2 + \frac{1}{2} \|u_i^t\|_{R_i}^2, \quad (2)$$

with  $u_i = (u_i^t)_{t \in \mathbb{Z}_T}$ ,  $x = (x^t)_{t \in \mathbb{Z}_T}$ , and  $Q_i, R_i \succ 0$ ; while enforcing the following state and input constraints at all times:

$$Dx^t \leq d, \quad (3a)$$

$$G_i u_i^t + \sum_{j \in \mathcal{I} \setminus \{i\}} G_j u_j^t \leq g. \quad (3b)$$

Note that the performance criterion in (2) depends implicitly on the decisions of the other agents  $u_{-i} := (u_j)_{j \in \mathcal{I} \setminus \{i\}}$ , as the evolution of the shared state in (1) is determined by the control action of all agents  $u := (u_i)_{i \in \mathcal{I}}$ . Overall, each agent  $i \in \mathcal{I}$  faces the following optimal control problem:

$$\forall i \in \mathcal{I} : \min_{u_i, \xi} \sum_{t \in \mathbb{Z}_T} \frac{1}{2} \|\xi^t\|_{Q_i}^2 + \frac{1}{2} \|u_i^t\|_{R_i}^2 \quad (4a)$$

$$\text{s.t. } \xi^{t+1} = A\xi^t + \sum_{j \in \mathcal{I}} B_j u_j^t, \quad \forall t \in \mathbb{Z}_T \quad (4b)$$

$$D\xi^t \leq d, \sum_{j \in \mathcal{I}} G_j u_j^t \leq g, \quad \forall t \in \mathbb{Z}_T \quad (4c)$$

$$\xi^0 = x^0, \quad (4d)$$

where  $\xi$  is an auxiliary variable that tracks the shared state evolution, whereas  $x^0$  is the actual initial state of system (1). The collection of these  $N$  interdependent optimal control problems constitutes a *dynamic game* with constraints.

We first focus on *open-loop Nash equilibria* (ol-NEs) of the dynamic game (4). This solution concept assumes that the agents formulate their strategies at the moment the system starts to evolve, and cannot change them once the system runs. An ol-NE describes a strategy profile for which the agents are robust against any attempt by another agent to unilaterally alter their strategy. In other words, at an ol-NE, any agent that unilaterally changes their strategy will always be worse off.

Before formally introducing this solution concept, we write the game (4) in a more compact form by using the dynamics (4b) to eliminate the auxiliary state variable  $\xi$ , obtaining

$$\forall i \in \mathcal{I} : \min_{u_i} f_i(u_i, u_{-i}, x^0) \quad (5a)$$

$$\text{s.t. } (u_i, u_{-i}) \in \mathcal{U}(x^0), \quad (5b)$$

where  $f_i$  is a quadratic cost, parametrized in  $x^0$ , of the form

$$f_i(u, x^0) = \frac{1}{2} \|u_i\|_{H_{ii}}^2 + u_i^\top F_i x^0 + \frac{1}{2} \|x^0\|_{Z_i}^2 + \sum_{j \neq i} u_i^\top H_{ij} u_j, \quad (6)$$

and  $\mathcal{U}(x^0)$  is a polyhedral set of feasible input sequences,

$$\mathcal{U}(x^0) := \{u \in \mathbb{R}^n \mid Cu + Dx^0 \leq c\}. \quad (7)$$

The constraints matrices  $C, D, c$  and the quadratic cost functions  $f_i$  are formally derived in Appendix A.

**Definition 1.** A feasible collective input sequence  $u^* \in \mathcal{U}(x^0)$  is an ol-NE at the initial state  $x^0$  if, for each agent  $i \in \mathcal{I}$ :

$$u_i^* \in \arg \min_{u_i} f_i(u_i, u_{-i}^*, x^0) \text{ s.t. } (u_i, u_{-i}^*) \in \mathcal{U}(x^0) \quad (8)$$

Since the feasible set  $\mathcal{U}(x^0)$  in (7) couples the decision of the agents, this definition corresponds to that of generalized Nash equilibrium (GNE) [36, §12]. Therefore, for finite-horizon LQ dynamic games as in (4), the ol-NEs correspond to the GNEs of the compact game (5). Exploiting this equivalence, we can find one of such ol-NE by solving an affine variational inequality, parametrized in  $x^0$ , namely

$$\text{AVI}(\mathcal{F}(\cdot, x^0), \mathcal{U}(x^0)), \quad (9)$$

where  $\mathcal{F}(\cdot, x^0) := (\nabla_{u_i} f_i(u, x^0))_{i \in \mathcal{I}}$  is the pseudo-gradient mapping of (5), obtained by stacking the partial gradients of the local functions  $f_i$ , and is an linear mapping of the form

$$\mathcal{F}(u, x^0) = Hu + Fx^0. \quad (10)$$

The matrices  $H$  and  $F$  are formally derived in Appendix A.

Whenever  $\mathcal{U}(x^0) \neq \emptyset$ , the AVI in (9) admits a unique solution if the pseudogradient  $\mathcal{F}(\cdot, x^0)$  is strongly monotone [8, Prop. 2.3.3]. Since  $\mathcal{F}$  is a linear mapping, strong monotonicity for all parameters  $x^0$  is equivalent to the matrix condition below [8, Prop. 2.3.2], which we assume to hold true.

**Assumption 1.** The matrix  $H$  in (10) is positive definite.

In the remainder of this paper, we focus on the offline computation of the parameter-to-solution mapping

$$\mathcal{S} : x^0 \mapsto \text{SOL}(\mathcal{F}(\cdot, x^0), \mathcal{U}(x^0)), \quad (11)$$

over a compact set of initial conditions  $\mathcal{X} \subset \mathbb{R}^n$ . We refer to the problem of finding an explicit representation for  $\mathcal{S}$  as a multiparametric AVI (mpAVI). The next sections presents two relevant scenarios where the mpAVI is of practical relevance.

**Remark 1.** The results presented in the remainder of the paper can be readily extended to objective functions with coupling terms between agents' inputs of the kind  $u_i^\top R_{ij} u_j$ , with  $j \neq i$ . We omit these terms for ease of exposition.

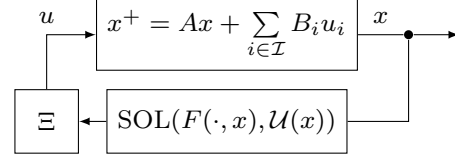


Fig. 1: In game-theoretic MPC, control actions are generated by solving a finite-horizon dynamic game and applying the first element of the ol-NE control sequence to the system.

#### A. Game Theoretic Model Predictive Control

Game-theoretic MPC employs recursive ol-NE of finite-horizon dynamic games to approximate infinite-horizon solutions, while remaining computationally tractable and responsive to unforeseen disturbances. At each sampling time, the agents compute an ol-NE by solving the AVI in (9), and then apply the first element of the ol-NE control sequence. This creates an implicit feedback policy  $\kappa$ , defined as

$$\kappa(x) = \Xi \mathcal{S}(x), \quad (12)$$

where  $\Xi$  is a selection matrix that extracts the first elements from the ol-NE sequence  $\mathcal{S}(x)$ . A schematic representation of this game-theoretic MPC controller is illustrated in Figure 1.

Critically, implementing the control law in (12) requires computing the solution to the AVI in (9) within the desired sampling interval of the controlled system. Existing solvers for dynamic games are based on iterative algorithms and their convergence may be too slow for high sampling rate systems. Computing the explicit solution mapping  $\mathcal{S}$  in (11) offline would enable on-line evaluation of ol-NEs with high accuracy without relying on costly iterative procedures.

#### B. The Game-theoretic Linear Quadratic Regulator

Ideally, one would solve the problem in (4) over an infinite control horizon, namely, for  $T \rightarrow \infty$ , as that allows the agents to optimize over the entire trajectory of the system. Unfortunately, the problem of computing an ol-NE of an infinite-horizon games cannot be immediately recast to a finite-dimensional AVI, as that in (9), due to the infinite number of decision variables and constraints. Nonetheless, under suitable assumptions on the game primitives, infinite-horizon ol-NEs can be recovered by solving a finite-horizon game equipped with appropriately chosen terminal ingredients [23]. Specifically, consider the fixed-point condition:

$$\forall i \in \mathcal{I} : u_i^* \in \arg \min_{u_i} f_i(u_i, u_{-i}^*, x^0) + \varphi_i^{\text{term}}(u^*, x^0) \text{ s.t. } (u_i, u_{-i}^*) \in \mathcal{U}(x^0), \quad (13)$$

obtained by augmenting (5) with the terminal cost

$$\varphi_i^{\text{term}}(u^*, x^0) := \frac{1}{2} \|\xi^T\|_{P_i}^2 + (S_i^\top x^{*,T})^\top \xi^T, \quad (14)$$

where  $P_i$  solves the algebraic Riccati equation

$$P_i = Q_i + A^\top P_i A - A^\top P_i B_i (R_i + B_i^\top P_i B_i)^{-1} B_i^\top P_i A, \quad (15)$$

the matrix  $S_i$  is given by  $S_i = X_i - P_i$ , with  $(X_i)_{i \in \mathcal{I}}$  solving the systems of coupled matrix equations

$$\forall i \in \mathcal{I} : \begin{cases} X_i = Q_i + A^\top X_i (A + B_i \sum_{j \in \mathcal{I}} K_j), \\ K_i = -R_i^{-1} B_i^\top X_i (A + B_i \sum_{j \in \mathcal{I}} K_j), \end{cases} \quad (16)$$

and  $x^{*,T}$  is defined recursively as  $x^{*,0} = x^0$ , and  $x^{*,t+1} = Ax^{*,t} + \sum_{i \in \mathcal{I}} B_i u_i^{*,t}$  for  $t \in \mathbb{Z}_T$ . This terminal cost is designed to reconstruct the contribution of the infinite-horizon cost-to-go truncated at time  $T$ . Under some additional technical assumptions, Theorem 1 in [23] shows that  $u_i^*$  which satisfies the fixed-point condition in (13) coincides with the truncation of an infinite-horizon ol-NE.

Crucially, the fixed-point condition in (13) does not correspond to that of a GNE, as  $u_i^*$  appears in the argument of the optimization problem for agent  $i$ . Nevertheless, the fixed-point condition augmented with the terminal cost in (14) remains a set of coupled LQ optimization problems, and it therefore admits a parametric AVI reformulation akin to (9). We refer the interested readers to [23, Prop. 2] for the formal proof of this statement and a complete derivation of the VI formulation. Finally, we derive the multiparametric AVI corresponding to this infinite-horizon case in Appendix A for completeness.

As a consequence, by constructing an explicit representation of the solution mapping of this AVI, we obtain an explicit state-feedback characterization of the infinite-horizon ol-NE.

### III. MULTIPARAMETRIC AFFINE VIs

For a fixed  $x^0$ , the AVI problem in (9) is equivalent to finding a primal-dual pair  $(u^*, \lambda^*)$  that solves the KKT system

$$0 = Hu + Fx^0 + f + C^\top \lambda \quad (17a)$$

$$0 \leq \lambda \perp -Cu - Dx^0 + c \geq 0, \quad (17b)$$

where  $\lambda$  is a dual variable associated with the constraints (7) [8, Prop. 1.2.1]. If  $\mathcal{U}(x^0) \neq \emptyset$ , then (17) admits a unique primal solution  $u^*$  following the existence and uniqueness of the solution to (9). Denote as  $\mathcal{A} \subseteq \{1, \dots, p\}$  the set of all active constraints at  $u^*$  and as  $C_{\mathcal{A}}, D_{\mathcal{A}}, c_{\mathcal{A}}$  the rows of  $C, D$  and  $c$  (respectively) associated with  $\mathcal{A}$ , that is,

$$C_{\mathcal{A}} u^* + D_{\mathcal{A}} x^0 = c_{\mathcal{A}}. \quad (18)$$

Furthermore, denote as  $\lambda_{\mathcal{A}}^*$  the elements of  $\lambda^*$  associated to the active constraints. By the complementarity slackness condition in (17b), the dual variables of the inactive constraints are 0 and, thus,  $C_{\mathcal{A}}^\top \lambda_{\mathcal{A}}^* = C^\top \lambda^*$ . Then, (17) implies the linear system

$$\begin{bmatrix} H & C_{\mathcal{A}}^\top \\ C_{\mathcal{A}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} u^* \\ \lambda_{\mathcal{A}}^* \end{bmatrix} + \begin{bmatrix} F \\ D_{\mathcal{A}} \end{bmatrix} x^0 + \begin{bmatrix} f \\ -c_{\mathcal{A}} \end{bmatrix} = 0. \quad (19)$$

We assume for now that  $\mathcal{A}$  is non-empty, and that  $u^*$  satisfies the linearly independent constraint qualifications (LICQ)

**Definition 2.**  $u^*$  satisfies the LICQ if the rows of  $C$  associated to its active set  $\mathcal{A}$ , namely,  $C_{\mathcal{A}}$ , are linearly independent.

Since  $C_{\mathcal{A}}$  is full row rank and  $H \succ 0$ , the system (19) admits a unique primal-dual solution [8, Ex. 1.8.9]

$$u^* = -H^{-1}(Fx^0 + f - C_{\mathcal{A}}^\top \lambda_{\mathcal{A}}^*) \quad (20a)$$

$$\lambda_{\mathcal{A}}^* = -(C_{\mathcal{A}} H^{-1} C_{\mathcal{A}}^\top)^{-1} (C_{\mathcal{A}} H^{-1} (Fx^0 + f) - D_{\mathcal{A}} x^0 + c_{\mathcal{A}}). \quad (20b)$$

Now, let us define the primal and dual solution mappings

$$\mathcal{S}_{\mathcal{A}}^u : x^0 \mapsto \begin{cases} u^* \text{ as in (20a)} & \text{if } \mathcal{A} \neq \emptyset \\ -H^{-1}(Fx^0 + f) & \text{if } \mathcal{A} = \emptyset \end{cases} \quad (21a)$$

$$\mathcal{S}_{\mathcal{A}}^\lambda : x^0 \mapsto \text{col}(\lambda_{\mathcal{A}}^*, \mathbf{0}_{p-|\mathcal{A}|}) \text{ with } \lambda_{\mathcal{A}}^* \text{ as in (20b)}, \quad (21b)$$

which extend the explicit primal solution (20a) to the case  $\mathcal{A} = \emptyset$ , and pad the dual (20b) with a vector of zeros associated with the inactive constraints. We will refer to  $\mathcal{S}_{\mathcal{A}} = \mathcal{S}_{\mathcal{A}}^u \times \mathcal{S}_{\mathcal{A}}^\lambda$  as the  $\mathcal{A}$ -active solution map.

If  $u^*$  does not satisfy the LICQ, then the matrix  $\begin{bmatrix} H & C_{\mathcal{A}}^\top \\ C_{\mathcal{A}} & \mathbf{0} \end{bmatrix}$  is singular [8, Ex. 1.8.9]. The primal solution is still unique [8, Prop. 2.3.3], and a primal-dual pair that solves (17) – and consequently solves (19) – still exists [8, Prop. 1.2.1]. However, by the rank-nullity theorem, (19) admits infinite dual solutions. Furthermore, the matrix  $(C_{\mathcal{A}} H^{-1} C_{\mathcal{A}})$  is singular, thus the  $\mathcal{A}$ -active solution map in (21b) is not well-defined. In this case, however, we show as part of the proof of the following statement that a dual solution to (19) exists with zero elements associated to the linearly dependent constraints, and that the primal and non-zero elements of such dual solution are given by the  $\bar{\mathcal{A}}$ -active solution, where  $\bar{\mathcal{A}}$  is a subset of  $\mathcal{A}$  with associated  $C_{\bar{\mathcal{A}}}$  full row rank.

**Proposition 1.** For each  $x^0 \in \mathcal{X}$ , either of the following holds:

- i)  $\mathcal{U}(x^0) = \emptyset$ ,
- ii)  $\mathcal{S}_{\mathcal{A}}^u(x^0)$  solves (9) for some  $\mathcal{A} \subseteq \{1, \dots, p\}$  such that  $C_{\mathcal{A}}$  is full-row rank.

The proof is provided in Appendix B. The next proposition shows that the same  $\mathcal{A}$ -active solution mapping remains valid within a polyhedral region of the state space.

**Proposition 2.** Let  $\mathcal{A} \subseteq \{1, \dots, p\}$  such that  $C_{\mathcal{A}}$  is full row rank. Then,  $\mathcal{S}_{\mathcal{A}}(x^0)$  solves the KKTs (17) for all  $x^0$  such that

$$C_{\mathcal{A}} u_{\mathcal{A}}^*(x^0) + D_{\mathcal{A}} x^0 \leq c \quad (22a)$$

$$\mathcal{S}_{\mathcal{A}}^\lambda(x^0) \geq 0, \quad (22b)$$

The proof is given in Appendix C. Since  $\mathcal{S}_{\mathcal{A}}$  is affine, the inequalities (22) define a polyhedron in the space of  $x^0$ , which is commonly known as *critical region* of the active set  $\mathcal{A}$ .

**Definition 3.** For  $\mathcal{A} \subseteq \{1, \dots, p\}$  such that  $C_{\mathcal{A}}$  is full row rank, its critical region is defined as

$$\mathcal{X}_{\mathcal{A}} := \{x^0 \in \mathcal{X} \mid (22) \text{ holds true}\}.$$

For a given  $\mathcal{A}$ , the corresponding critical region may be empty. Nevertheless, following Proposition 1, the set of all critical regions covers the state space  $\mathcal{X}$  for which the constraints in (7) are feasible. We denote as  $q$  the number of

---

**Algorithm 1** Combinatorial active-set exploration for mpAVI

---

INITIALIZATION

- 1: Find  $x^0$  such that  $\mathcal{U}(x^0) \neq \emptyset$
- 2: Compute  $u^*$  that solves (17)
- 3:  $\mathcal{A} \leftarrow \{i \in \{1, \dots, m\} \mid C_i u^* + D_i x^0 + c_i = 0\}$
- 4:  $\mathcal{R} \leftarrow \{\mathcal{A}\}$ ,  $\mathcal{S} \leftarrow \{\}$

MAIN LOOP

- 5: **while**  $\mathcal{R}$  is not empty **do**
  - 6:    $\mathcal{A} \leftarrow \text{FirstElement}(\mathcal{R})$
  - 7:    $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{A}$
  - 8:   **if**  $C_{\mathcal{A}}$  full row rank **and**  $\mathcal{X}_{\mathcal{A}} \neq \emptyset$  **then**
  - 9:     Compute  $\mathcal{S}_{\mathcal{A}}$  as in (21)
  - 10:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\mathcal{S}_{\mathcal{A}}, \mathcal{X}_{\mathcal{A}})\}$
  - 11:      $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{A} \cup \{i\}\}$ , for all  $i \notin \mathcal{A}$
  - 12:      $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{A} \setminus \{i\}\}$ , for all  $i \in \mathcal{A}$
  - 13:   **else if**  $C_{\mathcal{A}}$  not full row rank **and**  $\mathcal{X}_{\mathcal{A}} \neq \emptyset$  **then**
  - 14:      $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{A} \setminus \{i\}\}$ , for all  $i \in \mathcal{A}$
  - 15:   **end if**
  - 16: **end while**
  - 17: **return**  $\mathcal{S}$
- 

non-empty critical regions. The AVI solution mapping  $\mathcal{S}$  in (12) can be explicitly formulated as

$$\mathcal{S}(x) = \begin{cases} \mathcal{S}_{\mathcal{A}_1}^u(x) & \text{if } x \in \mathcal{X}_{\mathcal{A}_1}; \\ \vdots & \\ \mathcal{S}_{\mathcal{A}_q}^u(x) & \text{if } x \in \mathcal{X}_{\mathcal{A}_q}. \end{cases} \quad (23)$$

The mapping (23) is piecewise-affine (PWA), and can, in principle, be pre-computed offline by completing all possible combinations in  $\{1, \dots, p\}$ , where  $p$  is the number of constraints. However, the number of combinations grows combinatorially with  $p$ , which in practice limits the applicability of this brute-force approach to small-scale problems.

#### A. Efficient Construction of the Explicit Solution Mapping

We explain how to reduce the burden of computing the explicit mapping (23) by discarding combinations of constraints that are not instrumental in constructing a full-dimensional critical region. Following [37], we construct a tree in which each node is a set of constraints  $\mathcal{A} \subseteq \{1, \dots, p\}$ , and its child nodes are all the sets of the form  $\mathcal{A} \cup \{i\}$ ,  $i \in \{1, \dots, p\}$ .

For quadratic programs, the authors of [33] show that any two geometrically adjacent critical regions of the state space are connected in the tree and, crucially, the connecting path has some favourable structural properties that we formalize next, following [33, Def. 5] and [33, Lemma 2].

**Definition 4.** A sequence of subsets  $\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(n)}$  of  $\{1, \dots, p\}$  is combinatorially valid if, for all  $k$ :

- (i) the critical region  $\mathcal{X}_{\mathcal{A}^{(k)}}$  is non-empty;
- (ii) there exists  $i \in \{1, \dots, p\}$  such that either  $\mathcal{A}^{(k+1)} = \mathcal{A}^{(k)} \cup \{i\}$  or  $\mathcal{A}^{(k+1)} = \mathcal{A}^{(k)} \setminus \{i\}$ ;
- (iii) either  $C_{\mathcal{A}^{(k)}}$  or  $C_{\mathcal{A}^{(k+1)}}$  is full-row rank.

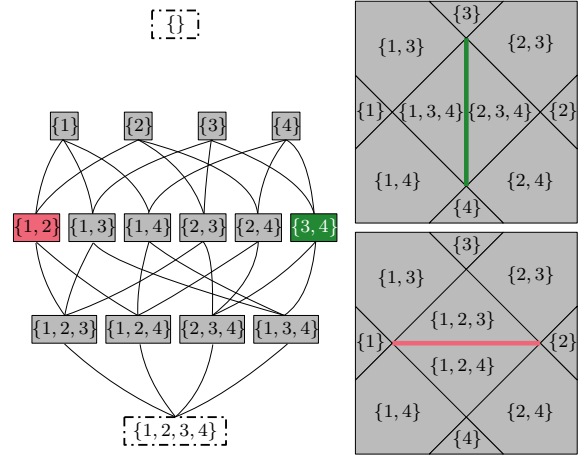


Fig. 2: Left: combinatorial tree of constraint in [37, Ex. 2]. Right: state space  $\mathcal{X} = [-1.5, 1.5]^2$  (drawn twice for clarity), partitioned in the critical regions found via Alg. 1. The constraints corresponding to  $\mathcal{A} = \{1, 2, 3, 4\}$  are not linearly independent, thus the corresponding (non-empty) critical region is excluded from the state-space partition. For  $\mathcal{A} = \{1, 2\}$  and  $\{3, 4\}$  we find a one-dimensional critical region: the colored lines segment. Any path on the graph is a combinatorially valid sequence as for Definition 4. There is no edge leading to  $\{\}$ , as it exhibits an empty critical region.

**Lemma 1.** Consider any two  $\mathcal{A}, \mathcal{A}' \subseteq \{1, \dots, p\}$  such that  $C_{\mathcal{A}}, C_{\mathcal{A}'}$  are full-row rank, and their respective critical regions  $\mathcal{X}_{\mathcal{A}}, \mathcal{X}_{\mathcal{A}'}$ . Let  $\mathcal{X}_{\mathcal{A}} \cap \mathcal{X}_{\mathcal{A}'} \neq \emptyset$ . Then, there exists a combinatorially valid sequence that begins in  $\mathcal{A}$  and ends in  $\mathcal{A}'$ .

The proof of Lemma 1 follows verbatim from that in [33, Lemma 2] and is thus omitted. In words, Lemma 1 states that, starting from any non-empty critical region of a constraint set  $\mathcal{A}$ , one can reach any other by iteratively adding or removing a single constraint to  $\mathcal{A}$ , ignoring the constraint sets that result in empty critical regions. Furthermore, in view of item (iii) in Definition 4, if by adding a constraint the full-rank condition is violated, then one should next remove a constraint, so that the condition is not violated for two successive active sets of the sequence. This informs the development of Algorithm 1. Note that its initialization requires a non-empty critical region (in Line 2). This can be computed by evaluating the active set at the solution for a particular initial state via any standard VI solver, e.g. [34]. For every constraint combination, the offline computation complexity amounts to the solution of the linear system in (20) and of a feasibility problem. In Figure 2, we illustrate an example of the critical regions computed via Algorithm 1, and the corresponding tree of constraints set.

## IV. NUMERICAL RESULTS

### A. Comparison against state-of-the-art iterative solvers

We investigate the on-line and off-line computational performance of the proposed approach over a randomly-generated set of 100 dynamic games, each of which is solved for 20 initial states. The code used for generating the dataset and producing

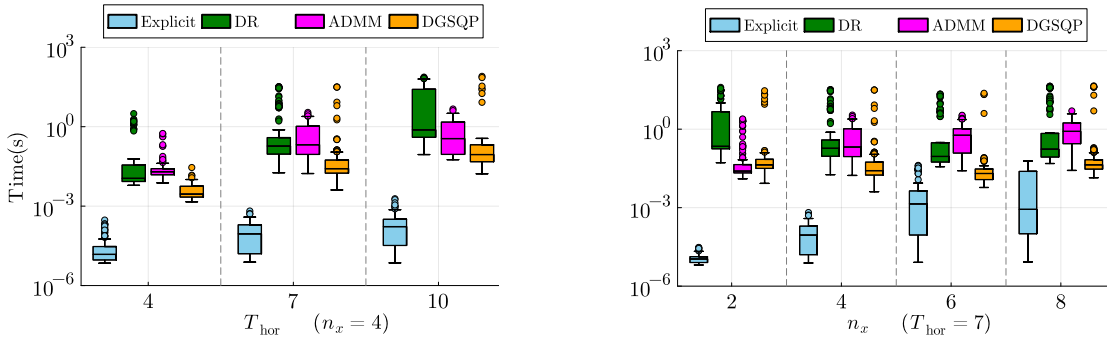


Fig. 3: Online evaluation time of the explicit solution mapping (23) constructed with Alg. 1, compared against the computation time for a high-quality solution ( $r(u) = 10^{-6}$ ) required by the Douglas–Rachford (DR) [34], ADMM [26], and DGSQP [25] solvers. The left plot shows the results in function of the horizon length, while the right one in function of the state dimension.

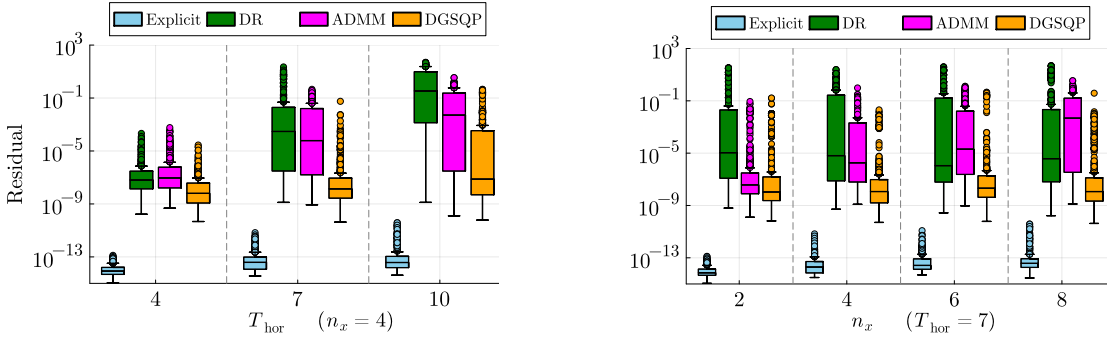


Fig. 4: Solution quality, measured by the natural residual [8, § 6], of the explicit mapping computed with Alg. 1, compared with that obtained by the DR [34], ADMM [26], and DGSQP [25] solvers under a computational budget of 0.1 s. The left plot reports the results as a function of the horizon length, and the right plot as a function of the state dimension.

the results of this section is available online at [https://github.com/bemilio/scripts\\_for\\_explicit\\_LQGames\\_paper](https://github.com/bemilio/scripts_for_explicit_LQGames_paper). All simulations are implemented in Julia and run on a Dell Latitude 7450 with an Intel Ultra 7 165U chip and 32 GB RAM.

In our first experiment, we compare the on-line evaluation time of the explicit solution mappings built with Algorithm 1 against three state-of-the-art iterative solvers [34], [26], [25] set to compute a high-precision solution, measured in terms of the natural residual  $r(u) = \|u - \text{Proj}_{\mathcal{U}(x^0)}(u - \mathcal{F}(u, x^0))\|$ . The iterations are terminated once  $r(u) < 10^{-6}$ , and the results are illustrated in Figure 3. We observe that the online evaluation time of our approach grows sharper with the state dimension: its worst-case value rises from approximately  $10^{-4}$  s, for  $n_x = 2$ , to 0.1 s, for  $n_x = 8$ . This is expected, as higher-dimensional problems generate more critical regions, making evaluation of the PWA mapping more demanding. Nevertheless, it remains faster than all solvers, the best of which, DGSQP [25], exhibits worst-case times of roughly  $10^2$  s across all tested dimensions. Online evaluation time of the explicit mapping shows no substantial dependence on the horizon length, whereas iterative solvers are more affected.

In the second case study, we compare the solution accuracy of the proposed approach against the iterative solvers, under a computational budget of 0.1 s. The results, illustrated in

| $T \backslash n_x$ | 2    | 4    | 6     | 8     |
|--------------------|------|------|-------|-------|
| 4                  | 100% | 100% | 100%  | 100%  |
| 7                  | 100% | 100% | 100%  | 66.7% |
| 10                 | 100% | 100% | 87.5% | 0%    |

TABLE I: Percentage of games whose solution mappings was computed within a 30 minutes runtime limit.

Figure 4, highlight a significant advantage of the explicit approach, as none of the tested solvers was able to compute reliably accurate solutions within the allotted time. This is to be expected, as our proposed method computes the NE exactly (up to machine precision) as the solution to a linear system.

The main limitation of the proposed approach is the offline computational effort required to construct the explicit solution mapping. For each game instance, Algorithm 1 is allocated a maximum of 30 minutes of offline computation time, after which it is terminated even if the mapping has not been fully constructed. In such cases, the algorithm returns a partially built mapping  $\mathcal{S}$  that does not cover the entire state space. Table I reports the percentage of instances for which the full solution mapping was entirely computed within the time limit. We observe that, for  $n_x = 8$  and  $T = 10$ , the computation time limit was exceeded in all tested instances.



| Maneuver phase            | Cost functions of leading and following vehicles                                                                                                                                                                                                                                                      | Active safety constraint                               |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| <i>Platooning</i>         | $J_1 = \frac{1}{2} \sum_{t=0}^{T-1} (\ v_1^t - v_1^{\text{ref}}\ ^2 + \ l_1^t - l_1^{\text{ref}}\ ^2 + \ u_1^t\ _{R_{11}}^2)$<br>$J_2 = \frac{1}{2} \sum_{t=0}^{T-1} (\ p_2^t - p_1^t - d^{\text{ref}}\ ^2 + \ v_2^t - v_2^{\text{ref}}\ ^2 + \ l_2^t - l_2^{\text{ref}}\ ^2 + \ u_2^t\ _{R_{22}}^2)$ | $p_1^t - p_2^t \geq d_{\min}$                          |
| <i>Initiate overtake</i>  | Same $J_1$ , $J_2 = \frac{1}{2} \sum_{t=0}^{T-1} (\ v_2^t - v_2^{\text{ref}}\ ^2 + \ l_2^t - l_2^{\text{ref}}\ ^2 + \ u_2^t\ _{R_{22}}^2)$ ,<br>with $l_2^{\text{ref}}$ set to left-lane center                                                                                                       | $p_2^t - p_1^t \leq -\gamma(l_2^t - l_1^t) - d_{\min}$ |
| <i>Perform overtake</i>   | Same as <i>initiate overtake</i>                                                                                                                                                                                                                                                                      | $l_1^t - l_2^t \geq d_{\min}$                          |
| <i>Complete overtake</i>  | Same as <i>initiate overtake</i> , with $l_2^{\text{ref}}$ set to right-lane center                                                                                                                                                                                                                   | $p_2^t - p_1^t \geq \gamma(l_2^t - l_1^t) + d_{\min}$  |
| <i>Clear-road driving</i> | Same as <i>complete overtake</i>                                                                                                                                                                                                                                                                      | none                                                   |

TABLE II: Summary of cost terms and safety constraints across the five overtake phases.

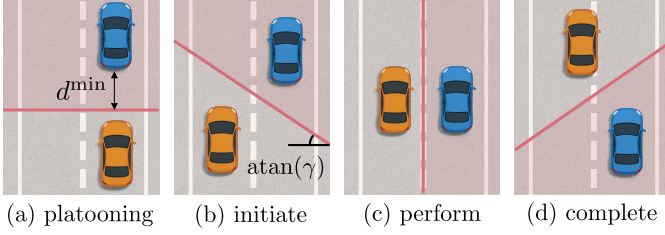


Fig. 5: Safety distance constraint in the manoeuvre phases.

### B. Illustrative example: autonomous driving

We demonstrate the proposed approach on an overtake maneuver involving two autonomous vehicles driving on a two-lane highway. Let  $v_i$ ,  $l_i$ ,  $p_i$ ,  $a_i$ , and  $\alpha_i$  denote respectively the speed, lateral position, longitudinal position, acceleration, and steering angle of vehicle  $i \in \{1, 2\}$ . Each vehicle is modeled as a discretized unicycle [38, Ex. 9.17] with sampling time  $\Delta t = 0.1$  s. The input vectors and the joint state are, respectively,  $u_i = \text{col}(a_i, \alpha_i)$  and  $x = \text{col}(v_1, l_1, p_2 - p_1, v_2, l_2)$ .

Depending on the relative positions of the two vehicles  $\Delta p = p_1 - p_2$ , we specify in Table II the costs and constraints according to five phases of the overtake maneuver: *platooning*, *initiate/perform/complete overtake*, and *clear-road driving*. The finite-state machine in Fig. 6 governs the transitions between these phases, using  $\Delta p$  as the switching variable.

*a) Platooning:* The leading vehicle aims at maintaining a reference speed  $v_1^{\text{ref}}$  and lateral position  $l_1^{\text{ref}}$ . The following vehicle aims at matching the leading vehicle's speed and maintaining a longitudinal distance  $d^{\text{ref}}$ . The safety distance constraint  $p_1 - p_2 \geq d^{\min}$  is imposed to avoid collisions.

*b) Initiate overtake:* The following vehicle aims at reaching the overtake lane and the overtake reference speed, while satisfying the safety distance constraint  $p_2 - p_1 \leq -\gamma(l_2 - l_1) - d^{\min}$  at all times, where  $\gamma > 0$  determines the diagonal separation between vehicles, as illustrated in Figure 5 (b).

*c) Perform overtake:* Same costs as for the *initiate overtake* phase, and with safety distance constraint  $l_1 - l_2 \geq d_{\min}$  to avoid lateral collisions between the vehicles.

*d) Complete overtake:* Same costs as for the *initiate overtake* phase, except that the reference lateral position for the overtaking vehicle (2) is the right lane, and the safety distance constraint is  $p_2 - p_1 \geq \gamma(l_2 - l_1) + d_{\min}$ .

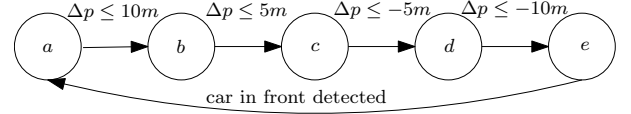


Fig. 6: Finite-state machine that switches between the overtake controller cases. We denote  $\Delta p = p_1 - p_2$ .

*e) Clear road driving:* Same as for the *complete overtake* phase, except that the safety distance constraint is removed.

The explicit game-theoretic MPC controllers obtained by applying Algorithm 1 to the 5 maneuver phases is able to complete the entire overtaking maneuver successfully.<sup>1</sup> Throughout the maneuver, the maximum online evaluation time is 0.0122 s, which is well below the sampling period required for 10 Hz operation. This demonstrates that the explicit feedback law computed offline enables reliable real-time execution of a nontrivial multi-phase autonomous driving task.

## V. CONCLUSION

We showed that the open-loop Nash equilibrium of a finite-horizon linear-quadratic dynamic game admits a piecewise affine dependence on the initial state. Leveraging this structure, we developed an algorithm that constructs the corresponding explicit solution mapping, effectively extending the paradigm of explicit MPC to non-cooperative multi-agent settings. As a result, linear-quadratic game-theoretic MPC becomes computationally feasible even at high sampling rates for multi-agent system of moderate size, as the explicit mapping removes the need for online iterative game-theoretic solvers.

## APPENDIX

### A. Derivation of the AVI

We here report the derivation of  $C, D, c, H_{ij}, F_i, Z_i$  introduced in section II. Denote:

$$\Theta := \text{col}(A^t)_{t \in \{1, \dots, T\}},$$

$$\Gamma_i := \begin{bmatrix} B_i & 0 & \dots & 0 \\ AB_i & B_i & \dots & 0 \\ \vdots & & \ddots & \\ A^{T-1}B_i & A^{T-2}B_i & \dots & B_i \end{bmatrix}$$

<sup>1</sup>Animations available at:  
[https://github.com/bemilio/scripts\\_for\\_explicit\\_LQGames\\_paper/blob/main/vehicle\\_animation.gif](https://github.com/bemilio/scripts_for_explicit_LQGames_paper/blob/main/vehicle_animation.gif)

The dynamics in (1) is equivalently rewritten as

$$x = \Theta x^0 + \left( \sum_{i \in \mathcal{I}} \Gamma_i u_i \right). \quad (24)$$

By stacking the constraints in (3) and substituting (24), (3) is equivalent to

$$\begin{aligned} (I_T \otimes D) (\Theta x^0 + \sum_{i \in \mathcal{I}} \Gamma_i u_i) &\leq \mathbf{1}_T \otimes d \\ \sum_{i \in \mathcal{I}} (I_T \otimes G_i) u_i &\leq \mathbf{1}_T \otimes g. \end{aligned}$$

From the latter, we find the expression for  $C, D, c$  in (7):

$$\begin{aligned} C &= \begin{bmatrix} (I_T \otimes D) \Gamma_1 & \dots & (I_T \otimes D) \Gamma_N \\ I_T \otimes G_1 & \dots & I_T \otimes G_N \end{bmatrix} \\ D &= \begin{bmatrix} (I_T \otimes D) \Theta \\ \mathbf{0} \end{bmatrix} \\ c &= \begin{bmatrix} \mathbf{1}_T \otimes d \\ \mathbf{1}_T \otimes g \end{bmatrix}. \end{aligned}$$

One can rewrite (4a) as

$$\frac{1}{2} \xi^\top \bar{Q}_i \xi + \frac{1}{2} u_i^\top \bar{R}_i u_i,$$

where we denoted

$$\bar{Q}_i = I_{T-1} \otimes Q_i, \quad \bar{R}_i = (I_T \otimes R_i). \quad (25)$$

By substituting (24), we obtain the following formulation for the matrices in (6)

$$\begin{aligned} H_{ii} &= \Gamma_i^\top \bar{Q}_i \Gamma_i + \bar{R}_i, \\ H_{ij} &= \Gamma_i^\top \bar{Q}_i \Gamma_j, \quad \forall j \neq i, \\ F_i &= \Gamma_i^\top \bar{Q}_i \Theta, \\ Z_i &= \Theta^\top \bar{Q}_i \Theta. \end{aligned} \quad (26)$$

With the terminal cost in (14), the problem in (8) is cast as the AVI in (9) by substituting  $\bar{Q}_i$  in (25) with

$$\begin{bmatrix} I_{T-1} \otimes Q_i & 0 \\ 0 & S_i + P_i \end{bmatrix} \quad (27)$$

and then constructing  $(H_{ij})_{i,j \in \mathcal{I}^2}$  and  $(F_i)_{i \in \mathcal{I}}$  as in (26).

### B. Proof to Proposition 1

Let us consider a generic  $x^0$  fixed such that  $\mathcal{U}(x^0) \neq \emptyset$ . Let  $u^*$  be the unique primal solution to (17),  $\mathcal{A}$  the active set at  $u^*$ ,  $C_{\mathcal{A}}$  the associated rows of  $C$ ,  $\lambda^*$  a dual solution to (17), and  $\lambda_{\mathcal{A}}^*$  the elements of  $\lambda^*$  with indexes in  $\mathcal{A}$ . We distinguish three cases:

a) *Case  $\mathcal{A} = \emptyset$ :* By the complementarity condition in (17), it must be  $\lambda^* = \mathbf{0}$ . After substituting  $\lambda^* = \mathbf{0}$  in (17), it is immediate to see that  $\mathcal{S}_{\mathcal{A}}^u(x^0)$  is the only primal solution.

b)  *$u^*$  satisfies the LICQ:* We showed in Section III that, if  $(u^*, \lambda^*)$  satisfy (17), then  $(u^*, \lambda_{\mathcal{A}}^*)$  satisfy (19).  $\mathcal{S}_{\mathcal{A}}(x^0)$  is a solution to (19). By the uniqueness of the solution to (19) [8, Ex. 1.8.9], it must be  $u^* = \mathcal{S}_{\mathcal{A}}^u(x^0)$ .

c)  *$u^*$  does not satisfy the LICQ:* We show that we can construct a set  $\bar{\mathcal{A}} \subset \mathcal{A}$  of linearly independent constraints such that the  $\bar{\mathcal{A}}$ -active solution map in (21) satisfies (17). Without loss of generality, assume  $\lambda_{\mathcal{A}}^* > 0$ . The argument applies unchanged after restricting  $\mathcal{A}$  to the indices corresponding to the nonzero elements of  $\lambda^*$ . Take  $k \in \mathcal{A}$ , and scalars  $(\alpha_j)_{j \in \mathcal{A} \setminus \{k\}}$  such that

$$C_k = \sum_{j \in \mathcal{A} \setminus \{k\}} \alpha_j C_j, \quad (28)$$

where  $C_j$  is the  $j$ -th row of  $C$ . Define the auxiliary multipliers

$$\lambda'_j := \lambda_j^* + \alpha_j \lambda_k^*, \quad \forall j \in \mathcal{A} \setminus \{k\}.$$

We substitute (28) in (17) to get

$$0 = Hu^* + Fx^0 + f + C^\top \lambda^* \Leftrightarrow \quad (29)$$

$$0 = Hu^* + Fx^0 + f + \sum_{j \in \mathcal{A}} C_j^\top \lambda_j^* \Leftrightarrow$$

$$0 = Hu^* + Fx^0 + f + \sum_{j \in \mathcal{A} \setminus \{k\}} C_j^\top (\lambda_j^* + \alpha_j \lambda_k^*) \Leftrightarrow$$

$$0 = Hu^* + Fx^0 + f + \sum_{j \in \mathcal{A} \setminus \{k\}} C_j^\top \lambda'_j \Leftrightarrow$$

$$0 = Hu^* + Fx^0 + f + C^\top \lambda', \quad (30)$$

where

$$\lambda' := \begin{bmatrix} \text{col}(\lambda'_j)_{j \in \mathcal{A} \setminus \{k\}} \\ \mathbf{0}_{p-|\mathcal{A}|+1} \end{bmatrix}.$$

If  $\lambda' \geq 0$ , then (30) implies that  $\lambda'$  is a dual solution with  $|\mathcal{A}| - 1$  non-zero elements. Otherwise, take

$$\gamma = \max_{j \text{ s.t. } \lambda'_j < 0} \frac{\lambda'_j}{\lambda'_j - \lambda_j^*}. \quad (31)$$

Note that  $\gamma \in (0, 1)$ . Construct the vector

$$\lambda'' = (1 - \gamma) \lambda' + \gamma \lambda^*. \quad (32)$$

For each  $j$ , following (31) it holds that

$$\lambda''_j = \lambda'_j + \gamma(\lambda_j^* - \lambda'_j) \geq \lambda'_j - \frac{\lambda'_j}{\lambda'_j - \lambda_j^*} (\lambda_j^* - \lambda'_j) \geq 0$$

and equality holds for the index that maximizes (31). By taking a convex combination of (29) and (30) with coefficient  $\gamma$ , the following holds true:

$$0 = Hu^* + Fx^0 + f + C^\top (\gamma \lambda^* + (1 - \gamma) \lambda'),$$

thus  $\lambda''$  is a dual solution to (17) with  $|\mathcal{A}| - 1$  non-zero elements. We repeat this procedure until we obtain a multiplier  $\bar{\lambda}$  such that the indexes of its non-zero elements  $\bar{\mathcal{A}} \subset \mathcal{A}$  define a full-row rank matrix  $C_{\bar{\mathcal{A}}}$ . Since  $(u^*, \bar{\lambda})$  satisfy (17), then  $(u^*, \bar{\lambda}_{\bar{\mathcal{A}}})$  satisfy (19) for the index set  $\bar{\mathcal{A}}$ . The unique solution to (19) is  $\mathcal{S}_{\bar{\mathcal{A}}}(x^0)$ , thus it must be  $\mathcal{S}_{\bar{\mathcal{A}}}^u(x^0) = u^*$ .

### C. Proof to Proposition 2

We show that the  $\mathcal{A}$ -active solution mapping computed at a generic  $x^0$  that satisfies (22) also satisfies (17). By definition,  $\mathcal{S}(x^0)$  satisfies (19). The primal condition in (17) is immediately satisfied. Following (22), the dual condition is satisfied if complementarity holds, that is,

$$(C_{\mathcal{A}} \mathcal{S}_{\mathcal{A}}^u(x^0) + D_{\mathcal{A}} x^0 - c_{\mathcal{A}})^\top \lambda_{\mathcal{A}}^* = 0. \quad (33)$$



From the second row-block of (19), we have

$$C_A S_A^u(x^0) + D_A x^0 - c_A = 0,$$

which implies (33) and concludes the proof.

## REFERENCES

- [1] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. No. 23 in Classics in Applied Mathematics, Philadelphia: SIAM, 2nd ed., 1999.
- [2] B. Nortmann, A. Monti, M. Sassano, and T. Mylvaganam, “Feedback Nash Equilibria for Scalar Two-Player Linear-Quadratic Discrete-Time Dynamic Games,” *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 1772–1777, 2023.
- [3] G. Salizzoni, R. Ouhamma, and M. Kamgarpour, “Nash equilibria in scalar discrete-time linear quadratic games,” in *2025 European Control Conference (ECC)*, pp. 2416–2421, IEEE, June 2025.
- [4] G. Salizzoni, S. Hall, and M. Kamgarpour, “Bridging Finite and Infinite-Horizon Nash Equilibria in Linear Quadratic Games,” *arXiv preprint arXiv:2508.20675*, 2025.
- [5] J. Engwerda, “Algorithms for computing Nash equilibria in deterministic LQ games,” *Computational Management Science*, vol. 4, pp. 113–140, Apr. 2007.
- [6] M. Sassano, T. Mylvaganam, and A. Astolfi, “OL-NE for LQ differential games: A Port-Controlled Hamiltonian system perspective and some computational strategies,” *Automatica*, vol. 171, p. 111953, Jan. 2025.
- [7] B. Nortmann, A. Monti, M. Sassano, and T. Mylvaganam, “Nash Equilibria for Linear Quadratic Discrete-time Dynamic Games via Iterative and Data-driven Algorithms,” *IEEE Transactions on Automatic Control*, pp. 1–15, 2024.
- [8] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research, New York: Springer, 2007.
- [9] B. Di and A. Lamperski, “Local First-Order Algorithms for Constrained Nonlinear Dynamic Games,” in *2020 American Control Conference (ACC)*, (Denver, CO, USA), pp. 5358–5363, IEEE, July 2020.
- [10] S. Hall, D. Liao-McPherson, G. Belgioioso, and F. Dörfler, “Stability Certificates for Receding Horizon Games,” *arXiv preprint arXiv:2404.12165*, Apr. 2024.
- [11] R. Spica, E. Cristofalo, Z. Wang, E. Montijano, and M. Schwager, “A Real-Time Game Theoretic Planner for Autonomous Two-Player Drone Racing,” *IEEE Transactions on Robotics*, vol. 36, pp. 1389–1403, Oct. 2020.
- [12] A. A. Do Nascimento, A. Papachristodoulou, and K. Margellos, “A Game Theoretic Approach for Safe and Distributed Control of Unmanned Aerial Vehicles,” in *2023 62nd IEEE Conference on Decision and Control (CDC)*, (Singapore, Singapore), pp. 1070–1075, IEEE, Dec. 2023.
- [13] S. Hall, G. Belgioioso, D. Liao-McPherson, and F. Dörfler, “Receding Horizon Games with Coupling Constraints for Demand-Side Management,” in *Conference on Decision and Control (CDC)*, (Cancun, Mexico), pp. 3795–3800, IEEE, Dec. 2022.
- [14] M. Maljkovic, G. Nilsson, and N. Geroliminis, “Receding-Horizon Games with Tullock-Based Profit Functions for Electric Ride-Hailing Markets,” in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, (Milan, Italy), pp. 8384–8389, IEEE, Dec. 2024.
- [15] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, “Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, (Paris, France), pp. 1475–1481, IEEE, May 2020.
- [16] S. Le Cleac’h, M. Schwager, and Z. Manchester, “LUCIDGames: Online Unscented Inverse Dynamic Games for Adaptive Trajectory Prediction and Planning,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 5485–5492, July 2021.
- [17] L. Peters, V. Rubies-Royo, C. J. Tomlin, L. Ferranti, J. Alonso-Mora, C. Stachniss, and D. Fridovich-Keil, “Online and offline learning of player objectives from partial observations in dynamic games,” *The International Journal of Robotics Research*, vol. 42, pp. 917–937, Sept. 2023.
- [18] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, “Game-Theoretic Planning for Self-Driving Cars in Multivehicle Competitive Scenarios,” *IEEE Transactions on Robotics*, vol. 37, pp. 1313–1325, Aug. 2021.
- [19] D. Kalaria, C. Maheshwari, and S. Sastry, “ $\alpha$ -RACER: Real-Time Algorithm for Game-Theoretic Motion Planning and Control in Autonomous Racing using Near-Potential Function,” in *Learning for Dynamics & Control Conference*, PMLR, 2025.
- [20] G. Fieni, M.-P. Neumann, F. Furia, A. Caucino, A. Cerofolini, V. Ravaglioli, and C. H. Onder, “Game Theory in Formula 1: Multi-agent Physical and Strategic Interactions,” *arXiv preprint arXiv:2503.0542*, Mar. 2025.
- [21] E. Benenati and S. Grammatico, “Probabilistic Game-Theoretic Traffic Routing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, pp. 13080–13090, Oct. 2024.
- [22] F. M. Hante and A. Topalović, “Stabilizing Model Predictive Control for Generalized Nash Equilibrium Problems using Approximation by  $\alpha$ -quasi-GNEPs and Lyapunov End Cost,” in *2025 European Control Conference (ECC)*, pp. 2427–2432, IEEE, June 2025.
- [23] E. Benenati and S. Grammatico, “Linear-Quadratic Dynamic Games as Receding-Horizon Variational Inequalities,” *IEEE Transactions on Automatic Control*, pp. 1–16, 2026 (accepted).
- [24] S. Le Cleac’h, M. Schwager, and Z. Manchester, “ALGAMES: A fast augmented Lagrangian solver for constrained dynamic games,” *Autonomous Robots*, vol. 46, pp. 201–215, Jan. 2022.
- [25] E. L. Zhu and F. Borrelli, “A Sequential Quadratic Programming Approach to the Solution of Open-Loop Generalized Nash Equilibria,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, (London, United Kingdom), pp. 3211–3217, IEEE, May 2023.
- [26] H. Min, P. Yi, W. Wang, J. Lei, and J. Wang, “ADMM-iCLQG: A Fast Solver of Constrained Dynamic Games for Planning Multi-Vehicle Feedback Trajectory,” in *2025 IEEE Intelligent Vehicles Symposium (IV)*, (Cluj-Napoca, Romania), pp. 2265–2272, IEEE, June 2025.
- [27] B. Liu and D. Liao-McPherson, “A Log-domain Interior Point Method for Convex Quadratic Games,” in *2025 American Control Conference (ACC)*, (Denver, CO, USA), pp. 4891–4896, IEEE, July 2025.
- [28] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, “The Computation of Approximate Generalized Feedback Nash Equilibria,” *SIAM Journal on Optimization*, vol. 33, pp. 294–318, Mar. 2023.
- [29] J. Kabzan, M. I. Valls, V. J. F. Reijgwart, H. F. C. Hendrikx, C. Ehmke, M. Prajapat, A. Bühler, N. Gosala, M. Gupta, R. Sivanesan, A. Dhall, E. Chisari, N. Karnchanachari, S. Brits, M. Dangel, I. Sa, R. Dubé, A. Gawel, M. Pfeiffer, A. Liniger, J. Lygeros, and R. Siegwart, “AMZ Driverless: The full autonomous racing system,” *Journal of Field Robotics*, vol. 37, pp. 1267–1294, Oct. 2020.
- [30] J. Betz, T. Betz, F. Fent, M. Geisslinger, A. Heilmeyer, L. Hermansdorfer, T. Herrmann, S. Huch, P. Karle, M. Lienkamp, B. Lohmann, F. Nobis, L. Ögretmen, M. Rowold, F. Sauerbeck, T. Stahl, R. Trauth, F. Werner, and A. Wischnewski, “TUM autonomous motorsport: An autonomous racing software for the Indy Autonomous Challenge,” *Journal of Field Robotics*, vol. 40, pp. 783–809, June 2023.
- [31] M. Liu and I. Kolmanovsky, “Input-to-State Stability of Newton Methods in Nash Equilibrium Problems with Applications to Game-Theoretic Model Predictive Control,” *arXiv preprint arXiv:2412.06186*, Dec. 2024.
- [32] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” *Automatica*, vol. 38, pp. 3–20, 2002.
- [33] D. Arnström and D. Axehill, “A High-Performant Multi-Parametric Quadratic Programming Solver,” *arXiv preprint arXiv:2404.05511*, Apr. 2024.
- [34] J. Eckstein and M. C. Ferris, “Operator-Splitting Methods for Monotone Affine Variational Inequalities, with a Parallel Application to Optimal Control,” *INFORMS Journal on Computing*, vol. 10, pp. 218–235, May 1998.
- [35] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.
- [36] D. P. Palomar and Y. C. Eldar, *Convex Optimization in Signal Processing and Communications*. Cambridge university press, 2010.
- [37] P. Ahmadi-Moshkenani, T. A. Johansen, and S. Olaru, “Combinatorial Approach Toward Multiparametric Quadratic Programming Based on Characterizing Adjacent Critical Regions,” *IEEE Transactions on Automatic Control*, vol. 63, pp. 3221–3231, Oct. 2018.
- [38] S. S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*. Springer Science & Business Media, 2013.